Enhancing Environmental Data Insights with Oracle Analytics Cloud

Alenka Baggia
Faculty of Organizational Sciences
University of Maribor
Kranj, Slovenia
alenka.baggia@um.si

Alenka Brezavšček
Faculty of Organizational Sciences
University of Maribor
Kranj, Slovenia
alenka.brezavscek@um.si

Robert Leskovar
Faculty of Organizational Sciences
University of Maribor
Kranj, Slovenia
robert.leskovar@um.si

Abstract— **In today's data-driven world, effective environmental management relies heavily on advanced analytics tools. This paper explores the application of Oracle Analytics Cloud for in-depth analysis and visualization of environmental data to improve decision-making processes in the municipality of Kranj. The study aims to use Oracle Analytics Cloud for processing and analysing large data sets collected by the municipality, with a focus on detecting trends and patterns that could support policymaking. Oracle Analytics Cloud enables the integration of various data sources and provides a comprehensive platform for cross-functional environmental analysis.**

Keywords— *data analysis, environmental data, Oracle Analytics Cloud*

I. INTRODUCTION

In the face of climate change, environmental data monitoring and analysis are among the key priorities for sustainable environmental management. However, environmental data is becoming increasingly complex and detailed. Environmental scientists therefore need the ability to utilize the available data and information for decision-making. As emphasised by [1], there is a need for interdisciplinary insights to harness the potential of AI for environmental sustainability. The effective integration of different data from different sources to enable comprehensive analysis and gain new insights requires the use of advanced analytical techniques and data science methods ([2], [3], [4]).

The rise of data analytics has created a demand for professionals who are proficient in big data and can use various software tools to extract useful information for decision-making. Specifically, this demand has been shown in smart city management, where data-driven decision-making is based on big data analytics [5]. To train such professionals, educational institutions need both the infrastructure and teachers who are familiar with data analysis. These objectives are in line with the Erasmus+ project Including EVERyone in GREEN Data Analysis (EverGreen), which aims to develop innovative teaching materials and make them available to lecturers and students. These outputs will improve the digital readiness, resilience and capacity of teachers and students and build their digital and sustainability skills.

To achieve the project outcomes, several real-world cases were developed using advanced data analysis tools to facilitate informed decision-making on environmental issues and conservation. This paper presents the results of a case study on the analysis of environmental data on air quality of the Municipality of Kranj, Slovenia (MOK), analysed with Oracle Analytics Cloud (OAC), one of the most powerful software tools for data analytics. The paper aims to illustrate the key data visualization features of the OAC tool using a real-life example, making it more accessible to students and the professional audience.

II. RELATED WORKS AND TOOLS

Data integration and visualisation tools are crucial for big data analysis, especially in environmental monitoring and smart cities. Gupta et al. [2] emphasise the need to integrate multiple data sources to enable comprehensive environmental analysis, while [3] highlights how combining machine learning with visualisation can uncover insights into pollution patterns. In relation to smart cities, [5] discusses how big data analytics support real-time decision making in urban planning and environmental monitoring.

Several tools are commonly used for data integration and visualisation. A Comprehensive review of tools is provided by [6], [7] or [8]. Beside OAC, the most popular tools are Tableau, Power BI or Qlik Sense. Tableau is known for its interactive visualisations and ease of use, which makes it accessible to non-technical users, although it offers limited data preparation capabilities compared to other platforms. Power BI, with its strong integration with Microsoft services, supports real-time analytics and is widely used in environmental monitoring, although it lacks advanced machine learning capabilities. Qlik Sense is characterised by its associative engine in data integration and exploration, making it ideal for big data, but requires more technical expertise than Tableau or Power BI.

III. ENVIRONMENTAL DATA AT THE MUNICIPALITY OF KRANJ

In 2022, the municipality of Kranj was selected for the European Commission's initiative to transform 100 cities into climate-neutral and smart urban centres by 2030 [9]. This recognition is the result of proactive measures taken by the municipality, including establishment of a smart city infrastructure using Internet of Things (IoT) technology. This initiative led to the installation of sensors at key locations to systematically monitor air quality, acoustic pollution and meteorological conditions. The potential for increased air pollution from traffic, industrial operations and combustion processes was taken into account when selecting these locations. The project also enabled citizens to access real-time environmental data via mobile applications and the municipality's digital portal, encouraging public participation and awareness.

Continuous monitoring of pollutants and weather parameters is carried out at each station, including particulate matter (PM10 and PM2.5), nitrogen dioxide (NO_2), carbon

979-8-3503-8768-1/24/$31.00 ©2024 IEEE

monoxide (CO), ozone (O_3) as well as temperature, humidity, air pressure and noise levels.

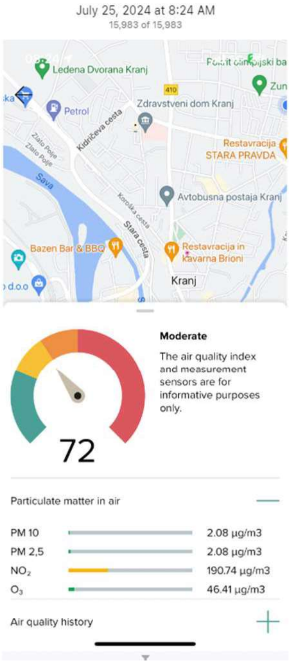
Slovenian environmental legislation, including the Regulation on Ambient Air Quality ([10], [11], [12], [13]), set critical limits and target values that are important for protection of the public health and the environment. These regulations distinguish between limit values, which are defined as threshold values to reduce harmful effects on the health and the environment within a certain period of time, and target values, which are desirable guideline values to reduce potential harm over a certain period of time. Table 1 provides an overview of the limit and the target values for the parameters monitored in the municipality of Kranj. Limit values are prescribed for all the parameters except for ozone, for which a target value has been set.

TABLE I. MONITORED PARAMETERS AND THEIR THRESHOLDS

Parameter	Unit	Measurement time period	Limit / target value
Nitrogen dioxide (NO_2)	µg/m3	1 hour	200, max 18 exceedances/year
		Year	40
PM10	µg/m3	24 hours	50, max 35 exceedances/year
		Year	40
PM2.5	µg/m3	Year	20
Carbon monoxide (CO)	Mg/m3	8 hours	10
Ozon (O_3)	µg/m3	8 hours	120, max 25 exceedances/year

[a] Source: Mestna Občina Kranj, 2023 [14]

The Air Quality Index (AQI) is used to provide real-time air quality data so that residents can adapt their activities to fluctuating pollution levels. Fig. 1 shows an example of the AQI display on the mobile "Pametni Kranj", showing moderate air quality due to the increased value of NO_2 in the air. In this example, the NO_2 level is most likely influenced by the higher summer temperatures, air stagnation and increased sunlight (as a precursor to O_3).



July 25, 2024 at 8:24 AM
15,983 of 15,983

Moderate

The air quality index and measurement sensors are for informative purposes only.

72

Particulate matter in air

PM 10	2.08 µg/m3
PM 2,5	2.08 µg/m3
NO_2	190.74 µg/m3
O_3	46.41 µg/m3

Air quality history

Fig. 1. Mobile app "Pametni Kranj" (Source: Mestna Občina Kranj, 2024).

In addition, the dataset includes acoustic measurements regulated by the Decree on the Assessment and Management of Environmental Noise. This regulation sets four levels of noise thresholds, with the second and third levels applying to residential areas, where the critical noise levels are set at 65 and 80 dBA, respectively.

IV. METHODOLOGY

The basic data set for the case study was provided by the MOK in .csv format and included air quality metrics, historical records and sensor data collected from November 23, 2022, to February 5, 2024. The data was thoroughly cleaned before analysis. The first problem was the notation of PM10 and PM2.5 concentrations below 2 micrograms per cubic meter, referred to as "<2". This symbol hindered the import process into the Oracle APEX [15] database as it was incorrectly read as text data, causing distortions. To resolve this issue, the "<2" entries were replaced with "1" to ensure accurate database integration.

Data cleaning and normalization was performed on the Oracle APEX platform, which is located on Oracle Cloud Infrastructure (OCI). In this phase, redundant columns were removed — three from the *air_quality* dataset and one from the *air_quality_sensors* dataset.

Further examination of the data revealed normalization issues and missing values in the *AIR_QUALITY* table. The measurements varied between different locations and time periods. In addition, some indices, such as the AQI, were calculated as moving averages and were only calculated when sufficient data was available. Normalization was essential to ensure the consistency of the data. The AQI was recorded consistently and therefore created in a separate AQI table together with the fields *sensor_id*, *created_at* and *updated_at*. Other data was assigned to specific tables based on their unique ID and parameter type. These tables were then migrated to the OAC for further processing

Within the OAC, the data underwent an initial review and attribute classification. Most of the data originally labelled as measures had to be reclassified to match the structure of the dataset. Geo-coordinates were converted to location attributes, and date-related information, such as day of week, month, time of year and time of day. were converted to support the planned analysis. After all cleaning activities, the data was ready for the OAC analyses.

V. ORACLE ANALYTICS CLOUD

The Oracle Analytics Platform [16] is a suite designed to fulfil various data analysis requirements. It provides a foundation for data ingestion, preparation, enrichment, visualization and collaboration. The platform helps organizations transition from basic data processing to advanced analytics. By integrating machine learning, Oracle Analytics facilitates the extraction of actionable insights from various data sources, whether in the cloud, on-premises or in a hybrid configuration.

Oracle Analytics offers a balanced approach that combines centralized control of analytics with the flexibility of self-service data exploration. This ensures that organizations can manage their data while allowing individual departments and users to independently derive insights.

The platform supports the entire analytics workflow, including connecting to multiple data sources, modelling complex data sets, enriching data with additional context and exploring data through intuitive visualizations. Storytelling and collaboration features enable teams to share insights and make informed decisions quickly.

Oracle Analytics Cloud (OAC) is the cloud-native version deployed on Oracle Cloud Infrastructure (OCI), providing seamless collaboration and advanced visualization without the need for an on-premises data centre [17]. Oracle Analytics Server (OAS) is an on-premises solution for those who prefer to keep their infrastructure and data in-house.

The Oracle Analytics Platform offers analytics for all roles within an organization. It leverages machine learning and offers a range of functions from no-code analytics to customizable algorithms for specific use cases. The design of the platform ensures that both, centralized reporting and self-service analytics, coexist to provide consistent and reliable data insights across the enterprise.

The Oracle Analytics Cloud (OAC) includes several important functions that are essential for comprehensive data analysis:

- **Data Visualization**: OAC offers a set of data visualization tools that allow users to create meaningful representations of their data, including maps, bar charts, and scatter plots. These visualizations make it easier to understand complex data sets and identify patterns or trends.
- **Data preparation**: OAC allows users to import data from various sources, clean it and prepare it for analysis. This process includes replacing values, defining data types and splitting columns. The prepared data can then be used to create visualizations or perform further analysis.
- **Data analysis**: OAC offers robust functions for in-depth data analysis, such as creating calculations from existing data and defining filters. In addition, OAC's Auto Insights feature offers automatic suggestions for insights based on the data.

These OAC functions are used to analyse environmental data on air quality in the municipality of Kranj. The data is visualized, processed and analysed to provide insights on the level of air pollution and its impact on the environment and human health. The document also highlights the use of OAC for educational purposes and shows its versatility and wide range of applications.

VI. Case Study: The Municipality of Kranj's Air Quality Analysis

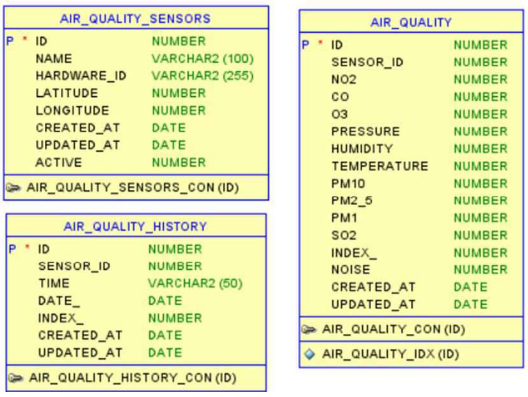
Using OAC to analyse environmental data on air quality in the Municipality of Kranj provided important insights. The data set was prepared and visualized using the robust features of OAC, as described in the Methodology section.

The input data for the analyses were provided in .csv format. The air quality data, the air quality history and the air quality sensors for the period from November 23, 2022, to February 5, 2024, were included in the present analysis. The municipality data is not publicly available. The initial dataset with 7 columns contains 71,103 rows. The columns include attributes *ID*, *sensor_id*, *time*, *date*, *index*, *created_at* and *updated_at*. In addition, input data on sensor locations were given in a separate file, including only 7 rows with basic sensor data, namely *ID*, *name*, *hardware_id*, *latitude*, *longitude*, *created_at*, *updated_at*, *deleted_at* and *active*.

To carry out an efficient data analysis, the data contained in the dataset must be cleaned. The first challenge arose even before the data was imported into the Oracle APEX database tables (Oracle, 2024b). The values of PM10 and PM2.5, which were smaller than 2 were represented with smaller signs and the number 2: <2. This caused major problems when importing the data, as the data was initially treated as text and some values were distorted. To avoid this, we first replaced all strings »<2« with 1 before importing the data into the database.

The .csv files were first uploaded to the Oracle APEX instance in the Oracle Cloud Infrastructure to be cleaned and normalised before analysis. The three empty columns in air_quality and one in air_quality_sensors were dropped.

The preliminary database tables created in Oracle APEX are shown in Fig. 2.



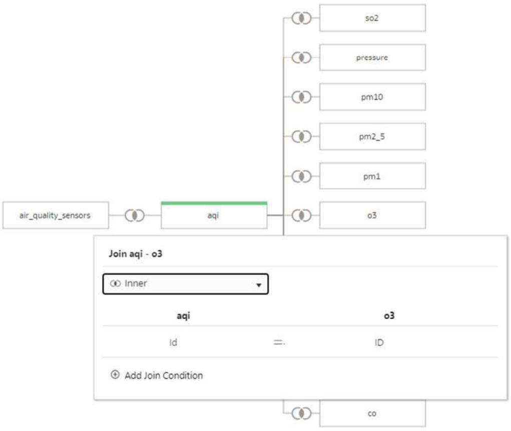
AIR_QUALITY_SENSORS		
P *	ID	NUMBER
	NAME	VARCHAR2 (100)
	HARDWARE_ID	VARCHAR2 (255)
	LATITUDE	NUMBER
	LONGITUDE	NUMBER
	CREATED_AT	DATE
	UPDATED_AT	DATE
	ACTIVE	NUMBER
	AIR_QUALITY_SENSORS_CON (ID)	

AIR_QUALITY_HISTORY		
P *	ID	NUMBER
	SENSOR_ID	NUMBER
	TIME	VARCHAR2 (50)
	DATE_	DATE
	INDEX_	NUMBER
	CREATED_AT	DATE
	UPDATED_AT	DATE
	AIR_QUALITY_HISTORY_CON (ID)	

AIR_QUALITY		
P *	ID	NUMBER
	SENSOR_ID	NUMBER
	NO2	NUMBER
	CO	NUMBER
	O3	NUMBER
	PRESSURE	NUMBER
	HUMIDITY	NUMBER
	TEMPERATURE	NUMBER
	PM10	NUMBER
	PM2_5	NUMBER
	PM1	NUMBER
	SO2	NUMBER
	INDEX_	NUMBER
	NOISE	NUMBER
	CREATED_AT	DATE
	UPDATED_AT	DATE
	AIR_QUALITY_CON (ID)	
◆	AIR_QUALITY_IDX (ID)	

Fig. 2. Preliminary database tables used for data cleaning.

When reviewing the data, it became clear that the data is not normalised, and many values are missing in the *AIR_QUALITY* table. Not all measurements were carried out at all locations and not all measurements were carried out in the same period. In addition, some measurements (e.g. AQI) are based on moving averages of other measurements and are therefore only calculated when the required data is available. Since we did not want to lose so much valuable data, we decided to normalise the output. The AQI is available in each row for each ID. Therefore, this measure was included in the AQI table, along with the *sensor_id*, *created_at* and *updated_at* data. All other data was split into individual tables based on the ID and the specific parameter value. The tables were exported from APEX and imported into the OAC dataset where the ID was defined as a join condition for internal links between the individual data sources. Based on the cleaned data, the dataset shown in Fig. 3. was prepared as the basis for the visualizations, with the ID facilitating the connection between the internal data sources.



Join aqi - o3

Inner

aqi		o3
Id	=	ID

Add Join Condition

Fig. 3. OAC Dataset for MOK.

The dataset was used to prepare sample visualizations of the environmental data provided by the municipality. The visualizations comprised five canvases presented in the following lines.

A. Map of Sensors

The first map shows the locations of the air quality sensors (Fig. 4), where the dot diameter represents the AQI values. This visualization helped to identify areas of high air pollution.

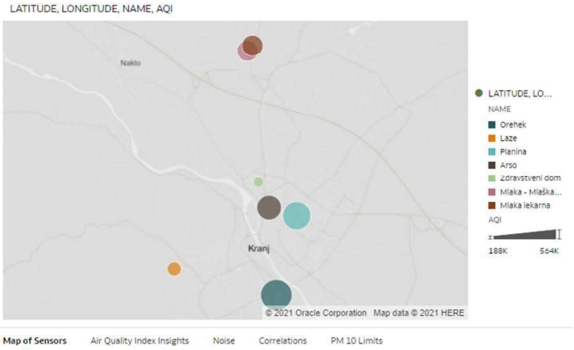


Fig. 4. Map of Sensors visualization.

B. Air Quality Index Insights

The AQI insights canvas contains four different visualizations of the AQI (Fig. 5). First, the average AQI by location, including conditional formatting of locations with exceeded average AQI. Monthly and hourly visualizations of seasonality with the average AQI values provide insight into the evolution of air quality over time. The visualization of hourly AQI values also includes the graphical representation of the AQI threshold, which is 51. The last visualization shows the average AQI value by month and sensor location to get an overall view of air pollution.

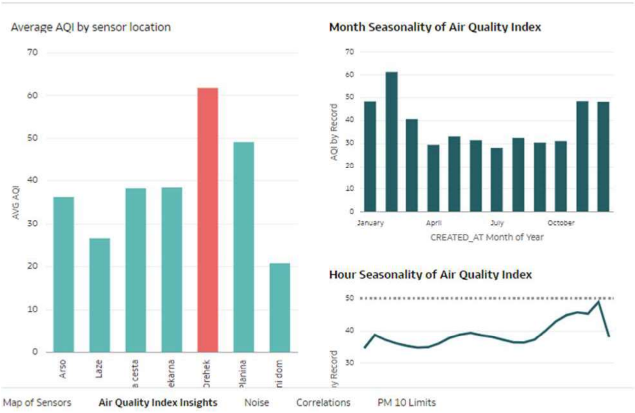


Fig. 5. AQI Details Canvas.

C. Noise

According to the Decree on the Assessment and Management of Environmental Noise, there are four levels of noise thresholds, with levels II and III including residential areas that have a critical noise level of 65 and 80 dBA, respectively. The visualizations on the Noise Canvas (Fig. 6) include the representation of the average noise at the sensor locations with the noise thresholds shown. Further visualizations by day of the week and hours of the day are available for each location. To get an insight into the noise level over the years, the average noise at the sensor locations is displayed by month. The last visualization delves deeper into the noise level exceedance and shows all the cases with a noise level above 65 dBA, with the location colour coded.

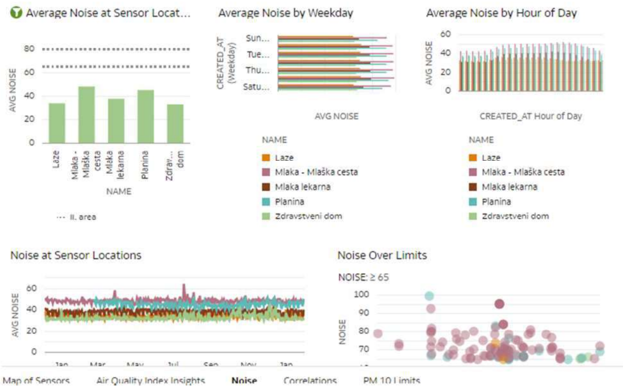


Fig. 6. Noise Canvas.

D. Correlations

The canvas presented in Fig. 7 is mainly used for the introduction of scatter diagrams. To explore the possible correlations, we want to show the relationship between the AQI and various pollutants, including nitrogen dioxide (NO_2) and particulate matter (PM1, PM2.5 and PM10). We also want to investigate the correlation between temperature and ozone (O3), taking into account the spatial distribution of the sensors. These correlations are shown in Fig. 7.

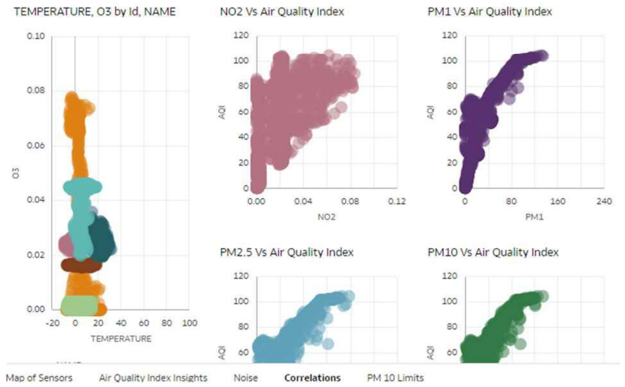


Fig. 7. Correlations Canvas.

The scatter plots show a strong correlation between the AQI values and the particulate matter values (PM1, PM2.5 and PM10), as can be seen from the purple, blue and green scatter plots on the right part of Fig 7. In any case, the relationship between the variables can be described by a monotonically increasing function. In contrast, the correlation between AQI and NO_2, represented by the red scatter plot on the right part of Fig. 7, appears weaker due to the greater dispersion of the data points.

The total scatter, which includes the data from all sensor sites (left-hand side of Fig. 7), shows a fairly homogeneous distribution of points across the sites, except the *Laze* site, which stands out slightly. To understand the reason for this discrepancy, a detailed analysis focusing on the *Laze* site would be required.

E. PM10 limits

This canvas contains visualizations showing PM10 levels (Fig. 8). Particulate matter with a maximum size of 10 micrometers (PM10) can accumulate in the human respiratory tract. Elevated PM10 levels are an indicator of air pollution and pose a risk to human health. According to [14], the limit value for PM10 is 50 µg/m3, whereby 35 exceedances per year are permitted. The average limit value during a year should not exceed 40 µg/m3. In our first visualization, we show the average values of PM10 per sensor location. This serves as a filter for the other two visualizations, namely the daily average averages for the selected site during the whole year 2023, and the graphical representation of the number of exceedances per month at this site.

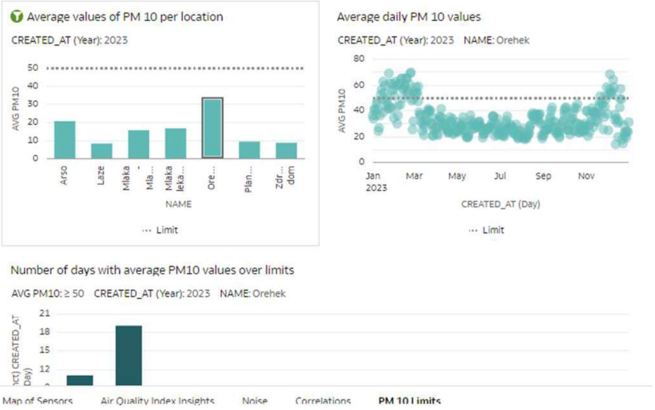


Fig. 8. PM10 Limits Canvas.

The presented visualizations facilitated a comprehensive understanding of air quality in the municipality of Kranj and enabled the identification of pollution sources, high pollution areas and the effectiveness of pollution prevention measures. The results also provided valuable information for the planning of infrastructure and transportation solutions as well as for the implementation of measures to reduce air pollution. In addition, the visualizations helped to communicate the results to local residents and increase public awareness and engagement.

In our case, the data provided by the municipality of Kranj was used to introduce students to the various data analysis possibilities offered by the Oracle Analytics Cloud Service. In addition, this analysis is relevant to policymakers, environmental scientists, urban planners and public health officials interested in using data-driven insights to improve air quality management, urban infrastructure and public health outcomes. The results can also benefit local government agencies, smart city developers and sustainability advocates working to create carbon-neutral urban environments.

VII. Conclusions

The analysis of the air quality data using Oracle Analytics Cloud (OAC) provided several insightful results. In particular, the visualisations revealed that particulate matter concentrations (PM10 and PM2.5) were consistently above the legal limits at certain times, especially in areas with high traffic volumes. In addition, the correlation between nitrogen dioxide (NO2) levels and the Air Quality Index (AQI) was less pronounced compared to particulate matter, suggesting that different pollutants contribute to overall air quality to different degrees. These findings highlight the critical areas where targeted interventions, such as traffic management and anti-pollution measures, could significantly reduce air pollution. Furthermore, seasonal variations in pollution levels were found, with pollution levels being higher during the colder months, likely due to increased combustion activity. These results provide a valuable basis for both policy development and public awareness initiatives.

The application of OAC for the analysis of environmental data on air quality in the MOK has shown significant benefits in data visualization, preparation and analysis. Data visualization simplifies complex information, facilitates quick identification of trends, patterns and outliers, and effectively communicates results to stakeholders. Data preparation automates the cleaning process, saving time and resources and improving data quality for accurate and reliable analysis. Users can manipulate and transform data to meet specific requirements. Data analysis provides deep insights for data-driven decisions, enables prediction of future trends based on historical data, and enables custom calculations and metrics tailored to specific requirements.

Overall, these functions contribute to a more efficient and effective analysis process, leading to better decision-making and strategic planning for organizations. In environmental analysis, they are particularly useful for monitoring pollutant levels, evaluating the effectiveness of environmental measures and planning actions to improve air quality and public health. The MOK air quality analysis case study serves as a practical example of the need for and benefits of analysing environmental data. It can be used as a teaching tool to illustrate the process of data analysis and visualization and serve as a starting point for more complex investigations and visualizations. All files used in this case and the exported OAC workbook are available in the public OCI bucket.

Acknowledgment

The authors would like to acknowledge the support given by the European Commission through the Action Erasmus+ Including EVERyone in GREEN Data Analysis (EverGreen), program ID 2022-1-SK01-KA220-HED-000089149, co-funded by the Erasmus+ program of the European Union. The European Commission's support for the production of this publication does not constitute endorsement of the contents which reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

EverGreen
DATA
Analytics

Co-funded by the
Erasmus+ Programme
of the European Union

The authors would also like to thank Oracle Academy Program Manager, Mr. Darko Jureković, for his continuous support in disseminating the project results.

References

[1] J. Shuford, ‘Interdisciplinary Perspectives: Fusing Artificial Intelligence with Environmental Science for Sustainable Solutions’, *J. Artif. Intell. Gen. Sci. JAIGS ISSN3006-4023*, vol. 1, no. 1, pp. 106–123, Jan. 2024, doi: 10.60087/jaigs.v1i1.87.

[2] S. Gupta, D. Aga, A. Pruden, L. Zhang, and P. Vikesland, ‘Data Analytics for Environmental Science and Engineering Research’, Environ. Sci. Technol., vol. 55, no. 16, pp. 10895–10907, Aug. 2021, doi: 10.1021/acs.est.1c01026.

[3] K. Gibert, J. S. Horsburgh, I. N. Athanasiadis, and G. Holmes, ‘Environmental Data Science’, *Environ. Model. Softw.*, vol. 106, pp. 4–12, 2018, doi: 10.1016/j.envsoft.2018.04.005.

[4] D. T. Hristopulos, B. Spagnolo, and D. Valenti, ‘Open challenges in environmental data analysis and ecological complex systems(a)’, *Europhys. Lett.*, vol. 132, no. 6, pp. 68001–68001, 2020, doi: 10.1209/0295-5075/132/68001.

[5] O. Olaniyi, O. J. Okunleye, and S. O. Olabanji, ‘Advancing data-driven decision-making in smart cities through big data analytics: A comprehensive review of existing literature’, *Curr. J. Appl. Sci. Technol.*, vol. 42, no. 25, pp. 10–18, 2023.

[6] A. Lavanya, S. Sindhuja, L. Gaurav, and W. Ali, ‘A Comprehensive Review of Data Visualization Tools: Features, Strengths, and Weaknesses’, *Int. J. Comput. Eng. Res. Trends*, vol. 10, no. 1, pp. 10–20, Jan. 2023, doi: 10.22362/ijcert/2023/v10/i01/v10i0102.

[7] A. J. Kadam and K. Akhade, ‘A Review on Comparative Study of Popular Data Visualization Tools’, *Alochana J.*, vol. 13, no. 4, pp. 532–538, 2024.

[8] H. M. Shakeel, S. Iram, H. Al-Aqrabi, T. Alsboui, and R. Hill, ‘A Comprehensive State-of-the-Art Survey on Data Visualization Tools: Research Developments, Challenges and Future Domain Specific Visualization Framework’, *IEEE Access*, vol. 10, pp. 96581–96601, 2022, doi: 10.1109/ACCESS.2022.3205115.

[9] MOK, ‘Strategic Council for Climate Neutrality and Smart Communities in Kranj’. Accessed: Jul. 25, 2024. [Online]. Available: https://kranj.si//en/strategic-council-for-climate-neutrality-and-smart-communities-in-kranj

[10] UL, ‘Uredba o kakovosti zunanjega zraka’, *Uradni List RS 92011*, pp. 964–964, 2011.

[11] UL, ‘Uredba o spremembah in dopolnitvah Uredbe o kakovosti zunanjega zraka’, *Uradni List RS 82015*, pp. 523–523, 2015.

[12] UL, ‘Uredba o spremembah Uredbe o kakovosti zunanjega zraka’, *Uradni List RS 662018*, pp. 10140–10140, 2018.

[13] UL, ‘Zakon o varstvu okolja (ZVO-2)’, *Uradni List RS 442022*, pp. 2341–2341, 2022.

[14] Mestna občina Kranj, ‘Kvaliteta zraka’, 2023, [Online]. Available: https://www.kranj.si/kranj-moje-mesto/kakovost-zraka

[15] Oracle, ‘Oracle APEX’, Apr. 2024, [Online]. Available: https://apex.oracle.com/en/

[16] Oracle, ‘Getting Started with Oracle Analytics Cloud’, Jan. 2023, [Online]. Available: https://docs.oracle.com/en/cloud/paas/analytics-cloud/acsgs/what-is-oracle-analytics-cloud.html

[17] Oracle, ‘Oracle Cloud Infrastructure’, Jan. 2024, [Online]. Available: https://www.oracle.com/cloud/

Versatile Function GPA

Tomas Brandejsky
Faculty of Electrical Engineering and Cybernetics
University of Pardubice
Pardubice, Czech Republic
0000-0001-8647-9849

Jan Merta
Faculty of Electrical Engineering and Cybernetics
University of Pardubice
Pardubice, Czech Republic
Jan.Merta@upce.cz

***Abstract*—This herein presented paper is dedicated to the introduction of the continuous versatile function Genetic Programming Algorithm (GPA) developed with respect to BigData processing. The basic structure of this hierarchical evolutionary algorithm and examples of versatile functions are presented.**

On the basis of experiments with the hybrid evolutionary algorithm (hybrid EA) providing symbolic regression of precomputed Lorenz attractor system data representing hybrid EA's behaviour, a discussion of examples of an obtained solution is presented. The versatile function concept GPA is applicable, but it requires the hybrid evolutionary algorithm application, as is demonstrated in the paper.

***Index Terms*—hybrid evolutionary algorithm, genetic programming algorithm, versatile function GPA, BigData, symbolic regression**

I. Introduction

The BigData boom stands especially on two fundamental technologies for its processing - Apache Hadoop and on its successor - Apache Spark. The Hadoop allowed us to divide BigData processing onto many small sub-tasks and then to process them in parallel on the Hadoop cluster. Spark then added in memory processing without storing partial results on memory media and significantly increased processing speed together with many other speed-ups and improvements.

In BigData, not only database systems are used, but also other technologies. At this moment, the Deep Artificial Neural Networks (Deep ANNs) are used in BigData analytics frequently, especially for modeling and situation recognition like computer vision. The main problem of Deep ANNs is complicated learning, which consumes a lot of computational power, time and energy.

In the history of computer science and cybernetics, Professor Lotfi Zadeh, a researcher and promoter of fuzzy sets, recognized that there were three techniques that had some common features and the potential to revolutionize artificial intelligence - evolutionary techniques, fuzzy sets and artificial neural networks (ANNs) - and gave them the common name of soft computing. Despite the different nature of these techniques, they can be easily combined. Fuzzy sets and neural networks are universal approximators, but EAs can also create models and learn their parameters. EAs (genetic algorithms, evolutionary strategies, genetic programming algorithms, and many others) are more suitable than ANNs for solving certain types of problems because of some interesting properties. These include not only optimization but also symbolic regression.

The Symbolic Regression is a method of machine learning capable of finding equations describing training data set. The representation in the form of an algebraic or differential equation can be much more compact than the representation of an ANN representing the same relationship. It makes obtained models easily understandable for humans. Models represented by the equation give a clear view on dependencies between system variables. The symbolic regression is typically provided by Genetic Programming Algorithms. Due to quadratic computational complexity, they are not suitable for large data sets, but the methods based on static or dynamic training data subsets are now studied, but they are not being the subject of this paper. Nowadays, some researchers also study the possibility of using Deep ANN for symbolic regression solving tasks like [1]. The remarks about used computer equipment imply that GPA can be more efficient.

The standard GPA in the sense of original Koza's works discussed later has many limitations when it is applied to BigData. They are related to working with a strongly nonlinear state space, where e.g. replacement of a multiplication binary operator with an additional one significantly changes resulting function properties. For example, such a change can transform a quadratic function into a linear one. This mutation invalidates related constants in the equation, and they have to be estimated again.

The paper brings the idea of GPA operating in a linear space with smooth movement from one function to another. Because the concept of gamma function GPA is novel, the small illustrative example will continue:

We have expression

$$x = a * y + z \tag{1}$$

Let the Γ be the binary function of three parameters. The first two are arguments y and z, third is γ parameter controlling Γ operator behavior. The magnitudes of γ are from interval $< 0, 1 >$. If the γ is 0, the result of $\Gamma(y, z, \gamma)$ is y+z, if the γ is 1, the result of $\Gamma(y, z, \gamma)$ is y*z. For any magnitude of γ , the result is between these defining functions

$$x = \gamma * (y * z) + (1 - \gamma) * (y + z) \tag{2}$$

979-8-3503-8768-1/24/$31.00 ©2024 IEEE

We can tell that the result of Γ operator is, in this case, linear interpolation between '+' and '*' operators. Thus, the equation (1) can be represented using Γ function as (3):

$$x = \Gamma(\Gamma(a, y,' *'), z,' +') \tag{3}$$

J. Koza formed Genetic Programming as an optimization problem in [2] by extending of the previous research of Nichael Lynn Cramer [3]. GPA in this task optimizes both the structure and parameters of an arising model on the basis of information in a training data set or fitness function. The name Genetic Programming is given by the first idea to develop evolutionary computer programs. One of the first applications published by J. Koza 1994 [4] was symbolic regression (discovering of model described by algebraic equation fitting training data set). Later, many different application domains were opened such as analogue electric circuit design [5], synthesis of topology for controller [6], application of GP to the synthesis of complex kinematic mechanisms [7] and many others.

II. Motivation

GPA works on discontinuous and nonlinear space of functions. E.g. in the case of symbolic regression, any change in builded expression can totally change whole expression behavior, like for example replacement of x^2 to $\sqrt{x}$. The idea of versatile function use is to change the nonlinear noncontinuous space of functions to a linear and continuous space of parameters. Thus, at the beginning of this work was the expectation that the GP task solving will be easier if the problem domain is continuous.

Nevertheless, it is not possible to eliminate GPA part of the algorithm and eliminate the solution structure evolution because in such case it is needed to produce full trees. In the case of building Γ function arity n and expected depth of the solution tree d, the number of nodes of the full tree would be

$$m = 1 + n + n^2 ... + n^d = \sum_{i=0}^{d} n^d \tag{4}$$

in many cases, an evaluation of such structures is inefficient because the solution is far of full tree and application of GPA for operator tree development might work faster.

The idea of a fuzzy gamma operator was at the beginning of this work. It was introduced in [8] to combine properties of fuzzy logic AND operator and OR operator. The way of versatile operator suggested in this work is the use of only one kind of versatile function capable to continuously change its properties depending on the magnitude of its control parameters represented by real numbers. A versatile continuous function GPA presented in this work is an extension of the original idea to any operator defined on R with any chosen number of arguments n controlled by an adequate number of control parameters $\gamma_0, ..\gamma_{n-1}$. The number of control parameters of this function is not limited. In this work, the structure of the solution created from versatile functions is developed by GPA and parameters are set up by the Evolutionary Strategy algorithm within the frame of the hybrid algorithm GPAes.

The main aim of presented work is a limitation of evolution on an solution describing equation structure development and transformation of the main part of symbolic regression problem to the continuous optimization of a parameter set. Thus, it is the transformation of the structure development problem onto the problem of estimation of the versatile function parameters. These versatile functions shall have a lot of parameters $\gamma_0, ..\gamma_{n-1}$ depending on their dimension n- the number of simple binary functions they combine into a final versatile one.

III. Versatile Function Concept

The term versatile function denotes in this paper a real function combining properties of several functions. Depending on parameter magnitudes $\gamma_0, ..\gamma_{n-1}$, properties of a versatile function are closer to one or another defining function, as in the case of fuzzy gamma operator [8]. There are many ways to form higher order Γ functions with more $\gamma_0, ..\gamma_{n-1}$ parameters. In the work [9], three versions of the versatile function were tested. They were in the following forms suitable for approximation of Lorenz attractor combining functions $+$, $-$ and $*$, or commonly binary functions f_1, f_2 and f_3: (5).

$$r1 = \gamma_1\gamma_2 f_1(x_1, x_2) + (1-\gamma_1) f_2(x_1, x_2) + (1-\gamma_2) f_3(x_1, x_2) \tag{5}$$

$$r2 = \gamma_1 f_1(x_1, x_2) + \gamma_2 f_2(x_1, x_2) + \gamma_3 f_3(x_1, x_2) \tag{6}$$

γ_i are from interval $< 0, 1 >$.
and

$$r3 = \gamma_1 p_1(x_1, x_2) + \gamma_2 p_2(x_1, x_2) + \gamma_3(x_1, x_2) \tag{7}$$

with diffrerent set of operators $x_1 + 0 * x_2$, $0 * x_1 + x_2$, $x_1 * x_2$ (equal to x_1, x_2 and $x_1 * x_1$) and γ_is are from interval $< -\infty, \infty >$.

IV. Used GPAes algorithm

While some authors understand SR as the way to find any description of training data set, other ones add requirements on precision of the model or its comparability to a solution produced by humans. Such approaches [10] are more computationally expensive but open new application domains. To solve these Highly Accurate SR (HASR) problems, it is sometimes efficient to use hybrid evolutionary algorithms described in the next paragraph. The hybrid Genetic Programming Algorithm with Evolutionary Strategy optimization (GPAes) used in this work, was originally developed for the second approach to SR (and concluding HASR), but in this work it is applied for quick search of approximate model.

A. Hybrid Evolutionary Algorithms in Symbolic Regression

Koza in his work [4] identified a weak point of the Genetic Programming application when it is applied to symbolic regression in the identification of parameter (constant) magnitudes. Till now, there have been many modifications to GPA extending its abilities by linear or nonlinear optimization

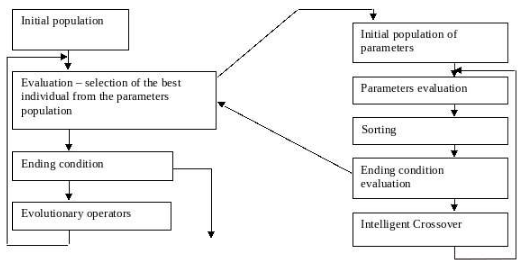

Initial population
Evaluation – selection of the best individual from the parameters population
Ending condition
Evolutionary operators
Initial population of parameters
Parameters evaluation
Sorting
Ending condition evaluation
Intelligent Crossover

Fig. 1. Structure of used hybrid genetic programming algorithm.

techniques such as [11] presented. The optimization techniques can be replaced by a genetic or evolutionary algorithm with similar results, but the consumption of computing resources of a hybrid algorithm with optimization of solution parameters is significant. The work [12] brings comparison of different constant optimization methods influencing hybrid evolutionary algorithm efficiency.

B. Used Hybrid GP Algorithm with Evolutionary Strategy for parameters optimization

An used GPAes algorithm [13] consists of a standard GPA without automatically defined functions, discussed in [4] and in each evolutionary cycle, the Evolutionary Strategy (ES) is applied to optimize the parameters of each individual in the population. The structure of the whole algorithm is outlined in Fig. 1.

C. The different setting of hybrid genetic programming algorithm for versatile function

It is legitimate to expect that a versatile function will ask different behavior of a hybrid algorithm than in the standard symbolic regression problem-solving because it has a lot of parameters to be optimized in nonlinear continuous space. On the opposite side, there is smaller pressure on the structure development due to the presence of only one versatile function and two terminal types (variable and constant) implemented as pointers to arrays of variables and constants. Thus, it is possible to expect that it will be useful to run the parameter optimization part of the hybrid algorithm with more iteration cycles than for a standard GP function set.

The following chapter will demonstrate how gamma function is applied in Lorenz attractor data symbolic regression using (7) form of γ-function style representation.

V. Experiments

As was mentioned above, the Lorenz attractor system served for the first testing of a versatile function concept. This system produces chaotic behaviors for some parameters, and that means it is sensitive to errors in the estimation of the model. The equations describing this model are as follows:

$$\begin{aligned} x'(t) &= \sigma(y(t) - x(t)) \\ y'(t) &= x(t)(r - z(t)) - y(t) \\ z'(t) &= x(t)y(t) - bz(t) \end{aligned} \tag{8}$$

TABLE I
Versatile function test parameters.

Parameter	Magnitude
GPA population size	100
GPA population number limit	20
ES population size	100
ES population number limit	100

Used parameters had magnitudes $\sigma = 16.0; r = 45.92; b = 4.0$ and initial point has position $\{x, y, z\} = \{19, 20, 50\}$.

A training data vector contains 500 samples and the ES cycle limit (number of nested ES optimizer cycles) was set to 100 cycles. The magnitude of constants is a result of optimization by a nested evolutionary strategy algorithm. The test parameters are represented by the Table I.

Example of results produced by the above-described hybrid GP algorithm applying gamma function in the form (7) is outlined by Table II for standard C++ pseudo random generator seed value equal to 2, and they can be transformed into standard representation in (9), (10) and (11).

$$x'(t) = 16.0425 * y - 16.0192 * x + 0.00356392 * (y * x) \tag{9}$$

$$\begin{aligned} y'(t) = &- 30.9282 * x + (-0.004649) * (0.302322 * x \\ &+ 0.541066 * z + 2.52825 * (x * z)) \\ &+ 0.702435 * (x * (0.302322 * x + 0.541066 * z \\ &+ 2.52825 * (x * z))) \end{aligned} \tag{10}$$

$$\begin{aligned} z'(t) = &- 2.33193 * (-9.68919) + 3.19204 * (15.738 * x \\ &+ 0.827112 * y + 0.235339 * (x * y)) + \\ &- 0.612886 * (-9.68919 * (15.738 * x + 0.827112 * y \\ &+ 0.235339 * (x * y))) \end{aligned} \tag{11}$$

After simplification, these symbolic regression results are the following equations (12), (13) and (14).

$$x'(t) = 16.0425 * y - 16.0192 * x + 0.00356392 * (y * x) \tag{12}$$

$$\begin{aligned} y'(t) = &1.775931x^2z + 0.212362x^2 + 0.36831xz \\ &- 30.929605x - 0.002515z \end{aligned} \tag{13}$$

$$\begin{aligned} z'(t) = &2.148741xy + 143.694375x + 7.551871y \\ &+ 22.594513 \end{aligned} \tag{14}$$

Thus, Versatile Function GPA is able in constrained time to produce an approximate but applicable solution of symbolic regression. It is caused by a bigger number of numerical

TABLE II
EXAMPLE OF MODEL PRODUCED BY HYBRID GP

```
x'=  ( larg=var(1));
     ( rarg=var(0));
       gamma1=const no(0 0):=16.0425
       gamma2=const no(0 1):=-16.0192
       gamma3=const no(0 2):=0.00356392
       {gamma1*larg+gamma2*rarg+
                gamma3*(larg*rarg)}
   fitness:=0.0545273

y'=  ( larg=const no(0 0):=-0.755116);
     ( rarg=
       ( larg=var(0));
       ( rarg=var(2));
         gamma1=const no(0 1):=-30.9282
         gamma2=const no(0 2):=-0.004649
         gamma3=const no(0 3):=0.702435
            {gamma1*larg+gamma2*rarg+
              gamma3*(larg*rarg)}
        )
          gamma1=const no(0 4):=0.302322
          gamma2=const no(0 5):=0.541066
          gamma3=const no(0 6):=2.52825
           {gamma1*larg+gamma2*rarg+
             gamma3*(larg*rarg)}
   fitness:=5.29507

z'=  ( larg=const no(0 0):=-9.68919);
     ( rarg=
       ( larg=var(0));
       ( rarg=var(1))
        gamma1=const no(0 1):=-2.33193
        gamma2=const no(0 2):=3.19204
        gamma3=const no(0 3):=-0.612886
         {gamma1*larg+gamma2*rarg+
           gamma3*(larg*rarg)}
        )
        gamma1=const no(0 4):=15.738
        gamma2=const no(0 5):=0.827112
        gamma3=const no(0 6):=0.235339
        {gamma1*larg+gamma2*rarg+
          gamma3*(larg*rarg)}
   fitness:=189.629
```

operations in Gamma function evaluation in comparison to a precise solution.

Of course, there more experiments were computed. The next Table III displays fitness values for different numbers of ES cycles from experiments repeated 10 times for each number ES cycles from set $(1, 3, 10, 32, 100)$ and for each variable $(x, y,$ and $z)$. Fitness vale magnitude represents average (not sum) of squares of value distances between model and training data obtained for each point of training dataset.

TABLE III
AVERAGE FITNESS FOR DIFFERENT NUMBER OF ES CYCLES

ES cycles	Variable		
	x	*y*	*z*
1	626.1064	1561.0518	1131.1116
3	222.4697	837.7314	980.3318
10	18.0633	224.5711	820.5328
32	0.6615	33.7889	428.2599
100	0.5450	8.8722	203.2751

TABLE IV
VERSATILE FUNCTION TEST PARAMETERS FOR MORE PRECISE SOLUTION.

Parameter	Magnitude
GPA population size	100
GPA population number limit	40
ES population size	100
ES population number limit	1000

With a bigger computing effort, it is possible to obtain a more precise solution, as is usual in hierarchical genetic programming applications. Even such a more precise solution does not need to be more complicated than the previous one. In the next example, there were 30 experiments computed with the setting described in Table IV. An example of such a more precise individuals are (15), (16) and (17). The average magnitudes of residual errors are in Table V.

$$\begin{aligned} x'(t) = &-16.0203 * x + 16.0205 * y \\ &+ 0.000992484 * (x * y) \end{aligned} \tag{15}$$

$$\begin{aligned} y'(t) =&((-0.0182939) * ((-13.6712) * z + (-14.5963) \\ &* (-13.1142) + (-0.585305) * (z * (-13.1142))) \\ &+ 12.4751 * x + 0.162529 * (((-13.6712) * z \\ &+ (-14.5963) * (-13.1142) + (-0.585305) \\ &* (z * (-13.1142))) * x)) \end{aligned} \tag{16}$$

$$\begin{aligned} z'(t) =&x(y(1.06327 - 0.00111305z) - 0.000346761z \\ &+ 0.331252) + y(0.000155169z - 0.148228) - 4.0561z \end{aligned} \tag{17}$$

After simplification, these symbolic regression results are the following equations (18), (19) and (20). The equation (15) is without change, but the others were significantly simplified.

$$x'(t) = 16.0425 * y - 16.0192 * x + 0.00356392 * (y * x) \tag{18}$$

$$y'(t) = x(43.5862 - 0.974425z) + 0.109679z - 3.5018 \tag{19}$$

$$\begin{aligned} z'(t) =&2.148741xy + 143.694375x + 7.551871y \\ &+ 22.594513 \end{aligned} \tag{20}$$

TABLE V
AVERAGE FITNESS FOR DIFFERENT NUMBER OF ES CYCLES

ES cycles	Variable		
	x	*y*	*z*
1000	0.04263	2.4782	40.2751

VI. Discussion

3 different shapes of versatile functions are presented. The best - third one - is used for above presented experiments. The application of a versatile function requires a different proportion between GPA and ES cycle limits than in the case of a typical function set consisting of e.g. '+', '-', '*', 'sin' etc. While for a typical function set containing many different functions, the number of ES cycles is limited between 1 and 40 (and the GPA cycle limit is between hundreds and many thousands). In the case of versatile function application, the situation is different. See Tables I and IV. These first results point that this concept is applicable in the area of symbolic regression with respect to future BigData applications.

VII. Conclusion

The presented paper discussed the idea of the novel versatile function genetic programming. It applies a hybrid genetic evolutionary algorithm combining a genetic programming algorithm for structural development and an evolutionary strategy algorithm for constant tuning. This algorithm is developed for big data applications because the main computational effort is concentrated on parameters optimization and due to the application of only one versatile function. It is suitable for use on computing accelerators like General-Purpose Graphics Processing Unit (GPGPU). The change of parameters on the place of whole structure change simplifies communication between processor and GPGPU in the case of future implementation using GPGPU accelerator. The algorithm is developed as an alternative approach to symbolic regression. We expect that due to its applicability to GPGPUs, it will be suitable for modeling systems described by very large data sets.

Acknowledgment

The work was supported by the Erasmus+ project: Project number: 2022-1-SK01-KA220-HED-000089149, Project title: Including EVERyone in GREEN Data Analysis (EVERGREEN) funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the Slovak Academic Association for International Cooperation (SAAIC). Neither the European Union nor SAAIC can be held responsible for them.

Co-funded by the European Union

EverGreen DATA Analytics

References

[1] B. K. Petersen, M. Landajuela, T. N. Mundhenk, C. P. Santiago, S. K. Kim and J. T. Kim, "Deep symbolic regression: Recovering mathematical expressions from data via risk-seeking policy gradients," Computer Research Repository, vol. abs/1912.04871, 2021, https://doi.org/10.48550/arXiv.1912.04871

[2] J. R. Koza, Genetic Programming: On the programming of computers by means of natural selection. MIT Press, Cambridge, Massachusetts, 1992. ISBN-13: 978-0262111706

[3] N. L. Cramer, A representation for the adaptive generation of simple sequential programs. In: Grefenstette, J. J. (ed) Proceedings of an International Conference on Genetic Algorithms and the Applications, 1985, pp.183–187. Carnegie-Mellon University, Pittsburgh, PA, USA

[4] J. R. Koza, Genetic programming II: Automatic discovery of reusable programs. The MIT Press, Cambridge, MA, 1994.

[5] J. R. Koza, D. Andre, F. H. Bennet III, and M. Keane, Genetic programming 3: Darwinian ivention and problem solving. Morgan Kaufman, 1999. ISBN 1-55860-543-6.

[6] J. R. Koza, M. A. Keane, M. J. Streeter, W. YU Mydlowec J., and G. Lanza, Genetic programming IV: Routine human-competitive machine intelligence. Kluwer Academic Publishers, 2003. ISBN 1-4020-7446-8.

[7] H. Lipson, How to draw a straight line using a GP: Benchmarking evolutionary design against 19th century kinematic synthesis. In: M. Keijzer (ed), Late Breaking Papers at the 2004 Genetic and Evolutionary Computation Conference, International Society for Genetic and Evolutionary Computation, CDROM.

[8] C. von Altrock, B. Krause, and H.-J. Zimmerman,"Advanced fuzzy logic control of a model car in extreme situations," Fuzzy Sets and Systems, vol. 48,no. 1, pp. 41–52, 1992.issn 0165-0114,https://doi.org/10.1016/0165-0114(92)90250-8.

[9] T. Brandejsky, "Versatile function in GPA," Neural Network World, 2020, pp. 379–392, DOI: 10.14311/NNW.2020.30.025

[10] M. Korns, "Extreme accuracy in symbolic regression," In: R. Riolo, J. H. Moore, and M. Kotanchek (eds), Genetic Programming Theory and Practice XI (Genetic and Evolutionary Computation). pp. 1–30. Springer, New York, 2014. ISBN-13: 978-1493903740, doi 0.1007/978-1-4939-0375-7-1

[11] B. McKay, M. J. Willis, and G. W, Barton, "Using a tree structured genetic algorithm to perform symbolic regression," In: Proc. of the 1st Int. Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications. UK, pp. 487–492, 1995.

[12] V. Hlavac, "Genetic programming with either stochastic or deterministic constant evaluation," Neural Network World, vol. 2/2018, pp. 119–131, 2018. ISSN 1210-0552. DOI 10.14311/nnw.2018.28.006.

[13] T. Brandejsky, "Nonlinear system identification by GPA-ES," In: Proceedings of the 2012 13th International Carpathian Control Conference, ICCC 2012, pp. 58–62, 2012.

Information System for Crime Monitoring in Europe

Marek Kvet, Michal Kvet, Zuzana Žillová, Erik Malina
University of Žilina, Faculty of Management Science and Informatics
Univerzitná 8215/1
010 26 Žilina, Slovakia
marek.kvet@fri.uniza.sk, michal.kvet@fri.uniza.sk, ORCID 0000-0001-5851-1530

***Abstract*—The goal of the project is to add features to information system, which has the major role in correct processing of crime data for the European region. The most important function of the information system is the statistical processing into a charts and representative tables form. In the information system, we want to focus on monitoring four categories of data, namely reported and convicted crimes, the situation of prisons and the number of police forces in each state. The information system allows administrators to log in to the application, where they can add and edit individual data to database. Data are processed in three main categories, namely data processing for Europe in the form of a report and an interactive map with a criminal index, for an individual state and for comparing states. The result of the work will be the developed web application that can be used by the countries in Europe, data analysts for searching interesting comparisons and the European Union for crime monitoring to increase the security of states, or it can be used by people who just want to expand their horizons in such a field.**

***Keywords*—reported criminal offences, prisons, criminal convictions, police forces, web application, statistics, crime monitoring, administration, crime predictions**

I. Introduction

The very rapid development in the field of information technologies has become an irreplaceable part of our lives. That is why many companies seek for young professionals, who are able to adapt to new or various changing trends very fast. One of the most affected subjects of these increasing changes are also the universities, which educate future experts in IT sector and various data specialists.

Data analysis plays an important role in many currently developed information systems [1, 2, 5, 6, 15]. The amount of data needed to be stored and processed has been rising very fast and thus, the attention of programmers and other scientifically oriented experts has been paid to the completely sophisticated spectrum of data science [3, 4, 7, 8]. Because of data analyzing necessity, the research project [18] has been implemented. Presented research is also a part on mentioned project. Its main idea is based on using advanced database technologies in sophisticated data analysis [9, 10, 11, 12, 13]

The information system project for monitoring criminality in Europe is part of a larger project that includes detailed monitoring and analysis of crime in the territory of the Slovak Republic, but also monitoring traffic accidents in the territory of the Czech Republic. Together with the monitoring of Europe, they will form a complete web application with multifunctional use and the possibility of analysis in each of these parts of the project.

In the past, the information system contained only the first two of the mentioned parts of the project, and crime monitoring in Europe was created as the newest part of the project and now it is becoming to be the main part, while it was implemented into the existing parts, which were unified into one project using the React framework for the frontend and .net for the backend implementation. Monitoring for crime in the territory of Europe fits perfectly into these frameworks because this information system must be created as a dynamic site that uses multiple graphs and reports to achieve the desired results for the users of the information system.

The goal of information system for crime monitoring in Europe was to create an information system that not only processes data but also improves user experience. The project aimed to analyse the data and provide valuable insights into crime trends and criminal justice systems in Europe.

II. Current Development

The development of an information system for crime monitoring in Europe has been undertaken as one part of a complex project. This project aims to analyse and evaluate crime data collected from the European Sourcebook of Crime and Criminal Justice Statistics, provided by the University of Lausanne in Switzerland. The data from official sources of individual countries in Europe offers valuable information into various aspects of criminal offenses, convicted crimes, police forces, and the state of prisons.

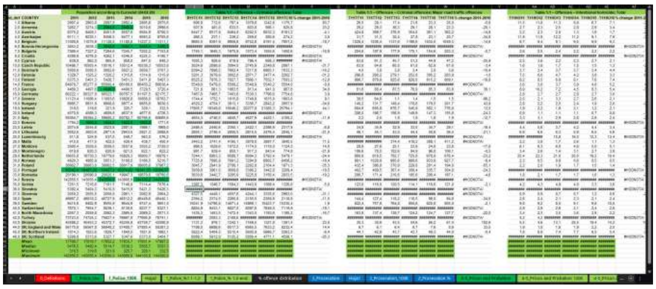


Fig. 1. Data in raw format

The initial phase of the project contained a detailed analysis of the data. The data was available in raw format, required processing for input to the information system. The data was divided into multiple .csv files, related with the database. Nowadays, the project contains crime data between the years 1995 and 2016 of criminal offenses, convicted crimes, and police forces.

979-8-3503-8768-1/24/$31.00 ©2024 IEEE

The development of the information system utilized the .NET platform, known for its speed, cross-platform, and compatibility. A relational database from Oracle was selected as the storage solution, providing a structured and efficient way of organizing the data into tables [5, 14, 16, 17].

In addition to data processing, the project also had a redesign of user interface. Using the React JavaScript library, the frontend was enhanced to improve usability and aesthetics. The navigation bar, and design were redesigned to create a user-friendly experience. The addition of English language as a universal allows accessibility to global users.

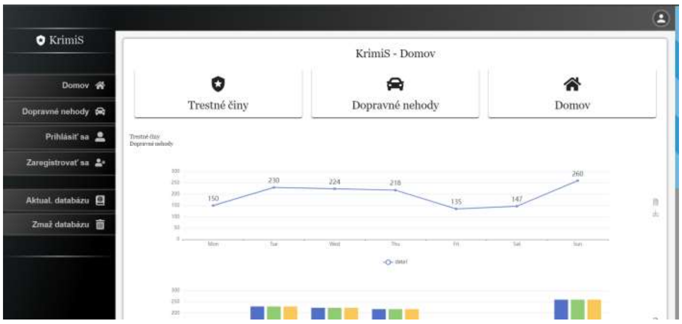


Fig. 2. Site before re-design

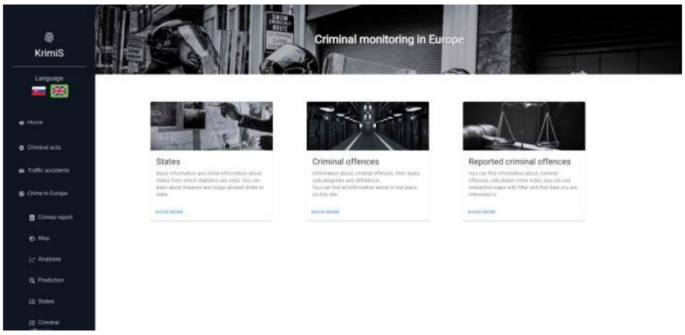
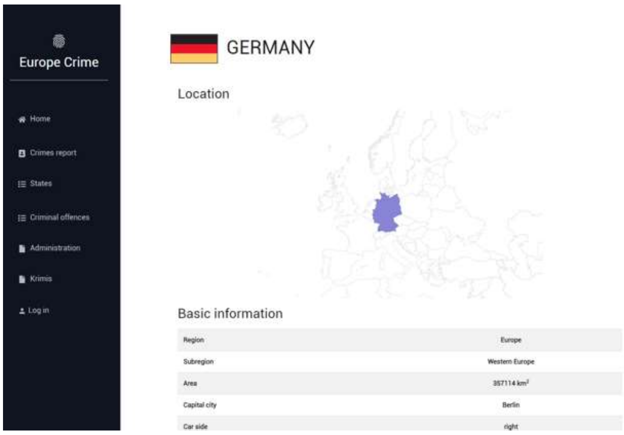


Fig. 3. Site after re-design

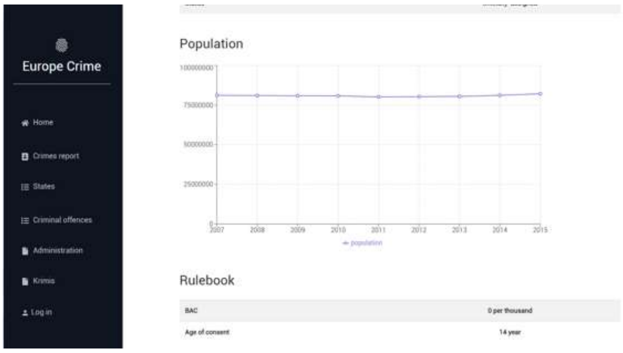
The project further expanded to include subpages dedicated to crime monitoring. A page is focusing on the categorization and definition of individual crimes was created, allowing users to understand specific offenses. Another subpage is providing information about individual states, including basic data, criminal laws, crime activity trends, drug limits, and firearm classifications.



Basic information

Region	Europe
Subregion	Western Europe
Area	357114 km^2
Capital city	Berlin
Car side	right

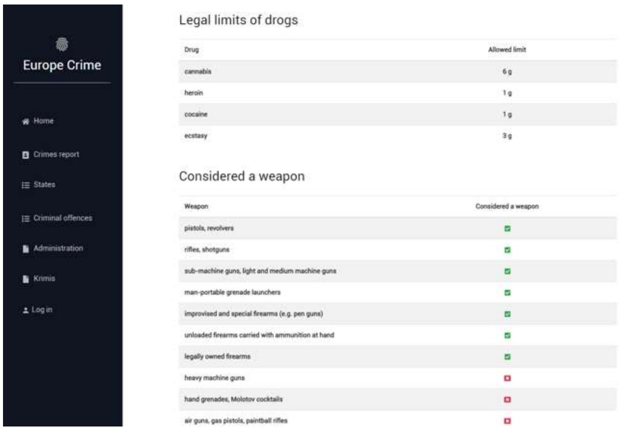
Fig. 4. Country page with basic information



Rulebook

BAC	0 per thousand
Age of consent	14 year

Fig. 5. Country page with population development in years



Legal limits of drogs

Drug	Allowed limit
cannabis	6 g
heroin	1 g
cocaine	1 g
ecstasy	3 g

Considered a weapon

Weapon	Considered a weapon
pistols, revolvers	☑
rifles, shotguns	☑
sub-machine guns, light and medium machine guns	☑
man-portable grenade launchers	☑
improvised and special firearms (e.g. pen guns)	☑
unloaded firearms carried with ammunition at hand	☑
legally owned firearms	☑
heavy machine guns	☒
hand grenades, Molotov cocktails	☒
air guns, gas pistols, paintball rifles	☒

Fig. 6. State page showing legal drug limits and gun information.

One of the pages is an interactive report featuring an integrated map. Users can apply filters to generate reports based on selected countries, years, and type of crime offence. The map represents the crime index, which indicates the number of crimes per 100 000 people. A corresponding table with sorted crime indexes allows overview of the data.

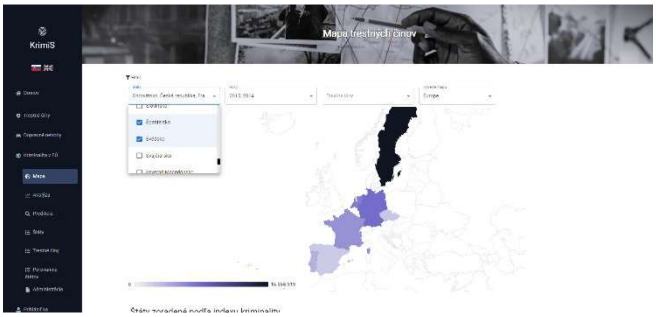


Fig. 7. Criminal report with map

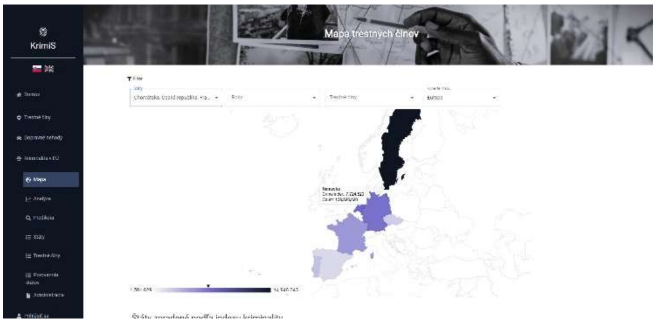


Fig. 8. Criminal report with an overview of the country

The project also prioritizes data analysis, aiming to extract informative insights and present them through various graphs and tables. The development of graphs depicting the trends in crime rates over time is underway, with the example of the number of crimes in the UK from 2007 to 2016 showcased.

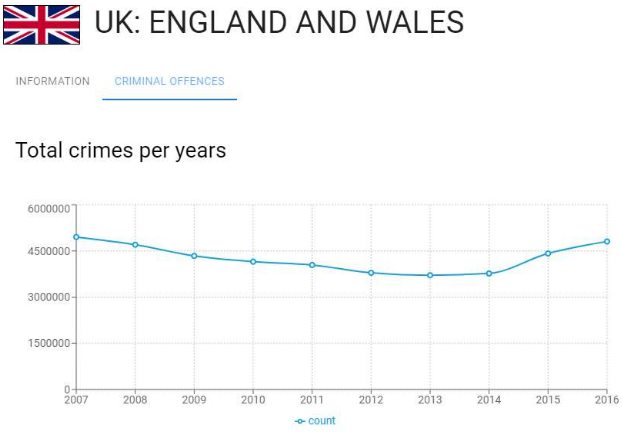


Fig. 9. Data analyses of total crimes per years for UK

III. Main Features of Website

A. *Addition Statistical processing*

The statistical measures in the Crime in Europe report and state crime analyses represents understanding of crime data. With the inclusion of metrics such as the maximum, minimum, mean, median, variance, kurtosis, standard deviation, skewness, and percentiles (25th, 50th, and 75th), the report now offers overview of the distribution and characteristics of crime activity in Europe.

The maximum value signifies the highest recorded crime rate in the dataset, the peak level of criminal activity. The minimum value represents the lowest recorded crime rate.

The mean, also known as the average, provides a measure of central tendency by summing up all the values and dividing them by the total number of data points. This metric offers a representation of the overall crime rate, giving researchers and policymakers an understanding of the average level of criminal activity in Europe.

The median, on the other hand, represents the middle value in the dataset when it is arranged in ascending or descending order. This measure shows the central tendency of crime data and helps identify the midpoint where half of the observations fall below and half above it.

Variance, as a statistical measure, quantifies the spread or dispersion of crime data. It indicates the degree to which crime rates deviate from the mean. A high variance suggests a wide range of crime rates, indicating significant variations across different regions or time periods. Conversely, a low variance implies a relatively consistent crime rate throughout the dataset.

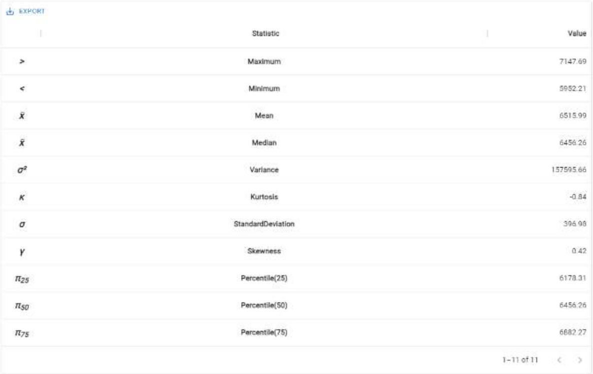
Kurtosis measures the degree of peakiness or flatness of the crime rate distribution. A positive kurtosis value indicates a distribution with heavy tails and a more peaked shape, suggesting the presence of outliers or extreme values. In contrast, a negative kurtosis value reflects a distribution with lighter tails and a flatter shape, indicating a more uniform or normal distribution of crime rates.

The standard deviation, calculated as the square root of the variance, provides a measure of the average amount by which individual crime rates deviate from the mean. It offers an understanding of the typical level of variation or dispersion in the dataset. A higher standard deviation suggests a greater degree of variability in crime rates, while a lower standard deviation indicates a more consistent pattern.

Skewness measures the symmetry of the crime rate distribution. A positive skewness value indicates a longer tail on the right side of the distribution, suggesting the presence of relatively high crime rates. Conversely, a negative skewness value signifies a longer tail on the left side of the distribution, indicating a prevalence of relatively low crime rates.

Additionally, the inclusion of percentiles (25th, 50th, and 75th) allows for a more detailed examination of crime data. These percentiles represent specific points in the dataset, dividing it into four equal parts. The 25th percentile indicates the value below which 25% of the data falls, while the 50th percentile corresponds to the median. The 75th percentile represents the value below which 75% of the data falls. These percentiles offer valuable insights into the distribution of crime rates, highlighting the range of values within specific quartiles.

By incorporating these statistical measures into the Crime in Europe report, researchers and policymakers gain a comprehensive understanding of the crime landscape.

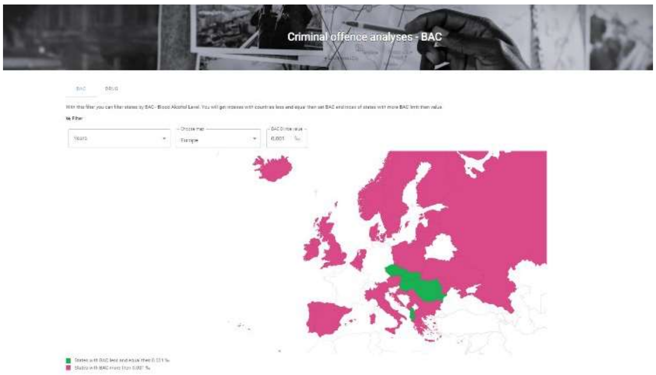


	Statistic	Value
>	Maximum	7147.69
<	Minimum	5952.21
$\bar{x}$	Mean	6515.99
$\tilde{x}$	Median	6456.26
σ^2	Variance	157595.66
κ	Kurtosis	-0.84
σ	StandardDeviation	396.98
γ	Skewness	0.42
π_{25}	Percentile(25)	6178.31
π_{50}	Percentile(50)	6456.26
π_{75}	Percentile(75)	6882.27

Fig. 10. Statistical measures in crime report

B. *Analyses of Blood alcohol level on traffic accidents index*

Next page in the European crime offense site, we have conducted an analysis of blood alcohol levels and their correlation with traffic offenses. This analysis includes a unique filtering option based on years and groups the data into two categories: blood alcohol limits exceeding the user-input value and limits below the user-input value.



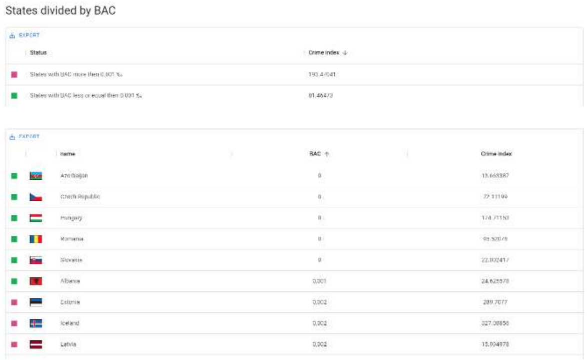
With this filter you can filter states by BAC - Blood Alcohol Level. You will get indexes with countries less and equal then set BAC and indexes of states with more BAC limit then value.

🟩 States with BAC less and equal then 0,001 ‰
🟥 States with BAC more then 0,001 ‰

Fig. 11. BAC analyses page

By utilizing this filter, users can explore the relationship between different blood alcohol limits and the occurrence of traffic offenses within specific time periods. Our findings indicate that there is no significant dependency between blood alcohol limits and the frequency of traffic offenses. This valuable information allows policymakers and authorities to assess the effectiveness of current blood alcohol limits in preventing traffic offenses across European countries.

With analysis and the ability to filter data based on specific years and blood alcohol limit ranges, we provide a powerful tool for researchers, policymakers, and individuals interested in understanding the impact of blood alcohol limits on traffic safety.


States divided by BAC

EXPORT

	Status	Crime index ↓
🟥	States with BAC more then 0,001 ‰	193.47041
🟩	States with BAC less or equal then 0,001 ‰	81.46473

EXPORT

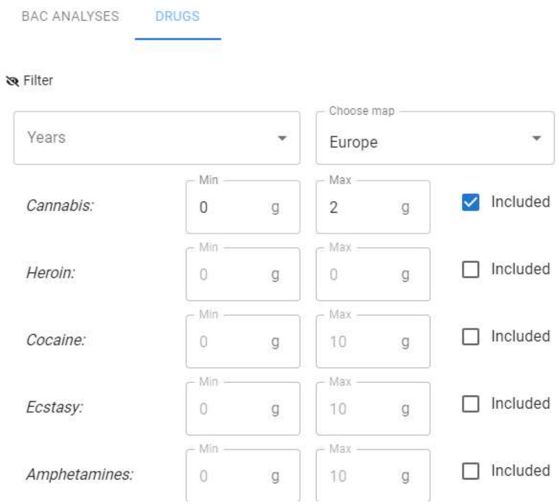
	name	BAC ↑	Crime index
🟩	Azerbaijan	0	13.663387
🟩	Czech Republic	0	72.11199
🟩	Hungary	0	174.71153
🟩	Romania	0	95.52079
🟩	Slovenia	0	22.002417
🟩	Albania	0,001	24.625578
🟥	Estonia	0,002	289.7077
🟥	Iceland	0,002	327.08858
🟥	Latvia	0,002	15.934978

Fig. 12. BAC analyses page statistics and information

C. *Analyses of drugs limits on drug crime index*

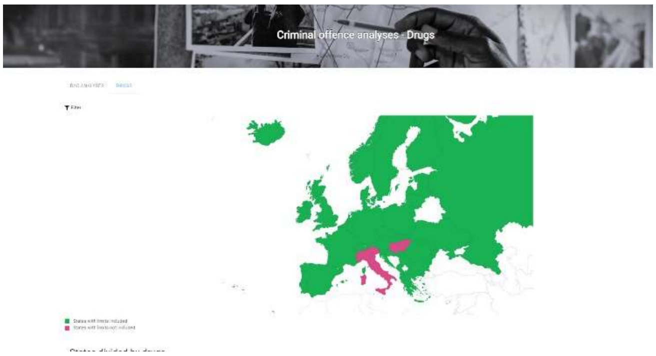
We are excited to introduce a new feature on our Europe Crime website that focuses on drug analyses. With this new addition, users can now explore and analyze the data related to drugs such as Cannabis, Heroin, Cocaine, Ecstasy, and Amphetamines. The feature offers advanced filtering options, allowing users to choose whether they want to include or exclude specific drugs within a range inputted by them. This filter functionality provides a clear distinction between data that falls within the specified range and data that falls outside of it, enabling users to examine the prevalence of drug-related crimes based on their specific criteria.

In addition to the filtering capabilities, the drug analysis feature also provides comprehensive statistics and crime indexes associated with the selected drugs.



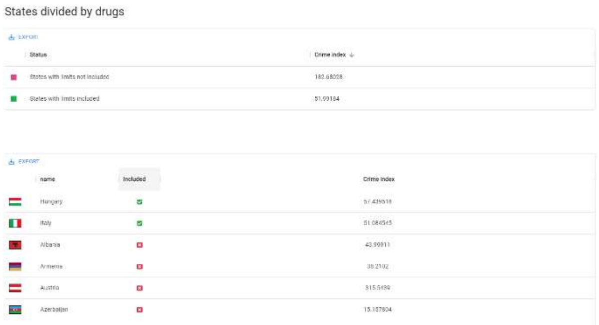
	Min	Max	
Cannabis:	0 g	2 g	☑ Included
Heroin:	0 g	0 g	☐ Included
Cocaine:	0 g	10 g	☐ Included
Ecstasy:	0 g	10 g	☐ Included
Amphetamines:	0 g	10 g	☐ Included

Fig. 13. Filter of drug limits



🟩 States with limits included
🟥 States with limits not included

Fig. 14. Map in drug analyses


States divided by drugs

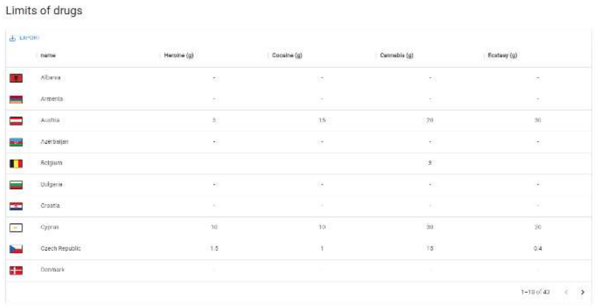
EXPORT

	Status	Crime index ↓
🟥	States with limits not included	182.68028
🟩	States with limits included	51.99184

EXPORT

	name	Included	Crime index
	Hungary	☑	57.439518
	Italy	☑	51.084545
	Albania	☒	43.99911
	Armenia	☒	38.2102
	Austria	☒	315.5499
	Azerbaijan	☒	15.137804

Fig. 15. Table with states and indexes in drug analyses


Limits of drugs

EXPORT

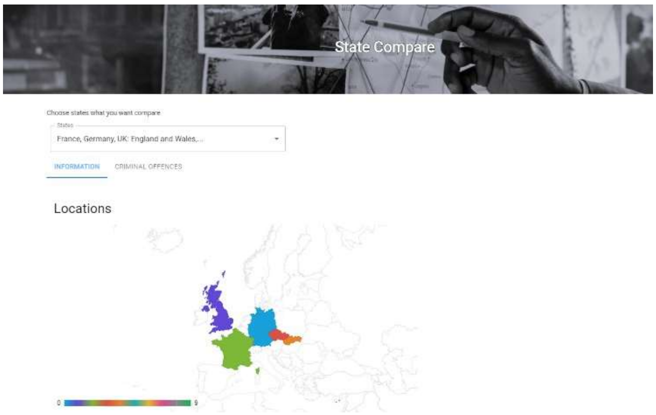
	name	Heroine (g)	Cocaine (g)	Cannabis (g)	Ecstasy (g)
	Albania	-	-	-	-
	Armenia	-	-	-	-
	Austria	3	15	20	30
	Azerbaijan	-	-	-	-
	Belgium			3	
	Bulgaria	-	-	-	-
	Croatia	-	-	-	-
	Cyprus	10	10	30	20
	Czech Republic	1.5	1	10	0.4
	Denmark				

1–10 of 43 ‹ ›

Fig. 16. Drug limits comparison table

D. *State information comparison*

Another page on our website is focusing on comparing information about different countries. This feature allows users to select specific states they want to compare, and then a map displaying these countries is shown. On the map, users can see the locations of individual states and easily compare their geographical positions.

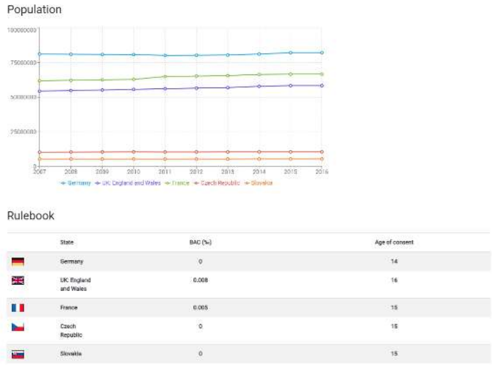


Choose states what you want compare

States: France, Germany, UK: England and Wales,...

Fig. 17. Map with location in state comparison page

In addition to the map, this page provides other information about the compared countries. Users can obtain statistical data about the population of each country and compare them with each other. There is also a rulebook available, which provides information about the applicable rules and laws in each state. Users can gain an overview of legal regulations that pertain to various areas such as road traffic, criminal law, and more.



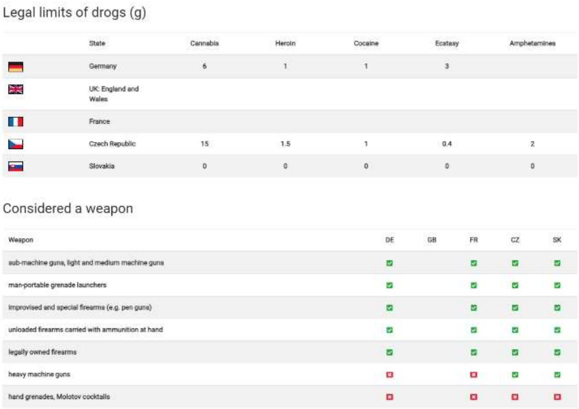
Rulebook

	State	BAC (‰)	Age of consent
🇩🇪	Germany	0	14
🇬🇧	UK: England and Wales	0.008	16
🇫🇷	France	0.005	15
🇨🇿	Czech Republic	0	15
🇸🇰	Slovakia	0	15

Fig. 18. Population and rulebook in state comparison page

Another useful feature is providing information about permitted legal weapons in individual states. Users can get an overview of the types of weapons that are allowed, and any potential restrictions or licenses associated with them. This information is valuable for those interested in firearms and who want to have an overview of their legality in different countries.

Furthermore, the website also provides information about drug limits in individual states. Users can learn about the quantity and types of drugs that are allowed in different countries and the potential legal consequences for their illegal possession or distribution.



Legal limits of drogs (g)

	State	Cannabis	Heroin	Cocaine	Ecstasy	Amphetamines
🇩🇪	Germany	6	1	1	3	
🇬🇧	UK: England and Wales					
🇫🇷	France					
🇨🇿	Czech Republic	15	1.5	1	0.4	2
🇸🇰	Slovakia	0	0	0	0	0

Considered a weapon

Weapon	DE	GB	FR	CZ	SK
sub-machine guns, light and medium machine guns	☑		☑	☑	☑
man-portable grenade launchers	☑		☑	☑	☑
improvised and special firearms (e.g. pen guns)	☑		☑	☑	☑
unloaded firearms carried with ammunition at hand	☑		☑	☑	☑
legally owned firearms	☑		☑	☑	☑
heavy machine guns	☒		☒	☑	☑
hand grenades, Molotov cocktails	☒		☒	☒	☒

Fig. 19. Legal limits of drugs and weapons allowed in state comparison page.

E. *State crime indexes comparison*

Another important page for our website is state crime activity comparison. On this page we can see states and their criminal offences counts and compare it.

This page allows users to select specific countries for comparison, and it will display graphs comparing the number of criminal offenses and crime indexes for each year.

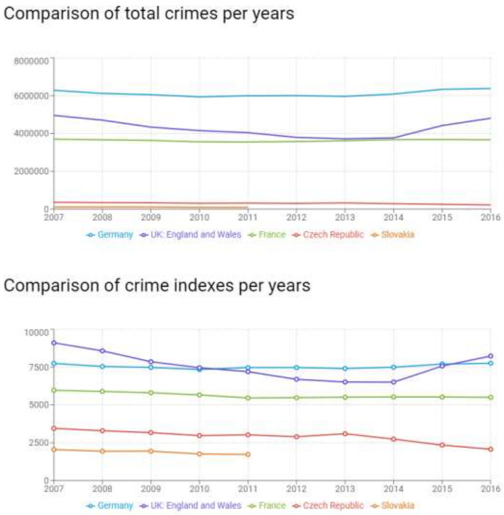
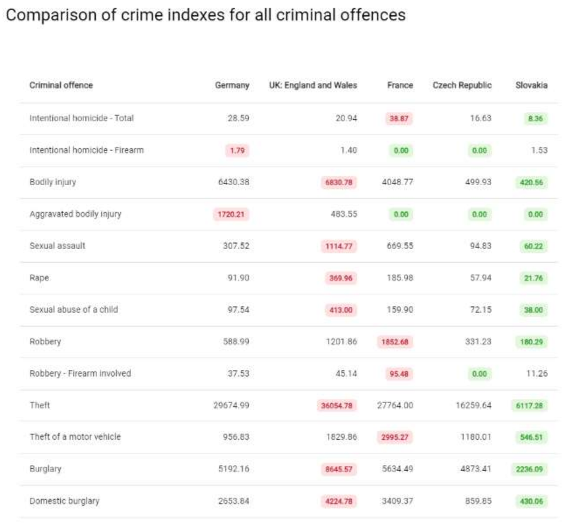


Fig. 20. Graphs with total count and crime index

In addition to the graphs, this page also includes a table that lists all types of criminal offenses and their crime indexes for each country. The best index (smallest value) is highlighted in green, while the worst index (highest value) is highlighted in red.



Comparison of crime indexes for all criminal offences

Criminal offence	Germany	UK: England and Wales	France	Czech Republic	Slovakia
Intentional homicide - Total	28.59	20.94	38.87	16.63	8.36
Intentional homicide - Firearm	1.79	1.40	0.00	0.00	1.53
Bodily injury	6430.38	6830.78	4048.77	499.93	420.56
Aggravated bodily injury	1720.21	483.55	0.00	0.00	0.00
Sexual assault	307.52	1114.77	669.55	94.83	60.22
Rape	91.90	369.96	185.98	57.94	21.76
Sexual abuse of a child	97.54	413.00	159.90	72.15	38.00
Robbery	588.99	1201.86	1852.68	331.23	180.29
Robbery - Firearm involved	37.53	45.14	95.48	0.00	11.26
Theft	29674.99	36054.78	27764.00	16259.64	6117.28
Theft of a motor vehicle	956.83	1829.86	2995.27	1180.01	546.51
Burglary	5192.16	8645.57	5634.49	4873.41	2236.09
Domestic burglary	2653.84	4224.78	3409.37	859.85	430.06

Fig. 21. Table with types of crime offences and highlighted indexes

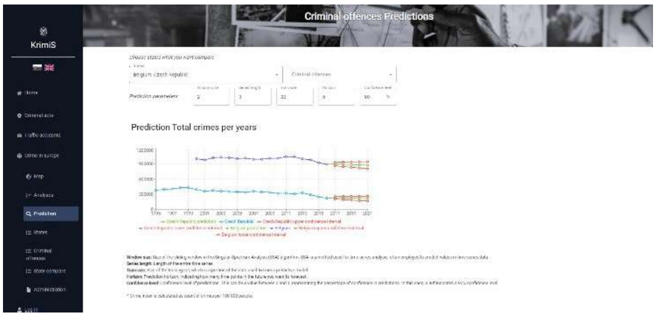
F. *Criminal offence indexes prediction*

Our next page is focused on predictions. It allows users to input state or more states and get prediction for type of crime offence, which he can choose in filter.

Predictions are made for 5 years, we used ml.NET library for forecasting. It allows us also to add confidence level interval.

In statistical terms, a confidence level of 90 percent implies that if we were to repeat the prediction process more times, 90 percent of the final intervals would contain the true value.

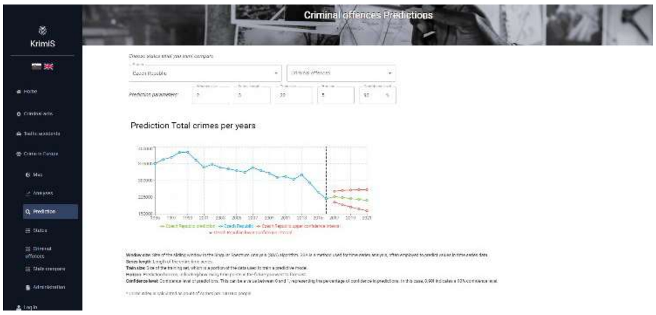
To set up the prediction, we allow the user to choose parameters of prediction. The first parameter is the window size. This parameter determines the size of the moving window used for prediction calculations. The second parameter is series length, where users can specify the length of the time series data used for training the prediction model. The third parameter is train size, which indicates the proportion of the dataset used for training the prediction model. Horizon defines the time horizon into the future for which predictions are made. The last parameter is confidence level, which will be shown in the chart.



Choose states what you want compare

Prediction parameters: 5 | 5 | 22 | 5 | 90 %

Fig. 22. Prediction of multiple states



Criminal offences Predictions
KrimiS
Home
Criminal acts
Traffic accidents
Crime in Europe
Map
Analyses
Prediction
States
Criminal offences
State compare
Administration
Log in
Prediction Total crimes per years
225000
150000
Czech Republic prediction
Czech Republic
Czech Republic upper confidence interval
Czech Republic lower confidence interval

Choose states what you want compare

Czech Republic

Criminal offences

Prediction parameters:

Window size: Size of the sliding window in the Singular Spectrum analysis (SSA) algorithm. SSA is a method used for time series analysis, often employed to predict values in time series data.
Series length: Length of the entire time series.
Train size: Size of the training set, which is a portion of the data used to train a predictive model.
Horizon: Prediction horizon, indicating how many time points in the future you want to forecast.
Confidence level: Confidence level of predictions. This can be a value between 0 and 1, representing the percentage of confidence in predictions. In this case, 0.90 indicates a 90% confidence level.

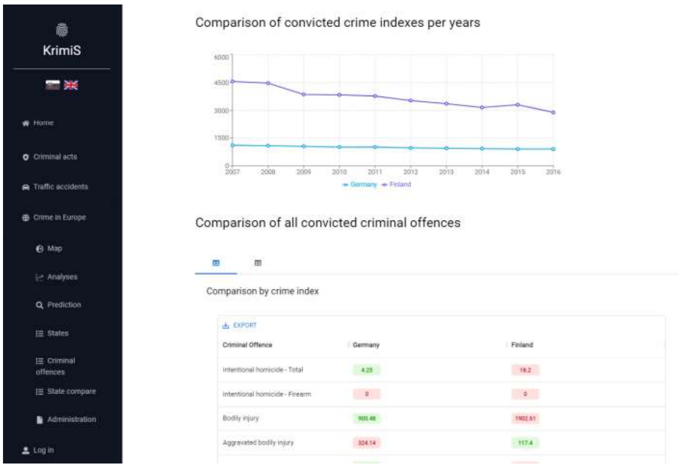
Fig. 23. Prediction of one state

G. *Convicted crime data analyses*

One of the significant components of the web application is a module dedicated to convicted crimes, which processes data and generates detailed reports on crime for a specific state in a given period. These reports offer a comprehensive view of the structure and extent of criminal activities, recording their development and trends.

The created reports have a flexible structure that allows users to dynamically filter data by type of crime and year. The cornerstone of these reports is the Index of Convicted Crimes, providing a normalized value of convictions per 100,000 inhabitants, enabling comparisons of criminality across different states and tracking changes over time.

Additionally, the reports include various tables and graphs that further support the analysis and understanding of crime data. Therefore, this application represents a powerful tool not only for legal and criminology professionals but also for the public interested in issues of justice and security.



Comparison of all convicted criminal offences

Comparison by crime index

Criminal Offence	Germany	Finland
Intentional homicide - Total	4.25	18.2
Intentional homicide - Firearm	0	0
Bodily injury	900.48	1902.61
Aggravated bodily injury	324.14	117.4

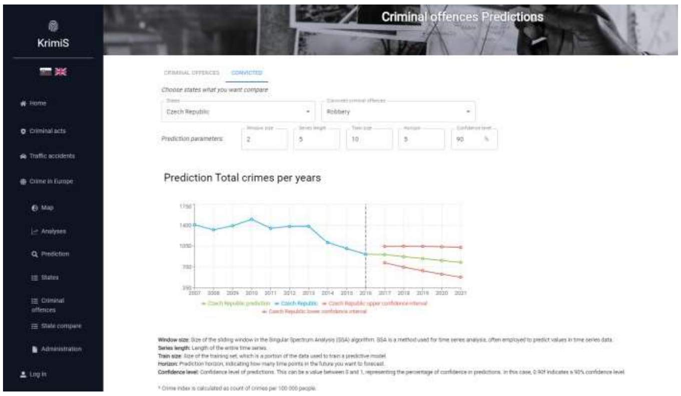
Fig. 24. Sample report of convicted crimes comparing the states of France and Germany

H. *Convicted Crime Offenses Data Prediction*

This page is focused on predicting Convicted Crime Offenses. It functions similarly to Criminal Offense Index Prediction. It allows us to use the same parameters, and we can build our own prediction model. These predictions can be a powerful tool for data analysts who know how to set up the model, providing valuable insights into convicted crime trends within the data. Additionally, we have enabled the ability to filter data by crime offense, allowing users to focus only on the fields they are interested in.

Similarly to the Criminal Offense Index Prediction, this page also offers the capability to compare convicted crime trends across multiple states.

The advantages of using this prediction tool extend beyond mere forecasting. By understanding convicted crime trends, analysts can better allocate resources, devise preventive measures, and inform policy decisions to address specific criminal activities effectively.



Choose states what you want compare

States: Czech Republic

Convicted criminal offences: Robbery

Prediction parameters: Window size 2, Series length 5, Train size 10, Horizon 5, Confidence level 90 %

Window size: Size of the sliding window in the Singular Spectrum Analysis (SSA) algorithm. SSA is a method used for time series analysis, often employed to predict values in time series data.
Series length: Length of the entire time series.
Train size: Size of the training set, which is a portion of the data used to train a predictive model.
Horizon: Prediction horizon, indicating how many time points in the future you want to forecast.
Confidence level: Confidence level of predictions. This can be a value between 0 and 1, representing the percentage of confidence in predictions. In this case, 0.90 indicates a 90% confidence level.

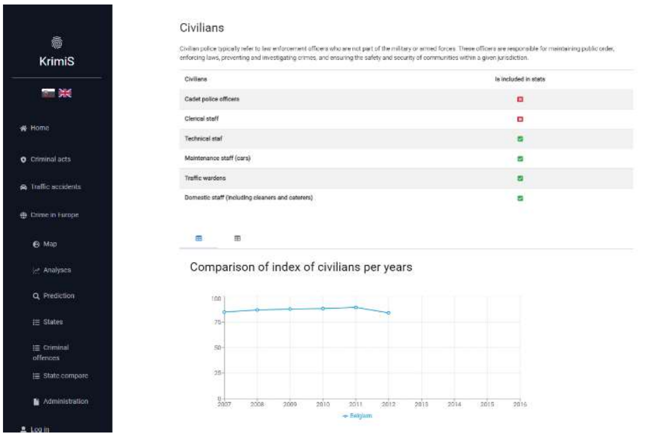
* Crime index is calculated as count of crimes per 100 000 people.

Fig. 25. Prediction of robbery convicted crimes in the Czech Republic

I. Police forces analyses

Our web application offers users the capability to compare or display counts of state police officers and civilians. On the state detail page, we provide a subpage dedicated to police forces, where users can access detailed counts over the years. Similar to other datasets, this feature allows users to view counts in both tables and charts, which are exportable. Additionally, the page provides a table detailing the types of police officers included in the counts, clarifying the composition of the state's law enforcement personnel.

Moreover, our application includes a comparison feature that compare civilian counts with police officer counts. This comparison sheds light on the ratio of civilians to police officers within each state, providing valuable insights into law enforcement resource allocation and community-police relations. By understanding this ratio, users can assess the balance of law enforcement personnel and civilian population, which can inform decisions regarding staffing levels, resource distribution, and community engagement initiatives.



Civilians

Civilian police typically refer to law enforcement officers who are not part of the military or armed forces. These officers are responsible for maintaining public order, enforcing laws, preventing and investigating crimes, and ensuring the safety and security of communities within a given jurisdiction.

Civilians	Is included in stats
Cadet police officers	☒
Clerical staff	☒
Technical staf	☑
Maintenance staff (cars)	☑
Traffic wardens	☑
Domestic staff (including cleaners and caterers)	☑

Fig. 26. Page showing civilians count in years in Belgium

J. Export of reports

Recognizing the importance of comprehensive data access, we've expanded our capabilities to include exporting page in PDF format. This feature enhances users' experience by allowing them to save and utilize visualizations beyond the confines of the platform.

In addition to generating reports, our platform now empowers users to seamlessly export visualizations, such as graphs and maps, in PDF format. This functionality enables users to retain and access visual data for future reference, ensuring that insights can be revisited and shared with ease.

IV. Planned Features of the System for Future Development

A. Incorporation of new data in the future

As part of our ongoing commitment to providing the most up-to-date information, we plan to incorporate new crime data as soon as it becomes available. This includes promptly integrating newly published crime statistics and relevant datasets to ensure that users have access to the latest information for their analysis and decision-making processes.

B. Incorporation of data on the state of prisons

Another part, where the web application could be expanded is data on the state of prisons.

When we think of prison, we imagine a guard place where criminals serve their prison sentences, and the accused undergo detention.

Such data would add a new dimension to the already existing data in terms of the number of links in individual countries and would enable another analytical dimension interesting from the user's point of view.

From such data, we would allow users to create new reports, also with normalized data.

Like the previous expansion, the data and system would need to be supplemented with new parts of the application.

C. Global filtering and save of the filter

A very useful functionality of the system would be the global use of filtering and the saving of such filters. We would allow all logged in users on the site to save filters like this and it would work as follows. The user would choose to use a global filter, if such a filter were selected, then changes in the filter of one subpage would also be reflected in other subpages of the system, which would clearly simplify the use of the system and make the system more user-friendly. In addition, if it were a user with an account, he could save such a filter in his saved filters, which would allow him to see these filters after logging into the application again, and he could simply select them again and use them again.

Another useful functionality of the use of filters could be that if the user has saved filters, he could download individual selected reports of the states he would like and already have them filtered according to his requirements.

D. Implementation of authorization

Since the current system does not include authorization, it is necessary to address this area as soon as possible. We would like to resolve the authorization area using a standard username and password method. After logging in, the user will be allowed to view administrative pages that would allow modification of data in the database - their addition, deletion, or modification. The system would thus be fully ready for deployment even with administrative pages.

For regular users, we would like to add the possibility to register and then authorize outside the administrative part of the application. Such an account would allow, for example, the storage of a filter, reports or other functionalities added over time.

E. Administration for individual states

After adding authorization, it will be convenient to expand the system with administration for individual countries. This means that each country would have its own administrator (instead of a global admin), who would secure data from his country, prepare it and upload it to the system.

This extension would ensure a fully functional stand-alone system for all countries, which would be very beneficial when putting such a system into practice. The administration would be simpler than from the point of view of a general global administrator, and the responsibility for the validity of the data would be taken by the administrator of the given country.

F. Interoperability with existing statistical systems

As mentioned, the proposed information system was created to obtain, store and analyse large amount of data interesting for public administration and other authorities responsible for keeping safety of served population. Therefore, the information system should provide its users also with advanced and sophisticated tools based on data analysis. One of possible ways of incorporating such features consists in interoperability with existing software like Matlab or many other. These systems usually offer different free API, which can be used to extend our system. Su summarize it, this could be one of directions the future development of our information system should follow.

V. Conclusion

The development and implementation of the crime offence page for Europe represent a significant step to understand crime data in the Europe region. The page provides a platform for accessing crime data, statistics, and reports, enabling users to get valuable insights into the state of criminality in European countries. The website offers many features, such as the drug analysis tool, comparison of crime rates and indexes, and country information, and the usability and relevance of the page. By utilizing advanced technologies, the page offers a valuable resource for policymakers, researchers, and the public in making decisions, formulating effective strategies, and addressing the challenges associated with crime. As the page continues to evolve and expand, it has the potential to contribute significantly to the field of crime monitoring and prevention in Europe, ultimately making safer and more secure societies.

As far as the future research directions are concerned, the development of existing information system will continue by incorporating several advanced statistical tools and solving those issues that raise from users of the system or the business community.

Acknowledgment

Presented research was supported by the Erasmus+ project: Project number: 2022-1-SK01-KA220-HED-000089149, Project title: Including EVERyone in GREEN Data Analysis (EVERGREEN) funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the Slovak Academic Association for International Cooperation (SAAIC). Neither the European Union nor SAAIC can be held responsible for them.

Co-funded by the European Union

EverGreen DATA Analytics

References

[1] Abhinivesh, A., Mahajan, N.: The Cloud DBA-Oracle, Apress, 2017

[2] Anders, L.: Cloud computing basics, Apress, 2021

[3] Cunningham, T.: Sharing and Generating Privacy-Preserving Spatio-Temporal Data Using Real-World Knowledge, 23rd IEEE International Conference on Mobile Data Management, Cyprus, 2022.

[4] Greenwald, R., Stackowiak R., and Stern, J.: Oracle Essentials: Oracle Database 12c, O'Reilly Media, 2013.

[5] Hansen, K.: Practical Oracle SQL: Mastering the Full Power of Oracle Database, Apress, 2020

[6] Idreos, S., Manegold S., and Graefe, G.: Adaptive indexing in modern database. In: ACM International Conference Proceeding Series, 2012

[7] Kuhn, D. and Kyte, T.: Expert Oracle Database Architecture: Techniques and Solutions for High Performance and Productivity. Apress, 2021.

[8] Kuhn, D. and Kyte, T.: Oracle Database Transactions and Locking Revealed: Building High Performance Through Concurrency, Apress, 2020.

[9] Kvet, M.: Developing Robust Date and Time Oriented Applications in Oracle Cloud: A comprehensive guide to efficient date and time management in Oracle Cloud, Packt Publishing, 2023, ISBN: 978-1804611869

[10] Kvet, M., Papán, J.: The Complexity of the Data Retrieval Process Using the Proposed Index Extension, IEEE Access, vol. 10, 2022.

[11] Lewis, J.: Cost-Based Oracle Fundamentals, Apress, 2005.

[12] Liu, Z., Zheng Z., Hou, Y. and Ji, B.: Towards Optimal Tradeoff Between Data Freshness and Update Cost in Information-update Systems, 2022 International Conference on Computer Communications and Networks (ICCCN), USA, 2022.

[13] Roske, E., McMullen, T., et. al: Look Smarter Than You Are with Oracle Analytics Cloud Standard Edition, Lulu.com, 2017

[14] Shanbhag, S.: Oracle Cloud Infrastructure 2023 Enterprise Analytics Professional, 2022

[15] Steingartner W., Eged, J., Radakovic, D., Novitzka V.: Some innovations of teaching the course on Data structures and algorithms, In 15th International Scientific Conference on Informatics, 2019.

[16] Su S.Y.W., Hyun S.J. and Chen, H.M.: Temporal association algebra: a mathematical foundation for processing object-oriented temporal databases, IEEE Transactions on Knowledge and Data Engineering, vol. 4, issue 3, 1998.

[17] Yao, X., Li, J., Tao, Y. and Ji, S.: Relational Database Query Optimization Strategy Based on Industrial Internet Situation Awareness System, 7th International Conference on Computer and Communication Systems (ICCCS), China, 2022.

[18] Erasmus+ project EverGreen dealing with the complex data analytics: https://evergreen.uniza.sk/

Comparative Study of Various Function Definitions Used in Oracle SQL Dialect

Michal Kvet
Department of Informatics, Faculty of Management Science and Informatics
University of Žilina
Žilina, Slovakia
Michal.Kvet@fri.uniza.sk

***Abstract*— SQL language is a non-procedural language with the defined statements and available clauses. Many times, conditions and format of the output depend on the functions. In this paper, Oracle relational database system (RDBMS) is used. Functions to be called are commonly defined in the PL/SQL, consequencing in context switches necessity between the SQL and PL/SQL environments. This paper deals with the various techniques for defining functions to be used in SQL statements. It provides an methodology for defining functions. For the performance evaluation study, temporal conversion functions are covered, focusing on the introduced SQL macro.**

Keywords—function definition, SQL macro, context switch, temporal database, Oracle

I. Introduction

Structured Query Language (SQL) is an important part of the data processing. Today we can no longer imagine an information system without data processing. Thus, the data layer, proper modeling and data management is a crucial part of the information technology. Over the decades, various models, enhancements and normalization techniques were proposed to ensure performance [6] [9]. Processing costs and time demands are the most significant and key factors influencing overall performance. It is evident, that the data amount is still rising and such an expansion is exponential [10].

Online Transaction Processing (*OLTP*) systems are characterized by the data changes, which must be handled. Thus, the focus is not only on the data retrieval, but also Insert, Update and Delete operations. In contrast to that, *Online Analytical Processing* (*OLAP*) is delimited by the data analysis as a keyword meaning, that large data sets are encapsulated by the complex analytically oriented queries. The normalization process is not so strict in that case to focus on the data retrieval process [4] [11].

In the context of the data processing and retrieval, SQL language is used in relational databases. It is a non-procedural language, consisting of various clauses (*Select, From, Where, Group By, Having* [8]) specifying the formats and conditions applied to the data to form the result set. User does not specifies, how to get the data, data access techniques nor the physical location of the data. All these aspects and tasks must be done by the database, currently defined by the cost-based approach [9]. The aim of the optimizer is to minimize the costs, processing time and response and to maximize the performance and throughput. When dealing with the data retrieval, it is worth mentioning functions, which can be called during the processing. Generally, function calls can be placed in any clause. Although there are many functions available in the database systems. In some database systems, like Oracle Database, functions can be called even automatically, without explicit user definition [6]. The reason is associated with the data types and format conditions. Implicit conversions change the original data type to be applied in the condition or to pass the requirements of the function calls. One way or another, individual functions are defined by the procedural extension of the SQL. Individual database systems have their own approaches and names for the procedural language. For Oracle Database, PL/SQL is used, as a bridge for building anonymous blocks executing the code only once, procedures, and functions, which can be optionally enclosed by the packages.

As evident from the definition, procedural language behaves differently compared to the SQL language. Considering that, it is vital to optimize the functions to be called from the SQL environment.

The aim of this paper is to provide the methodology for dealing with the function calls in the SQL environments. Various techniques and concepts are proposed, discussed and evaluated, to optimize the performance. Since it is part of the bigger project, the focus is done on the Oracle database. Therefore, we do not focus on database technologies in general, but emphasize a specific system. There are several reasons for this decision. Firstly, Oracle Database is a most complex systems and from the function definition perspective, it provides the most robust solutions. It can limit context switches and optimize the function for the SQL calling environment very easily and efficiently. Besides, these function results can be cached, if the function body is deterministic, limiting the necessity to execute the function with the same parameters multiple times [1] [2]. The validation of the cached results is done automatically ensuring that if the content of the function is changed, pre-stored data are automatically flushed away.

Secondly, it shares and propagates SQL macros, which encapsulate the code block directly in the SQL definition. Thirdly, Oracle Database provides scalable databases provisionable in Oracle Cloud Infrastructure characterized by the availability domains, automated administration and accessibility [1] [2]. Fourthly, it provides robust, complex and user-friendly environment for the data analytics in the cloud environment [14]. It forms an output of the Erasmus+ project EverGreen focusing on the environmental data analysis.

Finally, data in the Oracle Database can be enhanced by the data-driven applications placed directly on the data layer, which can significantly reduce data transfer demands and thus, significant performance improvements can be reached.

For the purposes of this paper temporal conversion functions are discussed, pointing to creating function definition management methodology. The data structure and temporal layer optimization [5] [12] [13] must be done, when dealing with the complex time evolving data [7].

The paper is structured as follows. Section 2 deals with the procedural language definition summary. Section 3 discusses

979-8-3503-8768-1/24/$31.00 ©2024 IEEE

context switches and compilation optimizing function to be primarily used in SQL. Section 4 points to the SQL macros. Performance evaluation study of the temporal conversion functions from MySQL to Oracle Database is present in section 5.

II. PL/SQL CODE BLOCKS

PL/SQL je structured language forming code into code blocks executing from the top. It combines the power of SQL with procedural language. All the statements of the block are passed to the engine at once, limiting the data transfer and operations. It offers wider opportunities for identifying and capturing exceptions. The block itself consists of the body, which is mandatory, optionally enhanced by the exception handler. Local variables are defined in the declaration section, starting with the *Declare* keyword up to the beginning of the content body. The body itself cannot be empty [11].

```
[DECLARE
    Variable declaration;]
BEGIN
    Statements & commands;
[EXCEPTIONS
    exception handlers;]
END;
/
```

PL/SQL code can be used in SQL in form of the functions, passing the following prerequisites [11]:

- the parameters and return value data type must be recognizable in SQL,
- function cannot influence the transaction itself by invoking transaction end explicitly or by using command, which considers transaction end internally (like DDL statements),
- cannot change the content of the table, from which the function is called.

The syntax of the function is following:

```
CREATE [OR REPLACE] FUNCTION function_name
 [( parameter1 [ mode1 ] datatype1,
    parameter2 [ mode2 ] datatype2, ... )]
RETURN datatype
IS|AS
  [ variable_name data_type [:=
init_value]; ]
BEGIN
  statements;
  RETURN expression;
  [ EXCEPTION
      WHEN exception_type1 THEN
        statements;
      [WHEN ...]
  ]
END [function_name];
/
```

If the function is deterministic (the provided result directly depend on the input parameters, by calling the function at any time, the same results are provided), the results can be cached in the instance memory.

To get the day element from the *Date* value, *Extract* function can be used in RDBMS Oracle. In contrast, *Day* function is used in MySQL. To make the migration smooth, *Day* function must be defined in RDBMS Oracle, as well. Particular function can be implemented in the following way:

```
CREATE OR REPLACE FUNCTION Day
                          (date_in Date)
  RETURN varchar IS
BEGIN
  RETURN extract(day from date_in);
END;
/
```

III. CONTEXT SWITCHES, USER DEFINED FUNCTIONS

Database code blocks are primarily intended to be used and referenced in the PL/SQL environment. Many times, *Select* statements are hidden behind the functions, evaluation, constraints and application rules, too. Sure, the code can be called from any position making it reusable [3] [11]. Furthermore, if the parameters are present and function is deterministic, results can be cached. However, if the function is called from the SQL environment, context switches are present between SQL and PL/SQL environment. Context switch is a mechanism for sharing system resources. PL/SQL engine passes the SQL statement with the bind variable values over the SQL engine, which evaluates that statement and passes provided result set back to the PL/SQL engine for the consecutive processing [11]. Analogously, if the function is called from the SQL environment, context swich must be present to provide PL/SQL engine to process the function and provide results, which are then used for the query processing. Generally, functions, which are directly bundled in RDBMS are generally optimized for PL/SQL usage. Thus, to optimize performance and reduce context switches, own functions must be developed shifting the processing directly to the SQL.

When dealing with the system migrations over various database systems, such a problem is even deeper. Namely, each database uses its own dialect, own function names, parameters and implementation. Code from one system cannot be directly run on another and many conversions need to be done [8] [9].

The pragma *UDF* navigates the compiler, that the function will be primarily used in SQL statements. This definition can improve performance of the SQL statements, however, if the function will be used in PL/SQL, as well, processing cost penalty would be present. Although such a pragma has been introduced more than ten years ago (in Oracle 12c), developers are not commonly aware of that and do not use that option resulting in adding many context switches. In an analytical environment, the overhead caused by the context switches can be enormous.

As already stated, this paper deals with the temporal functions, which are critical within the migration process making the code immediately runnable.

The function compiled for the SQL usage can look like following. Internally, it is treated as an inline function embedded to the statement itself, but making it generally available through the function name.

```
CREATE OR REPLACE FUNCTION Day
                          (date_in Date)
  RETURN varchar
     IS PRAGMA UDF;
BEGIN
  RETURN extract(day from date_in);
END;
/
```

IV. SQL MACRO

SQL macro is a PL/SQL function that returns SQL snippet, which is then inserted directly to the SQL statement at the beginning of its execution – even before the statement is parsed. Thanks to that, hitting context switches is no longer performance problem [11].

SQL macro looks very similar to the conventional functions defined in PL/SQL. However, rather than performing an action during the statement execution, the action occurs directly during the query optimization. The code substitution is done before executing the query to centralize the functionality without the costs of the context switching. Generally, table or scalar macros can be defined, but for the purposes of temporal function management used during the migration, only scalar functions are relevant.

Scalar type of SQL macro returns a piece of SQL text, which results in a scalar value. It cannot have table arguments, only scalar values are applicable.

For getting the day element extracted from the Date value, SQL macro can be implemented this way:

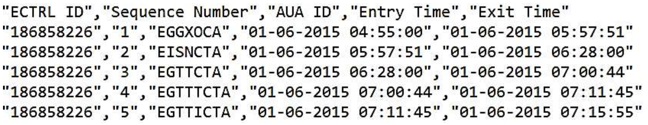
```
CREATE OR REPLACE FUNCTION Day
                          (date_in Date)
  RETURN varchar sql_macro(scalar) is
BEGIN
   RETURN q'[extract( day from date_in )]';
END;
/
```

Expand_sql_text procedure of the *Dbms_utility* package lets you expand the statement, which will actually be carried out. The function call will be replaced by the *Extract* function call:

```
-- Original statement
... where Day(entry_date) = 'Sunday';
-- Preprocessed statement
... where extract(day from entry_date)
                                = 'Sunday';
```

V. PERFORMANCE EVALUATION STUDY

The data set of flight monitoring in European region was used for the computational study. It consists of the positions of the airplanes and individual flight parameters. The whole flight is monitored over the whole journey starting from the departure airport, taxi, take-off up to parking position on the arrival airport. The data set consists of 5 million of rows. The position of the aircraft itself is assigned to the *Flight Information Regions* (*FIRs*) depicting entry and exit time. The data set consists of 1000 rows for the FIR assignment. The structure of the data set is shown in Fig. 1. To follow and evaluate the flight efficiency, expected and real route are recorded. When dealing with the FIRs, it is necessary to highlight, that the borders of the FIR are not static, but can evolve over time, forming the temporal database.



```
"ECTRL ID","Sequence Number","AUA ID","Entry Time","Exit Time"
"186858226","1","EGGXOCA","01-06-2015 04:55:00","01-06-2015 05:57:51"
"186858226","2","EISNCTA","01-06-2015 05:57:51","01-06-2015 06:28:00"
"186858226","3","EGTTCTA","01-06-2015 06:28:00","01-06-2015 07:00:44"
"186858226","4","EGTTTCTA","01-06-2015 07:00:44","01-06-2015 07:11:45"
"186858226","5","EGTTICTA","01-06-2015 07:11:45","01-06-2015 07:15:55"
```

Fig. 1. Data Set.

The parameters of the server used for the evaluation are following:

- Processor: AMD Ryzen 5 PRO 5650U, 2.30 GHz, Radeon Graphics.
- Physical memory: Kingston, DDR4 type, 2x 32GB, 3200MHz, CL20.
- Storage repository: 2TB, NVMe disc type, PCIe Gen3 x 4, 3500MB/s for read/write operations.
- Operating system: Windows Server 2022, x64.
- Database system: Oracle Database 23ai, release bundle Oracle 23ai Free, Developer Release Version 23.2.0.0.0, Windows version.

Provided evaluation study models a migration process from a MySQL to Oracle Database, pointing to the temporal conversion functions. Tab. 1 shows the list of evaluated functions. Each function was created in a conventional PL/SQL environment, optimized using Pragma UDF clause and SQL macro.

TABLE I. LIST OF USED FUNCTIONS

Function name	Description
CUR_DATE	Provides current date and time values using local client time zone.
NOW	Gets current date and time of the server (cloud).
DAY	Extracts day element in a numerical format extracted from the provided date value (parameter of the function).
DAYNAME	Produces the name of the day in an English version format.
DAYOFWEEK_SESSION	Provides a numerical representation of the day highlighting the client territory (whether the first day of the week is Sunday or Monday).
DAYOFWEEK_GLOBAL	Provides a numerical representation of the day – the first day of the week is strictly defined as Monday.
MONTH	Provides a textual representation of the month.
WEEK	Gets a week number reference.
YEAR_N	Produces a numerical representation of the year.
YEAR_T	Produces a textual representation of the year.
FORMAT	Gets the output in a defined format (default 'DD/MM/YYYY').

For the evaluation, three function architectures and calls were done. The first solution (*SOL1*) is a reference and is defined by the conventional approach defining function directly in PL/SQL. The second solution (*SOL2*) emphasizes shift and optimization for the SQL call using *PRAGMA UDF*. The last solution (*SOL3*) points to the newly introduced SQL macros. Compared to the original PL/SQL approaches, it produces a content definition, which is deterministic and directly embeddable in the query. The difference between *SOL1* and *SOL2* expresses the context switch impacts. The complete limitation of the PL/SQL core environment is done by SQL macro. The obtained results are shown in Tab. 2. Each function has been called 10 times for the whole position monitoring data set. The shown values express the average value for 10 experimental rounds. The last row expresses the total sum value. The values are expressed in second format (ss.ff).

TABLE II. PERFORMANCE RESULTS [SS.FF]

Function name	Conventional function	Pragma UDF	SQL macro
CUR_DATE	17.74	16.93	15.84
NOW	16.89	16.04	14.72
DAY	17.12	16.83	15.34
DAYNAME	17.65	16.34	15.92
DAYOFWEEK _SESSION	16.99	16.21	14.08
DAYOFWEEK _GLOBAL	16.30	15.87	13.36
MONTH	17.55	16.98	14.24
WEEK	17.24	16.70	13.99
YEAR_N	16.75	15.92	13.86
YEAR_T	17.44	16.84	14.31
FORMAT	21.04	17.12	16.62
TOTAL SUM	**192.71**	**181.78**	**162.28**

Most of the proposed functions reach almost the same results, varying up to 8.83%. From the reached results, it can be shown, that the resulting data type and format impact the performance. Namely, comparing numerical and textual representation of the *Year* function, the processing time demand difference is 4.11% for conventional approach, 5.78%, if Pragma UDF is used and 3.25%, if SQL macro is used. Similarly, the performance strongly depends on the session or server reference, which can be shown either in *CUR_DATE* and *NOW* function calls (the difference is 5.03% for conventional function, 5.55% for Pragma UDF and 7.61% for SQL macro) or by referring to the *DAYOFWEEK* function, which can point to the server or client territory and *National Language Support* (*NLS*) parameters. Namely, session reference impacts additional 4.23% for conventional approach, 2.14%, if function is optimized for SQL call using Pragma UDF and 5.39% for SQL macro. Always, server side reference provides better performance. Graphical representation of the results is in Fig. 2.

Overall, considering these three solutions by the total sum, conventional function requires 192.71 seconds, while Pragma UDF takes only 181.78 seconds. Thus, the context switches requires 10.93 seconds, which impacts more than 5.67% of the overall processing. SQL macro provides the best solution, the total processing time demands are significantly lowered by 15.78%, which is physically expressed by more than 30 seconds (referencing conventional function) and 19.5 seconds (making UDF reference).

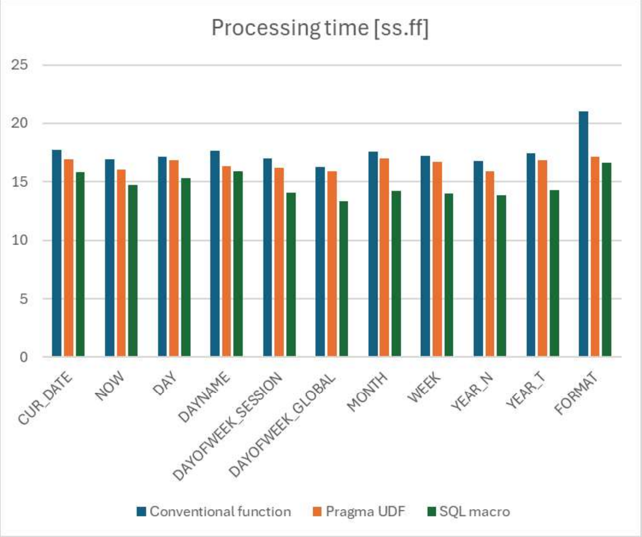


Fig. 2. Results – individual operations

The reached results in a graphical form are depicted in Fig. 3.

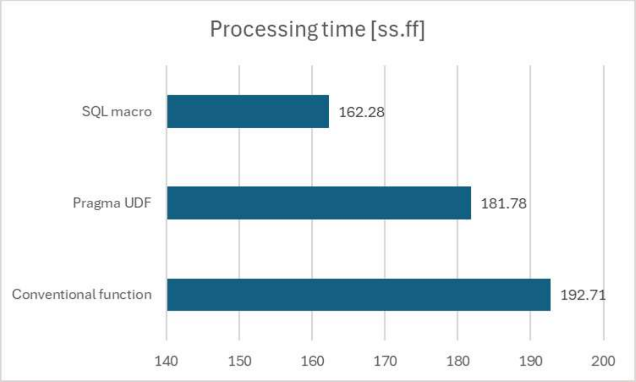


Fig. 3. Results – total sum

Fig. 4 shows the scalability of the solution. Based on the results, the impacts on the performance is almost linear for conventional and UDF function. With the increase in size of the date set, differences between the performance are more and more significant. For example, referring to the 500 million of rows, the context switches delimit 484.81 seconds, expressing 10.04%. Considering SQL macro, the improvement compared to conventional approach expresses 27.11%. By referencing UDF function, an reached improvement is 18.98%. Conversely, if the data set size is small, although there are performance differences, they are minimal and not as impactful.

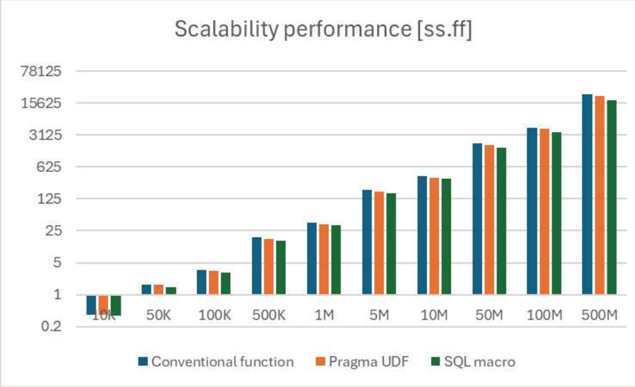


Fig. 4. Results – Scalability

VI. Conclusions

Data analysis is an inseparable part of the information systems. Reports, statistics and complex data analytics need to be provided to ensure proper decision-making, optimizing resources, consumption and last but not least, the overall performance of the system. When dealing with the data analysis, it is always worth mentioning SQL environment providing language for the relational data manipulation. Even in non-relational databases and big data concepts, the core part of the relational theory and SQL extensions are always present.

When dealing with the data processing and data management, functions are used for conditions and output formatting. They are commonly coded in procedural language, for RDBMS Oracle, PL/SQL is used. Regardless of the database system used, there are two environments – SQL on one hand and procedural on the other, requiring to switch between them. Such a context switch can significantly impact the performance of the queries. When the data sets to be handled are big, the problem is even deeper. Based on the proposed peformance study, more than 5% of the query processing can be devoted to the context switch for only one function reference. If the query required multiple function calls, or the parameters themselves depended on the result of the function, the performance would be significantly affected and the system could easily degrade.

This paper deals with the various concepts of function definitions and emphasizing calls in SQL environment. A newly introduced SQL macro provides the best performance. Based on the computational study, total demands are reduced by more than 15%.

In the future research, we will emphasize not only processing time, but also the server resource consumptions and overall costs. We will also focus on index structures incorporating functions, function result caching, as well as SQL macro used as a data source (considering tabular SQL macro).

Acknowledgment

It was supported by the Erasmus+ project: Project number: 2022-1-SK01-KA220-HED-000089149, Project title: Including EVERyone in GREEN Data Analysis (EVERGREEN) funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the Slovak Academic Association for International Cooperation (SAAIC). Neither the European Union nor SAAIC can be held responsible for them.

Co-funded by the European Union

EverGreen DATA Analytics

This paper was also supported by the VEGA 1/0192/24 project - Developing and applying advanced techniques for efficient processing of large-scale data in the intelligent transport systems environment.

References

[1] A. Abhinivesh and N. Mahajan, “The Cloud DBA-Oracle,” Apress, 2017.

[2] L. Anders, “Cloud computing basics,” Apress, 2021.

[3] T. Cunningham, “Sharing and Generating Privacy-Preserving Spatio-Temporal Data Using Real-World Knowledge,” In 23rd IEEE International Conference on Mobile Data Management, Cyprus, 2022.

[4] S. Idreos, S. Manegold, and G. Graefe, “Adaptive indexing in modern database,” In ACM International Conference Proceeding Series, 2012.

[5] J. Janáček and M. Kvet, “Shrinking fence search strategy for p-location problems,” In 2020 IEEE 20th International Symposium on Computational Intelligence and Informatics (CINTI), Hungary, 2020.

[6] D. Kuhn and T. Kyte, “Expert Oracle Database Architecture: Techniques and Solutions for High Performance and Productivity,” Apress, 2021.

[7] M. Kvet, “Developing Robust Date and Time Oriented Applications in Oracle Cloud: A comprehensive guide to efficient Date and time management in Oracle Cloud,” Packt Publishing, 2023, ISBN: 978-1804611869.

[8] M. Kvet and J. Papán, “The Complexity of the Data Retrieval Process Using the Proposed Index Extension,” IEEE Access, vol. 10, 2022.

[9] J. Lewis, “Cost-Based Oracle Fundamentals,” Apress, 2005.

[10] M. Malcher, D. Kuhn, “Pro Oracle Database 23c Administration: Manage and Safeguard Your Organization’s Data”, Apress, 2024

[11] B. Rosenzweig, E. Rakhimov, “Oracle PL/SQL by Example,” Oracle Press, 2023.

[12] W. Schreiner, W. Steingartner, V. Novitzká, “A Novel Categorical Approach to Semantics of Relational First-Order Logic,” In Symmetry-Basel, vol. 12, issue 10, MDPI, 2020.

[13] W. Steingartner, J. Eged, D. Radakovic, V. Novitzka, “Some innovations of teaching the course on Data structures and algorithms, ” In 15th International Scientific Conference on Informatics, 2019.

[14] Erasmus+ project EverGreen dealing with the complex data analytics: https://evergreen.uniza.sk/

Transforming and Analysing Oceanographic Data with Oracle Analytics Cloud: Insights from the World Ocean Database

Robert Leskovar
Faculty of Organizational Sciences
University of Maribor
Kranj, Slovenia
robert.leskovar@um.si

Alenka Brezavšček
Faculty of Organizational Sciences
University of Maribor
Kranj, Slovenia
alenka.brezavscek@um.si

Alenka Baggia
Faculty of Organizational Sciences
University of Maribor
Kranj, Slovenia
alenka.baggia@um.si

Abstract— **The environment is under severe threat from climate change, pollution, deforestation and biodiversity loss. Addressing these challenges requires collecting and analysing environmental data to gain actionable insights. This paper explores the use of Oracle Analytics Cloud (OAC) to analyse data from the World Ocean Database (WOD), a comprehensive collection of oceanographic data. By converting the WOD data from its native netCDF format to CSV, we show how OAC can be used to access, prepare and visualize this data. The study highlights the potential of cloud-based analytics platforms to better understand oceanographic trends and support informed decision-making in marine science.**

Keywords—data analytics, environmental data, world ocean database

I. INTRODUCTION

The environment is under unprecedented threat from climate change, pollution, deforestation and biodiversity loss. To combat these problems effectively, it is crucial to collect and analyse environmental data in order to gain actionable insights. Publishing this data ensures transparency and enables a collaborative approach to solving environmental problems. This requires not only qualified experts in data analysis, but also efficient software tools to process and interpret large amounts of data. The Erasmus+ Evergreen project aims to address both environmental challenges and the lack of data analysis professionals by promoting education and collaboration in this area.

To showcase the use of open environmental data analytics and its benefits for teachers and students, a dataset from the Oracle Open Data platform was selected to be used for a case study of the Erasmus+ Evergreen project. Oracle Open Data is a free repository of scientifically relevant datasets from trusted sources, aimed at researchers, educators, data scientists and analysts. It provides access to a variety of datasets, including genomic, climate, AI/ML and other public datasets from institutions such as NASA, DeepMind and Stanford [1], [2]. Among others, the World Ocean Database (WOD) dataset is available on the Oracle Open Data platform.

This paper focuses on using Oracle Analytics Cloud (OAC) to unlock the potential of WOD and provides valuable insights into accessing, transforming and analysing this vast dataset. The integration of WOD with OAC involves several key steps, starting with the retrieval of data from Oracle Open Data. Given the complexity and scale of the WOD, the data must be converted from its original netCDF format to an OAC-compatible format such as CSV. This conversion process is crucial for the subsequent steps of data import, preparation and visualization in OAC.

In today's data-driven world, processing and analysing environmental data faces significant challenges due to the scale and heterogeneity of the data sets involved. Advanced platforms such as Oracle Analytics Cloud (OAC) enable the integration and analysis of these large datasets in real time and provide practical solutions for researchers and decision makers in the field of environmental studies. The main objective of this paper is to demonstrate the practical applications of such cloud-based platforms in environmental data analysis and how these technologies can improve the accessibility and visualisation of data. Although this work is aimed at a broad audience — including researchers, analysts, lecturers and students — its main objective is to promote the principles and benefits of environmental data analytics and to show how these platforms can be applied in practise.

II. RELATED WORKS

The integration of big data analytics into environmental and urban research has made considerable progress in recent years. The development of frameworks such as CityPulse, which utilises big data analytics for smart cities, demonstrates the importance of handling diverse data sets from IoT infrastructures, social media streams and sensor networks. CityPulse overcomes challenges such as data heterogeneity, velocity and uncertainty by integrating different data sources to provide smart city services such as traffic and pollution monitoring [3].

In the environmental field, big data analytics has helped to address the problem of air pollution. For example, [4] developed a hybrid ARIMA neural network model augmented by optimisation algorithms to monitor and predict air quality using real-time data. This demonstrated the power of machine learning in processing large data sets and making accurate predictions [4]. In water management, big data analytics was applied to transboundary aquifers in southern Africa to model groundwater levels and support sustainable water use. This approach uses remote sensing data and machine learning algorithms to analyse groundwater data with spatial and temporal complexity [5].

Furthermore, [6] emphasises the role of big data in mitigating climate change by using machine learning for predictive modelling and trend analysis in environmental systems. These technologies help to monitor biodiversity, water quality and pollution levels and provide important insights for urban planning and environmental policy [6], [7].

979-8-3503-8768-1/24/$31.00 ©2024 IEEE

Overall, the application of big data analytics to environmental management and urban systems is transforming decision-making processes by enabling real-time monitoring, prediction and optimisation. However, the integration of different types of data, computational challenges and the need for advanced algorithms still pose significant hurdles [8], [9].

The exponential growth of environmental data has led to the development of several powerful tools and platforms tailored to process large and complex data sets. Tools such as Hadoop and Spark have become an integral part of the environmental data analytics landscape due to their ability to process large amounts of data quickly and efficiently. Hadoop, with its distributed storage and MapReduce processing model, is particularly useful for big data storage and batch processing, making it suitable for processing heterogeneous environmental data from various sources such as satellite observations, IoT sensors and climate modelling [9]. Spark, on the other hand, is favoured for its speed and in-memory processing capabilities, which are crucial for analysing environmental data in real time and for iterative machine learning tasks [8].

Other tools such as Python and R, which are equipped with extensive libraries for data manipulation and visualisation (e.g. Pandas, TensorFlow, ggplot2), are commonly used in scientific research for tasks such as air quality prediction, climate trend analysis and biodiversity monitoring (Vir6). More specialised tools such as KNIME and RapidMiner offer user-friendly interfaces and the integration of various machine learning algorithms that allow researchers to create complex data workflows without extensive programming knowledge [8]. These tools are crucial for transforming raw environmental data into actionable insights that support decision-making in areas such as urban planning, climate adaptation and resource management.

In addition to tools such as Hadoop, Spark and KNIME, Oracle Analytics Cloud (OAC) provides a robust platform for processing and analysing large-scale environmental data. OAC integrates advanced analytics, machine learning and data visualisation in a cloud-native environment, enabling efficient data transformation and real-time insights. Its ability to process diverse data sets, such as those used in environmental monitoring or urban systems analysis, makes it a valuable tool for researchers working with complex, heterogeneous data. Thanks to its flexibility and scalability, the platform is particularly suitable for analysing large datasets such as oceanographic or climatic data, enabling better decision making and resource management [10].

III. World Ocean Database

The World Ocean Database (WOD) is a comprehensive collection of scientifically quality-controlled ocean profile and plankton data. It includes measurements of various oceanographic variables such as temperature, salinity, oxygen, phosphate, nitrate, silicate, chlorophyll, alkalinity, pH, pCO2, TCO2, Tritium, Δ13Carbon, Δ14Carbon, Δ18Oxygen, Freon, Helium, Δ3Helium, Neon, and plankton [11]. This extensive dataset is crucial for understanding and monitoring the state of our oceans.

For data analysts, the WOD provides a rich source of high-quality, standardized data that can be used for various analytical purposes, including climate modeling, environmental impact assessments and marine ecosystem studies. As explained in [11], the database contains 20,547 different data sets from 216,845 oceanographic cruises on 8,215 different platforms. It contains 3.56 billion individual profile measurements, including 1.95 billion temperature measurements, 1.13 billion salinity measurements and 260 million oxygen measurements. The availability of quality flags and metadata ensures the reliability and reproducibility of the analyses, making the WOD an invaluable resource for scientific research and data-driven decision-making. The data, stored in netCDF-4 format, covers a historical period from 1792 to 2021 and provides invaluable insights into oceanic conditions over time.

In the WOD, the data is organized as follows: A profile is a set of measurements for a single variable at different depths. A cast includes one or more profiles taken at the same time, together with other data such as meteorological measurements. A station is a specific location where casts are taken. A cruise is the deployment of a platform for oceanographic research, identified by a unique cruise number and country code. Data sets that are archived at the NCEI are assigned an access number. Finally, a WOD dataset combines similar data types, which are stored in separate files for convenience.

WOD divides the data into different data sets depending on the type of data collection. For example, bottle data and low-resolution casts are grouped together, while high-resolution conductivity, temperature and depth data are stored separately due to their size. Each dataset is identified by a three-letter notation. For our analysis, we used the Ocean Station Data (OSD) dataset, which focuses specifically on surface temperature measurements. This dataset includes low-resolution conductivity, temperature and depth data, bottle data and plankton measurements and provides a comprehensive overview of oceanographic conditions.

A. NetCDF – Network Common Data Format

Scientific data is often stored in files because they are easy to manage, transfer and share. These files are structured and contain metadata to describe the data. There are numerous file formats, such as HDF5, netCDF4 and Zarr, each developed for specific tasks. In the case of the World Ocean Database, the netCDF4 (NC) format is used [12].

HDF5, the Hierarchical Data Format 5, is a file format designed for the organized storage of large amounts of data. NetCDF4 is a file format for storing array-oriented data, characterized by the file extension .nc. It uses a similar structure to HDF5, with groups containing other groups or variables. Unlike HDF5 datasets, netCDF4 variables cannot be resized after their creation, but they can be declared with an unlimited size in a given dimension [13]. As a self-describing format, netCDF4 contains metadata for both groups and variables.

IV. Methodology

WOD data was retrieved from the Oracle Open Data platform. This dataset includes various oceanographic measurements such as temperature, salinity and oxygen content. Various tools were tested to examine the netCDF files, but many were unsuccessful due to the complexity of the WOD files. The WOD data, originally in netCDF format, was converted to CSV format using Python scripts. In this step, the relevant data was extracted, cleaned and converted into a format that is compatible with OAC. Rows with missing temperature data or incorrect date formats were excluded to ensure data quality.

The transformed data was uploaded to Oracle Cloud Infrastructure (OCI) for secure storage. OCI's object storage service was used to create a bucket called WOD where all raw data was stored. The cleaned data was imported into an autonomous transaction processing database within OCI to enable efficient data management and retrieval for subsequent analysis in OAC. A connection was established between OAC and the OCI database using a wallet for secure access. The dataset from the database table was then defined via this connection. Various visualizations were created in OAC to explore the oceanographic data. OAC's advanced AI-based tools, such as Auto Insights and Explain options, were used to further explore the data. These tools automatically identified and highlighted key patterns and trends to improve the decision-making process.

A. *Examining and Converting the NetCDF file*

The integration of WOD with OAC involves several important steps, starting with the retrieval of data from Oracle Open Data. Given the complexity and volume of the WOD, the data must be converted from its native netCDF format to an OAC-compatible format such as CSV. This conversion process is crucial for the subsequent steps of data import, preparation and visualization in OAC. Several challenges were encountered during this process and solutions were implemented to overcome them, including the use of Python scripts for data extraction and cleaning.

First, an attempt was made to examine the netCDF file using HDFView 2.11 for Linux (Kubuntu 22.04 operating system). Although HDFView worked properly with the sample data, the tool did not recognize the file format despite registering the netCDF file format. In the next attempt, ncdump was used under Linux, which was able to print the structure (dimensions, variables) and the raw data. Therefore, we used ncview, a tool which can create simple diagrams of variables but is not able to handle more complex data structures such as WOD files. Hexdump under Linux did not deliver any useful results either.

Due to the failures of using tools in a Linux environment, we decided to use the web-based viewer and the Ocean Data View software. Despite the initial promising information about the ability to use C++ and Java APIs, there was a lot of excitement due to the need to install a 64-bit version of Qt 5 (which is notoriously buggy and version-dependent) or read an extensive Java manual to develop the app. Furthermore, despite following the instructions to install (extract) the files, there was no success.

It was therefore decided to check sample programs on the Oracle Open Data website. The first attempt to use Fortran resulted in a segmentation fault runtime error. The attempt to use the wodSURF program was also unsuccessful.

There was a successful attempt to represent the data structure with Matlab and R, but the file was too complex to process with either tool.

After preparing the environment (plugins and libraries in Apache NetBeans), an example provided by [1] was used to process sea surface temperature data. Several changes were made to the sample Python program (e.g. different data sources, saving the data frame as a .csv file and creating a map). Finally, the modified program displayed a map of the average sea surface temperature measurements. It also created the text file with the required surface temperature data (cast, lat, lon, time, temperature_row_size and avg_value), while excluding rows with missing temperature data or incorrect date format. For convenience, the Jupiter Lab workbook was used instead of Apache NetBeans for the final extraction of the data.

V. Data analysis with Oracle Analytics Cloud

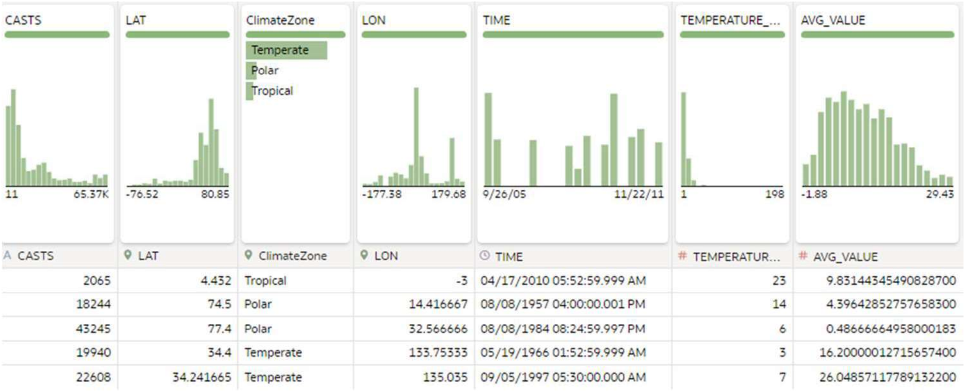
Once the data had been successfully extracted from Oracle Open Data and converted into a compatible format, it was uploaded to Oracle Cloud Infrastructure (OCI) for secure storage. OCI's Object Storage Service was used to create a bucket called WOD where all the raw data was stored. The visibility of this bucket was set to public to allow free access to the data.

The next step was to clean and prepare the data for analysis. This involved ensuring that the CSV files contained the correct headers and that the data was formatted correctly. The cleaned data was then imported into an autonomous transaction processing database within OCI, which enabled efficient data management and retrieval for subsequent analysis in OAC.

A connection was established between OAC and the OCI database using a wallet for secure access. The connection was used to define the dataset from the database table. Several changes were made to the dataset to enable smoother analysis. Firstly, the latitude and longitude were defined as locations, secondly, climate zones were defined based on the latitude, as shown in Figure 1.

Various visualizations were created in OAC to examine the oceanographic data. These visualizations include maps of measurement locations, average temperature trends over time, and analyses of temperature variations in different climate zones, as shown in the rows below. After the dataset was completed, the first visualization of the average sea temperature over the years was created. As this visualization was not very meaningful, the second visualization showing the monthly average temperatures over the years was added to the canvas (Figure 2).

Apart from the clear indication that the average temperatures increase in summer, the visualizations do not provide any further valuable insights. It was therefore decided to investigate the locations where the measurements were taken. Due to the extreme size of the dataset, it is not possible to display all locations on a single map (OAC requires the addition of filters). Therefore, the display of measurements was filtered to the period from 2001 to 2021. The map of measurements indicating the average temperature and the map of measurements by year are shown on the 2001 map in Figure 3.



CASTS	LAT	ClimateZone	LON	TIME	TEMPERATUR...	AVG_VALUE
2065	4.432	Tropical	-3	04/17/2010 05:52:59.999 AM	23	9.83144345490828700
18244	74.5	Polar	14.416667	08/08/1957 04:00:00.001 PM	14	4.39642852757658300
43245	77.4	Polar	32.566666	08/08/1984 08:24:59.997 PM	6	0.48666664958000183
19940	34.4	Temperate	133.75333	05/19/1966 01:52:59.999 AM	3	16.20000012715657400
22608	34.241665	Temperate	135.035	09/05/1997 05:30:00.000 AM	7	26.04857117789132200

Fig. 1. Extended dataset with sea surface temperatures

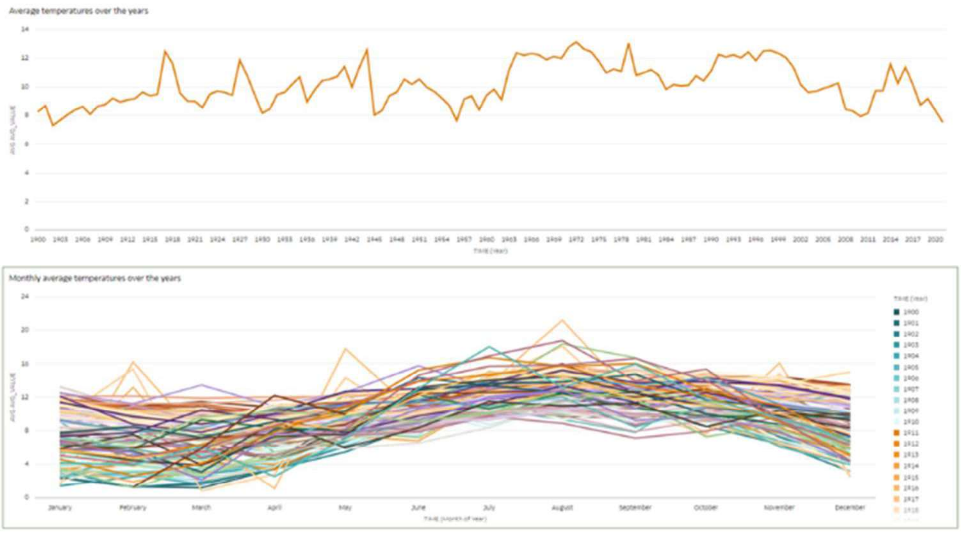


Fig. 2. Canvas with average temperatures over the years

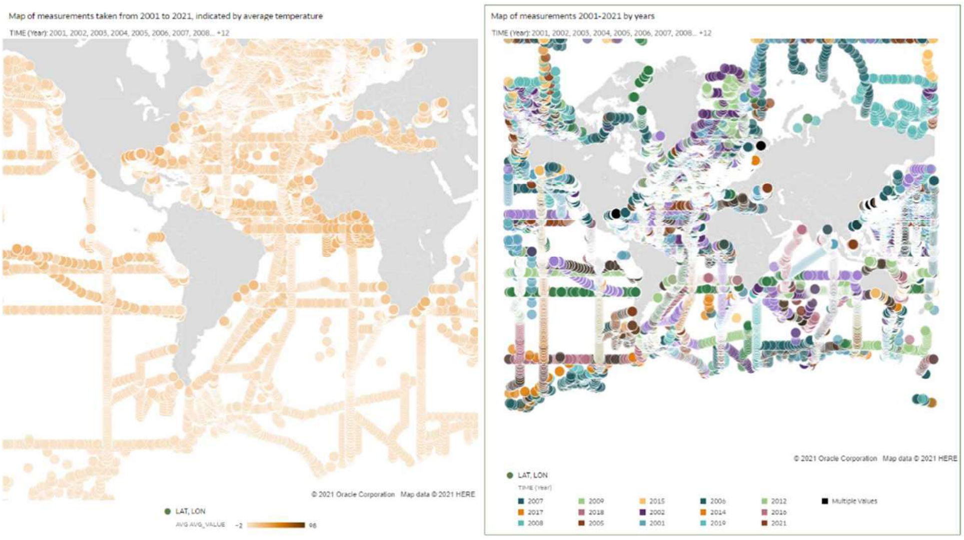

Map of measurements taken from 2001 to 2021, indicated by average temperature
TIME (Year): 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008... +12
© 2021 Oracle Corporation Map data © 2021 HERE
LAT, LON
AVG AVG_VALUE
Map of measurements 2001-2021 by years
TIME (Year): 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008... +12
© 2021 Oracle Corporation Map data © 2021 HERE
LAT, LON
TIME (Year)
2007
2017
2008
2009
2018
2005
2015
2002
2001
2006
2014
2019
2012
2016
2021
Multiple Values

Fig. 3. Locations of measurements taken between 2001 and 2021

Our analyses further focused on the period between 2011 and 2021, based on climate zones. As expected, the lowest average temperatures were measured in the polar climate zone. Since the average temperature in the polar climate zone increased significantly in 2016 compared to other years (Figure 4), we wanted to find out whether the reason for this lies in the location of the measurements. We therefore supplemented the map of measurement locations in the polar zone by year, as shown in Figure 4.

Since global warming of the oceans is a hot topic, we wanted to find locations where data was collected over several years to show the actual warming. To identify the most frequently measured fixed locations over the years, we created a table with the selection of locations and filtered the number of measurements to over 10,000. This allowed us to identify seven locations, all located in the Baltic Sea, shown in Figure 5.

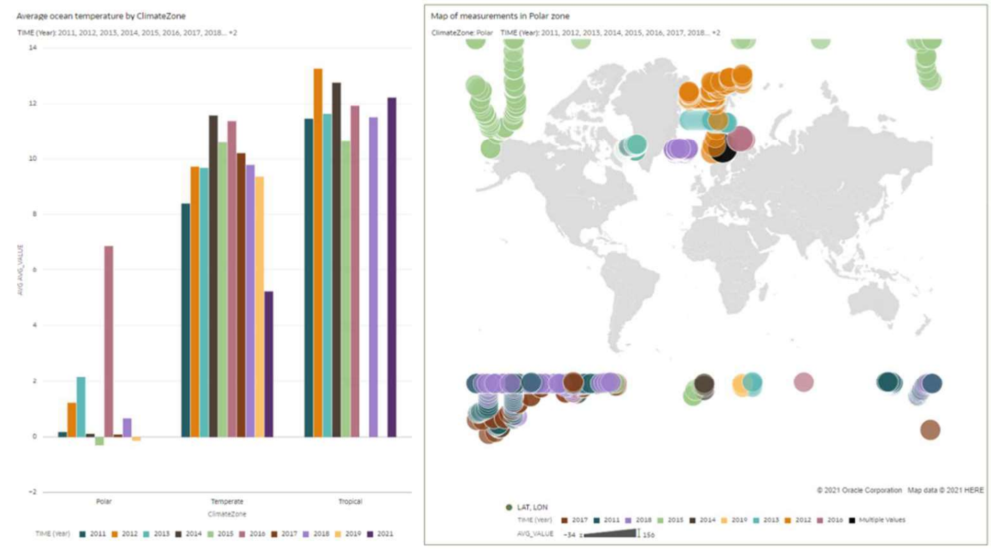
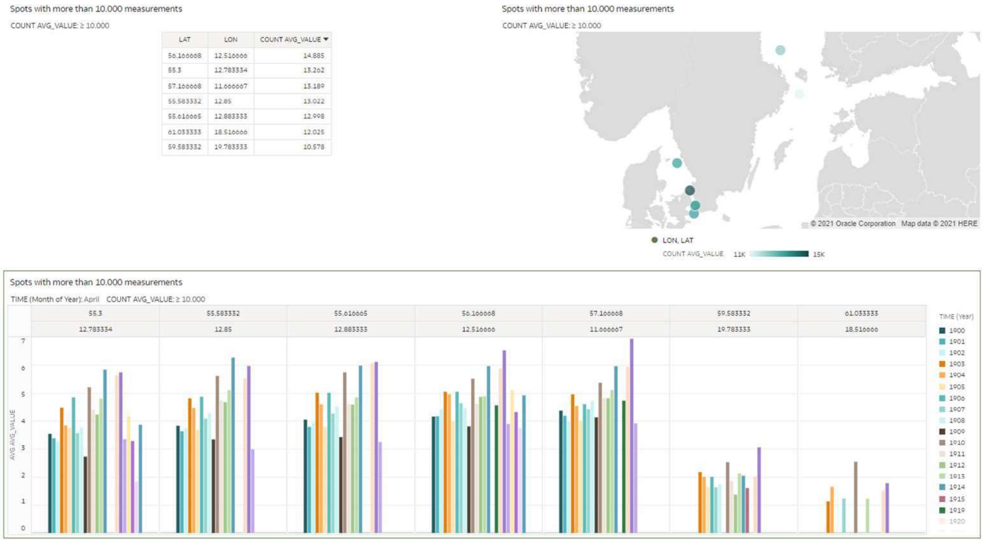


Fig. 4. Temperatures in the polar zone and measurement locations


Spots with more than 10.000 measurements

COUNT AVG_VALUE: ≥ 10.000

LAT	LON	COUNT AVG_VALUE ▼
56.166668	12.516666	14.885
55.3	12.783334	13.262
57.166668	11.666667	13.189
55.583332	12.85	13.022
55.616665	12.883333	12.998
61.033333	18.516666	12.025
59.583332	19.783333	10.578

Fig. 5. Locations with over 10,000 measurements

The average temperature over the years was also included in the list of the best-measured locations. Since the measurements were only taken from 1900 to 1957, our visualization cannot confirm the global warming of the oceans. Instead, we have focused on the locations for which measurements are available up to the present day. We limited the data to the southern part of the Adriatic Sea due to the location (the filter was set to 45° in the north, 40.5° in the south, 15° in the west and 20° in the east). Figure 6 shows the average annual temperature in the southern Adriatic and a map of the measurements taken.

OAC offers an advanced option to use the visualization data as a filter for another visualization. Figure 7 shows the year 1984 selected in the first visualization and the corresponding map of measurements taken in that year in the lower part of the canvas.

OAC also offers an advanced AI-based tool which can be used to examine the data. When the Auto Insights option is selected, OAC provides examples of visualizations for the entire dataset, while the Explain option provides feedback based on the selected measure. Figure 8 shows two visualizations suggested by the Auto Insights and Explain options The Auto Insights option in Oracle Analytics Cloud (OAC) is useful because it automatically identifies and highlights important patterns and trends in your data, saving time and improving decision-making.

.

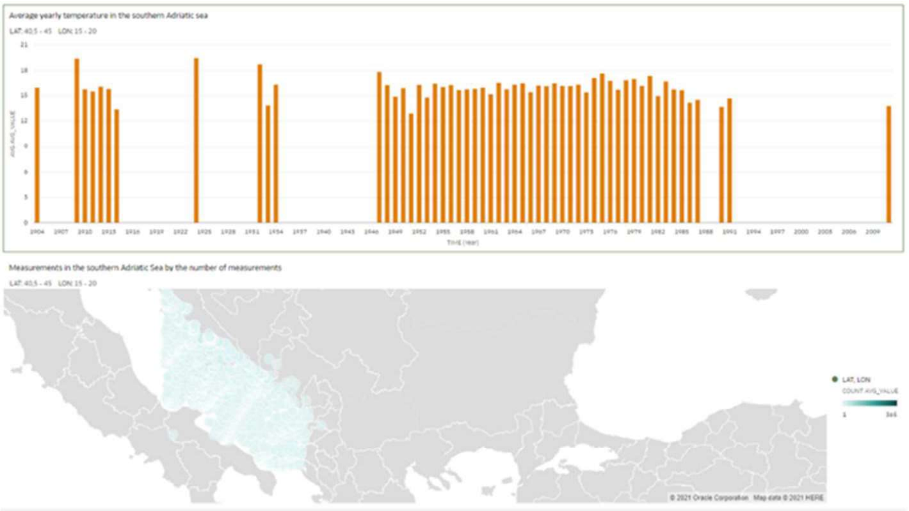

Average yearly temperature in the southern Adriatic sea
LAT: 40,5 - 45 LON: 15 - 20
AVG_AVG_VALUE
21
18
15
12
9
6
3
0
TIME (Year)
Measurements in the southern Adriatic Sea by the number of measurements
LAT: 40,5 - 45 LON: 15 - 20
LAT, LON
COUNT AVG_VALUE
1
366
© 2021 Oracle Corporation Map data © 2021 HERE

Fig. 6. Measurements taken in the southern Adriatic

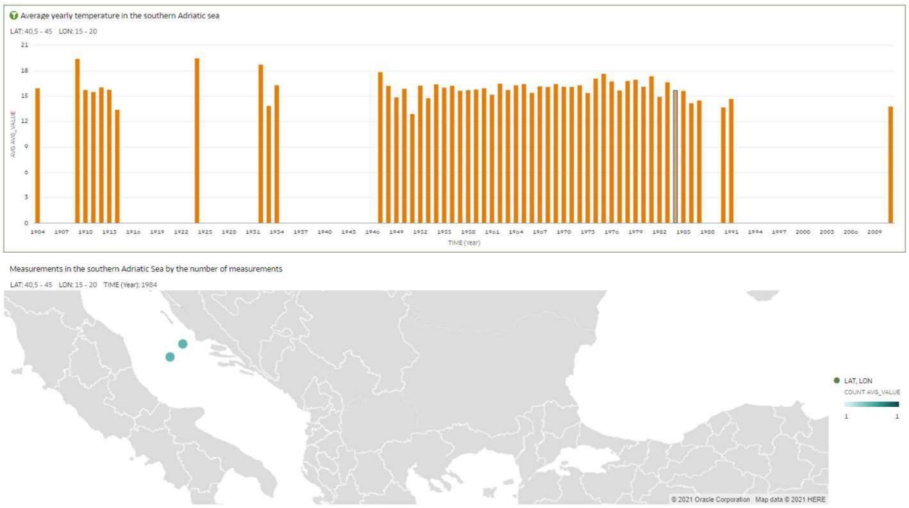

Average yearly temperature in the southern Adriatic sea
LAT: 40,5 - 45 LON: 15 - 20
AVG AVG_VALUE
21
18
15
12
9
6
3
0
1904 1907 1910 1913 1916 1919 1922 1925 1928 1931 1934 1937 1940 1943 1946 1949 1952 1955 1958 1961 1964 1967 1970 1973 1976 1979 1982 1985 1988 1991 1994 1997 2000 2003 2006 2009
TIME (Year)
Measurements in the southern Adriatic Sea by the number of measurements
LAT: 40,5 - 45 LON: 15 - 20 TIME (Year): 1984
LAT, LON
COUNT AVG_VALUE
1
1
© 2021 Oracle Corporation Map data © 2021 HERE

Fig. 7. Filtering measurement locations based on the year

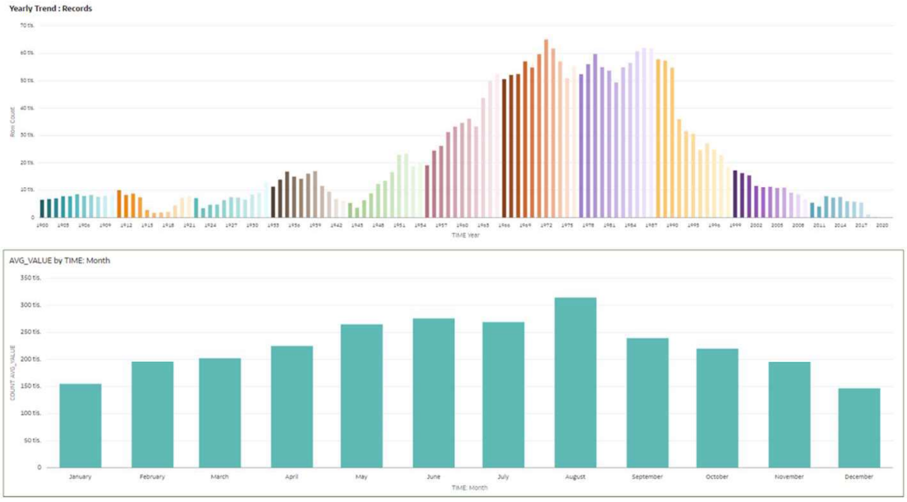


Fig. 8. Visualizations based on the Auto Insights optio

VI. CONCLUSIONS

This study successfully demonstrates the application of Oracle Analytics Cloud (OAC) to the World Ocean Database (WOD) and provides a comprehensive overview for researchers and analysts. The paper describes the steps to retrieve, transform and analyse oceanographic data and highlights the challenges and solutions encountered during the process.

The integration of WOD with OAC not only facilitated the effective visualization and analysis of large oceanographic datasets, but also highlighted the potential of cloud-based analysis platforms in scientific research. By making large datasets more accessible and usable, this approach can improve the understanding of oceanographic trends and support informed decision-making in marine research. Future work could focus on expanding the scope of the analysis to include additional parameters from the WOD and exploring the use of advanced analytical techniques to gain deeper insights into oceanic conditions.

Finally, the paper has shown how Oracle Analytics Cloud and similar platforms can facilitate the integration and analysis of environmental data. While the focus has been on practical applications, the results highlight the broader potential of these tools for environmental research. Platforms such as OAC provide robust solutions for researchers and analysts working with large, complex datasets and enhance data processing and visualisation capabilities. In addition, their accessibility and flexibility also make them valuable in education, where students and teachers can gain hands-on experience with advanced data analysis tools. In the future, we aim to address more specific environmental problems and make a deeper scientific contribution while continuing to promote the practical benefits of these platforms.

Acknowledgment

The authors would like to acknowledge the support given by the European Commission through the Action Erasmus+ Including EVERyone in GREEN Data Analysis (EverGreen), program ID 2022-1-SK01-KA220-HED-000089149, co-funded by the Erasmus+ program of the European Union. The European Commission's support for the production of this publication does not constitute endorsement of the contents which reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

EverGreen DATA Analytics

Co-funded by the Erasmus+ Programme of the European Union

The authors would also like to thank Oracle Academy Program Manager, Mr. Darko Jureković, for his continuous support in disseminating the project results.

References

[1] Oracle, ‘Oracle Open Data’, Connect with open and freely distributed data sets. Accessed: Jul. 29, 2024. [Online]. Available: https://opendata.oraclecloud.com/ords/r/opendata/opendata/home

[2] Oracle, ‘Oracle Open Data’, Oracle Help Center. Accessed: Jul. 29, 2024. [Online]. Available: https://docs.oracle.com/en/programs/research/oracle-open-data/

[3] Puiu. D *et al.*, ‘CityPulse: Large Scale Data Analytics Framework for Smart Cities’, *IEEE Access*, vol. 4, pp. 1086–1108, 2016, doi: 10.1109/ACCESS.2016.2541999.

[4] Hamza, M *et al.*, ‘Big Data Analytics with Artificial Intelligence Enabled Environmental Air Pollution Monitoring Framework’, *Comput. Mater. Contin.*, vol. 73, no. 2, pp. 3235–3250, 2022, doi: 10.32604/cmc.2022.029604.

[5] Gaffoor, Z, Pietersen, K, Bagula, A, Jovanovic, N, Kanyerere, T, and Wanangwa, G, ‘Big Data Analytics and Modelling: Localising transboundary data sets in Southern Africa: A case study approach’, Water Research Commission, Pretoria, Report to the Water Research Commission WRC Report No. TT 843/20, 2021. [Online]. Available: https://wrcwebsite.azurewebsites.net/wp-content/uploads/mdocs/Report%20TT%20843%20final%20web.pdf

[6] van Emmerik, ‘The Complexity and Scale of Environmental Datasets’, *J. Environ. Occup. Health*, vol. 13, no. 11, pp. 01–02, 2023.

[7] Gupta, S, Aga, D, Pruden, A, Zhang, L, and Vikesland, P, ‘Data Analytics for Environmental Science and Engineering Research’, *Environ. Sci. Technol.*, vol. 55, no. 16, pp. 10895–10907, Aug. 2021, doi: 10.1021/acs.est.1c01026.

[8] Bonthu, S and Bindu, H, *Review of Leading Data Analytics Tools*, vol. 7. 2018. doi: 10.14419/ijet.v7i3.31.18190.

[9] Acharjya, D P and Kauser, A, ‘A Survey on Big Data Analytics: Challenges, Open Research Issues and Tools’, *Int. J. Adv. Comput. Sci. Appl.*, vol. 7, no. 2, 2016, doi: 10.14569/IJACSA.2016.070267.

[10] Harvey, R, Gill, S, and Rhone, S, ‘Oracle® Cloud: Visualizing Data and Building Reports in Oracle Analytics Cloud’. Oracle, 2024. [Online]. Available: https://docs.oracle.com/en/cloud/paas/analytics-cloud/acubi/visualizing-data-and-building-reports-oracle-analytics-cloud.pdf

[11] Boyer, T P *et al.*, ‘WORLD OCEAN DATABASE 2018’, 2018. [Online]. Available: https://www.ncei.noaa.gov/data/oceans/woa/WOD/DOC/wod_intro.pdf

[12] National Centers for Environmental Information, ‘NCEI Standard Product: World Ocean Database (WOD)’. Accessed: Jul. 29, 2024. [Online]. Available: https://www.ncei.noaa.gov/access/metadata/landing-page/bin/iso?id=gov.noaa.nodc:NCEI-WOD

[13] Ambatipudi, S and Byna, S, ‘A Comparison of HDF5, Zarr, and netCDF4 in Performing Common I/O Operations’, 2022, *arXiv*. doi: 10.48550/ARXIV.2207.09503.

Performance Testing of Intelligent Data Layer

1st Filip Majerik
Faculty of Electrical Engineering and Informatics
University of Pardubice
Pardubice, Czech Republic
filip.majerik@upce.cz

2nd Monika Borkovcova
Faculty of Electrical Engineering and Informatics
University of Pardubice
Pardubice, Czech Republic
monika.borkovcova@upce.cz

***Abstract*— With the growing popularity of Object-Realational Mapping (ORM) Frameworks, the performance requirements of the applications that implement them are also growing. Thus, it is necessary to examine whether our proposed Intelligent Data Layer (IDL) can handle application requests efficiently enough compared to conventional ORM Framework implementations. Thus, an environment has been created to simulate the behavior of the application on a production server. From the queries performed and the data collected, it will then be evaluated whether IDL is suitable for use in a production environment and how much of an asset it can be in a practical environment.**

Keywords*— *database application architecture, ORM and performance of data layer, entity joining

I. Introduction

As the need to create new digital services grows, so does the need to accelerate their development. For this purpose, various tools are being created, and these tools inseparably include object-relational mapping technologies called ORM frameworks.

These serve as a communication layer between the application and the database system. At the same time, they also serve as an abstraction layer to shield the database system in use and to provide a simple interface that allows the developer to have almost no database knowledge and still develop database-dependent applications. However, with the implementation of such an abstraction comes all sorts of performance issues, which are the focus of this paper [2], [5], [6], [9], [11].

As part of our long-term research, we have designed and previously validated an Intelligent Data Layer to help developers optimize data access when using the ORM Framework. Now we have reached the stage where we need to verify that IDL is usable in practice. In order to decide if our IDL is usable in practice, we had to build an experiment that should be as close as possible to a commonly used production environment.

The IDL performance testing was performed on the same hardware on which the IDL architecture was built. The data layer of the backend system of a commercial entity dedicated to the development and operation of an IPTV platform serves as the main case study for this work. Currently, it manages around 150,000 end devices on this backend layer, which are asking for new data at periodic times and the current solution with ORM has insufficient performance.

This backend system manages individual operators, paid services, content sources and last but not least takes care of data distribution for the devices. To diagnose the problem, a minimalist model of the relational database was developed by dropping information from the model that is not critical and does not affect the eventual data retrieval [8].

In this study, the relational database is connected to the application layer using the Doctrine ORM framework. This framework is used here to a very limited extent and is basically used only for basic operations that a regular database allows. The entire data layer solution is encapsulated in a model layer consisting of a "query system", repositories and service components. The model layer was designed in this way to allow a developer who does not have primary knowledge of databases to participate in the development.

However, the ORM framework was also left with a fairly fundamental responsibility. Namely, that of converting relational data into objects and vice versa. The operations that are performed through ORM are then the creation, modification and deletion of entities. Joining entities and other operations on entities are performed at the repository and model layer level. This model layer, thanks to the implemented query system, makes it relatively easy to filter the required data, sort this data and join the data through a specially implemented relational interface.

II. Environmental Conditions for Experiments

The relational database model from the article Design of Data Access Architecture using ORM Framework was used for the experiments [8].

The relational database for performance testing contained the same data that had been used previously for index or partitioning experiments. An attempt was made to keep the environment as similar as possible so that it would be easier to decide on the effectiveness of IDL. No special operations were needed to modify or change the data in any way. The following table shows all the database tables and the number of rows they contain for the experiment [1], [6], [7], [10].

TABLE I. Number of Rows in Each Database Table

Table	Number of Rows
brand	36
device_type	3
device_profile	4
operator	10
subscriber	311.847
device	1.480.661

The original experiment did not need to be modified in any way to perform this work. It was taken exactly as previously. However, within this work, stress tests were performed to reveal whether or not IDL is usable in practice or directly in a production environment.[12]

As in previous experiments, four models were used and subjected to testing, it means firstly native SQL query through ORM, ORM and Lazy loading, ORM and Eager loading and the last model ORM and implemented IDL. For performance testing, however, it was necessary to test a varying number of web workers.

979-8-3503-8768-1/24/$31.00 ©2024 IEEE

Therefore, we performed testing with the following configuration:

- 2 web workers + 2 php-fpm processes
- 4 web workers + 4 php-fpm processes
- 8 web workers + 8 php-fpm processes
- 16 web workers + 16 php-fpm processes
- 32 web workers + 32 php-fpm processes
- 64 web workers + 64 php-fpm processes

Next, the numbers of expected results were determined so that it could be decided whether IDL is worthwhile for smaller or larger datasets. The result thresholds were therefore set at 10, 100, 1.000, 10,000 and 100.000 results.

These configurations and limits were chosen entirely experimentally. In the end, it will be evaluated which of the configurations was the most ideal for the hardware used and whether IDL is worthwhile for these limits.

The combinations were also chosen to verify how IDL behaves in the case of "less" server load and then in the case of extreme server load, when for example CPU cores are no longer available and processes have to "compete" for system resources. [4], [5]

A previously published experiment served as the core of our current experiment. Where information from the device, device_profile, device_type, brand, subscriber and operator tables are combined within the output.

This information is then converted into JSON and returned to the user. For the purpose of this experiment, the query was extended to limit the output data even further.

Native SQL query:

```
SELECT d.id AS device_id,
d.name AS device_name,
d.mac_address, dp.name AS device_profile_name,
dt.name AS device_type_name,
d.last_start, br.name AS brand_name,
CONCAT_WS(' ', s.name, s.surname) AS subscriber_name,
o.name AS operator_name
FROM device d
LEFT JOIN device_profile dp
ON d.device_profile_id = dp.id
LEFT JOIN device_type dt
ON d.device_type_id = dt.id
LEFT JOIN brand br
ON br.id = d.brand_id
LEFT JOIN subscriber s
ON d.subscriber_id = s.id
LEFT JOIN operator o
ON s.operator_id = o.id;
```

As part of the experiment, it was then necessary to prepare an application for simulating clients. The client was built in JAVA. Using JAVA Thread we created 512 threads to simulate user queries. We synchronized these threads using CylicBarrier so that they start querying the target web server at the same time.

We then started collecting data about the queries made, which was inserted into the response headers on the Symfony side. To collect statistical data, we needed to create a custom EventListener on the Symfony side, which ran the necessary timers after the request started and then collected statistics on the SQL queries executed. We then read these headers within the test JAVA application and compiled the aggregated statistics below.

The following statistics were transmitted within the headers memory peak in MB, query processing time on Symfony side, number of executed SQL queries, number of unique SQL queries, total query time and answer size in kB.

On the JAVA application side, the total request time was also monitored. This time was measured from the initiation of the request to the web server, to the reading of the complete response. Measured time also included the time consumed by the request waiting for the server workers to be released, as well as the time it took to transfer the data.

When evaluating the performance of a database application, it is crucial to track various metrics that provide a comprehensive view of its efficiency and reliability.

Request time [ms] (RT) is an essential metric that measures the total time from query initiation to retrieval of all data. This metric directly affects user satisfaction and can be an indicator of the overall effectiveness of the system.

Execution time [ms] (ET) focuses on the time it takes to execute a query on the server, which helps to identify potential problems on the server side, such as inefficient algorithms or insufficient resources.

Memory Peak [MB] (MP) tracks the maximum memory usage by the PHP process, which is important for efficient resource management and preventing memory problems that can lead to application crashes.

Number of Database Queries (Q) shows how many queries have been executed, and a smaller number usually means more efficient processing and less load on the database. This is important for optimizing performance and reducing latency.

Number of Different Queries (DQ) measures the number of unique queries, which can indicate how well the application is optimized for cache utilization, which is key to improving response speed.

Query Time [ms] (QT) provides an overview of the total time it takes to process queries, which is important for identifying whether queries or the database server needs to be optimized.

The QT/ET ratio is useful for determining whether the application or the database is causing the delay, and an ideal value of around 0.5 indicates balanced processing.

The Number of Requests per second (Req./s) shows how many requests an application can process per second, which is key to its scalability and ability to handle high load.

Together, these metrics provide a comprehensive picture of application performance and allow you to identify areas that require optimization.

Measuring these metrics is essential to ensure high performance and reliability of the database application, which is key to its successful deployment in a production environment.

III. Experiments

TABLE II. 2 NGINX Workers, 2 PHP-FPM Processes

Number of rows	Method	RT	ET	MP	Q	DQ	QT	QT/ET	Req./s
10	Native Query	13522	15,088	8	1	1	1,01	0,07	27,78
	Lazy Loading	15545	31,38	12	18	6	3,15	0,1	22,93
	Eager Loading	15897	30,134	12	2	2	1,96	0,07	22,64
	IDL	15083	30,24	12	6	6	2,24	0,07	23,55
100	Native Query	13939	16,54	8	1	1	1,42	0,09	26,72
	Lazy Loading	16635	39,98	12	59	6	7,17	0,18	20,8
	Eager Loading	15700	34,45	12	2	2	2,95	0,09	22,25
	IDL	15379	32,5	12	6	6	2,48	0,08	22,89
1.000	Native Query	14117	20,31	10	1	1	5,624	0,28	25,74
	Lazy Loading	25067	105,27	16	412	6	38,05	0,36	12,38
	Eager Loading	20600	71,48	16	2	2	10,26	0,14	15,6
	IDL	18478	54,55	16	6	6	4,86	0,09	17,99
10.000	Native Query	21227	63,59	24,64	1	1	33,95	0,53	15,21
	Lazy Loading	117258	810,58	64,87	4018	6	367,13	0,45	2,27
	Eager Loading	68790	438,17	68,85	2	2	68,23	0,16	3,95
	IDL	48302	275,29	56,73	9	6	26,78	0,1	5,83
100.000	Native Query	79444	473,26	144,79	1	1	297,74	0,63	3,41
	Lazy Loading	925852	7084,63	525,09	36220	6	3153,06	0,45	0,27
	Eager Loading	1359422	10083,21	516,56	2	2	6703,21	0,66	0,19
	IDL	348583	2527,16	437,37	42	6	208,74	0,08	0,75

TABLE III. 4 NGINX Workers, 4 PHP-FPM Processes

Number of rows	Method	RT	ET	MP	Q	DQ	QT	QT/ET	Req./s
10	Native Query	7554	16,64	8	1	1	0,89	0,05	49,52
	Lazy Loading	8342	36,43	12	18	6	2,92	0,08	41,39
	Eager Loading	8566	34,38	12	2	2	1,94	0,06	40,73
	IDL	8358	35,11	12	6	6	2,16	0,06	41,64
100	Native Query	7594	17,99	8	1	1	1,36	0,08	48,72
	Lazy Loading	9203	46,99	12	59	6	7,37	0,16	36,66
	Eager Loading	8447	38,07	12	2	2	2,61	0,07	40,78
	IDL	8522	38,82	12	6	6	2,51	0,06	40,31
1.000	Native Query	7688	22,6	10	1	1	5,29	0,23	46,67
	Lazy Loading	14444	128,19	16	412	6	44,01	0,34	21,02
	Eager Loading	11324	81,21	16	2	2	9,68	0,12	28,06
	IDL	10234	65,35	16	6	6	4,89	0,07	31,68
10.000	Native Query	12682	81,16	24,64	1	1	45,95	0,57	25,23
	Lazy Loading	69359	967,08	64,87	4018	6	428,16	0,44	3,84
	Eager Loading	36793	464,97	68,85	2	2	76,28	0,16	7,47
	IDL	26590	305,7	56,73	9	6	31,59	0,1	10,61
100.000	Native Query	54033	670,84	144,79	1	1	470,95	0,7	4,95
	Lazy Loading	534754	8096,51	513,1	36220	6	3667,59	0,45	0,48
	Eager Loading	682780	10327,7	518,56	2	2	6937,05	0,67	0,38
	IDL	183794	2658,79	437,37	42	6	268,81	0,1	1,42

TABLE IV. 8 NGINX Workers, 8 PHP-FPM Processes

Number of rows	Method	RT	ET	MP	Q	DQ	QT	QT/ET	Req./s
10	Native Query	4887	21,51	8	1	1	1,12	0,05	75,73
	Lazy Loading	5513	46,35	12	18	6	3,5	0,08	62,94
	Eager Loading	5658	43,46	12	2	2	2,16	0,05	62,7
	IDL	5442	44,28	12	6	6	2,48	0,06	63,81
100	Native Query	4824	21,58	8	1	1	1,54	0,07	76,74
	Lazy Loading	6019	61,71	12	59	6	9,67	0,16	55,87
	Eager Loading	5637	49,72	12	2	2	2,97	0,06	61,57
	IDL	5762	48,94	12	6	6	2,87	0,06	60,74
1.000	Native Query	4920	28,46	10	1	1	6,42	0,23	71,65
	Lazy Loading	10096	183,03	16	412	6	60,13	0,33	29,85
	Eager Loading	7761	112,43	16	2	2	11,75	0,1	41,06
	IDL	6921	88,62	16	6	6	5,75	0,06	47,18
10.000	Native Query	7885	100,95	24,64	1	1	57,21	0,57	40,37
	Lazy Loading	50910	1420,49	64,87	4018	6	576,5	0,41	5,25
	Eager Loading	27781	718,33	68,85	2	2	104,13	0,14	9,81
	IDL	19683	469,52	56,73	9	6	33,45	0,07	14,21
100.000	Native Query	36255	869,95	144,79	1	1	595,25	0,68	7,51
	Lazy Loading	447658	13532,97	525,09	36220	6	5462,04	0,4	0,58
	Eager Loading	441080	13343,82	516,56	2	2	8206,97	0,62	0,59
	IDL	146177	4229,41	437,37	42	6	306,11	0,07	1,79

TABLE V. 16 NGINX Workers, 16 PHP-FPM Processes

Number of rows	Method	RT	ET	MP	Q	DQ	QT	QT/ET	Req./s
10	Native Query	3559	30,42	8	1	1	1,6	0,05	100
	Lazy Loading	3799	70,64	12	18	6	7,77	0,11	85,05
	Eager Loading	3828	64,82	12	2	2	4,06	0,06	84,97
	IDL	3859	66,58	12	6	6	5,46	0,08	86,68
100	Native Query	3654	32,58	8	1	1	2,61	0,08	98,23
	Lazy Loading	4381	91,3	12	59	6	16,13	0,18	75,7
	Eager Loading	3982	73,04	12	2	2	5,24	0,07	84,63
	IDL	4124	74	12	6	6	6,49	0,09	82,58
1.000	Native Query	3959	46,68	10	1	1	11,81	0,25	90,25
	Lazy Loading	7422	274,15	16	412	6	106,73	0,39	39,84
	Eager Loading	5456	158,29	16	2	2	15,82	0,1	56,7
	IDL	5110	129,42	16	6	6	11,23	0,09	62,47
10.000	Native Query	10481	202,67	24,64	1	1	125,78	0,62	34,96
	Lazy Loading	37769	2070,44	64,87	4018	6	992,69	0,48	7,15
	Eager Loading	21536	1080,35	68,85	2	2	154,96	0,14	12,89
	IDL	15273	718,67	56,73	9	6	64,74	0,09	18,31
100.000	Native Query	40142	2032,22	144,79	1	1	1627,79	0,8	6,81
	Lazy Loading	318840	18782,27	523,09	36220	6	8979,5	0,48	0,83
	Eager Loading	353733	20956,69	518,56	2	2	12536,25	0,6	0,75
	IDL	121547	6897,43	437,37	42	6	559,06	0,08	2,19

TABLE VI. 32 NGINX WORKERS, 32 PHP-FPM PROCESSES

Number of rows	Method	RT	ET	MP	Q	DQ	QT	QT/ET	Req./s
10	Native Query	3511	60,55	8	1	1	2,36	0,04	106,62
	Lazy Loading	3843	141,58	12	18	6	35,47	0,25	90,2
	Eager Loading	3689	115,45	12	2	2	12,75	0,11	93,16
	IDL	3432	124,28	12	6	6	19,25	0,15	95,99
100	Native Query	3223	56,95	8	1	1	2,42	0,04	115,44
	Lazy Loading	3845	182,61	12	59	6	73,43	0,4	85,15
	Eager Loading	3492	132,15	12	2	2	14,08	0,11	93,91
	IDL	3520	129,26	12	6	6	19,01	0,15	96,15
1.000	Native Query	3456	75,13	10	1	1	11,38	0,15	105,65
	Lazy Loading	6583	510	16	412	6	304,3	0,6	45,5
	Eager Loading	4975	286,18	16	2	2	24,33	0,09	63,09
	IDL	4311	223,82	16	6	6	25,73	0,11	73,37
10.000	Native Query	11396	229,66	24,64	1	1	113,13	0,49	35,68
	Lazy Loading	32682	3494,03	64,87	4018	6	2370,97	0,68	8,58
	Eager Loading	18221	1804,75	68,85	2	2	209,96	0,12	15,41
	IDL	13226	1172,48	56,73	9	6	80,2	0,07	22,19
100.000	Native Query	42237	3048,33	144,79	1	1	2260,17	0,74	7,21
	Lazy Loading	289843	32530,88	525,09	36220	6	22056,03	0,68	0,96
	Eager Loading	311404	36168,96	518,56	2	2	22842,12	0,63	0,87
	IDL	97749	10628,96	449,37	42	6	564,8	0,05	2,83

TABLE VII. 64 NGINX WORKERS, 64 PHP-FPM PROCESSES

Number of rows	Method	RT	ET	MP	Q	DQ	QT	QT/ET	Req./s
10	Native Query	3340	110,96	10	1	1	2,41	0,02	110,01
	Lazy Loading	3592	262,31	12	18	6	58,89	0,22	96,44
	Eager Loading	4091	234,31	12	2	2	42,49	0,18	90,94
	IDL	3676	247,6	12	6	6	38,95	0,16	94,78
100	Native Query	3261	108,54	8	1	1	3,41	0,03	115,29
	Lazy Loading	4051	356,96	12	59	6	138,62	0,39	84,75
	Eager Loading	3665	246,68	12	2	2	22,52	0,09	95,47
	IDL	3724	261,49	12	6	6	38,97	0,15	93,53
1.000	Native Query	3759	135,59	10	1	1	10,64	0,08	102,24
	Lazy Loading	7440	1104,31	16	412	6	736,62	0,67	41,73
	Eager Loading	5085	546,64	16	2	2	37,25	0,07	64,14
	IDL	4631	446,61	16	6	6	47,87	0,11	71,17
10.000	Native Query	13167	409,59	24,64	1	1	24,64	0,06	31,97
	Lazy Loading	38363	7757,99	64,87	4018	6	6099,36	0,79	7,69
	Eager Loading	19533	3545,65	68,85	2	2	385,79	0,11	15,56
	IDL	13041	2282,91	64,73	9	6	126,58	0,06	22,59
100.000	Native Query	45987	8032,03	144,79	1	1	7354,9	0,92	6,15
	Lazy Loading	334354	71403,84	523,1	36220	6	58280,42	0,82	0,87
	Eager Loading	306350	69011,13	516,56	2	2	42397,33	0,61	0,91
	IDL	103287	21492,22	449,37	42	6	846,71	0,04	2,76

"Request time" and "Requests per second" were selected as the most important performance indicators for each experiment. These are visualized in the following graphs.

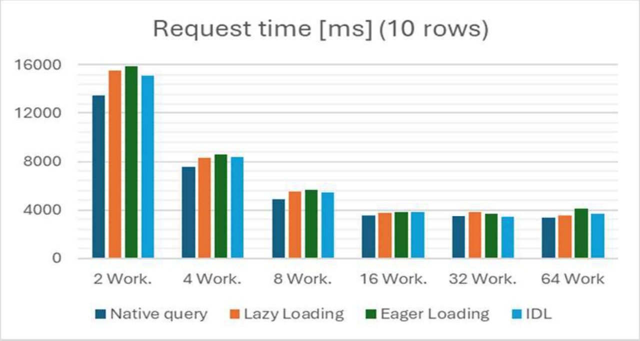


Fig. 1. Comparison of request times – 10 rows

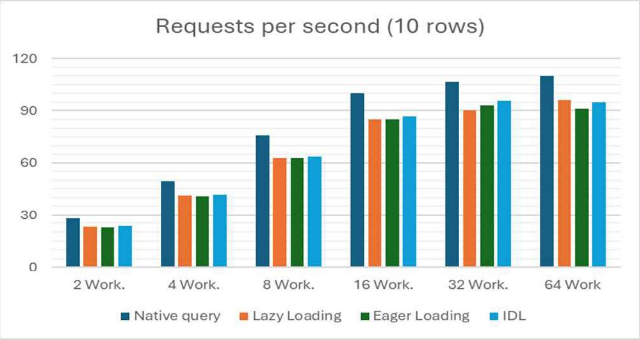


Fig. 2. Comparison of requests per second – 10 rows

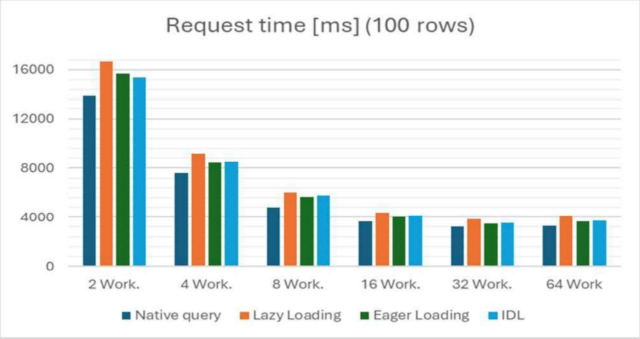


Fig. 3. Comparison of request times – 100 rows

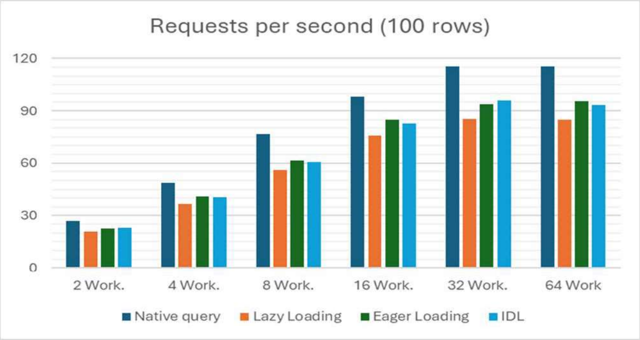


Fig. 4. Comparison of requests per second – 100 rows

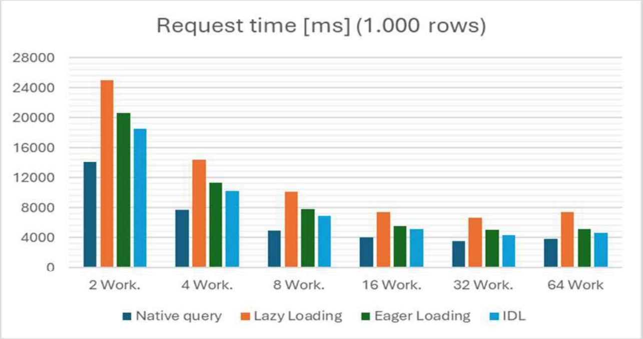


Fig. 5. Comparison of request times – 1.000 rows

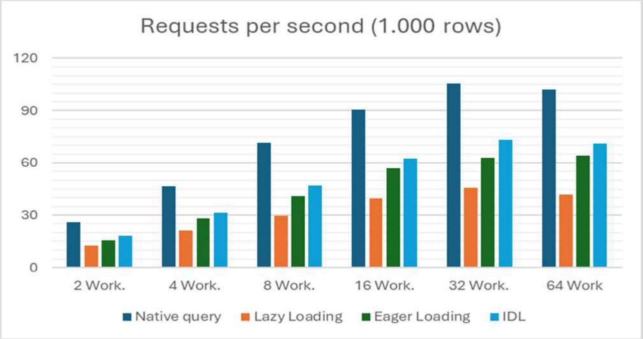


Fig. 6. Comparison of requests per second – 1.000 rows

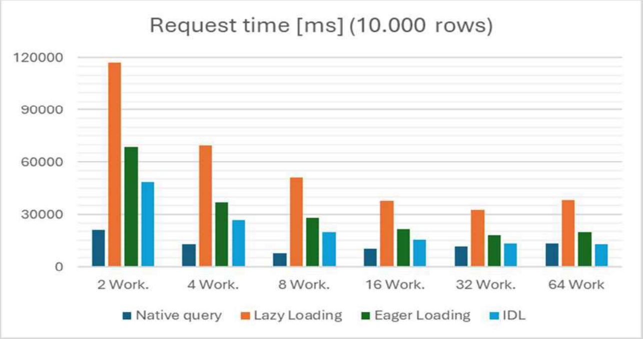


Fig. 7. Comparison of request times – 10.000 rows

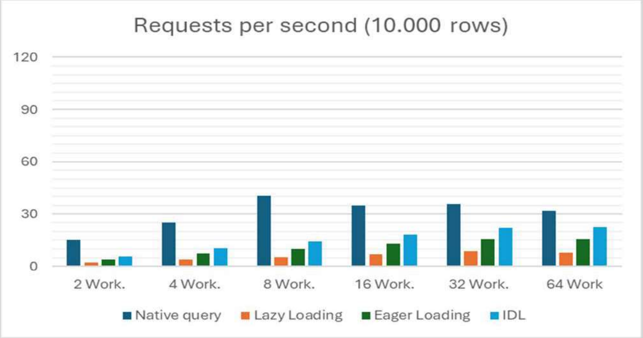


Fig. 8. Comparison of requests per second – 10.000 rows

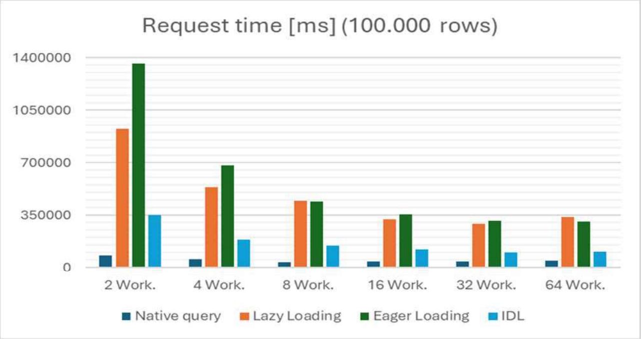


Fig. 9. Comparison of request times – 100.000 rows

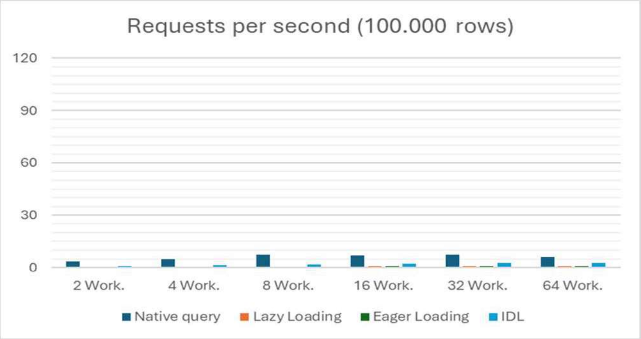


Fig. 10. Comparison of requests per second – 100.000 rows

IV. Discussion of Results

A. *Overall evaluation of 2 NGINX WORKERS and 2 PHP-FPM PROCESSES*

Native Query:

- It has the lowest number of queries (Q) and miscellaneous queries (DQ), indicating efficient use of queries.
- It achieves the highest number of requests per second (Req./s) in most cases, indicating high efficiency.
- It has relatively low Query Time (QT) and QT/ET ratio, indicating efficient query processing.

Lazy Loading:

- It has a high number of queries (Q), which may lead to higher load on the database and longer processing time.
- Significantly higher memory peak (MP) and execution time (ET) for larger datasets, indicating that it is not suitable for large volumes of data.
- Low requests per second (Req./s) for larger datasets, indicating lower efficiency.

Eager Loading:

- Has a lower number of queries (Q) than Lazy Loading, which may lead to better efficiency.
- Medium performance between Native Query and Lazy Loading, but with higher memory requirements.
- The QT/ET ratio is relatively low, indicating efficient query processing.

IDL:

- It has a balanced number of queries (Q) and miscellaneous queries (DQ), which can be a trade-off between Lazy and Eager Loading.
- It achieves better performance than Lazy Loading, especially for larger datasets.
- The QT/ET ratio is low, indicating efficient query processing.

For small datasets (e.g., 10 and 100 rows): Native Query is the most efficient choice due to its low processing time and high number of requests per second.

For medium datasets (e.g., 1,000 rows): Eager Loading and IDL offer a good compromise between performance and memory requirements.

For large datasets (e.g., 10,000 and 100,000 rows): Native Query remains the most efficient, but due to the use of ORM, IDL is a good alternative if performance and memory requirements need to be balanced.

B. Overall evaluation of 4 NGINX WORKERS and 4 PHP-FPM PROCESSES

Native Query:

- Requests per second (Req./s): Significantly higher in all cases, indicating an improvement in overall processing performance and efficiency.
- Query Time (QT): Slightly lower, contributing to a better QT/ET ratio and higher efficiency.

Lazy Loading:

- Number of requests per second (Req./s): Significantly higher for smaller datasets, indicating an improvement in performance, but still remains low for larger datasets.
- Execution Time (ET): Increased for larger datasets, still indicating unsuitability for large data volumes.

Eager Loading:

- Requests per second (Req./s): Significantly higher for all dataset sizes, indicating improved performance.
- Execution Time (ET): Slightly improved, contributing to better efficiency.

IDL:

- Requests per second (Req./s): Significantly higher, especially for larger datasets, indicating improved performance and efficiency.
- Execution Time (ET): Slightly improved, contributing to better efficiency.

Overall, there is an improvement in performance for all methods, especially Native Query and Eager Loading, which now achieve higher requests per second. Lazy Loading still remains unsuitable for large volumes of data, but shows improvement for smaller datasets. IDL shows significant performance improvements, making it a suitable alternative for different dataset sizes.

C. Overall evaluation of 8 NGINX WORKERS and 8 PHP-FPM PROCESSES

Native Query:

- Requests per second (Req./s): Even higher than before, indicating further performance improvements.
- Query Time (QT): Slightly higher, but still maintains an effective QT/ET ratio.

Lazy Loading:

- Requests per second (Req./s): Improvement for smaller datasets is more pronounced, but performance remains low for larger datasets.
- Execution Time (ET): Significantly higher for larger datasets, confirming unsuitability for large data volumes.

Eager Loading:

- Requests per second (Req./s): Significant performance improvement for all dataset sizes, a positive change.
- Execution Time (ET): Slightly improved, contributing to better efficiency.

IDL:

- Requests per second (Req./s): Significant improvement, especially for larger datasets, indicating better performance than before.
- Execution Time (ET): Slightly improved, contributing to better efficiency.

Overall, there has been a further improvement in performance for all methods, especially Native Query, which now achieves even higher requests per second. Lazy Loading still remains unsuitable for large volumes of data, but shows more significant improvements for smaller datasets. Eager Loading and IDL show further performance improvements, making them even more suitable alternatives for different dataset sizes.

D. Overall evaluation of 16 NGINX WORKERS and 16 PHP-FPM PROCESSES

Native Query:

- Requests per second (Req./s): Slightly lower for larger datasets, but still very high for smaller datasets. Performance remains strong, although there has been a slight decrease for the largest datasets.
- Query Time (QT): Increase in QT/ET for larger datasets, indicating that processing efficiency has decreased slightly.

Lazy Loading:

- Requests per second (Req./s): Significant improvement for smaller datasets, but still low performance for larger datasets. The improvement is noticeable, but the method remains unsuitable for large data volumes.
- Execution Time (ET): Significantly higher for larger datasets, confirming unsuitability for large data volumes.

Eager Loading:

- Requests per second (Req./s): Slight improvement for smaller datasets, but performance for larger datasets remains similar to before.
- Execution Time (ET): Slightly improved, contributing to better performance on smaller datasets.

IDL:

- Requests per second (Req./s): Significant improvement for smaller datasets, but performance for larger datasets remains similar to before.
- Execution Time (ET): Slightly improved, contributing to better performance on smaller datasets.

Overall, there is a slight improvement in performance on smaller datasets for most methods, especially Lazy Loading and IDL. Native Query remains the most efficient choice, although there is a slight performance drop on the largest datasets. Lazy Loading still remains unsuitable for large data volumes, but shows improvement for smaller datasets. Eager Loading and IDL show a slight performance improvement, making them suitable alternatives.

E. *Overall evaluation of 32 NGINX WORKERS and 32 PHP-FPM PROCESSES*

Native Query:

- Requests per second (Req./s): Slight improvement for smaller datasets, performance remains stable for larger datasets.
- Query Time (QT): Improvement in processing efficiency, especially for smaller datasets.

Lazy Loading:

- Requests per second (Req./s): Slight improvement for smaller datasets, but still poor performance for larger datasets. Performance has improved, but the method remains unsuitable for large data volumes.
- Execution Time (ET): Significantly higher for larger datasets, confirming unsuitability for large data volumes.

Eager Loading:

- Requests per second (Req./s): Slight improvement for smaller datasets, indicating better performance. Performance remains stable for larger datasets.
- Execution Time (ET): Slightly improved.

IDL:

- Requests per second (Req./s): Significant improvement for smaller datasets, performance remains stable for larger datasets.
- Execution Time (ET): Slightly improved.

Overall, there was a slight improvement in performance on smaller datasets for most methods, especially Native Query and IDL. Native Query remains the most efficient choice, although there was a slight performance drop on the largest datasets. Lazy Loading still remains unsuitable for large data volumes, but shows improvement for smaller datasets. Eager Loading and IDL show another slight performance improvement.

F. *Overall evaluation of 64 NGINX WORKERS and 64 PHP-FPM PROCESSES*

Native Query:

- Requests per second (Req./s): Slight decrease for larger datasets, indicating a performance degradation.
- Query Time (QT): Increase in QT/ET for larger datasets, indicating a decrease in processing efficiency.

Lazy Loading:

- Requests per second (Req./s): Slight improvement for smaller datasets, but still low performance for larger datasets. Performance has improved, but the method remains unsuitable for large data volumes.
- Execution Time (ET): Significantly higher for larger datasets, confirming unsuitability for large data volumes.

Eager Loading:

- Requests per second (Req./s): Slight improvement for smaller datasets, but performance for larger datasets remains similar to before.
- Execution Time (ET): Slightly improved, contributing to better performance on smaller datasets.

IDL:

- Requests per second (Req./s): Slight improvement for smaller datasets, but performance for larger datasets remains similar to before.
- Execution Time (ET): Slightly improved, contributing to better performance on smaller datasets.

Overall, there is a slight improvement in performance on smaller datasets for most methods, especially Lazy Loading and IDL. Native Query remains the most efficient choice, although there is a slight performance drop on the largest datasets. Lazy Loading still remains unsuitable for large data volumes, but shows improvement for smaller datasets.

V. Conclusion

At the beginning of our work we set ourselves the task of evaluating whether the architecture we have built is suitable for use in practice and whether it really shows signs of optimizing access to database data. From the results, we can obtain quite interesting information about the behavior of the ORM framework and IDL.

The first interesting thing that can be gleaned from the results tables is that the total request length is often not affected by the server-side execution time. Another interesting fact that can be seen from the tabulated results and subsequently the visualization is that almost none of the observed parameters improve significantly when reaching 16 workers. On the contrary, a dramatic deterioration can be observed, for example, in the query time, which increases >1.5x in almost all cases. In some cases, even much larger increases can be observed. For example, 100,000 records with 16 workers needed 8,979.5 ms, with 32 workers it was already 22,056.03 ms (~146% increase) and with 64 workers it was even 58,280.42 ms (~164% increase). However, the total query time is still increasing in the order of 10%. If we look at the results in terms of requests per second, we can objectively evaluate that IDL outperformed Lazy and Eager Loading in the vast majority of the experiments performed. For queries >=1000 records, it performed better in all cases, except for the comparison with the experiment where Native Query was used. For lower number of records, the results of IDL and Eager loading differ in low query units. Only in the case of 64 workers and 10 records was Lazy loading the most effective.

Regarding IDL and other observed parameters, a similar evaluation can be made as in the previous case. Starting from >=1000 records, this approach was the most efficient after the Native Query approach was used to load the data. Thus, IDL was more efficient in execution time per query, maximum memory required, query time, ratio of query processing time to total execution time, and also in number of queries per second.

On the other hand, for Lazy Loading and Eager Loading, we can observe an extreme increase in values as the number of records increases for any number of workers. Thus, these two approaches simply cannot be recommended for use in any competitive environment combined with a large number of records loaded. The results show that against IDL, the worst case (2 workers, 100,000 records, Eager loading) required almost four times more time to execute the entire request.

Based on these results, we can say that IDL can be used as an optimization technique. In almost all the experiments presented, the measured values of IDL were better than those of standard ORM approaches.

For a definitive evaluation of whether or not to deploy this data layer in practice, it would be advisable to perform further measurements, e.g. on server hardware.

Acknowledgment

It was supported by the Erasmus+ project: Project number: 2022-1-SK01-KA220-HED-000089149, Project title: Including EVERyone in GREEN Data Analysis (EVERGREEN) funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the Slovak Academic Association for International Cooperation (SAAIC). Neither the European Union nor SAAIC can be held responsible for them.

Co-funded by the European Union

EverGreen DATA Analytics

References

[1] M. Bandle, J. Giceva, T. Neumann, “To Partition, or Not to Partition, That is the Join Question in a Real System,“ International Conference on Management of Data, SIGMOD 2021 168-180. Online: Association for Computing Machinery. 2021. doi:10.1145/3448016.3452831.

[2] Z. Dong et al. “Database Deadlock Diagnosis for Large-Scale ORM-Based Web Applications,“ 2023 IEEE 39th International Conference on Data Engineering (ICDE). Anaheim, CA, USA. 2864-2877. 2023. doi: 10.1109/ICDE55515.2023.00219.

[3] C. Pitt, “Pro PHP 8 MVC: Model View Controller Architecture-Driven Application Development,“ 2nd edition, Apress, 2021. ISBN: 978-1484269565.

[4] W. Khan, C. Zhang, B. Luo, T. Kumar, E. Ahmed. 2021. “Robust Partitioning Scheme for Accelerating SQL Database.“ IEEE International Conference on Emergency Science and Information Technology, ICESIT 2021. Chongqing: Institute of Electrical and Electronics Engineers Inc. 369-376. doi:10.1109/ICESIT53460.2021.9696761

[5] S. Kläbe, K. Sattler, “Patched Multi-Key Partitioning for Robust Query Performance.“ 26th International Conference on Extending Database Technology, EDBT 2023. Ioannina: OpenProceedings.org. 324-336. doi:10.48786/edbt.2023.26

[6] M. Kvet, “Database Index Balancing Strategy,” 2021 29th Conference of Open Innovations Association (FRUCT), Tampere, Finland, 2021, 214-221. doi: 10.23919/FRUCT52173.2021.9435452

[7] M. Kvet and J. Papan, “The Complexity of the Data Retrieval Process Using the Proposed Index Extension,” in IEEE Access, vol. 10, 46187-46213, 2022. doi: 10.1109/ACCESS.2022.3170711

[8] F. Majerik and M. Borkovcova, “Design of Data Access Architecture Using ORM Framework,“ 34th Conference of Open Innovations Association (FRUCT), Riga, Latvia, 2023, 93-99. doi: 10.23919/FRUCT60429.2023.10328151.

[9] V. Salgova, K. Matiasko, “The Effect of Partitioning and Indexing on Data Access Time,” 29th Conference of Open Innovations Association FRUCT, FRUCT 2021. Tampere: IEEE Computer Society. 301-306. doi:10.23919/FRUCT52173.2021.9435500

[10] G. Vial, “Lessons in Persisting Object Data Using Object-Relational Mapping,” in IEEE Software, vol. 36, no. 6, 43-52. Nov.-Dec. 2019, doi: 10.1109/MS .2018.227105428.

[11] Z. Xu, J. Zhu, a L. Yang, “Mining the Relationship between Object-Relational Mapping Performance Anti-patterns and Code Clones,” Proceedings of the 35th International Conference on Software Engineering, San Francisco, 2023, 136-141. doi: 10.18293/SEKE2023-161.

[12] Yan, Cong, Alvin Cheung, Junwen Yang, a Shan Lu. 2017. “Understanding Database Performance Inefficiencies in Real-world Web Applications.“ ACM Conference on Information and Knowledge Management. New York: Association for Computing Machinery. 1299-1308. doi:10.1145/3132847.3132954.

Verify the Effectiveness of the Intelligent Data Layer on Different Hardware Configurations

1st Filip Majerik
Faculty of Electrical Engineering and Informatics
University of Pardubice
Pardubice, Czech Republic
filip.majerik@upce.cz

2nd Monika Borkovcova
Faculty of Electrical Engineering and Informatics
University of Pardubice
Pardubice, Czech Republic
monika.borkovcova@upce.cz

***Abstract*—The article builds on research focused on the design and construction of an Intelligent Data Layer (IDL) using Object-Relational Mapping (ORM) frameworks, which can sometimes be inefficient. The aim of this article is to verify whether IDL functions as an optimization technique in different hardware environments. Previous research has shown that IDL can significantly improve the performance of applications, especially in loading and processing large amounts of data. However, this improvement was observed on standard test hardware. Therefore, we conducted this testing, which includes tests on different hardware configurations, specifically a server and a test workstation. The tests focus on data retrieval speed, memory usage efficiency, and overall application performance when using IDL compared to traditional approaches. The results of the tests will help determine whether IDL is a universal optimization technique or whether its effectiveness depends on specific hardware conditions. This knowledge is crucial for us for the future development and deployment of IDL in real-world applications. The experiments were conducted in an environment with an Nginx web server, PHP, and a MySQL database. The tested implementation runs on Symfony and Doctrine. For the individual experiments, a relational database model from previous work was used, which included tables with varying numbers of rows.**

***Keywords*— database application architecture, ORM and performance of data layer, entity joining**

I. Introduction

ORM (Object-Relational Mapping) frameworks allow developers to map object-oriented programming to relational databases. This means that objects in the code can be directly stored in and retrieved from the database without the need to write SQL queries. ORM frameworks simplify working with databases and enable developers to work with data at a higher level of abstraction [13]. ORM frameworks are becoming increasingly popular among developers, evolving into more robust solutions for managing database data not only in desktop applications but also across a wide range of other environments, such as web and mobile applications.

These frameworks enable developers to work more efficiently with databases by providing tools for easy mapping of objects to database tables, which simplifies the process of storing and retrieving data.

This reduces the need for writing complex SQL queries and increases developer productivity. Moreover, with the growing demands for performance and scalability of applications, ORM frameworks are continuously innovating and adding new features that allow for better optimization and data management, which is crucial for modern software projects [14].

In our experiments, we address the issue of data management at the data layer using just ORM framework and optimizing data retrieval using our proposed Intelligent Data Layer (IDL). In [7], it is crucial to use systematic approaches to effectively analyze the performance of object-oriented software such as our IDL.

Maplesden et al. [8] have conducted an extensive systematic mapping of performance analysis techniques for object-oriented software, which provides a valuable overview of methods that can be applied to our IDL [3]. When optimizing performance, it is important to focus not only on improving speed, but also on identifying and removing software "bloat". Xu et al. [12] emphasize the importance of software bloat analysis for finding, removing, and preventing performance problems in modern large-scale object-oriented applications. This approach is relevant to our work with IDL, especially in optimizing data retrieval.

In our previous experiments, we focused on testing IDL with different numbers of processes, and in this paper we focus on load with different numbers of processes and different disk storage configurations. Woodside et al [11] in their paper on the future of software performance engineering stress the importance of integrating performance engineering into the entire software development lifecycle.

This perspective is particularly relevant to our research as it suggests that performance optimization, such as our IDL, should be considered from the early stages of design and development. In previous experiments, we have conducted experiments over IDL, which showed that all the parameters under study were indeed improved within a single request.[4]

For the purpose of this article, we have extended the previously presented IDL performance test to a real production environment. Thus, the originally presented environment is extended to include a separate "client" machine and a separate server that handles all requests. The connection between the computer and the server is implemented using a LAN with 1gbps throughput. Furthermore, the environment for the Symfony framework is set to "production" [9], [10].

Changing the environment for the Symfony framework in this case will cause a complete file cache to be built with the first request. This cache then speeds up the retrieval of the necessary files, so there is no need to repeatedly parse the metadata for the ORM with each request, and the dependency tree that does not need to be built with each request is resolved.

In addition, all debug messages, including logging of executed SQL queries from the ORM Doctrine, for example, are disabled within this environment [1], [2].

The impact of ORM frameworks on database query performance was comprehensively described in [15], where the authors present undesirable recurring patterns that negatively affect the performance of the data layer.

979-8-3503-8768-1/24/$31.00 ©2024 IEEE

II. Experiments

A. Foundations of Experiments

The test station chosen was a previously used machine with an Intel i7-7820X processor, 64GB DDR4 2666MHz with a Samsung 970 EVO 500GB NVME drive. A JAVA mini-application was used to run the benchmarks, which performed all the experimental requests to the server and then collected the necessary statistical data and aggregated the necessary results. The server environment for the experiments was built on a DELL PowerEdge R630 server. The server used 2x Intel Xeon E5-2620 v3 processor with a base frequency of 2.4GHz and 128GB DDR4 1866MHz. Samsung SSD 870 500GB was used as system disk. PERC H730 Mini 1GB cache was used as RAID Controller.

The following disks were used to store the database data:

- 3x Seagate ST300MM0008 300GB 2.5" Enterprise SAS HDDs 10k RPM and 128MB cache, connected via 12 Gbps SAS interface
- 3x Verbatim Vi550 S 512GB SSDs (rev. U0506A0), connected via 6 Gbps SATA interface

The operating system for the server was chosen to be Debian GNU/Linux 12 (bookworm). In addition, Docker was installed. The previously presented environment, which was ported from earlier experiments, was then run within Docker. The environment consists of a Nginx web server (1.27.0), PHP (8.1.29) running as FPM processes and a MySQL database (8.0.26 Community). The tested implementation then runs on Symfony 6.3 and Doctrine 2.10.

All application and database data was synchronized using rsync between the source client machine and the server so that all disks contained the same data. The relational database model from earlier work was used to perform the experiments [7]. The experiment builds on previous result and includes several database tables, each with a specific number of rows. The 'brand' table contains 36 rows, the 'device_type' table has 3 rows, the 'device_profile' table includes 4 rows, and the 'operator' table consists of 10 rows. Additionally, the 'subscriber' table holds 311,847 rows, while the 'device' table contains 1,480,661 rows. As in the previous experiments, the experiment was based on a model with a native SQL query through ORM and on models with ORM for Lazy Loading, Eager Loading, and implemented IDL.

Lazy Loading and Eager Loading are two different approaches to data loading that are often used in programming and database management. Lazy Loading is a technique that delays the loading of data until it is actually needed. This approach can improve application performance by minimizing the amount of data loaded into memory and reducing initial load time. Lazy Loading is particularly useful in cases where large datasets or complex objects are involved, which may not always be needed [5]. Eager Loading is a technique that loads all necessary data at once, usually upon the first access to the data. This approach can be advantageous if all data will be needed, as it reduces the number of database queries and can improve application performance in situations where query latency is high [6]. For testing parallel performance, as in the previous experiment, testing was conducted on web workers (WW) and php-fpm-processes (PFP) in combinations of 2+2, 4+4, 8+8, 16+16, 32+32, 64+64.

In accordance with previous testing, we adhered to the originally set limits of results, also to allow for comparison based on dataset size, these limitations are 10^{n}; n=1,2,3,4,5. All of the combinations were then tested with different source disks for database and application data. This was done in the following configurations:

- 1x SAS HDD ST300MM0008
- 1x SATA SSD Vi550
- RAID1 SAS HDD ST300MM0008
- RAID1 SATA SSD Vi55O

The disk configurations were chosen so that the effectiveness of each disk configuration on overall performance in parallel polling could be easily compared. Tests were then performed for all of the above disk configurations using all of the above combinations of models, server worker and process settings, and varying numbers of records in the resulting dataset. The aim of testing with various disk configurations is primarily to determine whether SSDs, which are generally faster than HDDs, and RAID configurations have an impact on the overall performance of the IDL. This can also help determine whether the higher costs of SSDs or RAID configurations provide sufficient benefits in terms of performance and reliability. At the same time, we wanted to test whether different disks might have varying resilience to load. Testing will help to find out how the IDL behaves under high load with different disk configurations and whether it is able to maintain the required performance. To test the performance of IDL on different disk configurations, the experiments presented earlier were used. For testing, we used the same measurement units, the same combination of outputs, and the native SQL query as in the second paper at this conference titled "Performance Testing of Intelligent Data Layer". In addition, other values were collected on the server side. These were the total query execution time, i.e. from the start of the request on the application side to the download of the complete response from the server, and then the measurement of the number of queries executed per second, subsequently, this data was aggregated on the client side.

B. Results of Experiments

Because there were 120 combinations of parameters in the output, only selected outputs will be shown bellow.

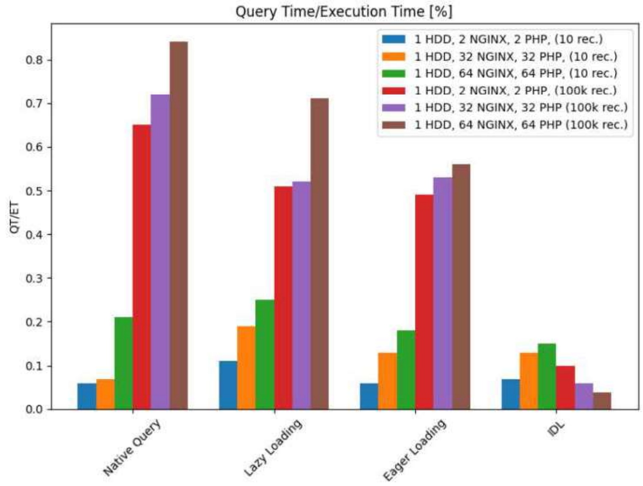


Fig. 1. Summary of QT/ET – 1xHDD

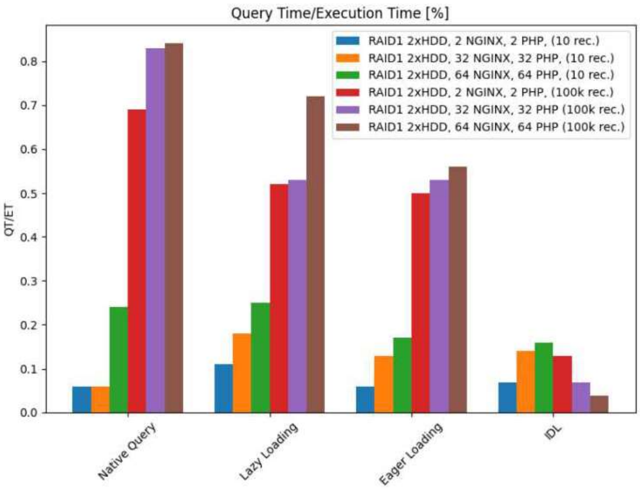


Fig. 2. Summary of QT/ET – RAID 1(2xHDD)

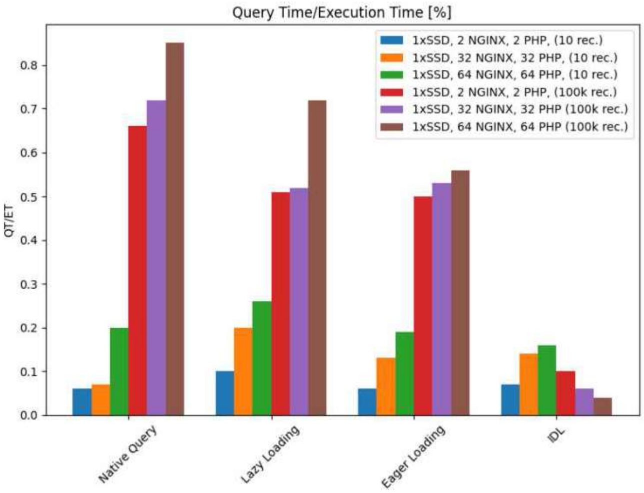


Fig. 3. Summary of QT/ET – 1xSSD

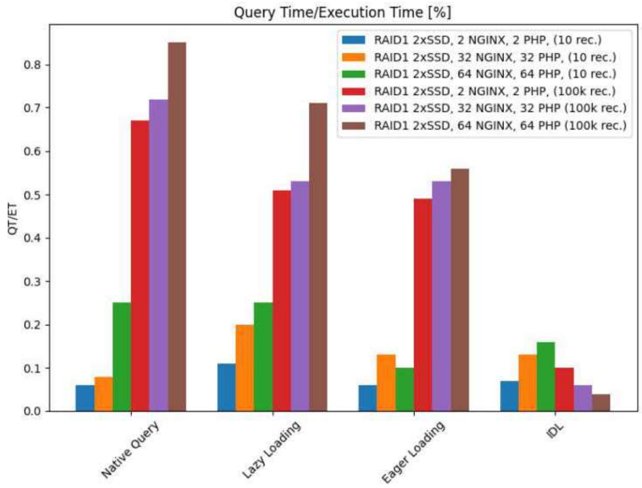


Fig. 4. Summary of QT/ET – RAID 1 2xSSD

III. Discussion of Results

Network graphs were used to visualize the observed data. They show very well the behaviour of the tested environment with an immediate comparison of the different configurations of processes, disks and the model used.

A. *Visualization of execution time vs. number of processes, experiment and disk configuration*

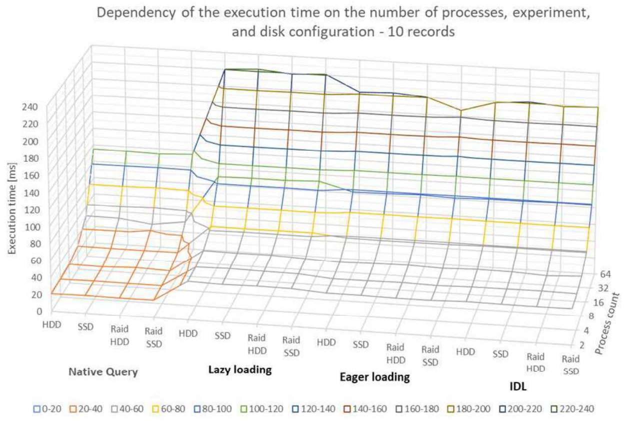


Fig. 5. Summary experimental results for 10 records

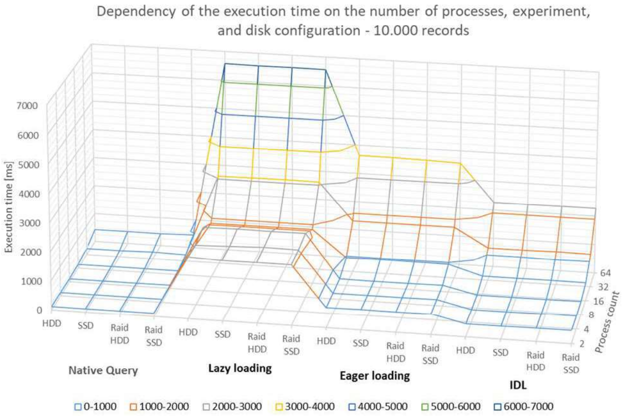


Fig. 6. Summary experimental results for 10.000 records

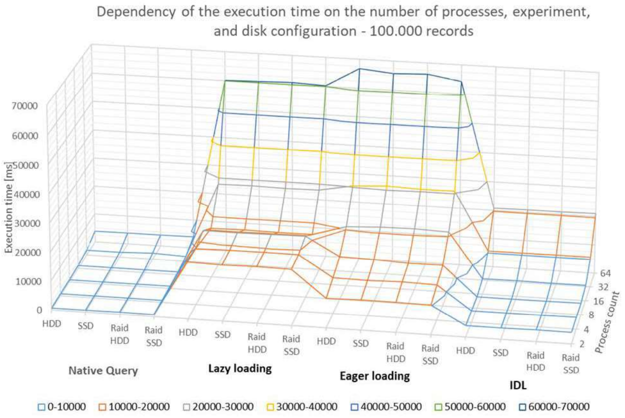


Fig. 7. Summary experimental results for 100.000 records

B. *Visualization of database query execution time vs. number of processes, experiment and disk configuration*

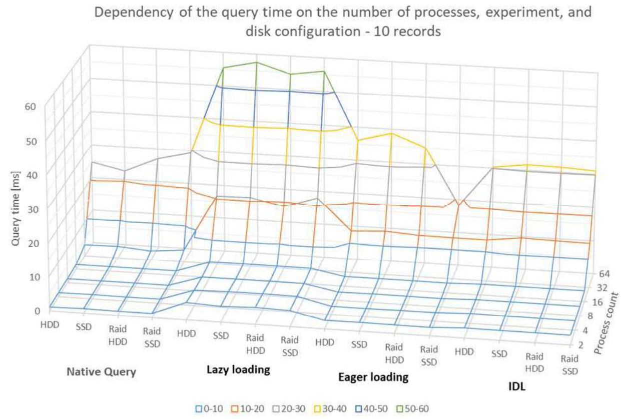


Fig. 8. Summary experimental results for 10 records

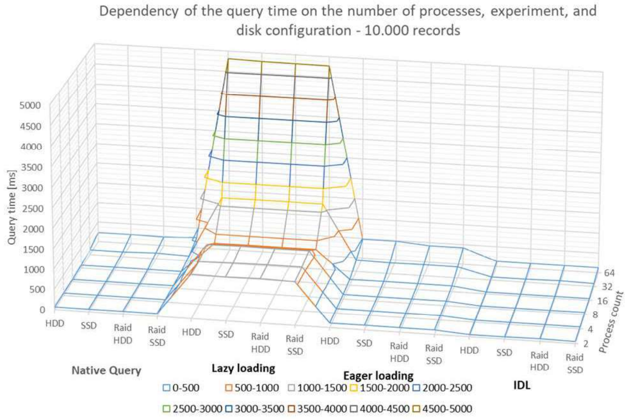


Fig. 9. Summary experimental results for 10.000 records

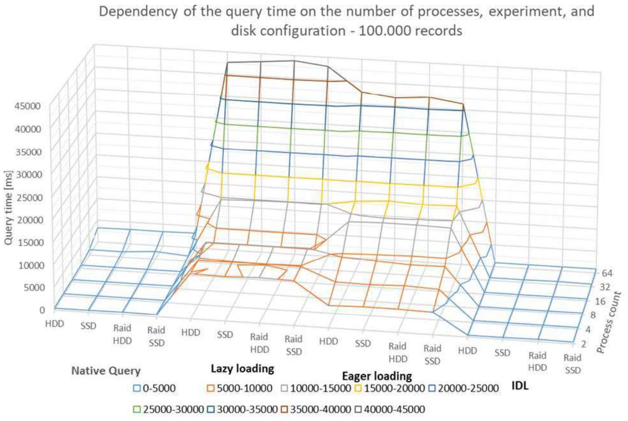


Fig. 10. Summary experimental results for 100.000 records

C. *Visualization of the number of requests processed depending on the number of processes, experiment and disk configuration*

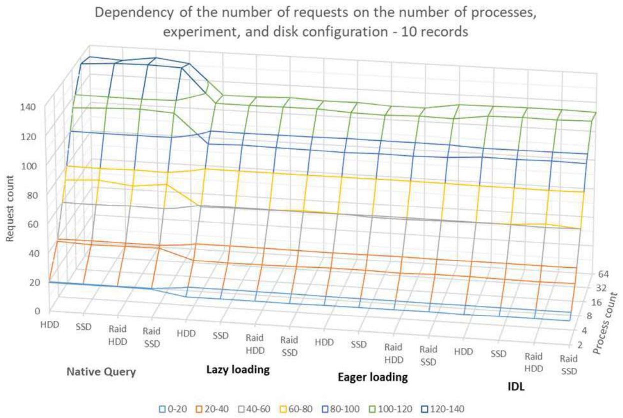


Fig. 11. Summary experimental results for 10 records

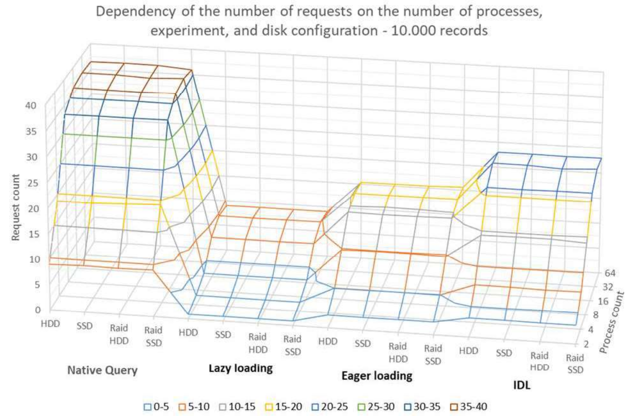


Fig. 12. Summary experimental results for 10.000 records

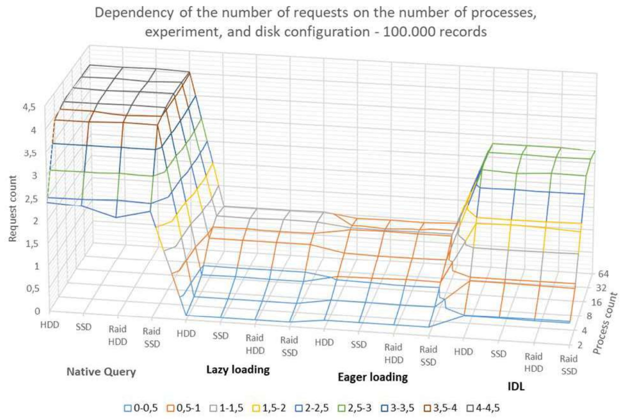


Fig. 13. Summary experimental results for 100.000 records

From the above visualizations, it is easy to read information about the behavior of IDL and very easy to compare which solution with which configuration is the most suitable. The network plots shown always represent the dependency of the measured variable on the disk configuration, the type of model experiment and the number of server processes.

The first network graphs (Figs. 5-7) show the dependence of request execution time on the number of processes, experiment and disk configuration (less is better). From the visualized results, it is very clear that different disk configurations, have little or no effect on the resulting execution times. The difference in execution times for 10 records and 2 processes was ±0.5~1ms (±1~2.5%) for all experimental variants. Such a difference can be considered almost an error related to network usage or additional load on the server side. For the 64 processes, these differences were in the range of ±2~17ms. However, when converted to percentages, we arrive at a result of 1.6~9%. The highest variation in execution time was observed when using RAID1 SSD and the Eager Loading experiment. There was no fluctuation in the other three experiments. To verify that this was not a "coincidence", this measurement was repeated 3 times each time with similar results. However, if we look at these numbers in terms of the number of processes we would easily come to the conclusion that the Lazy Loading method is the worst in terms of execution times. Eager loading and IDL were almost comparable for 10 records. Furthermore, it is certainly interesting to note that even with such a small number of data to output, the execution time between 16, 32, 64 processes increases substantially. We are thus able to observe, between 16 and 64 processes, an increase, even in the case of native querying.

If we then look at the graph (Fig. 7) for 100,000 records, we can see the extreme increase in execution time when using Eager loading. At the same time, we can then see the high efficiency of IDL, which needed approximately one third of the execution time to process queries against Lazy and Eager loading in all cases. Here again we can observe a large increase in execution time for all methods between 16, 32 and 64 processes. Consistent with the 10 records, a decrease in execution time was observed when using RAID1 SSD in combination with Eager Loading. The difference between the highest and lowest values was 2460ms (≈ 0.38%).

With further visualizations (Figs. 8-10), we can look at the result in terms of execution time of database queries (less is better). Again, we discuss the experiments for 10 records and then from the other end for 100,000 records.

For 10 records, we can see in the data that they behave less "stable" against execution times. There are much larger percentage fluctuations between the minimum and maximum values of the execution time of database queries than for the previous execution time queries. For example, in the case of Eager loading, the difference between the average fastest and worst database query processing time was 20.51ms with 64 processes. This represented a difference of ≈47%. For the other experiments, even with a different number of processes, such an extreme difference did not occur. The other differences ranged from 0.1~3.12ms. Which amounted to ≈3.5~16%. In terms of the number of processes, again extreme increases in time can be observed between 16, 32 and 64 processes, and again for the native query experiment. This increase was 628% between 16 and 32 processes. For IDL, it was 884%. In the case of Lazy loading, it was even 1112%. We attribute these extreme increases mainly to the extreme CPU load, with an average 5-minute CPU load of 37.8 (for 24 threads) during the test.

For 100,000 records, we can observe the same behavior for Eager loading as for execution times. Here too, we can observe an extreme increase in the time required to process database queries against 10,000 records. Thus, as in the previous case, Eager loading cannot be recommended for large datasets. However, IDL performed very well for these values and beat even the native query approach in absolutely all combinations. The average query execution time with 64 processes and IDL was 907.19ms. However, the approach with native query required 3316.29ms on average. For Eager loading, the measured values were approximately 41 times higher than for IDL. For Lazy loading, the values were even almost 48 times higher. The interesting point here is that for IDL, the execution time required did not increase between 16 and 32, nor between 32 and 64 processes as it did for the other experiments. This too can be considered a very good result. Thus, the database server was not subject to extremely increased load as in the other cases.

The most important results in terms of performance are then those focused on the number of requests handled depending on the experiment, the disk configuration and the number of processes (more is better). Here too, IDL and Eager loading were expected to perform very similarly. At a glance, we can see that for 10 records, indeed IDL, Eager loading and lazy loading performed almost identically. For all the experiments mentioned, the measured number of requests handled for the 2 threads oscillated between 16 - 16.5 requests. Through the native query, then, ~4 more requests were handled on average. Here again, interesting behavior can be observed between 32 and 64 processes, where in almost all cases the number of cleared requests deteriorated. It can be assumed that this is due to "server overload", where requests required much more database and execution time and thus there was no further increase in the number of cleared requests. IDL did not perform extremely better against Eager and Lazy loading for any number of processes. It can be seen from the table that up to 16 processes were at most 1 to 2 requests, and for 32 and 64 processes, 3 to 5 requests. Thus, it can be evaluated that there is no winner among the selected experiments in terms of number of requests handled for 10 records.

For 100,000 records, the differences between the experiments are already more interesting. At first glance, it can be seen in the graph that the experiment with native querying reached its maximum results very quickly and there was almost no more increase between 4, 8, 16, 32 and 64 processes. On average, it was an improvement of 0.12 req./s. However, which directly indicates that the number of processes played almost no role anymore with such a large dataset. For lazy loading we can talk about an increase of ~900% with respect to 2 processes when using 32 processes, but in the result we are still talking about approximately 1 processed query per second. Eager loading is better off with a percentage increase between 2 and 32 processes, but the number of requests handled for 32 threads still stabilizes at around 1 request per second. The differences between in disk configurations are again negligible. IDL then managed 0.48 requests per second for 2 processes and the values stabilized at 2.8 requests per second for 16 processes. The increase at 32 processes was already less than 0.1 requests, and at 64 processes, as with Eager loading and Lazy loading, there was a deterioration. However, IDL was able to handle 185% (+1.87 req./s) more requests than Eager loading and 165% (+1.78 req./s) more requests than Lazy loading at 32 processes. IDL then performed very well when compared to native querying, handling only 29% fewer requests. Here again, we can talk about IDL being a suitable optimization technique for large datasets.

IV. Conclusion

From these results and visualizations, it is very easy to state that IDL can be considered as an optimization technique. With a small number of data, it achieves the same values as the "natively" used ORM techniques Eager loading and Lazy loading in almost all the parameters under study.

For large datasets, it beats even native querying in some of the monitored parameters, e.g. the parameter of time required for processing database queries discussed here. The results also show that the different disk configurations used had minimal or almost no effect on the improvement or deterioration of the parameters. Fundamentally, however, these parameters were affected by the number of processes running in parallel, with throttling on the CPU side occurring at high load on the server under test and thus the measured values were fundamentally affected for the worse. The work could be further extended by automated IDL compilation, e.g. by reading ORM annotations, or by focusing on improving data processing on the IDL side.

Acknowledgment

It was supported by the Erasmus+ project: Project number: 2022-1-SK01-KA220-HED-000089149, Project title: Including EVERyone in GREEN Data Analysis (EVERGREEN) funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the Slovak Academic Association for International Cooperation (SAAIC). Neither the European Union nor SAAIC can be held responsible for them.

Co-funded by the European Union

EverGreen DATA Analytics

References

[1] R, Anido, AR. Cavalli, LP. Lima Jr, N. Yevtushenko, “Test suite minimization for testing in context”, 2003, Software Testing, Verification and Reliability, 13(3): 141–155.

[2] J. Armas, P. Navas, T. Mayorga, P. Rengifo and B. Arévalo, “Optimization of code lines and time of access to information through object-relational mapping (ORM) using alternative tools of connection to database management systems (DBMS),” 2017 2nd International Conference on System Reliability and Safety (ICSRS), Milan, Italy, 2017, pp. 500-504, doi: 10.1109/ICSRS.2017.8272872.

[3] R. Garcia and M. T. Valente, “Object-Business Process Mapping Frameworks: Abstractions, Architecture, and Implementation,” 2014 IEEE 18th International Enterprise Distributed Object Computing Conference, Ulm, Germany, 2014, pp. 160-169, doi: 10.1109/EDOC.2014.30.

[4] G. K. Kaminski and P. Ammann, “Using Logic Criterion Feasibility to Reduce Test Set Size While Guaranteeing Fault Detection,” 2009 International Conference on Software Testing Verification and Validation, Denver, CO, USA, 2009, pp. 356-365, doi: 10.1109/ICST.2009.14.

[5] H. Guo et al., “Lazy-WL: A Wear-aware Load Balanced Data Redistribution Method for Efficient SSD Array Scaling,” 2021 IEEE International Conference on Cluster Computing (CLUSTER), Portland, OR, USA, 2021, pp. 157-168, doi: 10.1109/Cluster48925.2021.00030.

[6] P. Rani, J. Zellweger, V. Kousadianos, L. Cruz, T. Kehrer and A. Bacchelli, “Energy Patterns for Web: An Exploratory Study,” 2024 IEEE/ACM 46th International Conference on Software Engineering: Software Engineering in Society (ICSE-SEIS), Lisbon, Portugal, 2024, pp. 12-22.

[7] F. Majerik and M. Borkovcova, “Design of Data Access Architecture Using ORM Framework,” 2023 34th Conference of Open Innovations Association (FRUCT), Riga, Latvia, 2023, 93-99. doi: 10.23919/FRUCT60429.2023.10328151.

[8] D. Maplesden, E. Tempero, J. Hosking and J. C. Grundy, “Performance Analysis for Object-Oriented Software: A Systematic Mapping,” in IEEE Transactions on Software Engineering, vol. 41, no. 7, pp. 691-710, 1 July 2015, doi: 10.1109/TSE.2015.2396514

[9] Y. Marchuk, I. Dyyak and I. Makar, “Performance Analysis of Database Access: Comparison of Direct Connection, ORM, REST API and GraphQL Approaches,” 2023 IEEE 13th International Conference on Electronics and Information Technologies (ELIT), Lviv, Ukraine, 2023, pp. 174-176, doi: 10.1109/ELIT61488.2023.10310748.

[10] V. Mukhin, Y. Kornaga, Y. Bazaka, M. Bazaliy and A. Yakovleva, “Modified Method of Software Testing for Distributed Computer System,” 2018 IEEE First International Conference on System Analysis & Intelligent Computing (SAIC), Kyiv, UKraine, 2018, pp. 1-4, doi: 10.1109/SAIC.2018.8516724.

[11] M. Woodside, G. Franks and D. C. Petriu, “The Future of Software Performance Engineering,” Future of Software Engineering (FOSE '07), Minneapolis, MN, USA, 2007, pp. 171-187, doi: 10.1109/FOSE.2007.32.

[12] G. Xu, N. Mitchell, M. Arnold, A. Rountev and G. Sevitsky, “Software bloat analysis: Finding removing and preventing performance problems in modern large-scale object-oriented applications”, Proc. FSE/SDP Workshop Future Softw. Eng. Res., pp. 421-425, 2010.

[13] C. Richardson. “ORM in Dynamic Languages,” in Communications of the ACM. vol. 52, 0001-0782, 2009. doi: 10.1145/1498765.1498783

[14] A. Torres, R. Galante, M. S. Pimenta, and A. J. B. Martins, “Twenty years of object-relational mapping: A survey on patterns, solutions, and their implications on application design,” Information and Software Technology, vol. 82, pp. 1 - 18, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0950584916301859

[15] D. Colley, C. Stanier and M. Asaduzzaman, “The Impact of Object-Relational Mapping Frameworks on Relational Query Performance,” 2018 International Conference on Computing, Electronics & Communications Engineering (iCCECE), Southend, UK, 2018, pp. 47-52, doi: 10.1109/iCCECOME.2018.8659222.

Use of two-layer genetic programming for multidimensional symbolic regression

Jan Merta
Electrical Engineering and Informatics
University of Pardubice
Pardubice, Czech republic
jan.merta@upce.cz
0000-0002-1521-5050

Tomáš Brandejský
Electrical Engineering and Informatics
University of Pardubice
Pardubice, Czech republic
tomas.brandejsky@upce.cz
0000-0001-8647-9849

***Abstract*— This paper focuses on exploring the potential benefits and advantages or disadvantages of a two-layer approach in genetic programming. The first section describes two-layer genetic programming itself and how it differs from its basic version. The Python programming language framework DEAP was used for the implementation. The focus of the paper is also to compare the results obtained by using this two-layer genetic programming with different configurations of the parameters with ordinary basic genetic programming on different multidimensional datasets and benchmarks.**

Keywords—two-layer genetic programming, symbolic regression, multi-dimensional data, benchmarks

I. Introduction

This paper focuses on a two-layer genetic programming approach for symbolic regression on multidimensional data. Genetic programming is a technique for program evolution, and there are still no clear answers to some fundamental questions about the field. Genetic programming is a white-box machine learning technique that offers human-interpretable results, making it easier to identify dependencies between variables compared to neural networks. This makes it promising for environmental analysis, such as climate prediction and air quality analysis. However, genetic programming performance needs improvement, and a potential enhancement could be a multi-layered approach inspired by the priciples of ensemble learning.

A. *Symbolic regression*

Symbolic Regression (SR) is a machine learning method for solving regression problems that is able to provide analytical equations purely from data. Symbolic regression can reveal and explain deep relationships in the data that are invisible at first sight. It has applications in many different fields ranging from science, technology, economics to social sciences. Symbolic regression is a type of regression analysis in which a mathematical function describing a given set of data is derived. While conventional regression methods (e.g., linear, quadratic, etc.) have a predetermined independent variable(s) and attempt to adjust a combination of numerical coefficients to achieve a perfect fit, symbolic regression, on the other hand, attempts to find the parameters and equations simultaneously. It is usually implemented using evolutionary algorithms and genetic programming [1].

B. *Genetic programming*

Genetic programming (GP) [2] uses evolution to evolve computer programs. Most computer programs can be thought of as executing sequences of functions with arguments. A large number of language compilers first compile the program into a derivation tree and then generate a sequence of machine instructions that can be executed on the computer. Derivation trees are therefore a natural choice of representation for computer programs [3].

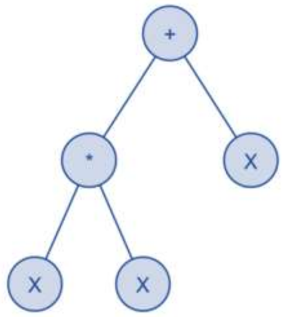


Fig. 1. Example of equation $x^2 + x$ in the form of syntactic tree

Individuals in a population of typical genetic programming are represented by a hierarchical composition of: primitive functions and terminals appropriate to the problem domain. The set of primitive functions used typically includes arithmetic operations, mathematical functions, conditional logic operations, or domain-specific functions. The set of terminals used typically includes inputs appropriate to the problem domain and various numerical constants. The composition of primitive functions and terminals corresponds directly to computer programs created in programming languages such as LISP (where they are referred to as symbolic expressions or S-expressions for short). These expressions can be represented by syntax trees (k-dimensional trees), in which all internal points and the root of the tree are labeled as functions and leaves of the tree are labeled as terminals.

To be able to apply the genetic programming method to a specific problem, we first need to define the problem at a high level. This definition can be summarized in the following steps:

1. Specification of the set of terminals.
2. Specification of the set of functions (non-terminals).
3. Selection of the fitness function - The value of this function for a given inidividual represents its probability of being selected for crossover.
4. Specification of parameters and hyper-parameters of the genetic programming run.
5. Selection of termination criteria.

In genetic programming, terminals are represented as variables or constants (from integers to Boolean values). The set of terminals can include: named variables (e.g., *x*, *y*, *speed*, etc.), constants or parameterless functions (e.g. random number, etc.).

979-8-3503-8768-1/24/$31.00 ©2024 IEEE

In genetic programming, the list of functions is governed by the type of problem. For simple numerical problems, arithmetic functions (+, -, *, /) are sufficient. Functions require a specific number of arguments, called arity. The set of functions may include: mathematical functions (+, -, *, sin, cos, log, exp, etc.), boolean operations (AND, OR, NOT), conditional operators (if, else if, else), iterative functions (do-while, for, while loops) and specific functions adapted to particular problems.

In genetic programming, the set of terminals and the set of functions should be chosen to satisfy the requirements of closure and sufficiency [4]. Closure in genetic programming requires that each function can accept any value and data type returned by another function, and the same applies to terminals. This requirement introduces two main problems: type consistency, where all functions must handle the same data type (e.g., returning 1 for true and 0 for false instead of Boolean values), and evaluation safety, which involves handling exceptions (e.g., division by zero) by creating safe versions of functions or reducing the fitness value on error. These measures ensure the correct and safe operation of the algorithm. Sufficiency in genetic programming means that a combination of different terminals and functions must be able to produce a solution to a given problem. While in some areas this identification is easy, in others it can only be assured by theoretical calculations, experience or trial and error. One big difference from most other evolutionary algorithms, if we focus on the fitness function, is in the context of genetic programming its evaluation. Since the structures generated by genetic programming are computer programs, it is necessary to run all the programs in the population to evaluate them, usually multiple times to eliminate the element of chance. For this reason, in practice, interpreters are used that execute tree nodes in such an order that a node is not executed until the values of all its arguments are known.

The genetic programming (GP) cycle is the iterative process through which programs evolve to solve a problem. It involves the following steps:

1. Initialization: Create an initial population of random programs (individuals).
2. Fitness Evaluation: Each program is evaluated based on how well it solves the given problem (its fitness).
3. Selection: Select the best-performing programs (based on fitness) for reproduction (crossover).
4. Crossover (Recombination): Combine parts of two parent programs to create new offspring, mixing their characteristics.
5. Mutation: Randomly alter parts of some programs to introduce diversity.
6. Replacement: Replace less-fit individuals with new offspring, forming a new generation.
7. Termination Check: If a stopping condition (e.g., a satisfactory solution or maximum number of generations) is met, the cycle stops; otherwise, repeat the process from Step 2.

This cycle continues until a satisfactory solution is evolved or any other termination condition is met.

The first random generation of individuals is created using these key strategies: Full Method (which creates balanced trees where all branches reach a set maximum depth, filling nodes with functions and leaves with terminals), Grow Method (which builds trees of varying shapes and sizes, allowing nodes to be functions or terminals, resulting in irregular structures, and Ramped Half-and-Half, (which combines both methods to generate a diverse population with a mix of balanced and irregular tree structures). These approaches ensure variety in the initial population, improving the evolutionary process. Once the initial generation of individuals is established, the fitness function of the individuals is evaluated and then the selection process takes place.

Selection in genetic programming is the process of choosing the best individuals (programs) from the population to become parents for the next generation. The goal is to favor individuals with higher fitness, so their traits are passed on. Common selection methods include: Tournament Selection, in which a small group of individuals is randomly chosen, and the best among them is selected as a parent, Roulette Wheel Selection, where individuals are selected based on their fitness probability, with fitter individuals having a higher chance) and Rank Selection, where individuals are ranked by fitness, and selection is based on rank, not absolute fitness.

Selection ensures that better-performing programs have a higher chance of reproducing. Crossover combines parts of two parent programs to create offspring. Common types include: Subtree Crossover (swaps random subtrees between parents), One Point Crossover (swaps parts after a single point in both parents or (Uniform Crossover: Randomly mixes nodes from both parents for more varied offspring). These methods promote diversity while preserving useful traits.

C. Two-layer genetic programming approach

Two-layer genetic programming [5] is a method that was inspired by the success of ensemble learning methods in the field of machine learning. The principle is to divide genetic programming into two layers. The combination of these layers should help genetic programming to exploit the search space and thus increase the resulting accuracy. Architecture of two-layer GP (2L-GP) is shown in the Fig. 2.

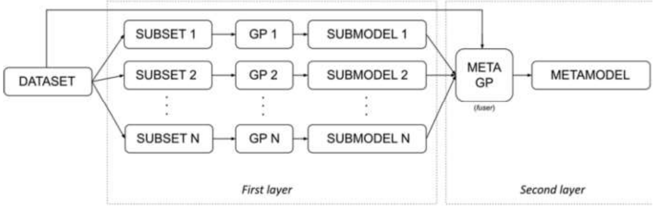


Fig. 2. Two-layer genetic genetic programming scheme

In the first layer, multiple genetic programming runs simultaneously to generate submodels that are then used as terminals in the second layer. In the second layer, these submodels obtained from the genetic programming runs in the first layer are used as a kind of building blocks that should already represent a fairly accurate solution to the problem at hand, and by combining these blocks we should achieve an even more accurate solution.

The whole two-layer genetic programming can be summarized in the following steps:

1. Creating submodels from individual independent runs of a simple GP.
2. Adding the created submodels to the terminal set of the second layer of the two-layer GP.
3. One run of the second layer using the simple GP including the added submodels.
4. Obtaining the final result from the second layer.

The two layers have independent parameters, so it is possible (and often advantageous to set these parameters differently for these layers to get better results). These parameters include, for example, the number of generations, the population size, the set of terminals, the set of features, and others.

The main motivation for developing this algorithm was to avoid bloat (the creation of extremely large trees without significantly improving the fitness value) by restarting the GP, in which the trees created from the GP runs in the first layer are used as building blocks in a new constructive way. This approach was also intended to increase the expressive power of genetic programming by allowing more efficient use of existing code and its variants. Last but not least, the possibility of parallel processing of independent GPs in the first layer and the associated speeding up of the algorithm's runtime.

D. *Benchmarks for genetic programming*

Benchmarking is the process of evaluating the performance of algorithms on a set of problems from different domains and comparing it with other algorithms. Academic researchers typically design a new algorithm and compare its performance with state-of-the-art algorithms on benchmark tests [6]. In 2012, the paper "Genetic Programming Needs Better Benchmarks" [7] was published to address the lack of good and reproducible benchmarks in genetic programming (GP). At that time, GP often used simple benchmarks taken from historical evolutionary algorithms that did not reflect real-world problems.

However, the situation has improved over the last 12 years and a large number of benchmark sets have been developed that attempt to remedy the shortcomings of the original benchmarks and thus provide a better foundation for future research [8]. A number of benchmarks have emerged that have attempted to fill this gap: PSB and PSB2 [9], SRBench [10], DIGEN [11] and symbolic regression problems from [12].

II. Experiments

This section describes the individual settings of the experiments. The Mean Squared Error (MSE) was chosen as the metric for the accuracy of the models produced (and also as fitness function for GP), and each setting was tested using 500 independent runs to produce the resulting statistics. Following sections describe individual settings of three experiments with three different benchmark functions. All of the baseline experiments used the entire dataset to create submodels, i.e., they did not create submodels from subsets of the original dataset.

A. *Simple function with two variables*

The first data set that was chosen for the approximation is the following equation (1) from [13]:

$$(x * y) * (x * y) \tag{1}$$

TABLE I. displays common parameters values for first set of experiments in the form of table. For the first function, values of the variables x and y were generated from -2 to 2 with step 0.1. All individual settings were tested by 500 independent runs.

TABLE I. Initial parameters values of experiments finding first function

Name	Value
Crossover	subtree crossover
Mutation	one point mutation
Selection	tournament selection
Tournament size	2
Elitism size	1
Crossover probability	0.9
Mutation probability	0.01
Max tree length	17
Min tree initial length	2
Max tree initial length	6
Terminal set	*x*, *y*, -1.0, 1.0, 2.0, 3.0
Function set for basic GP	+, -, *, sqrt, pow2, pow3
Function set for first layer in two layer GP	+, -, *
Function set for second layer in two layer GP	+, -, *, sqrt, pow2, pow3

The population size for the base GP was chosen to be 200 and the number of generations to be 100. For the two-layer GP, the population size was set to 50 with different values for number of generations in the first layer, and in the second layer, the population size was set to 100 or 200 and the number of generations to 10. A varying number of submodels produced by the first layer were also tested.

B. *Function with sine component*

The second data set that was chosen for the approximation is the following equation (2) with sine component from [14]:

$$(x-3) * (y-3) * 2\sin((x-4) * (y-4)) \tag{2}$$

TABLE II. displays common parameters values for second set of experiments in the form of table. For the second function, values of the variables *x* and *y* were generated from 0.05 to 6.05 with step 0.25. All individual settings were tested by 500 independent runs.

TABLE II. Initial parameters values of experiments finding second function

Name	Value
Crossover	subtree crossover
Mutation	one point mutation
Selection	tournament selection

Name	Value
Tournament size	2
Elitism size	1
Crossover probability	0.95
Mutation probability	0.05
Max tree length	8
Min tree initial length	2
Max tree initial length	6
Terminal set	*x*, *y*, -4.0, -3.0, -2.0, -1.0, 1.0, 2.0, 3.0, 4.0
Function set for basic GP	+, -, *, /, *sqrt*, *pow2*, e^x , e^{-x}, *sin*, *cos*
Function set for first layer in two layer GP	+, -, *
Function set for second layer in two layer GP	+, -, *, /, *sqrt*, *pow2*, e^x , e^{-x} , *sin*, *cos*

The population size for the base GP was chosen to be 250 and the number of generations to be 100. For the two-layer GP, the population size was set to 100 or 200 with different values for number of generations in the first layer, and in the second layer, the population size was set to 200 and the number of generations to 25. A varying number of submodels produced by the first layer were also tested.

C. *Function with more variables*

The third data set that was chosen for the approximation is the following equation (3) with three variables which was also taken from [14]:

$$30 * \frac{(x-1)*(z-1)}{y^2*(x-10)} \qquad (3)$$

TABLE III. INITIAL PARAMETERS VALUES OF EXPERIMENTS FINDING THIRD FUNCTION

Name	Value
Crossover	subtree crossover
Mutation	one point mutation
Selection	tournament selection
Tournament size	2
Elitism size	1
Crossover probability	0.95
Mutation probability	0.05
Max tree length	12
Min tree initial length	2
Max tree initial length	6
Terminal set	*x*, *y*, *z*, 5.0, -4.0, -3.0, -2.0, -1.0, 1.0, 2.0, 3.0, 4.0, 5.0
Function set for basic GP	+, -, *, /, *sqrt*, *pow2*, e^x
Function set for first layer in two layer GP	+, -, *
Function set for second layer in two layer GP	+, -, *, /, *sqrt*, *pow2*, e^x

The population size for the base GP was chosen to be 125 and the number of generations to be 100. For the two-layer GP, the population size was set to 100 with different values for number of generations in the first layer, and in the second layer, the population size was set to 225 and the number of generations to 20. A varying number of submodels produced by the first layer were also tested.

TABLE III. displays common parameters values for second set of experiments in the form of table. For the second function, values of the variables *x* and *z* were generated from -0.05 to 2.05 with step 0.15, and variable *y* from 0.95 to 1.95 with step 0.1. All individual settings were tested by 500 independent runs.

III. RESULTS

A. *Simple function with two variables*

The results of the first experiment performing two-layer genetic programming with the basic version on the simple function with two variables (1) are shown in Fig. 3:

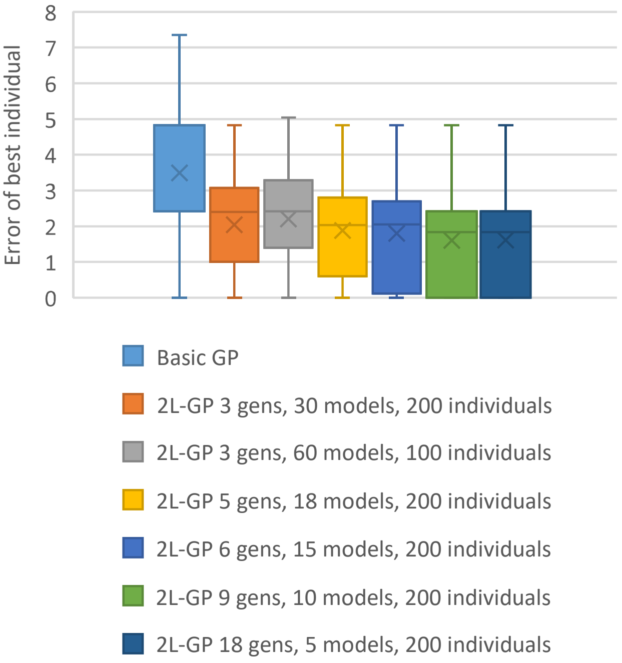


Fig. 3. Results of experiments on a simple function with two variables

The far left will always show the result using the single layer approach and the far right will always show the different configurations of the first layer parameters for the double layer approach. By comparing the results from the first function, it can be seen that the two-layer approach provided better results and was even able to find the exact function we are trying to approximate in quite a large number of cases, achieving a fitness function value of 0.

It seems that in these particular configurations, increasing the number ofgenerations in the first layer had a positive effect on the accuracy of the resulting models. Conversely, a larger number of submodels generated from the first layer did not have much effect in this case.

B. *Function with sine component*

The results of the second experiment performing two-layer genetic programming with the basic version on the function with sine component (2) are shown in Fig. 4.

In these particular configurations, increasing the number of generations in the first layer had a positive effect on the accuracy of the resulting models. Increasing the number of submodels generated from the first layer did not have much effect, but reducing the number of submodels to 5 submodels (dark grey, 9th box) somewhat limits the performance of GP, which even with 40 generations in the first layer does not generally perform better than 20 submodels generated from 10 generations (orange, 2nd box) or a configuration with 8 submodels and 25 generations in the first layer (brown, 8th box). For this function, it can be seen that the two-layer approach outperformed the single-layer approach, yielding more concentrated results with a smaller interquartile range.

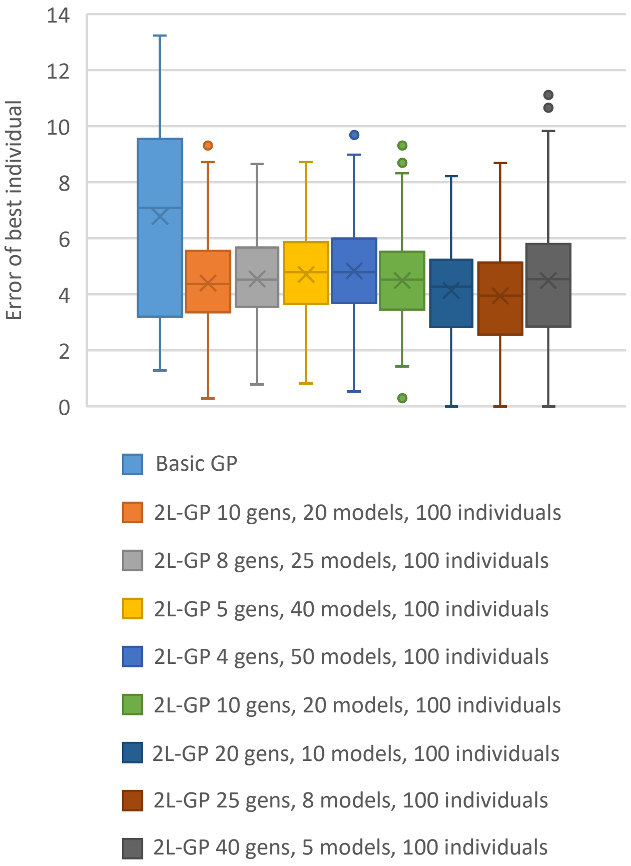


Fig. 4. Results of experiments on a simple function with two variables

C. *Function with more variables*

The results of the second experiment performing two-layer genetic programming with the basic version on the function with three variables (3) are shown in Fig. 5. In this case, it seems that the number of submodels played a larger role. Comparing box 2 (orange) with box 3 (grey), it seems that the reduction in the number of submodels, despite a small increase in the number of generations of the first layer, had a negative effect on the accuracy of the resulting models. The situation is similar when comparing box 4 (yellow) with box 6 (green) or box 4 (yellow) with box 7 (dark blue). Comparing box 4 (yellow) with box 5 (blue) is slightly more complicated, although box 5 has a smaller first quartile, box 4 has a better average and less inter-quartile variance, so the 4th configuration presents more compact and balanced results on average. It seem that 5 or 4 submodels are not enough to make up for lost performance with an increased number of generations. Nevertheless, the configurations with fewer models still had better accuracy than the configurations of the classical single-layer approach.

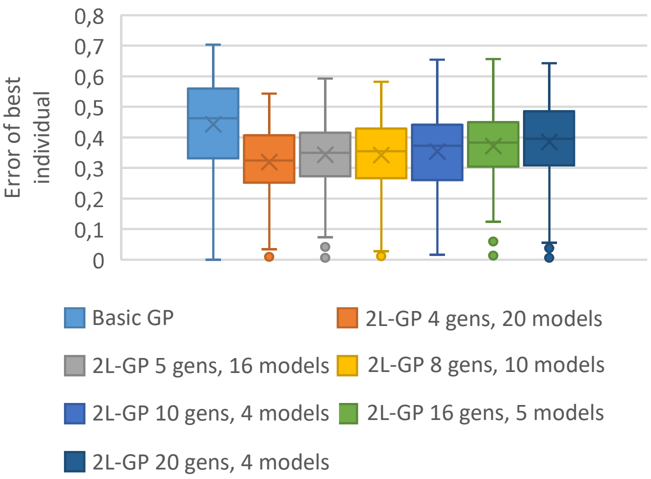


Fig. 5. Results of experiments on a function with three variables

After these experiments we also tried to compare different methods for creating data subsets for the first layer. We used bootstrapping, which made data subsets from fractions of original data. We tried bootstrapping with size of 30 %, 50 % and 70 % of original data size and we compared it with basic GP and 2L-GP without bootstraping (20 submodels, 4 generations and 100 individuals in the first layer).

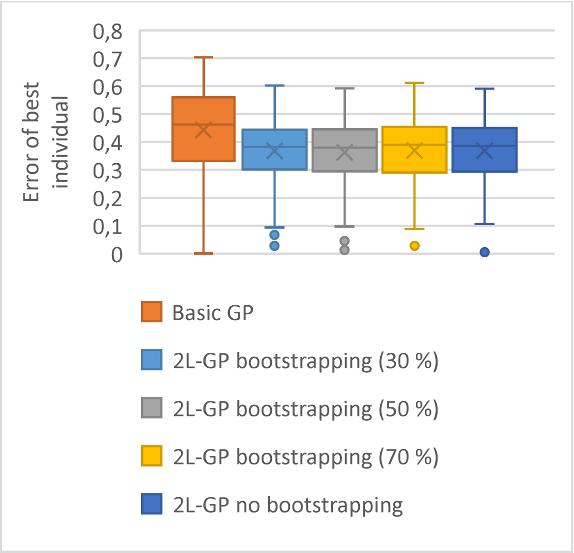


Fig. 6. Comparison of different bootstrapping percentages for a configuration with 4 generations and 20 models in the first layer (3rd function)

TABLE IV. COMPARISON OF RUN TIMES WITH DIFFERENT BOOTSTRAPPING SIZES (THIRD FUNCTION)

Bootstrap percentage	Elapsed time (minutes)
30 %	269
50 %	272
70 %	294
No bootstrapping (whole dataset) size	300

As you can see in TABLE IV. , all 2L-GP variants gave us very similar results. But when we compared the elapsed time of each bootstrapping experiment (300 runs), we saw noticeable differences between the configurations.

IV. Discussion

The aim of this paper was to explore the dynamics and benefits of two-layer genetic programming on multi-dimensional data data. Therefore, experiments were designed using recommended benchmark functions that include multiple variables as suggested by selected publications. This comparison was tested on a total of three symbolic regression problems. For a total of three of these problems, it can be directly stated that the two-layer approach achieved better results than the single-layer approach when using specific configurations.

The two-layer approach brought with it several advantages. One of the main advantages is the ability to simply parallelize the first layer. Thus, the runs of two-layer genetic programming are able to use the processor very efficiently, making the overall time consumption less and the runs shorter compared to the single-layer approach. This fact can be further enhanced by introducing some form of resampling. Another advantage that was observed for all problems tested is the smaller interquartile ranges of the best individuals produced by two layer genetic programming. Due to this fact, it can be stated that the results obtained from the two-layer approach are more consistent and often fall in a smaller interval than those obtained from the basic single-layer architecture.

V. Conclussion

In the context of environmental analysis, genetic programming (GP) stands out as a valuable machine learning method due to its transparency and ability to produce interpretable results. Its advantage over black-box models, like neural networks, lies in the ease of identifying relationships between environmental variables, which is critical for applications such as climate prediction and air quality analysis. The proposed two-layer GP could overcome the shortcomings of the basic GP and thus improve the application of GP to environmental analytics and data science in general.

In future research on the two-layer approach, it would be useful to try other types of problems other than symbolic regression. There is a wide range of problems that genetic programming can be applied to, and it would be good to see if a two-layer architecture would be as successful as it has been on the tested symbolic regression problems. But this does not mean that all problems related to symbolic regression are solved by this paper. Even the problems that have been solved in this paper using the two-layer approach probably contain room for improvement.

Acknowledgment

It was supported by the Erasmus+ project: Project number: 2022-1-SK01-KA220-HED-000089149, Project title: Including EVERyone in GREEN Data Analysis (EVERGREEN) funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the Slovak Academic Association for International Cooperation (SAAIC). Neither the European Union nor SAAIC can be held responsible for them.

Special thanks belongs to Václav Hrubý for their essential lab work and valuable contributions to this research. Their support was crucial to the success of this study.

References

[1] D. Angelis, F. Sofos, T. E. Karakasidis, “Artificial Intelligence in Physical Sciences: Symbolic Regression Trends and Perspectives”. Arch Computat Methods Eng 30, 2023, pp. 3845–3865. https://doi.org/10.1007/s11831-023-09922-z

[2] J. R. KOZA, “Genetic programming: On the programming of computers by means of natural selection”. Cambridge: Bradford Book, 1992. ISBN 978-0262111706

[3] M. L. WONG, K. S. Leung, “Data Mining Using Grammar Based Genetic Programming and Application”. 2nd Edition. Springer, 2000. ISBN 978-0792377467.

[4] R. Poli, W. B. Langdon, N. F. McPhee, J. R. KOZA, “A field guide to genetic programming”. Lulu Press, 2008. ISBN 1409200736.

[5] J. Merta, T. Brandejský, Two-layer genetic programming. Online. Neural Network World. 2022, vol. 32(4), pp. 215-231. ISSN 23364335. Available: https://doi.org/10.14311/NNW.2022.32.013.

[6] J. Woodward, S. MARTIN, J. Swan, “Benchmarks that matter for genetic programming”. Proceedings of the Companion Publication of the 2014 Annual Conference on Genetic and Evolutionary Computation. 2014-07-12, vol. 2014, pp. 1397-1404. ISBN 9781450328814. DOI: 10.1145/2598394.2609875, Available: https://doi.org/10.1145/2598394.2609875.

[7] J. McDermott, D.R. White, S. Luke, L. Manzoni, M. Castelli, L. Vanneschi, et al. “Genetic programming needs better benchmarks”. Proceedings of the 14th annual conference on Genetic and evolutionary computation. 2012, vol. 2012, pp. 791-798. ISBN 9781450311779. Available: https://doi.org/10.1145/2330163.2330273.

[8] J. McDermott, G. KRONBERGER, P. Orzechovwski, L. Vanneschi, L. Manzoni, et al. “Genetic programming benchmarks: Looking back and looking forward”. ACM SIGEVOlution. 2022, vol. 15(3), pp. 1-19. ISSN 1931-8499. DOI: 10.1145/3578482.3578483, Available: https://doi.org/10.1145/3578482.3578483.

[9] T. Helmuth, P. Kelly, “PSB2: The Second Program Synthesis Benchmark Suite”. Proceedings of the Genetic and Evolutionary Computation Conference. 2021, vol. 2021, pp. 785-794. Available: https://doi.org/10.48550/arXiv.2106.06086.

[10] CAVA LAB. “SRBench: A Living Benchmark for Symbolic Regression”. Online. CAVA LAB. SRBench. 2018. Available: https://cavalab.org/srbench/.

[11] P. Orzechovwski, J. H Moore,. “Generative and reproducible benchmarks for comprehensive evaluation of machine learning classifiers”. Online. Science Advances. 2022, vol. 8(47). Available: https://doi.org/10.48550/arXiv.2107.06475.

[12] HEURISTICLAB. HeuristicLab A Paradigm-Independent and Extensible Environment for Heuristic Optimization. Online. HEURISTICLAB. Symbolic Regression Benchmark Functions. 2012. Available: https://dev.heuristiclab.com/trac.fcgi/

[13] A. Hintze, J. Schossau, C. Bohm. The Evolutionary Buffet Method. In: Banzhaf, W., Spector, L., Sheneman, L. (eds) Genetic Programming Theory and Practice XVI. Genetic and Evolutionary Computation. Springer, Cham. 2019. Available: https://doi.org/10.1007/978-3-030-04735-1_2

[14] E. J. Vladislavleva, G. F. Smits and D. den Hertog, "Order of Nonlinearity as a Complexity Measure for Models Generated by Symbolic Regression via Pareto Genetic Programming," in IEEE Transactions on Evolutionary Computation, vol. 13, no. 2, pp. 333-349, April 2009, doi: 10.1109/TEVC.2008.926486

The Rise of Generative Artificial Intelligence in Business

Diana Mudrinić, Ms.Sc.
Chief executive officer
Incubator for new technologies Trokut Šibenik LLC.
Šibenik, Croatia
diana@trokut.eu

Ivan Šoda, Ms.Sc.
Assistant to director for information technologies and education
Incubator for new technologies Trokut Šibenik LLC
Šibenik, Croatia
ivan@trokut.eu

***Abstract*— Generative artificial intelligence (hereinafter: AI) has emerged as a transformative force across industries, driving significant advancements in automation, efficiency, and innovation. This paper explores the impact of generative AI on business operations, focusing on its ability to enhance productivity, reduce costs, and improve customer experiences. By leveraging foundation models such as GPT-3.5, businesses are automating tasks that previously required human intervention, leading to substantial cost savings and operational efficiencies. Furthermore, this paper discusses the ethical implications of AI deployment, particularly in the areas of data privacy, algorithmic bias, and accountability. The paper also addresses the future trends in generative AI, including the development of multimodal models and their potential applications. This study provides a comprehensive overview of how businesses can harness generative AI to gain a competitive edge while mitigating the associated risks.**

Keywords— Generative AI, business transformation, automation, productivity, GPT-3.5, data privacy, algorithmic bias, multimodal AI, innovation.

I. Introduction

Generative Artificial intelligence (AI) has brought profound changes to how businesses operate by fundamentally altering the dynamics of content creation, decision-making, and interaction. At its core, generative AI systems such as OpenAIs GPT-3.5 are models trained on vast amounts of data, which enable them to create human-like text, music, images, and even video. The leap from assistive AI, designed to support specific tasks, to generative AI is the equivalent of transitioning from basic automation to systems capable of originating creative outputs [1].

As businesses have sought to leverage AI in their operations, the generative capacity of these models has shifted from being a cutting-edge novelty to an integral part of how companies approach problem-solving, marketing, customer service, and research. By democratizing access to powerful AI tools, generative AI has made complex operations, once the purview of experts, accessible to general users [1].

Startups no longer need to invest heavily in infrastructure or hire large teams to handle tasks like data analysis, marketing, or customer support. Instead, they can leverage AI tools to automate these processes, allowing entrepreneurs to focus on product development, business strategy, and scaling their operations.

The integration of AI in business processes is reshaping the workforce. While there are concerns about job displacement, AI is also creating new roles and opportunities. Entrepreneurs are focusing on reskilling and upskilling their teams to work alongside AI, fostering a more dynamic and adaptable workforce.

This accessibility has driven rapid adoption across industries such as healthcare, finance, media, and retail. The AI landscape is evolving at a pace that businesses must keep up with or risk falling behind in an increasingly competitive marketplace [1].

II. THE MOST COMMON TYPES OF GENERATIVE AI MODELS

Many types of generative AI models are in operation today, and the number continues to grow as AI experts experiment with existing models. Among the many types of generative AI models are text-to-text generators, text-to-image generators, image-to-image generators, and image-to-text generators. It's possible for a model to fit into multiple categories—for example, the latest updates to ChatGPT and GPT-4 make it a transformer-based, large language, and multimodal model. Some of the most common types include the following:

Generative Adversarial Networks (GANs): Best for image duplication and synthetic data generation. The basic principle involves pitting two different algorithms against each other. One is known as the 'generator,' and the other is known as the 'discriminator,' and both are given the task of getting better and better at out-foxing each other. The generator attempts to create realistic content, and the discriminator attempts to determine whether it is real or not. Each learns from the other, becoming better and better at its job until the generator knows how to create content that's as close as possible to being 'real.' [12]

Diffusion Models: Best for image generation and video/image synthesis. Diffusion models are widely used in image and video generation, and work via a process known as 'iterative denoising'. Starting from a text prompt which the computer can use understand what it has to create an image of, random

979-8-3503-8768-1/24/$31.00 ©2024 IEEE

'noise' is generated – you can think of this as starting to draw a picture by scribbling randomly on a piece of paper. [12]

Large Language Models: Designed to generate and complete written content at scale, these are the most popular and well-known type of generative AI model right now. Fundamentally, they are neural networks that are trained on huge amounts of text data, allowing them to learn the relationship between words and then predict the next word that should appear in any given sequence of words. They can then be further trained on specific texts related to specialized domains – known as 'fine-tuning' to enable them to carry out specific tasks. [12]

Neural Radiance Fields (NeRFs): Emerging neural network technology that can be used to generate 3D imagery based on 2D image inputs. Unlike the other generative technologies, they are specifically used to create representations of 3D objects using deep learning. This means creating an aspect of an image that can't be seen by the 'camera' – for example, an object in the background of an image that's obscured by an object in the foreground or the rear aspect of an object that's been pictured from the front. [12]

III. KEY IMPACTS OF GENERATIVE ARTIFICIAL INTELLIGENCE

A. Enhanced Efficiency and Productivity

Generative AI significantly enhances productivity by automating various time-consuming and repetitive tasks that were previously done by humans. In marketing, for instance, AI-driven platforms are creating customized advertising campaigns tailored to specific audiences without human intervention. This personalization happens at a scale and speed unimaginable a few years ago [3].

In customer service, AI-powered chatbots, such as those used by companies like Amazon and Alibaba, can handle millions of customer queries per day with minimal human oversight. These systems use natural language processing (NLP) models like GPT-3 to understand customer intent, provide accurate responses, and even upsell products based on the user's history and preferences [3]. This dramatically reduces operational costs while increasing customer satisfaction and engagement.

Even in highly specialized industries like law and medicine, AI tools are proving invaluable. Legal firms are using AI to draft contracts, analyze legal documents, and search for case precedents, allowing human lawyers to focus on more complex legal strategies. In healthcare, generative AI assists in diagnosing diseases by analyzing medical records, imaging scans, and even genetic data. These applications are pushing the boundaries of human productivity and effectiveness [4].

B. Cost reduction

The financial benefits of generative AI cannot be overstated. By automating manual, repetitive tasks, companies can drastically reduce their labor costs. A McKinsey report found that companies employing AI could see reductions in operating costs by up to 30% in certain functions, including customer service, back-office operations, and logistics [2].

Generative AI has also changed the game in terms of content production. AI-driven platforms like Jasper and Writesonic allow businesses to generate marketing content, blog posts, social media updates, and even product descriptions with minimal human input [2]. This not only cuts down on the time spent on these tasks but also reduces the need to hire additional staff for content creation. Companies like Canva have integrated AI tools that help users design graphics, presentations, and social media posts, further streamlining operations and reducing costs associated with creative production [3].

In industries such as manufacturing and supply chain management, generative AI models predict demand, manage inventories, and optimize production schedules. This reduces waste, shortens lead times, and ensures that resources are allocated efficiently. These applications demonstrate how AI can significantly cut operational costs across various sectors [4].

C. Improved Customer Experience

AI's ability to analyze vast amounts of data in real time has revolutionized the customer experience. Businesses can now use AI to offer highly personalized recommendations, interact with customers through chatbots, and resolve customer queries instantly. Companies like Netflix, for example, use AI algorithms to recommend shows and movies based on users' viewing habits, creating a more engaging and personalized user experience [2].

Similarly, in e-commerce, retailers like Amazon and Walmart leverage AI to analyze purchasing behavior and offer product recommendations tailored to individual users. This personalization extends beyond the digital space; in physical retail environments, companies are using AI to track customer behavior, optimize store layouts, and provide personalized in-store recommendations via mobile apps [2]. These developments have led to greater customer satisfaction, loyalty, and, ultimately, higher revenue.

AI's role in improving customer experience is particularly evident in the travel industry. Companies like Expedia and Airbnb use AI to provide personalized travel recommendations and streamline the booking process. AI-powered virtual assistants can help travelers find accommodations, recommend activities, and even resolve issues during their stay, all in real time [3].

D. Innovation and New Business Models

Generative AI is not only transforming existing business models but also paving the way for entirely new ones. Companies are leveraging AI to create platforms and services that did not exist a few years ago. For example, Synthesia, an AI-driven video production platform, allows businesses to create professional-quality videos with virtual presenters by simply typing a script. This eliminates the need for expensive production teams and studios [4].

In healthcare, AI is being used to analyze large datasets from clinical trials to develop new drugs. AI models can simulate how different drug compounds interact with human cells, drastically speeding up the drug discovery process. Similarly, AI-powered diagnostic tools are helping doctors identify diseases more quickly and accurately, improving patient outcomes and reducing healthcare costs [4].

In the financial sector, companies like Wealthfront and Betterment are using AI to offer personalized investment advice, allowing users to manage their portfolios with minimal human interaction. These robo-advisors analyze market trends, user risk profiles, and financial goals to provide tailored investment strategies [4]. AI is also disrupting traditional lending models by providing more accurate credit risk assessments, enabling lenders to make better-informed decisions about loans and investments [3].

IV. Workforce transformation

The advent of AI has undeniably impacted the global workforce. While AI has automated many routine tasks, leading to concerns about job displacement, it has also created new opportunities. Many companies are investing in reskilling and upskilling their employees to ensure they can work effectively alongside AI systems [3]. The future workforce will likely be one where human creativity and strategic thinking are complemented by AI-driven tools.

Keep your text and graphic files separate until after the text has been formatted and styled. Do not use hard tabs, and limit the use of hard returns to only one return at the end of a paragraph. Do not add any kind of pagination anywhere in the paper. Do not number text heads the template will do that for you.

A. Reskilling and upskilling

Reskilling programs, like those implemented by AT&T and IBM, have become a cornerstone of corporate strategy as companies prepare their workforces for an AI-driven future. AT&T's "Future Ready" initiative, for instance, provides employees with access to online courses in data science, AI, and other relevant skills. This ensures that workers who may be displaced by automation can transition to more strategic, higher-level roles [4].

In addition to technical skills, soft skills such as creativity, leadership, and emotional intelligence will become more important. AI tools can handle many analytical and repetitive tasks, but they still lack the ability to think creatively or make intuitive decisions in the same way humans do. As such, the ability to work collaboratively with AI systems will become a key skill for future employees [4].

V. Business ethics in artificial intelligence

The rapid advancement of generative AI raises several ethical questions that businesses must address to ensure responsible usage. Issues like algorithmic bias, data privacy, and the accountability of AI decisions are increasingly coming under scrutiny. Companies must balance the benefits of AI with the need for ethical oversight to avoid potential harm to users and society at large [1].

Using AI is really important to staying competitive these days. But, it's getting more obvious that many people who use generative AI don't know about the possible risks it brings. As more users incorporate generative AI into their daily routines without conducting proper assessments, the potential risks and repercussions of deploying these AI models are starting to outweigh the benefits.

In July 2023, the White House announced seven large AI firms had committed to "develop robust technical measures to ensure that users know when content is AI-generated, such as watermarking". Given that foundation AI models have started to be trained on AI-generated data, these tools will have a role to play in documenting the provenance of training data as well as the integrity of downstream outputs from AI. [10]

Generative AI models' unique attributes pose a range of risks that we don't always see with other kinds of models. Here are the risks that business leaders must keep in mind as they consider generative AI projects.

A. Ethical Concerns

Generative AI introduces ethical dilemmas that stem from its ability to create content autonomously that raises questions of fairness, bias, and accountability.

One of the foremost concerns with Generative AI is that it might make things unfairly. For example, if it learns from data that mostly shows one group of people, it might favor that group over others. This could cause unfair treatment in things like hiring or financial decisions.

A recent op-ed in the Guardian argued that companies are using 'speculative fears' to "stop people asking awkward questions about how this particular technological sausage has been made". [11]

Generative AI opens the door to the creation of deepfake content that looks real but isn't such as manipulating images, videos, or audio recordings. This can be a big serious issue because it can hurt people's reputations or spread lies that cause trouble.

B. Security Risks

In addition to ethical concerns, Generative AI presents significant security risks that threaten data privacy, cybersecurity, and the integrity of digital systems.

Generative AI systems often require access to large datasets for training, which may contain sensitive or personally identifiable information. This makes people worry about keeping their information private and the chance that someone might get into it without permission and use it the wrong way.

Malicious actors may exploit vulnerabilities in Generative AI models to launch cyberattacks, like messing with data or breaking into systems. These attacks can compromise the confidentiality, integrity, and availability of data and systems Which can pose significant risks to organizations and individuals.[14]

C. Legal Implications

Generative AI presents various legal challenges related to intellectual property rights, liability issues, and regulatory compliance.

Determining ownership and protection of intellectual property rights related to AI-generated content can be complex and contentious. This leads to questions about copyrights, who gets to use the creations, and giving credit to the right people.

Foundation models are trained on large collections of data, much of which is gathered from across the web. The training of these models "depends on the availability of public, scrapable data that leverages the collective intelligence of humanity, including the painstakingly edited Wikipedia, millennia's worth of books, billions of Reddit comments, hundreds of terabytes' worth of images, and more". [13]

In public, companies have used different arguments to justify the lack of transparency around their training data. In documentation published at the launch of its GPT-4 model, OpenAI (2023) stated that it would not share detailed information about 'data set construction' and other aspects of the model's development due to "the competitive landscape and the safety implications of large-scale models." [15]

Generative AI models help businesses by doing things automatically and making tasks easier, which helps move things forward and lets creativity flow. But, we have to understand the security issues it causes. These problems might come from uncontrolled commands, accidentally showing secret information, problems with storing data, following complicated global rules, and the chance of data getting out. Businesses should make detailed plans to build trust in AI, mitigate risks, and make AI systems more secure.

Generative AI models learn from the data they are trained on, which can sometimes include biased information. This can lead to models making biased decisions, reinforcing stereotypes, or discriminating against certain groups. For example, AI models used in hiring processes have been shown to favor male candidates over female candidates when trained on historical hiring data that reflects gender bias [5].

To mitigate these issues, companies are investing in AI fairness research and developing guidelines for ethical AI deployment. Google's "AI Fairness Principles" outline how the company aims to minimize bias in its AI models, ensuring that they make fair and unbiased decisions. [5]

AI models rely on massive datasets, which often contain sensitive personal information. As such, companies must ensure that they are complying with data privacy regulations like the General Data Protection Regulation (GDPR) in Europe and the California Consumer Privacy Act (CCPA) in the U.S. [7] These regulations require businesses to obtain consent before collecting user data and to provide users with the ability to delete their data if they choose. [6]

Data breaches are a significant concern for businesses using AI. Hackers can target AI models to exploit vulnerabilities in the data they are trained on. Therefore, companies must invest in robust cybersecurity measures to protect their AI systems and the data they process.

As AI systems become more autonomous, it is essential to establish clear guidelines for accountability. For example, in the financial sector, AI is being used to make investment decisions and assess credit risk. However, when an AI-driven decision negatively impacts a user, it is important to determine who is responsible: the developer, the user, or the AI itself. [8]

Transparency is also critical to building trust in AI systems. Users need to understand how AI decisions are made, particularly in sensitive areas like healthcare and criminal justice. Many companies are now working on developing explainable AI (XAI) systems that provide insights into how models make decisions. [5]

AI has the potential to disrupt job markets by automating tasks traditionally performed by humans. While this can lead to increased efficiency, it also raises concerns about job displacement4. Ethical considerations include developing strategies for workforce reskilling and ensuring that the benefits of AI are broadly shared. [6]

AI technologies can be misused for malicious purposes, such as creating deepfakes or automating cyberattacks. Ensuring the ethical use of AI involves developing safeguards against misuse and promoting the responsible deployment of AI technologies. [8]

VI. Future trends in generative artificial intelligence

The future of generative AI holds tremendous potential. As models become more sophisticated, businesses will have access to tools that can generate increasingly complex outputs. New developments in multimodal models, which combine text, image, and video generation, are already being used in creative industries, entertainment, and media production [4]. These models will soon become commonplace in fields like education, where they will assist in content generation and even personalized lesson plans. [4]

Generative AI will have impact on Education since they will become more adept at generating content, they will be able to be used for more specific needs and learning styles of individual students. In addition, AI-powered tutors could interact with students in real time, offering explanations, answering questions, and even generating personalized practice problems or assignments based on the student's progress.

While challenges related to ethics and regulation remain, the potential for AI to transform how we create, learn, and innovate is boundless. Businesses and industries that embrace these advancements and responsibly implement AI technologies will be well-positioned to lead in a rapidly changing digital landscape. [4]

In the broader context, the widespread use of generative AI will prompt shifts in the workforce, where roles evolve to accommodate the increasing presence of AI systems. Workers will need to acquire new skills to effectively collaborate with AI and leverage its capabilities. As generative AI systems take on more tasks, organizations will need to focus on upskilling employees to ensure that human talents are directed towards activities that require emotional intelligence, leadership, and complex decision-making—areas where AI is less effective. [8]

In the long term, the proliferation of generative AI will create a gradual but significant evolution in the way industries operate. As businesses learn to incorporate AI into their workflows, they will discover new ways of innovating, problem-solving, and creating value. This evolution will be characterized by a more integrated and symbiotic relationship between humans and machines, where each complements the other's strengths. [8]

The future of generative AI promises to be one of transformation and opportunity. As these models become more powerful and integrated across various industries, they will redefine how businesses approach challenges, drive innovation, and create value. While the path forward presents certain challenges—particularly in terms of ethics, governance, and workforce adaptation—the potential benefits of generative AI will far outweigh these hurdles. The organizations that embrace this technology responsibly will be at the forefront of the next wave of industrial and creative evolution.

VII. Is artificial intelligence just hype or a proper change

We can talk about artificial intelligence (AI). Is it merely a trendy term in the tech business, or is it a concept that every Product Manager should be giving significant attention to? The reality is that AI is not simply a superficial buzzword but rather a transformative force that can completely transform how we develop and oversee products. However, there is a condition – to utilize AI efficiently, we must abandon certain conventional methods and adopt some lesser-known tactics. [8]

One of the main issues with AI hype is that it creates unrealistic expectations among the public and investors. When companies make bold claims about their AI-powered products or services, they often fail to deliver on those promises, leading to disappointment and erosion of trust. [8]

Like many new technologies, generative AI has been following a path known as the Gartner hype cycle, first described by American tech research firm Gartner.

This widely used model describes a recurring process in which the initial success of technology leads to inflated public expectations that eventually fail to be realized. After the early "peak of inflated expectations" comes a "trough of disillusionment," followed by a "slope of enlightenment," which eventually reaches a "plateau of productivity." [9]

Generative AI is rapidly improving, primarily driven by the increasing size of language models, more data, and greater computing power, while neural network architecture plays a smaller role. AI is being used to support humans, improving efficiency, reducing costs, and enhancing product quality. Smaller, more affordable AI models, like OpenAI's GPT-4o Mini, are being developed to optimize performance and cut costs. There's also a growing emphasis on AI literacy and workforce education to ensure ethical and effective use. The AI revolution will evolve gradually, transforming human activities without replacing them. [9]

VIII. Conclusion

Generative AI has demonstrated immense potential in revolutionizing business operations, offering a wide array of benefits ranging from increased efficiency to enhanced customer experiences. By automating routine tasks, personalizing customer interactions, and enabling innovation at scale, AI-powered tools are helping businesses remain competitive in an increasingly digital world. However, with these advancements come significant challenges, particularly in the areas of data privacy, algorithmic bias, and accountability. As companies continue to adopt AI technologies, it is imperative that ethical frameworks and governance structures are established to ensure responsible AI use.

Moreover, the future of generative AI promises even greater innovations, with multimodal models expanding the horizons of what is possible in industries such as healthcare, finance, and entertainment. Businesses that invest in AI today are likely to be the frontrunners in the next wave of digital transformation, but they must balance this innovation with a commitment to ethical and transparent AI practices.

References

[1] I. Goodfellow, Y. Bengio, and A. Courville, "Deep Learning," Cambridge, MA, USA: MIT Press, 2016.

[2] Exploring Explainable AI: Techniques for Interpretability and Transparency in Machine Learning Models**,** *Journal of Innovative Technologies*, vol. 7, no. 1, 2024. [Online]. Available: https://academicpinnacle.com/index.php/JIT/article/view/237. [Accessed: Sep. 5, 2024]

[3] T. Brown et al., "Language Models are Few-Shot Learners," in Proc. of the 34th International Conference on Machine Learning (ICML), 2020.

[4] OpenAI, "GPT-3: Language Models are Few-Shot Learners," OpenAI, 2020. [Online]. Available: https://openai.com/research/gpt-3. [Accessed: Sep. 15, 2024].

[5] Google AI, "AI Fairness Principles," *Google AI Blog*, 2022.

[6] General Data Protection Regulation (GDPR) – Official Legal Text, *GDPR Info*, [Online]. Available: https://gdpr-info.eu/. [Accessed: Sep. 10, 2024].

[7] [7] California Consumer Privacy Act (CCPA), *Office of the Attorney General, California Department of Justice*, [Online]. Available: https://oag.ca.gov/privacy/ccpa. [Accessed: Sep. 10, 2024]

[8] B. Marr, AI Hype Or Reality: The Singularity—Will AI Surpass Human Intelligence?, *Forbes*, 26-Jun-2024. [Online]. Available: https://www.forbes.com/sites/bernardmarr/2024/06/26/ai-hype-or-reality-the-singularity-will-ai-surpass-human-intelligence/. [Accessed: Sep. 7, 2024]

[9] Gartner Hype Cycle Research Methodology, *Gartner*, [Online]. Available: https://www.gartner.com/en/research/methodologies/gartner-hype-cycle. [Accessed: Sep. 7, 2024].

[10] Leibowicz, C. (2023, August 9). Why watermarking AI-generated content won't guarantee trust online. *Technology Review*. https://www.technologyreview.com/2023/08/09/1077516/watermarking-ai-trust-online/

[11] Naughton, J. (2023, August 19). The world has a big appetite for AI—but we really need to know the ingredients. *The Guardian*. https://www.theguardian.com/commentisfree/2023/aug/19/the-world-has-a-big-appetite-for-ai-but-we-really-need-to-know-the-ingredients

[12] https://www.forbes.com/sites/bernardmarr/2024/04/29/the-4-types-of-generative-ai-transforming-our-world/

[13] Huang, S., & Siddarth, D. (2023, February 6). Generative AI and the Digital Commons. Working paper. Collective Intelligence Project. https://cip.org/research/generative-ai-digital-commons

[14] https://www.linkedin.com/pulse/what-risks-associated-generative-ai-how-mitigate-them-auxiliobits-d9oie/

[15]] OpenAI. (2023). *GPT-4*. Retrieved November 2023 from https://openai.com/research/gpt-4

Deploying Oracle Machine Learning AutoML Models for Oracle APEX Analytics

Ivan Pastierik
Faculty of Management Science and Informatics
University of Žilina
Žilina, Slovakia
pastierik2@stud.uniza.sk

Abstract—**Oracle Application Express (APEX) is a powerful, low-code development platform that enables the rapid creation of scalable and secure enterprise applications. Integrating seamlessly with Oracle databases, Oracle APEX provides a user-friendly interface and robust functionality, making it an attractive solution for businesses aiming to reduce development time and costs. This paper explores the utilization of Oracle APEX for various applications, including data analytics, project management, and customer service, highlighting its versatility and efficiency. Additionally, the integration of Oracle Machine Learning (OML) within Oracle APEX is examined, demonstrating how advanced analytics and machine learning models can be developed and deployed to enhance business operations. The discussion is contextualized with a practical example of developing a weather prediction AutoML model using historical data and visualizing real-time predictions through an Oracle APEX application, showcasing the platform's capability to streamline complex processes and drive innovation in enterprise environments.**

Keywords—Analytics, Oracle APEX, Oracle Cloud, Oracle Machine Learning, AutoML, Weather Prediction

I. Introduction

The application of machine learning is becoming more and more important for organizations seeking to utilize data-driven insights to maintain a competitive advantage. The rapid evolution of machine learning technologies has facilitated the automation of complex analytical tasks, thereby enhancing decision-making processes across various industries. The motivation behind deploying machine learning models lies in their ability to uncover hidden patterns within vast datasets, predict future trends, and optimize operations. This technological advancement is not merely a trend but a necessity for companies aspiring to thrive in an increasingly data-centric world [1].

Machine learning, a subset of artificial intelligence, focuses on the development of algorithms that enable computers to learn from and make predictions based on data. This field encompasses various techniques, including supervised learning, unsupervised learning, and reinforcement learning, each designed to tackle different types of problems. The significance of machine learning lies in its capacity to transform raw data into actionable insights, automate repetitive tasks, and enhance the accuracy of predictive models [2]. As businesses accumulate ever-growing amounts of data, the importance of machine learning continues to grow, positioning it as a crucial part of modern analytics [1].

This is why Oracle Application Express (APEX) is becoming an important tool for various companies. Oracle APEX is a low-code data-centric development platform that enables users to build scalable and secure enterprise applications with minimal coding. This platform is particularly attractive to companies due to its user-friendly interface, robust functionality, and seamless integration with Oracle databases [3]. Organizations opt for Oracle APEX because it allows for rapid application development, reducing the time and cost associated with traditional software development. Additionally, Oracle APEX's cloud-ready architecture ensures that applications are easily deployable and maintainable, providing businesses with a flexible and efficient solution to meet their evolving needs.

Oracle APEX is not only valuable for analytics but also for developing a wide range of applications. Its capabilities extend beyond data analytics to include creating data entry forms, dynamic reports, and interactive dashboards. Oracle APEX can be used for managing projects, tracking customer interactions, and even building complex transactional applications that handle sales orders, inventory management, and customer service requests. This versatility makes Oracle APEX a comprehensive solution for companies looking to address various business needs through a single, integrated platform [4].

While Oracle APEX offers some data analytic opportunities, more complex analyses often require the use of Oracle Machine Learning (OML), a suite of machine learning tools integrated within the Oracle ecosystem. OML streamlines the creation, deployment, and management of machine learning models, leveraging the computational power of Oracle databases. This integration allows data scientists and analysts to build models directly where the data resides, enhancing workflow efficiency and ensuring data security and governance [1]. By utilizing OML within Oracle APEX, companies can conduct sophisticated analyses and develop intelligent applications that drive innovation and operational efficiency. For even better efficiency, deploying solutions in Oracle Cloud Infrastructure provides scalability, performance, and security, offering a robust and flexible environment for running Oracle APEX and OML applications and allowing businesses to scale resources according to demand while protecting sensitive data [3].

We will show the process of training an OML model utilizing AutoML feature directly within OCI on an example of weather prediction, where we will develop models to predict temperature, relative humidity, rain, snowfall, and snow depth based on historic data from the preceding seven days. Following the model training, we will create a simple Oracle APEX application capable of visualizing these predictions in real-time based on a specified date range. This application will retrieve predictions from the trained models through a REST API, showcasing the integration and practical application of Oracle Machine Learning within OCI and Oracle APEX. The reason why we have chosen this use case is that weather forecasting had been widely studied, with numerous models and methods applied to predict variables such as temperature, humidity, precipitation, and snow

979-8-3503-8768-1/24/$31.00 ©2024 IEEE

metrics, leveraging machine learning techniques to enhance prediction accuracy [5]. Another reason is, that the data and prediction results are quite easy to understand and interpret.

II. Introduction to Oracle APEX

Oracle Application Express (APEX) is an enterprise low-code application development platform developed by Oracle Corporation, evolving since the early 2000s. Created by Mike Hichwa, APEX simplifies cloud, mobile, and desktop application creation through a web-based IDE featuring wizards, drag-and-drop layout, and property editors. It is a no-cost feature of the Oracle Database, available on Oracle Cloud services, including the Autonomous Database Cloud Services. The latest version, 24.1, was released on June 17, 2024, with a complete history of changes available on Oracle’s website [6]. Currently Oracle APEX is available in Oracle Cloud Free Tier [7].

Designed for scalable, secure, and feature-rich enterprise applications, Oracle APEX allows developers and business users to rapidly build and deploy applications that leverage Oracle databases' capabilities. Its user-friendly interface and extensive functionality make it an attractive solution for organizations aiming to streamline their application development processes and boost productivity. The intuitive drag-and-drop interface simplifies the development process, enabling quick web-based application design with pre-built components and templates, thereby lowering the barrier for non-technical users and accelerating development timelines [8].

Oracle APEX’s seamless integration with Oracle databases ensures efficient data access and manipulation, leveraging Oracle’s robust data processing capabilities for handling large data volumes and delivering high-performance outcomes. This integration also enhances data security and integrity by inheriting Oracle databases' advanced security features [9]. The platform's cloud-ready architecture allows businesses to deploy applications on-premises or in the cloud, offering scalable and resilient solutions that meet modern enterprises' dynamic needs [10]. Furthermore, Oracle APEX supports a wide range of data analytics and reporting capabilities, enabling interactive dashboards and data visualizations that drive informed decision-making and operational efficiency [8].

III. Oracle Machine Learning

Oracle Machine Learning (OML) is an integrated suite of tools and technologies designed to streamline the development, deployment, and management of machine learning models within the Oracle ecosystem. By leveraging the robust computational power of Oracle databases, OML enables data scientists and analysts to build sophisticated models directly where the data resides, thus eliminating the need for extensive data movement and reducing latency. This in-database machine learning approach ensures high performance and scalability, making it suitable for handling large datasets and complex analytical tasks typical in enterprise environments [1].

OML offers a variety of machine learning algorithms and techniques, including classification, regression, clustering, anomaly detection, and time series analysis. These algorithms are optimized for use within Oracle databases, providing efficient and scalable solutions for diverse business problems [11]. Additionally, OML supports automated machine learning (AutoML) capabilities, which help streamline the model development process by automating tasks such as feature selection, algorithm selection, and hyperparameter tuning. This automation accelerates the creation of accurate models and makes advanced analytics more accessible to users with varying levels of expertise [12].

One of the key advantages of Oracle Machine Learning is its seamless integration with other Oracle products and services. For instance, OML can be used in conjunction with Oracle Autonomous Database, Oracle Analytics Cloud, and Oracle APEX to create comprehensive data-driven applications. Deploying OML models in Oracle Cloud Infrastructure (OCI) offers additional benefits such as scalability, high availability, and performance. OCI provides a resilient and flexible environment that can scale resources dynamically based on demand, ensuring that machine learning applications remain responsive and efficient. The integration of OML with OCI also facilitates the deployment of machine learning models as RESTful services, enabling easy consumption by other applications and services, which is particularly valuable for developing real-time predictive applications and integrating machine learning insights into business processes [1].

IV. Dataset Description

For weather prediction, we will use dataset built using Open-meteo API [13]. Through this API we have collected historic meteorologic data from weather stations situated near every regional city in Slovakia. These regional cities consist of Bratislava, Trenčín, Trnava, Nitra, Žilina, Banská Bystrica, Prešov, and Košice. The data retrieved from these weather stations includes measurements of temperature, relative humidity, snowfall, rain, and snow depth in hourly interval from January 1, 2012, to December 31, 2023, that is 105 192 measurements of temperature, relative humidity, snowfall, rain, and snow depth.

The data is organized into a data model, as seen in Fig. 1, consisting of two tables: WEATHER_DATA and WEATHER_LOCATIONS. The WEATHER_LOCATIONS table contains information about the position of weather stations, their altitudes, and descriptions, which represent the names of the regional cities closest to the weather stations. The WEATHER_DATA table contains records of temperature, relative humidity, rain, snowfall, and snow depth for these individual weather stations, along with timestamps indicating when the data was measured. This data was loaded directly into the database from CSV files downloaded from the Open-meteo API using Oracle APEX's data workshop.

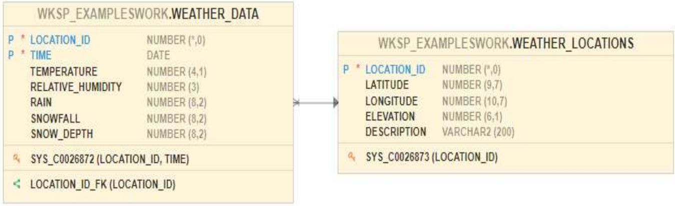

WKSP_EXAMPLESWORK.WEATHER_DATA
P * LOCATION_ID NUMBER (*,0)
P * TIME DATE
TEMPERATURE NUMBER (4,1)
RELATIVE_HUMIDITY NUMBER (3)
RAIN NUMBER (8,2)
SNOWFALL NUMBER (8,2)
SNOW_DEPTH NUMBER (8,2)
SYS_C0026872 (LOCATION_ID, TIME)
LOCATION_ID_FK (LOCATION_ID)
WKSP_EXAMPLESWORK.WEATHER_LOCATIONS
P * LOCATION_ID NUMBER (*,0)
LATITUDE NUMBER (9,7)
LONGITUDE NUMBER (10,7)
ELEVATION NUMBER (6,1)
DESCRIPTION VARCHAR2 (200)
SYS_C0026873 (LOCATION_ID)

Fig. 1. Data model for storing data from weather stations.

For simplicity, this study focuses exclusively on the weather station near the regional city of Žilina, which has location id equal to 0. All models are trained only on the data related to Žilina, providing a detailed and localized analysis of weather patterns for this specific area. This approach allows for a more concentrated examination of the prediction models and their accuracy in forecasting weather conditions based on historical data from Žilina.

V. Training AutoML Models

Training of AutoML models and all examples will be showed directly in Oracle Cloud Infrastructure. Before we begin with the training of models, we first need to provision database inside of OCI. For our purposes, we have provisioned Oracle Autonomous Transaction Processing Database. Next, we created OML user with role of OML developer, Oracle APEX user, and Oracle APEX workspace. Here it is necessary for the OML user to have role OML developer and not OML administrator, otherwise it would not be possible for the user to deploy models. Now when we have our OML user ready, we can proceed by creating the first model using AutoML UI.

A. AutoML Overview

The AutoML UI in Oracle Machine Learning is a no-code solution designed to automate machine learning model creation, enhancing productivity and potentially increasing model accuracy and performance. AutoML UI automates essential steps in the machine learning workflow, including algorithm selection, adaptive sampling, feature selection, model tuning, and feature prediction impact, allowing business users without extensive data science expertise to efficiently create and deploy machine learning models [12].

To begin using AutoML UI, users can create an experiment by specifying the data source, prediction target, and prediction type. The experiment ranks models by quality based on selected metrics, allowing users to deploy the best models or generate a Python notebook with the settings used. Supported metrics for classification include Balanced Accuracy, ROC AUC, and F1 scores, while regression metrics include R2, Negative Mean Squared Error, and Negative Mean Absolute Error. AutoML UI provides tools for monitoring and managing experiments, such as a progress bar, options for faster results or better accuracy, and a leaderboard showing top-performing models with detailed information, including prediction impact and confusion matrices. The features grid displays statistical information for the selected table, highlighting the target column and showing feature importance, which helps in understanding the model's behavior and improving its accuracy. Overall, Oracle's AutoML UI simplifies the machine learning process by automating complex tasks and providing intuitive tools for model creation and deployment, making it accessible to users across various skill levels and enabling them to derive valuable insights from their data efficiently.

B. AutoML Supported Classification Algorithms

1) Decision Tree: A Decision Tree is a model that splits the data into branches to make predictions based on feature values. It is easy to interpret and can handle both numerical and categorical data. The model creates a tree-like structure where each node represents a decision rule, and each branch represents the outcome of the rule [14].

2) Generalized Linear Model (GLM): GLM is an extension of linear regression that allows for response variables that have error distribution models other than a normal distribution. It supports various link functions to model different types of response variables, making it flexible for various types of classification problems [14].

3) Generalized Linear Model (Ridge Regression): This version of GLM incorporates ridge regression to handle multicollinearity among predictor variables. It adds a penalty term to the loss function to shrink the coefficients, which helps in improving the model's generalization by reducing overfitting [14].

4) Neural Network: A Neural Network is a computational model inspired by the human brain, consisting of layers of interconnected nodes (neurons). It is capable of learning complex patterns from the data and is particularly useful for handling non-linear relationships and high-dimensional data in classification tasks [14].

5) Random Forest: Random Forest is an ensemble learning method that constructs multiple decision trees during training and outputs the mode of the classes for classification. It improves prediction accuracy and controls overfitting by averaging the results of multiple trees, each trained on different parts of the data [14].

6) Support Vector Machine (Gaussian): A Support Vector Machine (SVM) with a Gaussian (or Radial Basis Function) kernel is a powerful regression model that finds the optimal hyperplane to separate different classes. The Gaussian kernel allows the model to handle non-linear relationships by mapping the input features into higher-dimensional space [13].

7) Support Vector Machine (Linear): An SVM with a Linear kernel is used for linearly separable data, finding the best hyperplane that maximizes the margin between classes. It is efficient and works well with large feature spaces but may not perform well with non-linear data [14].

C. AutoML Supported Regression Algorithms

1) Generalized Linear Model (GLM): For regression tasks, GLM extends linear regression to model different types of response variables, including those with non-normal error distributions. It supports various link functions to handle different types of dependent variables, making it versatile for regression analysis [14].

2) Generalized Linear Model (Ridge Regression): In the context of regression, this version of GLM applies ridge regression to address multicollinearity by adding a regularization term to the loss function. This technique helps to improve model generalization and reduce overfitting by shrinking the coefficients [14].

3) Neural Network: Neural Networks for regression tasks are used to model complex relationships between input features and a continuous target variable. They consist of layers of interconnected nodes that can learn non-linear patterns and interactions, making them suitable for complex regression problems [14].

4) Support Vector Machine (Gaussian): An SVM with a Gaussian kernel for regression, known as Support Vector Regression (SVR), aims to find a function that deviates from the actual observed values by a value less than a specified margin. The Gaussian kernel allows the model to capture non-linear relationships between the features and the target variable [14].

5) Support Vector Machine (Linear): Support Vector Regression (SVR) with a Linear kernel is used for linear regression tasks. It finds a linear function that fits the data while minimizing the prediction error within a specified margin. It is efficient for problems where the relationship between features and target is linear [14].

D. Preparing training data

Before we begin with the training of models itself, we need to create table with training data:

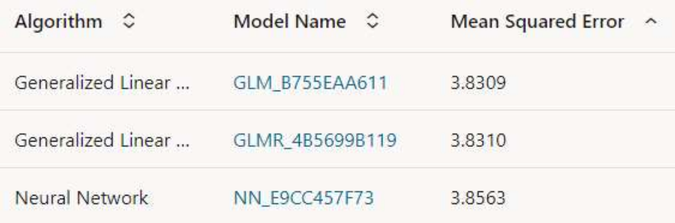
```
CREATE TABLE WEATHER_PREDICTIONS_OML_TRAIN
AS
SELECT * FROM
 (SELECT   time,   rain,   relative_humidity,   snowfall,   snow_depth,
    temperature,
    LAG(rain, 1) OVER(ORDER BY time) RAIN_lag_1,
    LAG(rain, 2) OVER(ORDER BY time) RAIN_lag_2,
    ………………………………………………………………..,
    LAG(temperature,        6)       OVER(ORDER        BY        time)
                                   TEMPERATURE_lag_6,
    LAG(temperature,        7)       OVER(ORDER        BY        time)
                                   TEMPERATURE_lag_7
FROM (
    SELECT TRUNC(time, 'DD') time,
        ROUND(AVG(temperature), 2) temperature,
        ROUND(AVG(relative_humidity), 2) relative_humidity,
        ROUND(SUM(rain), 2) rain,
        ROUND(SUM(snowfall), 2) snowfall,
        ROUND(AVG(snow_depth), 2) snow_depth
    FROM WEATHER_DATA
    WHERE location_id = 0 AND
            time < TO_DATE('1.1.2023', 'DD.MM.YYYY')
    GROUP BY TRUNC(time, 'DD') )
)
WHERE TEMPERATURE_lag_7 is not NULL;
```

This select statement creates new table “WEATHER_PREDICTIONS_OML_TRAIN”, by selecting and transforming data from the “WEATHER_DATA” table. It aggregates temperature, relative humidity, rain, snowfall, and snow depth, on a daily basis, which is necessary, because we only have hourly intervals of measurements, but we need daily intervals. This aggregation is only done for weather station near Žilina (location_id = 0) up to January 1, 2023, exclusive. The outer query introduces lagged variables, using the LAG analytical function to include weather data from up to seven days prior, which is essential for training AutoML models on time series. The final WHERE clause ensures that only rows with non-null values for the seven days prior (“TEMPERATURE_lag_7”) are included, ensuring completeness of the lagged data for model training purposes. “WEATHER_PREDICTIONS_OML_TRAIN” table has 4011 rows, with represents the size of training dataset.

E. Training Process and Results of Models

All models were trained using “WEATHER_PREDICTIONS_OML_TRAIN” table as train data and all models were trained as regression models, because we want to predict specific values of different weather variables. As metric we use Mean Squared Error, and we select three best algorithms for each model based on their Mean Squared Error value.

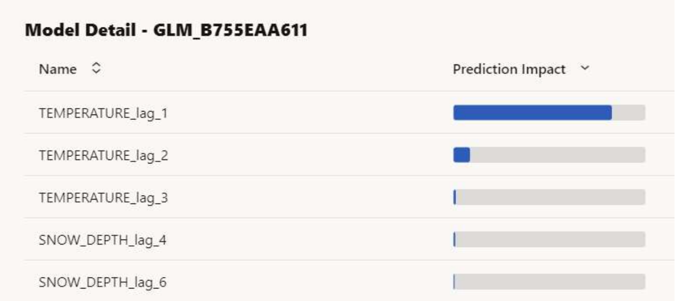
1) Temperature Prediction Model: Fig. 2 shows a table listing the algorithms, model names, and their Mean Squared Error (MSE) values. The top two algorithms are Generalized Linear Model and Generalized Linear Model (Ridge Regression) with slightly different configurations, having MSE values of 3.8309 and 3.8310 respectively. The third algorithm is a Neural Network model with an MSE of 3.8563. These MSE values indicate that the GLM models performed slightly better in terms of prediction accuracy compared to the Neural Network model.



Algorithm ⌃⌄	Model Name ⌃⌄	Mean Squared Error ⌃
Generalized Linear ...	GLM_B755EAA611	3.8309
Generalized Linear ...	GLMR_4B5699B119	3.8310
Neural Network	NN_E9CC457F73	3.8563

Fig. 2. Temperature prediction model algorithm leaderboard.

Fig. 3 details the prediction impact for the best-performing model, GLM_B755EAA611. The prediction impact measures how much each input feature influences the model's predictions. For this GLM model, the feature “TEMPERATURE_lag_1” has the highest impact on predicting the target variable, followed by “TEMPERATURE_lag_2” and “TEMPERATURE_lag_3”. Other features like “SNOW_DEPTH_lag_4” and “SNOW_DEPTH_lag_6” have lesser influence. This suggests that recent past temperatures (up to three days prior) are the most significant predictors for the current temperature, whereas the influence of snow depth and other features is relatively minor.

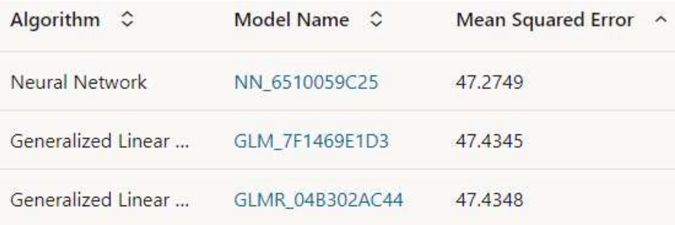


Model Detail - GLM_B755EAA611

Name ⌃⌄	Prediction Impact ⌄
TEMPERATURE_lag_1	
TEMPERATURE_lag_2	
TEMPERATURE_lag_3	
SNOW_DEPTH_lag_4	
SNOW_DEPTH_lag_6	

Fig. 3. Highest prediction impact features in best temperature prediction algorithm.

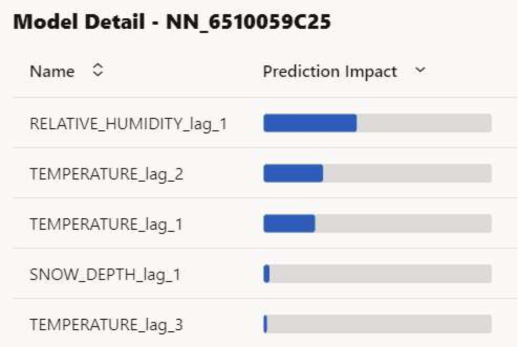
2) Relative Humidity Prediction Model: In Fig. 4, we can see that the best algorithm for predicting relative humidity is Neural Network model with MSE values of 47.2749, Generalized Linear Model and Generalized Linear Model (Ridge Regression) are slightly worse.



Algorithm ⌃⌄	Model Name ⌃⌄	Mean Squared Error ⌃
Neural Network	NN_6510059C25	47.2749
Generalized Linear ...	GLM_7F1469E1D3	47.4345
Generalized Linear ...	GLMR_04B302AC44	47.4348

Fig. 4. Relative humidity prediction model algorithm leaderboard.

In Fig. 5, we can see, that, the most important features when it comes to prediction impact are “RELATIVE_HUMIDITY_lag_1”, with quite high significance and then “TEMPERATURE_lag_2” and “TEMPERATURE_lag_1” with slightly lower significance. Influence of other features is minor.

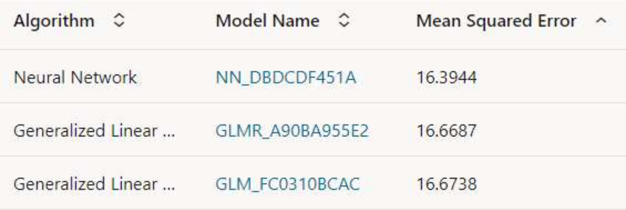


Model Detail - NN_6510059C25

Name	Prediction Impact
RELATIVE_HUMIDITY_lag_1	
TEMPERATURE_lag_2	
TEMPERATURE_lag_1	
SNOW_DEPTH_lag_1	
TEMPERATURE_lag_3	

Fig. 5. Highest prediction impact features in best relative humidity prediction algorithm.

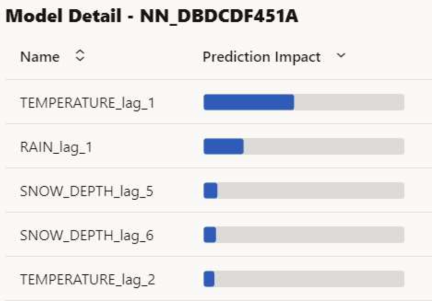
3) Rain Prediction Model: In case of rain prediction model the best algorithm is Neural Network model with MSE of 16.3944. All top three models can be seen in Fig. 6.



Algorithm	Model Name	Mean Squared Error
Neural Network	NN_DBDCDF451A	16.3944
Generalized Linear ...	GLMR_A90BA955E2	16.6687
Generalized Linear ...	GLM_FC0310BCAC	16.6738

Fig. 6. Rain prediction model algorithm leaderboard.

Five most important features are “TEMPERATURE_lag_1”, “RAIN_lag_1”, “SNOW_DEPTH_lag_5”, “SNOW_DEPTH_lag_6” and “TEMPERATURE_lag_2” as seen in Fig. 7.

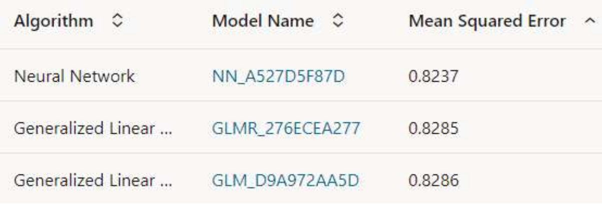


Model Detail - NN_DBDCDF451A

Name	Prediction Impact
TEMPERATURE_lag_1	
RAIN_lag_1	
SNOW_DEPTH_lag_5	
SNOW_DEPTH_lag_6	
TEMPERATURE_lag_2	

Fig. 7. Data model for storing data from weather stations.

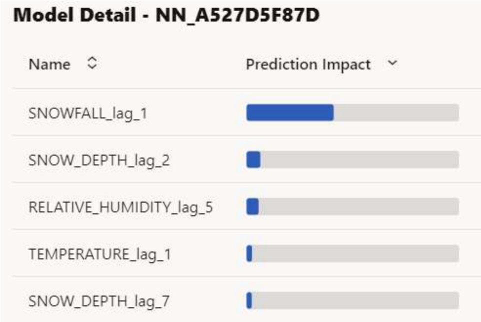
4) Snowfall Prediction Model: Best algorithm in leaderboard is Neural Network algorithm with MSE of 0.8237, rest of algorithms and their MSE could be seen in Fig. 8.



Algorithm	Model Name	Mean Squared Error
Neural Network	NN_A527D5F87D	0.8237
Generalized Linear ...	GLMR_276ECEA277	0.8285
Generalized Linear ...	GLM_D9A972AA5D	0.8286

Fig. 8. Snowfall prediction model algorithm leaderboard.

Most significant feature is “SNOWFALL_lag_1”. Features “SNOW_DEPTH_lag_2”, “RELATIVE_HUMIDITY_lag_5”, “TEMPERATURE_lag_1” and “SNOW_DEPTH_lag_7” are less significant, but still important, as seen in Fig. 9.

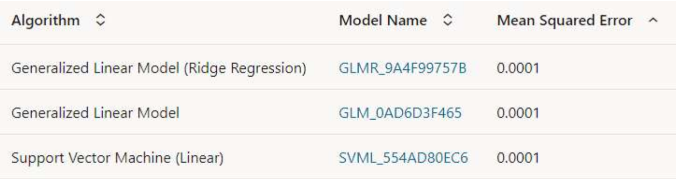


Model Detail - NN_A527D5F87D

Name	Prediction Impact
SNOWFALL_lag_1	
SNOW_DEPTH_lag_2	
RELATIVE_HUMIDITY_lag_5	
TEMPERATURE_lag_1	
SNOW_DEPTH_lag_7	

Fig. 9. Highest prediction impact features in best snowfall prediction algorithm.

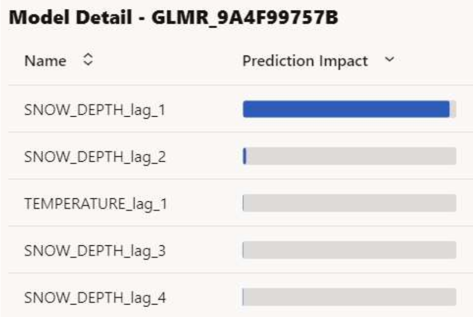
5) Snow Depth Prediction Model: MSE of all models are the same and have value of 0.0001. We have chosen first model from the ranking seen in figure, even though they are practically the same.



Algorithm	Model Name	Mean Squared Error
Generalized Linear Model (Ridge Regression)	GLMR_9A4F99757B	0.0001
Generalized Linear Model	GLM_0AD6D3F465	0.0001
Support Vector Machine (Linear)	SVML_554AD80EC6	0.0001

Fig. 10. Snow depth prediction model algorithm leaderboard.

When it comes to prediction impact of features, “SNOW_DEPTH_lag_1” is the most significant feature, rest of features are nearly insignificant as seen in Fig. 11.



Model Detail - GLMR_9A4F99757B

Name	Prediction Impact
SNOW_DEPTH_lag_1	
SNOW_DEPTH_lag_2	
TEMPERATURE_lag_1	
SNOW_DEPTH_lag_3	
SNOW_DEPTH_lag_4	

Fig. 11. Highest prediction impact features in best snow depth prediction algorithm.

F. Deploying Models in OML

Models inside AutoML UI can be deployed directly by selecting the model we want to deploy and pressing deploy button. Here all we need to do is to fill the name of the model, URI, version, namespace and optionally, it is also possible to write comments for the model.

After deploying the model, it is possible to look at model metadata, where it is possible to find all necessary information about model REST API endpoint, so all important information about getting predictions from model. All deployed models can be accessed from external sources by using following URL: “<oml-cloud-service-location-url>/omlmod/v1/deployment/<model-URI>”.

This way we have deployed models for prediction of temperature, relative humidity, rain, snowfall and snow depth.

VI. Deploying AutoML Models in Oracle APEX

Deploying AutoML models in Oracle APEX is straightforward thanks to its robust support for REST API integration. APEX, a low-code development environment, allows applications to communicate with machine learning models deployed in AutoML UI via REST API calls. This enables APEX applications to fetch real-time predictions from AutoML models, integrating advanced machine learning capabilities seamlessly. Let us now look at the basics of getting the predictions from models.

A. Retrieving Oracle Authorization Token

To retrieve the Oracle Authorization Token necessary for accessing Oracle Machine Learning (OML) services, we send a request to the URL: "<oml-cloud-service-location-url>/omlusers/api/oauth2/v1/token". The request must include the header:

```
{
  "grant_type": "password",
  "username": "<your-OML-user-username>",
  "password": "<your-OML-user-password>"
}
```

This token is valid for one hour. To streamline this process, the token is automatically synchronized into the "OML_TOKEN" table on an hourly basis. This setup ensures that a new token is fetched and replaces the existing one in the "OML_TOKEN" table every hour, maintaining uninterrupted access to OML services.

B. Configuring REST API Sources for Prediction Model

First, it is important to increase the rate limit for API calls in Oracle APEX. Default rate limit in Oracle APEX is 1000, which is not enough. It is possible to increase or disable these rate limits within the Oracle APEX workspace settings.

Once the rate limits are appropriately configured, the next step involves creating REST Sources within Oracle APEX. The REST Source will utilize the URL: "<oml-cloud-service-location-url>/omlmod/v1/deployment/<model-URI>/score" to connect with the deployed prediction model. This setup allows Oracle APEX to communicate directly with the OML model for real-time predictions.

To create a REST Source in Oracle APEX through Shared Components, it is required to navigate to Shared Components in the Oracle APEX application builder, and here find REST Data Sources, Next, provide a meaningful name and importantly "static_id" for your REST Data Source, which will be used for accessing this data source in SQL. As base URL we will use "<oml-cloud-service-location-url>", and as URL path prefix: "omlmod/v1/deployment/<model-URI>/score". Inside operations, we need to use POST operation since predictions need to be retrieved through a POST call and choose.

Next, we add the necessary headers for the authorization token by creating parameter of type HTTP Header with name Authorization. This parameter will be dynamically filled by the token from "OML_TOKEN" table. In the body of the POST request, include the input data for the prediction model structured as follows:

```
{
  "inputRecords": [{
    "RAIN_lag_1": #RAIN_lag_1#,
    "RAIN_lag_2": #RAIN_lag_2#,
    ..........................................................,
    "TEMPERATURE_lag_6": #TEMPERATURE_lag_6#,
    "TEMPERATURE_lag_7": #TEMPERATURE_lag_7#
  }],
  "topN": #topN#,
  "topNdetails": #topNdetails#
}
```

C. Passing Data to Prediction Models

Passing data to prediction models can be achieved by creating a function called "predict_temperature_automl", which allows various lagged weather parameters to be inputted and provides real-time predictions. This function constructs a set of parameters, retrieves the authorization token from the "OML_TOKEN" table, and then makes a REST API call to the prediction model. The function adds necessary parameters such as the authorization token and weather variables lag values into a parameter object. It then executes the REST API call, retrieves the response, extracts the predicted temperature value from the JSON response, and returns it rounded to two decimal places:

```
CREATE OR REPLACE FUNCTION predict_temperature_automl(
    p_model VARCHAR2,
    p_rain_lag_1 NUMBER,
    p_rain_lag_2 NUMBER,
    …………………………………,
    p_temperature_lag_6 NUMBER,
    p_temperature_lag_7 NUMBER
) RETURN NUMBER IS
    l_params apex_exec.t_parameters;
    l_token CLOB;
    l_json_string CLOB;
    l_regression NUMBER;
BEGIN
    SELECT 'Bearer ' || ACCESSTOKEN INTO l_token
    FROM OML_TOKEN;
    apex_exec.add_parameter(l_params, 'Authorization', l_token);
    apex_exec.add_parameter(l_params, 'topN', 35);
    apex_exec.add_parameter(l_params, 'topNdetails', 35);
    apex_exec.add_parameter(l_params, 'RAIN_lag_1',
format_number(p_rain_lag_1));
    apex_exec.add_parameter(l_params, 'RAIN_lag_2',
format_number(p_rain_lag_2));
    ………………………………………………………….,
     apex_exec.add_parameter(l_params, 'TEMPERATURE_lag_6',
format_number(p_temperature_lag_6));
     apex_exec.add_parameter(l_params, 'TEMPERATURE_lag_7',
format_number(p_temperature_lag_7));
     apex_exec.execute_rest_source(
      p_static_id => p_model,
      p_operation => 'POST',
      p_parameters => l_params
  );
  l_json_string := apex_exec.get_parameter_clob(l_params,
'RESPONSE');
  l_regression := JSON_VALUE(l_json_string,
'$.scoringResults[0].regression');
RETURN ROUND(l_regression, 2);
END;
/
```

This function begins by declaring necessary local variables to hold parameters, the authorization token, the JSON response, and the final regression value. It retrieves the authorization token from the "OML_TOKEN" table, concatenates it with the "Bearer" prefix, and stores it in "l_token". The function proceeds by adding the authorization token and other parameters, including "topN" and "topNdetails", to the parameter object "l_params". It then formats and adds each of the weather parameter inputs to "l_params". Using the "apex_exec.execute_rest_source" procedure, the function makes a POST request to the REST Data Source associated with the prediction model specified by "p_model", which will contain static id of REST Data Source, that we want to call. After executing the request, it retrieves the JSON response string and parses it to extract the predicted temperature value using the "JSON_VALUE" function, which it then rounds and returns. To utilize the "predict_temperature_automl" function, it can be called within the SELECT clause of an SQL query, passing in the appropriate model static id and lagged weather parameters. This allows for seamless integration of real-time model predictions into SQL queries, facilitating data-driven decision-making and analysis. This approach simplifies the process of obtaining predictions from machine learning models and makes it accessible through standard SQL operations.

D. *Prediction Results*

The data shown in the charts are testing data consisting of 358 measurements from the weather station of Žilina spanning across year 2023. The models were trained using AutoML UI in faster results mode, based on measurements from the prior 7 days. The charts display both the real values and the predictions for various weather parameters: temperature, humidity, rain, snowfall, and snow depth. The performance of the models can be evaluated by comparing the predicted values to the actual measurements.

In the temperature chart seen in Fig. 12, the model predictions closely follow the real temperature values, indicating that the model is well-calibrated for temperature prediction. The model captures the seasonal variations and the overall trend accurately, which suggests that the training data and the features used were appropriate for predicting temperature.

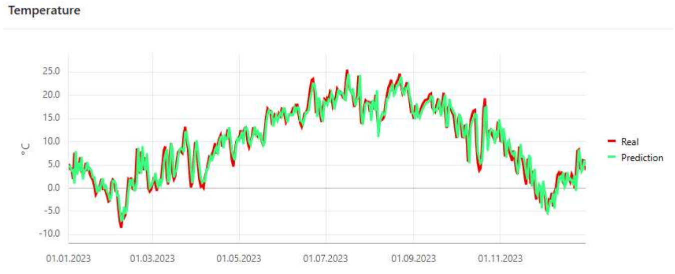


Fig. 12. Visualisation of real and predicted temperature values.

The humidity chart in Fig. 13 also shows a good fit between the real and predicted values. The predictions are generally in line with the actual humidity measurements, demonstrating the model's ability to capture the day-to-day fluctuations in humidity. This indicates that the model is effective in predicting humidity based on the prior 7 days of measurements.

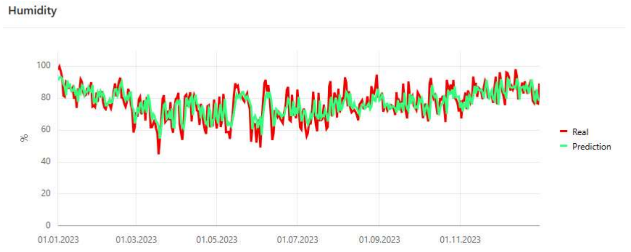


Fig. 13. Visualisation of real and predicted relative humidity values.

For rain in Fig. 14, the predictions are less accurate. While the model captures some of the spikes in rain, it often incorrectly estimates the actual amount of rain. This discrepancy might be due to the inherently sporadic and unpredictable nature of rain, which can be challenging for models to predict accurately based on historical data alone, especially with only basic weather variables.

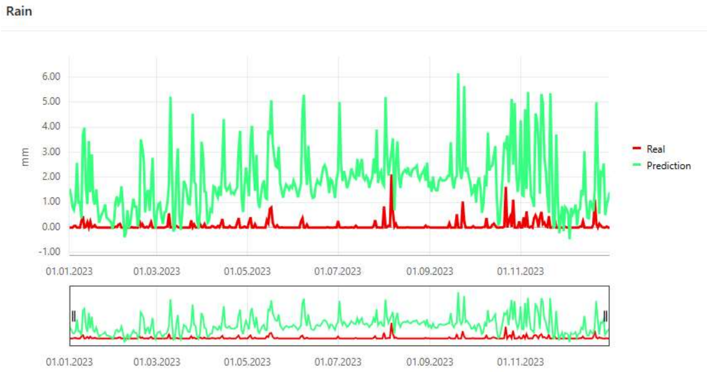


Fig. 14. Visualisation of real and predicted rain values.

In the snowfall chart in Fig. 15, the model does not accurately correspond to actual measurements. This could be because snowfall is a relatively rare event, and the model might not have had enough training examples to learn the patterns effectively. The sporadic nature of snowfall events makes it difficult for the model to predict them accurately.

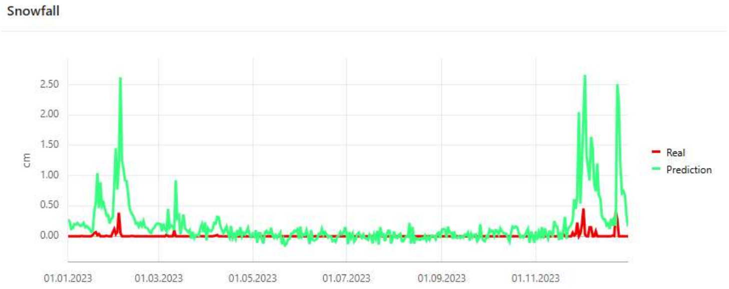


Fig. 15. Visualisation of real and predicted snowfall values.

The snow depth chart in Fig. 16 shows that the model successfully captures the actual snow depth accurately. This suggests that the model can capture the complexity of factors that influence snow accumulation and melting.

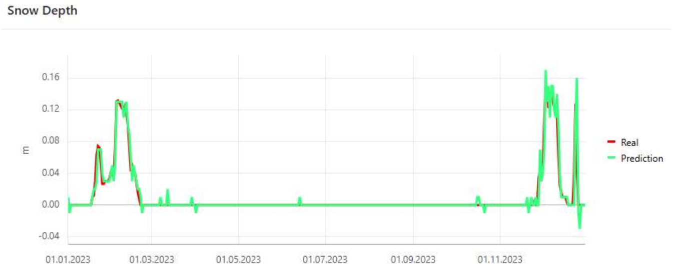


Fig. 16. Visualisation of real and predicted snow depth values.

Overall, while the models perform well for temperature, humidity and snow depth, they struggle with rain and snowfall predictions. The variability and rarity of these events, combined with possibly insufficient training data for such events, likely contribute to the poorer performance in these areas.

VII. Conclusion

In conclusion, utilizing the AutoML UI for deploying and integrating machine learning models within Oracle APEX offers a streamlined and efficient approach to analytics. The AutoML UI significantly reduces the complexity and time required for model development by automating critical tasks such as algorithm selection, feature selection, and model tuning. This no-code solution is particularly advantageous for business users and analysts who may not have a deep background in data science, enabling them to build and deploy sophisticated models with ease.

On the downside, the AutoML UI operates as a black box, meaning users have limited visibility into the internal workings of the models it generates. This lack of transparency can be both a strength and a weakness. As an advantage, it abstracts the complexity of machine learning algorithms, making them accessible to non-experts. Conversely, the black-box nature may be a disadvantage for those who require detailed insights into the model or need to customize specific aspects of the modeling pipeline. Also, another limitation is, that AutoML UI models can only have a single output, so you can't for example predict both temperature and humidity in the same model. Overall, while AutoML UI makes the access to advanced machine learning capabilities easier and integrates seamlessly with Oracle APEX, it is essential to be aware of its limitations and the trade-offs involved in using a black-box system for model generation and deployment.

Acknowledgment

It was partially supported by the Erasmus+ project: Project number: 022-1-SK01-KA220-HED-000089149, Project title: Including EVERyone in GREEN Data Analysis (EVERGREEN).

Co-funded by the Erasmus+ Programme of the European Union

EverGreen DATA Analytics

References

[1] A. Png and H. Helskyaho, *Extending Oracle Application Express with Oracle Cloud Features*. Apress, 2022.

[2] Y. Zhang, *New Advances in Machine Learning*. InTech, 2010.

[3] M. Kvet, K. Matiaško and Š. Toth, *Practical SQL for Oracle Cloud*. EDIS-Publishing House of the University of Žilina, 2022.

[4] E. Sciore, *Understanding Oracle APEX 20 Application Development: Think Like an Application Express Developer*. Apress, 2020.

[5] D. Fister, J. Pérez-Aracil, C. Peláez-Rodríguez, J. Del Ser, S. Salcedo-Sanz, "Accurate long-term air temperature prediction with Machine Learning models and data reduction techniques," *Applied Soft Computing*, vol. 136, pp. 110118, 2023.

[6] *Oracle APEX History*, [Online]. Available: https://apex.oracle.com/pls/apex/r/apex_pm/apex-community/history. Accessed: July 15, 2024.

[7] *Oracle Cloud Free Tier*, [Online]. Available: https://www.oracle.com/cloud/free/. Accessed: July 15, 2024.

[8] A. Geller and B. Spendolini, *Oracle Application Express (APEX): Build Powerful Data-Centric Web Apps with APEX*. McGraw Hill, 2017.

[9] *Oracle APEX Application Security*, [Online]. Available: https://docs.oracle.com/en/database/oracle/apex/24.1/htmdb/managing-application-security.html. Accessed: July 15, 2024.

[10] *Oracle APEX Deployment*, [Online]. Available: https://apex.oracle.com/en/platform/deployment/. Accessed: July 15, 2024.

[11] *Oracle Database Machine Learning*, [Online]. Available: https://www.oracle.com/artificial-intelligence/database-machine-learning/. Accessed: July 15, 2024.

[12] *Oracle AutoML User Interface*, [Online]. Available: https://docs.oracle.com/en/database/oracle/machine-learning/oml-automl-ui/index.html. Accessed: July 15, 2024.

[13] *Open-meteo weather API*, [Online]. Available: https://open-meteo.com/. Accessed: July 15, 2024.

[14] G. James, D. Witten, T. Hastie, R. Tibshirani and J. Taylor, *An Introduction to Statistical Learning: with Applications in Python*. Springer Cham, 2023.

Data-driven decision-making options for less tech-savvy users

1st Miroslav Potočár
Faculty of management science and informatics
University of Žilina
Žilina, Slovakia
Miroslav.Potocar@fri.uniza.sk

2nd Michal Kvet
Faculty of management science and informatics
University of Žilina
Žilina, Slovakia
Michal.Kvet@fri.uniza.sk

***Abstract*—This study explores the application of data-driven decision-making tools for users with limited technical expertise, focusing on the usability and effectiveness of Power BI and Oracle Analytics in real estate investment scenarios. A custom dataset of real estate transactions in Žilina, Slovakia, was collected and preprocessed to evaluate how these platforms enable non-programmers to perform data transformation, analysis and predictive modeling. Both tools were assessed on their ability to handle common data processing tasks, train predictive models, and generate actionable insights. Results show that while Power BI is more intuitive for data preprocessing, Oracle Analytics excels in model training and prediction. Combining the strengths of both platforms could provide a comprehensive solution for users with limited technical skills. This research offers practical recommendations for selecting and using these analytics tools in real-world decision-making contexts, particularly in the domain of real estate investment.**

***Index Terms*—Power BI, Oracle Analytics, data analytics, decision-making, real estate, non-programmers**

I. Introduction

We make a number of decisions every day. Some are minor and their impact is not that significant ("Which phone should I buy?"). In other cases, they are very important decisions whose consequences can reach far into the future ("Which property should I invest my money in?"). It is in the individual's interest to make the right decisions. However, this is not so simple. There are many ways in which the decision-making process can be approached [1].

One of the decision-making approaches is data-driven decision-making, which has found its application in many industries including, for example, business [2], healthcare [3] or education [4], [5]. Data-driven decision-making is not only reserved for industry and business [6], where the right data-driven decision-making leads to increases in output and productivity [7] but with the right data it can also benefit individuals. Duggan [8] believes that a large portion of user decision-making problems can be solved using simple models in conjunction with access to lots of data. Predictive models are inherently linked to making good decisions. These often provide the opportunity to get at least an approximate picture of important characteristics that would otherwise remain hidden.

Obtaining the right data to make decisions is not as difficult as it was in the past. The barrier today is represented primarily by data analysis, data processing, feature engineering and the design of appropriate prediction models. For people with experience in programming or for data scientists it is no problem at all. However, making good decisions should not be accessible only to a narrow group of people. This capability should be available to everyone. With the development of new technologies, this is becoming possible. Thanks to available analytics tools like Power BI [9] or Oracle Analytics [10], anyone can conveniently analyse the collected data, visualise it, make simple predictions and make better decisions based on it. These platforms are primarily designed for business analysts, data scientists and IT professionals. But can less tech-savvy users also use them to process their data, predict values and make data-driven decisions?

This research is intended to explore the opportunities that can be used by users without programming skills to make data-driven decisions. It focuses primarily on how accurate predictions can be obtained using available analytical tools (namely Power BI and Oracle Analytics). The focus is on the ease of making these predictions, as it is expected that users without a background in programming or data analytics will not have the necessary knowledge to manually create advanced models and tune their parameters. Secondly, this study focuses on whether and how these analytical tools can be used to transform and preprocess the data needed to train the available prediction models.

The focus of the research is to make it as close as possible to real-life use cases. For this reason, the selection of an investment real estate was chosen as the use case. In order to make the study as close as possible to real-world conditions, no existing real estate dataset was used, but a custom dataset was collected from a local real estate advertiser's website. This ensured that the study addressed data processing issues that a user may encounter in common practice. Some of the most common problems include inconsistencies between column names carrying the same information, removing duplicate rows, extracting data from columns, removing outliers, and others. Therefore, the study focuses on whether and how these problems can be solved on the chosen platforms.

As mentioned previously, predictive models and associated value predictions are also an important tool for analysis and can be used to provide additional information in the decision-

979-8-3503-8768-1/24/$31.00 ©2024 IEEE

making process. Therefore, this study focuses on whether the selected platforms make it possible to easily create prediction models and what this creation process looks like.

II. Use Case - Real Estate Investing

These days, investing is becoming more and more popular. There are many opportunities in which we can invest money. For many people, ETFs and shares are too abstract and therefore they prefer to avoid investing in them. Such investors prefer to invest in something tangible and easy to understand. For example, commodities, artwork or real estate fall into this category. Real estate in particular represents a very interesting investment opportunity, not only for conservative investors but also for investors who want to diversify their portfolio. However, choosing the right property is not as easy as it may seem at first sight. First of all, you need to choose a region with a growing population - buying an investment property in areas where people are moving away is not the best investment decision. Once you have chosen a suitable area, you still haven't won. Even within cities with growing populations, there are suitable and less suitable locations for living. When we choose the wrong location, we run the risk that our investment will not achieve the expected returns. An important parameter is whether the real estate is located near the city centre, public transport stops or, on the contrary, whether it is far enough away from an air polluting factory or a landfill.

After selecting a suitable location in the region, you still need to choose the right real estate. Each has certain parameters with varying degrees of importance. Each parameter has its specific weight on the overall price of the property and also on the rental price at which the property could be offered. In order to maximise the return on the investment, we need to choose the right property with the right price and a high enough potential rental price. However, the question is what is the right price for the property and what rental prices we can expect for it. Here we can use the analytical tools and mathematical models that are available to help us approximate the right prices. With the right model, we are able to predict reasonable prices given the parameters and location of the property and also identify properties that are selling below their value. If we have rental data, we can also create a mathematical model that will predict the rental price. We can apply this model to the data on the real estate being sold and predict the potential rental income given the parameters of the property. With this kind of data, we can make truly informed decisions.

III. Methodology

A. Data Collection

As mentioned in the introduction, the real estate domain was chosen for this study. Although there are a number of datasets focusing on real estate, a new dataset was created as part of this research. This decision was made to ensure that the research reflects as closely as possible the real-world conditions that ordinary users may encounter. For our purposes, we collected data on sold and rented apartments located in the city of Žilina.

In order to obtain real estate data, a web scraper was developed and operated for several months on the website of a local real estate advertiser. This was the site *www.nehnutelnosti.sk*. Every 6 hours this tool crawled all the adverts that met the specified criteria. As much property information as possible was collected about each advert, along with additional details such as the URL and the date the listing was last accessed. On each pass, new data was saved to the MongoDb database. For each advert, the URL was checked to see if it was already in the database. If the URL of an advert was already in the database, the last date and time that the data collector encountered that advert was updated.

Although a bot was used to collect the data, which requires some programming knowledge to create, and a MongoDb database to store the data, thanks to user-friendly automation tools such as Power Automate, it is possible even for a user with no programming skills to build some form of such a collector. Using similar tools, it is possible to extract data from web pages on a regular basis and then store this data in some sort of storage, e.g. excel sheets.

B. Data Enhancement

When making big decisions, as buying an investment property certainly is, it is important to gather as much relevant information as possible to help us in our decision-making. The records collected included information about the GPS coordinates of the property. Based on these coordinates, additional data was extracted using the OpenStreetMap API. Specifically, for each property, the distance to the nearest hospital, grocery store, public transportation stop, and distance to the city center were obtained. Other data could also be obtained in this way, such as distance to schools, restaurants, and more. The user has the possibility to add information about points of interest according to his needs and preferences.

C. Dataset Information

As mentioned, two sets of data were collected. The first focused on apartments sold in Žilina, the second focused on rental apartments in Žilina. Both sets were extended with additional information obtained through OpenStreetMap. The data contained in these datasets date from February 2024 to July 2024.

TABLE I
Description of raw datasets

	Dataset	
	zilina_rent	**zilina_sell**
Number of Entries	1032	2049
Number of Columns	98	110

D. Used Analytical Platforms

The two platforms used in this study were Power BI Desktop [11], [12] and Oracle Analytics Cloud. The effort was to find tools that would be available for free. This condition is not met by Oracle Analytics Cloud, which requires a credit to create an instance. The decision to select Oracle Analytics Cloud was made because the May 2024 version of Oracle Analytics Desktop was experiencing issues when training the prediction model that could not be resolved. In the Oracle Analytics Cloud version, the training of the prediction model passed without issue.

The first platform used in this study is Power BI Desktop version 2.131.901.0. Power BI is a powerful business analytics tool developed by Microsoft that allows users to visualize and analyze data with greater efficiency and understanding. It provides interactive visualizations and business intelligence capabilities with an intuitive interface that is simple enough for end users to create custom reports and dashboards. Power BI supports a wide range of data sources including databases, spreadsheets, cloud services, local data warehouses, web APIs or text files. It supports connecting to and retrieving data from all widely used database platforms such as MS SQL, Oracle Database, PostgreSQL and more. Data sources in this case can also be Excel sheet files, CSV, JSON files and more. Power BI also allows you to access and retrieve data from web pages or web APIs, which makes it a powerful tool not only for data analysis and visualization but also for automated data retrieval and extraction. In terms of working with data, the Power BI application can be divided into two basic parts. The first part is for creating visualizations. The second part is the **Power Query** tool [13]. The latter allows performing various operations on data. Using it, data can be cleaned, transformed, filtered and linked between different data sources.

The second platform used is Oracle Analytics Cloud. Oracle Analytics is a comprehensive suite of business intelligence and data analytics tools designed to help organizations make data-driven decisions. The platform provides powerful features that enable users to collect, process, analyze, and visualize data from a variety of sources, making it easier to gain insights and plan strategically. Oracle Analytics supports seamless integration with a wide range of data sources, including databases, cloud storage, and real-time streaming data. It also allows loading files in various formats such as excel sheet, CSV, txt but does not support loading and processing files in JSON or XML formats. The tool can be divided into several parts in terms of working with data, with **Dataset**, **Workbook** and **Data flow** being three of the main ones. In the Dataset section, you can load data from various sources, transform columns, remove them, change their types, extract data from them, join them with other data sources, add and remove columns, change column names, and so on. Another part is Data flow. This requires already a specific dataset to work with on the input. The functionality of this part largely overlaps with the functionality available in the Dataset part. Also here it is possible to add and remove columns, add data sources, rename them and so on. In addition, it allows you to filter the data as needed and also to use the modified data to create one of the pre-built machine learning models designed for numerical prediction, classification or clustering [14]. The advantage of Data flows is that once a data flow is created, it can be simply applied to new datasets with the same structure. The last important part is the Workbook, which allows the creation of different types of visualizations and reports.

E. Dataset Transformation

The raw datasets do not have the required format suitable for creating a prediction model. The records need to be preprocessed. Preprocessing, transformation and extraction of data from dataset columns is performed using individual analytical tools (Power BI, Oracle Analytics). However the main ideas behind the processing steps are the same for each of the cases.

F. Predictive Model Training

After processing the datasets, the data will be used to create prediction models that will predict rent levels given the parameters of the property. Power BI does not include any option for easy creation of prediction models, but it can be used to construct at least a simple linear prediction model. In the case of Oracle Cloud, the range of available prediction models is wider. For numerical predictions, it provides linear regression model, elastic net linear regression model, random forest and decision tree. In the case of linear regression, lasso or ridge regularization can also be used. Most of these models provide hyperparameters that can be tuned. The scikit-learn library is used to train the tuned models. From this library, implementations for linear regression, linear regression using lasso or ridge, decision tree, random forest and k-nearest neighbors are used. For each model, a subset of hyperparameters is selected and tuned using grid search. The selected subset of hyperparameters, is identical to the hyperparameters available in the models on Oracle Analytics Cloud. The parameters obtained by tuning the models in scikit-learn learn will also be used in the creation of the Oracle Analytics Cloud models to see if the models need to be tuned on these platforms or if they perform well enough with the default settings.

Before the actual process of training the prediction model, it is necessary to divide the data into training and test sets. There are many recommendations that should be taken into account when splitting the data. One of the most important is data shuffling, which ensures that the data is random, thus eliminating any natural order or structure that could bias the performance of the model. This recommendation is exactly what is needed to be ensured. When splitting the dataset, the standard 80/20 ratio will be followed, i.e. 80% of the original data will be used to train the model and the remaining 20% will be used to test the model.

Data normalization is also an important step in preparing data for prediction model training. It is assumed that less technically savvy users who choose to use tools such as Power BI to process the data and create a simple prediction model will not have a mathematical background and therefore normalization could contribute to confusion for these users. A further complication is that normalized data is more difficult for an inexperienced user to interpret. In order to be as close as possible to the realistic use of such tools by ordinary users, this step will only be applied if it is easily achievable on the platform. Oracle Analytics Cloud offers this option, so data normalization will be applied during training of models with partial hyperparameter tuning. In the case of models from the scikit-learn library, normalization will also be applied.

IV. Data Preprocessing

By examining the Table I it is possible to observe a mismatch between the number of columns in each dataset. This is due to the inconsistent structure of the advertisements. Not every advert contained all the parameters. Another problem is the inconsistency in the names of the parameters, which provide the same information about the same key property characteristic (or can be used to approximate or infer it), but their names and types are different. Such columns need to be merged into a single column and the original columns should be removed from the dataset.

Datasets contain columns with a categorical feature. However, a prediction model based on linear regression is not able to handle categorical variables. One solution is to transform such a column containing a categorical feature into multiple binary columns - this is an application of the one hot encoding method.

Although data were collected systematically and efforts were made to minimize duplicate records, duplication could not be completely avoided. Advertisers often removed the original record and then added it with the same parameters (or changed the original name), thus changing the URL of the advertisement. Since the URLs of the adverts were the unique identifier of the record, once they were changed, it was no longer possible to determine whether or not the property was already in the database. Thus, URLs could not be used to filter out duplicate rows. However, it can be assumed that based on certain key parameters of the property (area, price, GPS coordinates, etc.), duplicates can be identified and subsequently removed.

To train good prediction models, it is necessary to have good quality data that will not bias the model's predictions in a significant way. For this reason, it will be necessary to identify any outliers at key features of the model and then filter out these undesirable entries. A histogram will be used to identify outliers and an eyeballing method will be used to identify values that will then be filtered out.

The resulting datasets contain features that are irrelevant to the prediction model, or there is only a small percentage of records that have such a feature present. Such features need to be removed from the datasets.

To build a prediction model, it is necessary that all the attributes needed for prediction take not-null values. Thus, in datasets it will be necessary to solve the problem of missing data in columns. There are several ways to deal with missing values. In this research, the simplest option is chosen, which is to remove the records with missing value.

Performing these preprocessing steps results in datasets with the following characteristics.

TABLE II
Description of preprocessed datasets

	Dataset	
	zilina_rent	**zilina_sell**
Number of Entries	440	735
Number of Columns	42	42

After processing, the datasets contain the following columns:

- **street** - categorical feature - the street on which the real estate is located. It is not used in predictions, but can be used in visualizations.
- **realEstateState** - categorical feature - the state of the real estate. It takes the values “Complete reconstruction”, “Partial reconstruction”, “New building” and “Original state”. This feature is not used in its original form for predictions, but is suitable for use in visualizations and data filtering.
- **gps_cordsLatitude** - decimal number - the latitude coordinate of the real estate. It is not used in predictions but can be used in creating visualizations.
- **gps_cordsLongitude** - decimal number - the longitude coordinate of the real estate. It is not used in predictions but can be used to create visualizations.
- **area** - decimal number - represents the area of the property.
- **priceNum** - decimal number - in the case of the sold flats dataset, this feature represents the sale price of the property. In the case of the dataset on rented properties, this attribute represents the rental price. The role of the prediction model will be to predict the rental price.
- **nearestHospitalDistance** - decimal number - distance to the nearest hospital.
- **nearestGroceryDistance** - decimal number - distance to the nearest grocery store.
- **nearestPublicTransportStopDistance** - decimal number - distance to the nearest public transport stop.
- **cityCenterPositionDistance** - decimal number - distance from the city centre.
- **numberOfParkingPlaces** - whole number - number of parking places.
- **numberOfBalconies** - whole number - number of balconies.
- **numberOfLodges** - whole number - number of lodges.
- **numberOfTerraces** - whole number - number of terraces.
- **numberOfCellars** - whole number - number of cellars.
- **numberOfRooms** - whole number - number of rooms.
- **completeReconstruction** - binary - column is based on the categorical feature “realEstateState” and indicates whether the property is a completely reconstructed property.
- **partialReconstruction** - binary - column created based on the categorical feature “realEstateState” and indicates whether the property is a partially reconstructed property.

- **newBuilding** - binary - column was created based on the categorical feature “realEstateState” and indicates whether it is a new property.
- **originalState** - binary - column was created based on the categorical feature “realEstateState” and indicates whether the property is in its original state.

V. Achieving Objectives On Specific Platforms

A. Data Loading

The data that needed to be loaded comes from the local MongoDb database. This data was in JSON format and contained nested JSON objects.

Power BI does not directly provide a connector in itself that would allow direct connection to the local MongoDb database. The database data was therefore exported to a text file in JSON format. Loading of the JSON file went without any problems. Power BI automatically recognized that these were JSON objects and loaded them correctly. Once loaded, Power BI automatically performed several steps which included expanding the nested JSON objects and changing the data types for the columns where it could be determined.

Oracle Analytics Cloud did not provide the ability to connect to the local MongoDb database directly. Therefore, the data was exported to a text file in JSON format. Oracle Analytics provided the option to load the text file but after loading it did not identify that it was a JSON format and was not able to parse the data correctly. In order to continue with the study, the JSON file was parsed using Python and saved as an Excel worksheet. This file has now been successfully loaded. In the loaded data, there were columns that contained nested JSON objects. Oracle Analytics did not provide an easy way to expand these JSON objects, or an easy way to manually extract the values. Although there is a possibility of replacing and extracting data using regular expressions, which definitely pleases programmers, but it can be assumed that ordinary users would not be able to use this option easily. Thus, data extraction from JSON objects could not be realized easily. Using Python, a new Excel worksheet was created that now contained expanded JSON objects.

B. Merging Columns

There were several columns in the dataset that carried the same information but differed in their names. For these columns it was necessary to create a new column that would combine the values from the original columns under specific conditions. An example is the subset of columns informing about the number of cellars for a given real estate. The following columns were present:

- **Number of cellars** - number of cellars.
- **Cellar area** - cellar area.
- **Cellar** - existence of a cellar.

In such cases, the goal is to create a *“numberOfCellars”* column that contains the number of cellars for the real estate. The procedure for combining the columns is as follows, if a value *“Number of cellars”* is defined for a record this value is taken to the output column *“numberOfCellars”*. If it is not defined, it is checked to see if a value is defined for the *“Cellar area”* column, if so, the real estate is assumed to have 1 cellar. If neither is defined, it is checked whether there is a non-null value in the *“Cellar”* column and whether it is different from the *“No”* string. In this case, it is assumed that 1 cellar belongs to the property. In other cases, a value of 0 is assigned.

Power BI in the Power Query section allows you to create a new column. A simple combination of columns according to certain rules could not be achieved by using a predefined function. However, in the Power Query section for adding a new column, an expression can be defined using a simple combination of *“IF condition THEN expression ELSE expression”* statements that will provide the desired result.

Oracle Analytics also provides an easy way to add a new column in the Dataset or Data flow sections. Again, the desired functionality can be achieved through the column value editor, where a simple combination of *“CASE WHEN condition THEN expression ELSE expression END”* statements can be used to achieve the desired result.

C. Categorical Column Transformation

There is a categorical feature *“realEstateState”* in the data, which needs to be transformed into multiple binary columns using the one hot encoding method for the purpose of training the prediction model.

In Power BI, the same technique was used as in the case of column merging. For each categorical value, a new binary column was defined and using the *“IF condition THEN expression ELSE expression”* statement, a value of 0 or 1 was assigned to a particular record depending on the column value.

Oracle Analytics was able to solve this problem by using the “CASE WHEN condition THEN expression ELSE expression END” statement when creating a new column. In the case where such columns are created only for the training of a prediction model, Oracle Analytics provides an option in the Data flow section when training a linear regression model to automatically convert the categorical features. Two automatic encoding options, Indexer and Onehot, are available there.

D. Removing Duplicates

After preprocessing the data, it is necessary to remove duplicates that could cause problems when training the prediction model.

Power BI in the Power Query section allows you to select columns and then find and remove duplicate records based on those selected columns.

Oracle Analytics does not provide any direct and easy procedure for removing duplicate rows. It is probable that record deduplication can also be achieved by a combination of several steps, such as sophisticated use of aggregation, or the creation of a column containing a string constructed from all the values in the columns on whose basis we want to perform deduplication. It can be assumed that an ordinary user would not have figured out these non-intuitive methods of deduplication and would thus not be able to perform duplication removal.

E. Outliers

The process of outlier removal can be divided into two steps - outlier identification and outlier removal. For outlier identification, a straightforward way is to use the eyeballing method, where the data is displayed in a suitable type of graph and the presence of outliers is visually identified. In this study, histogram plotting is used to identify outliers. Using it, values that are significantly different from the others are identified and these are subsequently removed from the dataset.

There is no easy way to plot a histogram in Power BI at the time of this study. Respectively, it does not provide an easy way to define bin counts and ranges based on which the values would be automatically divided. However, there is the possibility of using third party solutions. Once the histogram has been plotted, it is possible to easily identify outliers and then remove these values in the Power Query section using filters.

Oracle Analytics in the Dataset section plots a histogram for each integer column and also allows you to define the number of bins along with their ranges, which greatly simplifies the process of identifying outliers. The disadvantage is that these outliers cannot be filtered out directly in the Dataset section but need to be switched to the Data flow section where we can define a specific filter.

F. Prediction Model Training

Based on the processed data, it is necessary to create a prediction model. Before the model can be trained, the data must be properly split 80/20 into training and test sets. The training set will be used to train the prediction model and the test set will be used to evaluate the predictive ability of the model.

In Power BI, there is no easy way to simply shuffle and split data into groups of the desired size. However, this can be achieved by a sequence of several steps. In Power Query, a new column is created and a random value from 0 to 1 is assigned to it. Based on this value, the entire set is sorted to ensure randomness. An index column will be added to the sorted records, assigning each column a value from the sequence, starting from 1. Then, depending on the number of records in the set, a split index can be computed, which will be located in 80% of the dataset. A copy of the dataset will be created and only records with an index lower than the split index value will be kept in this copy. This created the training set. Only records with index greater than the split index value will be retained in the original dataset, which will create a test set. Although this splitting process is tedious, it is intuitive and is not expected to be a problem for non-programmers.

The generated training dataset will be used to train linear regression model. In the visualization part of Power BI, a new table containing the intercept and coefficients of the predictors is computed based on the training dataset using the DAX function *LINEST*, which accepts the dependent variable and independent variables as input. These model parameters are then used in the prediction of the values. Power BI will not directly provide a performance evaluation of the trained model. Creating a linear regression model in Power BI is non-intuitive and may be a challenge for non-programmers. Power BI allows you to show the resulting model parameters. The individual coefficients are listed as well as the intercept value.

In Oracle Analytics, splitting the data and then training a linear regression model is easy. Model training is done in the Data flow section, where model training is added as the final step of data processing. As mentioned before Oracle Analytics Cloud provides several numerical prediction models. For each of the models, the initialization procedure is almost the same. In the modal window for setting the training parameters, it is necessary to select the target variable to be predicted. In this section it is also possible to set the hyperparameters of the model. In addition, it is possible here to split the data into training and test sets in a specified ratio, apply data normalization, select the desired transformation method for categorical variables, and select a method to deal with missing values if such are present in the dataset. Finally, the name under which the model is to be stored is defined. Once the Data flow is executed, the model training is started and when it is finished, the new model is available on the home page in the *''Machine Learning"* section. By clicking on the trained model, the *"Inspect"* option can be used to examine the performance of the model along with its coefficients. The processes of model definition, data partitioning, replacing missing values, standardization, model training, and evaluation are intuitive in the Oracle Analytics environment and therefore suitable for non-programmers. Oracle Analytics provides a way to view the resulting linear regression model parameters. However, the output lacks the *intercept* parameter and the individual coefficients are rounded to three decimal places.

In order to compare the prediction models created by each platform as accurately as possible, a training and testing dataset was created using Python. This was used as input to the training process on each platform. In Oracle Analytics, the user has to define the size of the training and test sets. It is not possible to define that 100% of the data is to be used for training. For this reason, the portion of data to be used for training has been set to 99%. For comparison, several prediction models were trained on the training set in Python using the scikit-learn library. The resulting model parameters are shown in the Table III. The performance evaluation on the exact test set is shown in Table IV.

The parameters trained determined by *LINEST* in Power BI are much the same as those trained by scikit-learn. They differ only in the case of the columns created by encoding the categorical column *"realEstateState"* and in the value *intercept*. The parameters determined during training in Oracle Analytics differ significantly from the others. Additionally, for the columns *"nearestHospitalDistance"* and *"numberOfRooms"*, the coefficients were not specified at all. Similarly, it was not possible to determine the value of *intercept*.

G. Predicting Values

The trained models are used to predict the rent for each record in the test set in order to evaluate their performance.

TABLE III
TRAINED MODELS PARAMETERS

Coefficient	Platform		
	Power BI	OAC	scikit
area	6.586	7.006	6.586
cityCenterPositionDistance	-0.005	-0.016	-0.005
nearestGroceryDistance	0.02	0.003	0.02
nearestHospitalDistance	-0.024	*NA*	-0.024
nearestPublicTransportStopDistance	-0.025	-0.022	-0.025
numberOfRooms	13.834	*NA*	13.834
numberOfBalconies	-19.602	-19.274	-19.602
numberOfCellars	-36.891	-36.983	-36.891
numberOfParkingPlaces	73.189	54.498	73.189
numberOfLodges	7.263	11.214	7.263
numberOfTerraces	73.925	75.757	73.925
newBuilding	53.097	113.382	59.905
originalState	0	61.069	6.808
completeReconstruction	-30.504	25.696	-23.696
partialReconstruction	-49.825	5.058	-43.017
intercept	270.184	*NA*	263.376

TABLE IV
MODELS EVALUATION

Model	Metric				
	MAE	MSE	RMSE	R^2	Δ MSE(%)
scikit-linear	98.23	22020.83	148.39	0.64	129.52
scikit-lasso	96.08	21940.78	148.12	0.65	128.69
scikit-ridge	98.22	22021.62	148.4	0.64	129.53
scikit-tree	78.02	11003.83	104.9	0.82	14.69
scikit-forest	72.35	**9594.1**	97.95	0.85	0.0
scikit-knn	77.51	12142.62	110.19	0.8	26.56
PowerBI-linear	100.82	**22131.78**	148.77	0.64	130.68
OAC-linear	100.81	22133.83	148.77	0.64	130.7
OAC-lasso-def	98.87	22020.07	148.39	0.65	129.52
OAC-ridge-def	100.8	22148.71	148.82	0.64	130.86
OAC-elastic-def	109.47	27838.2	166.85	0.55	190.16
OAC-tree-def	112.39	30224.29	173.85	0.51	215.03
OAC-forest-def	151.47	52970.23	230.15	0.15	452.11
OAC-lasso	98.61	22043.54	148.47	0.64	129.76
OAC-ridge	100.81	22135.27	148.78	0.64	130.72
OAC-tree	95.31	**18494.03**	135.99	0.7	92.76
OAC-forest	116.67	25069.5	158.33	0.6	161.3

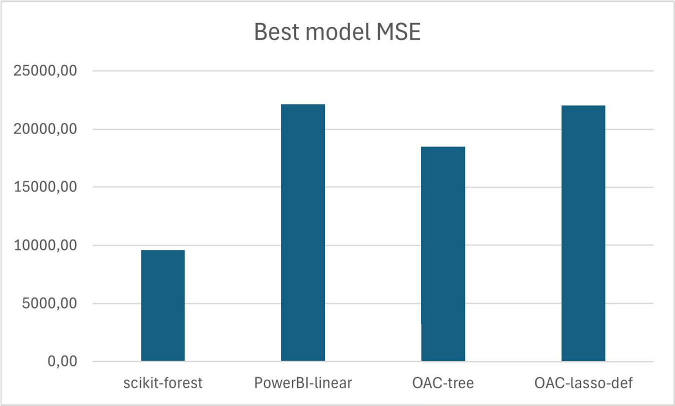


Fig. 1. The best models on the platform with respect to MSE.

After evaluation these models can be used to predict rent rates for each of the records in the sold property dataset.

In Power BI, to predict values, it is necessary to create a new column for the desired dataset in the visualization section, into which the predicted value for the record is inserted via a DAX expression. This DAX function needs to be manually defined according to $slope_1 * x_1 + \ldots + slope_n * x_n + intercept$. The values $slope_i$ and $intercept$ come from a table that was created as output when training the prediction model. The values of x_n represent the corresponding values within the record. This procedure is dependent on the dataset used and it is not possible to simply change the source dataset. When predicting values on a different dataset, the values of x_i need to be correctly referenced. The prediction process in Power BI is non-intuitive and prone to errors.

In Oracle Analytics, value prediction is performed in the Data flow section. Here, the data source is defined at the beginning and then the already trained model is added. If required, it is possible to define a mapping of the columns of the input data source to the inputs of the prediction model. The last step in the Data flow section is to save the dataset enhanced with the predicted value. After running the created Data flow, a new dataset will appear on the main screen in the Datasets section. To change the source data, simply open the saved Data flow and change the data source. The other components of the Data flow remain unchanged. Predicting new values is simple and user-friendly on the Oracle Analytics platform.

VI. RESULTS

For each platform, the available prediction models were trained. For scikit-learn, a linear regression model, a lasso linear regression model, a ridge linear regression model, a decision tree, a random forest, and k-nearest neighbors were trained and tuned. The hyperparameters of the models were tuned using grid search. In PowerBI, only the linear regression model was trained. In Oracle Analytics Cloud, the available models for predicting numerical values were trained. Specifically, these were the linear regression model, lasso linear regression model, ridge linear regression model, elastic net regression model, decision tree, and random forest. These models (except for the linear regression) were trained in two configurations, where in the first case they were trained using the default hyperparameters and in the second case hyperparameters similar to those found during the process of tuning the models from the scikit-learn library were used. It was not possible to use the same hyperparameters as the hyperparameter scales differed between the OAC and scikit-learn models.

For each trained model, metrics of mean absolute error (MAE), mean squared error (MSE), root mean square error ($RMSE$), and R squared (R^2) were evaluated. MSE was chosen as a most important indicator of model performance.

The results can be seen in Table IV. Here, for each platform, the lowest MSE value achieved by any of the models on that platform is highlighted. The best value is marked in bold.

The table also contains a column ΔMSE, which gives the percentage difference between the MSE for the given model and the MSE of the overall best model found.

In the case of scikit-learn, scikit-forest, which is a tuned random forest model, was found to be the best model. It achieved MSE with a value of 9594.1, making it also the best performing model among all the trained models. In PowerBI, only the linear regression model was trained, and its MSE was 22131.78, a difference of more than 130% with respect to the best model. In Oracle Analytics Cloud, the best model was OAC-tree, which was a decision tree model partially tuned using the hyperparameters obtained while training the decision tree model in scikit-learn. OAC-tree achieved a MSE of 18494.03, which is approximately 93% difference to the best model. Among the OAC models trained without tuning the hyperparameters, OAC-lasso-def achieved the best results with MSE 22020.07 and thus a 129% difference with respect to the best model.

VII. Discussion and Conclusion

The study compared the Power BI and Oracle Analytics Cloud platforms in the context of how these platforms can be used by less tech-savvy users to make data-driven decisions. The purchase of an investment property was chosen as a use case. In order to get as close as possible to real use cases, a dataset containing data on sold and rented apartments in Žilina, obtained from the website of a local advertiser, was collected. For the chosen platforms, it was examined whether the platform can be used to perform the common tasks required to transform and preprocess the data. Subsequently, the prediction models trained using these platforms were compared with the results of the trained and tuned models from the scikit-learn library.

The study reveals that both platforms allow performing basic data operations and training some sort of prediction models. However, the platforms differ in their ease of use in achieving each objective. They also differ in the options available for prediction models.

Power BI has generally proven to be more intuitive and user-friendly when preprocessing data. Power BI provided a large number of usable data sources. It made it easy to combine columns, extract values, remove columns, and filter records based on various criteria. The large community and the existence of a number of tutorials to solve many of the problems is also a major advantage for this platform. A drawback was the inability to plot a histogram, which is necessary when analyzing data and identifying outliers. A major drawback is the absence of prediction models in the free version of this platform. However, it is possible to build a linear regression model, but the creation of prediction model and subsequent prediction of the values using the Power BI platform is challenging and error prone. The MSE values obtained by this model differed by 130% from the results of the best model, which is a scikit-learn random forest with tuned hyperparameters.

The Oracle Analytics platform is also user-friendly, but not as easy to use as Power BI. A possible reason for this is the division of the platform into several parts, namely Dataset, Data flow and Workbook. Some functionality over the data could be done in the Dataset part but also in the Data flow part. However, some functionalities related to data transformation were available in either one or the other part, which required frequent switching between the parts. The advantage was the automatic creation of histogram for each numeric column in the Dataset section. The user could easily define the number of bins, their ranges and thus easily identify outliers. However, the disadvantage was that outlying values could not be directly filtered out in the Dataset section, but it was necessary to switch to the Data flow section, where such filtering on the basis of specific values was possible. A significant shortcoming of the platform was the inability to easily remove duplicate records. Another shortcoming of Oracle Analytics was the inability to use a data source that contained data in JSON or XML format. The platform was unable to process data in these formats or at least easily extract the desired values from them. Another disadvantage of this platform is the small community and the lack of guidelines or tutorials on how to solve some tasks. The Oracle Analytics platform excelled in the task of creating a prediction model and in using the trained models to predict values. It provides several models to predict numerical values. Some hyperparameters can also be defined for these models. However, these hyperparameters cannot be easily tuned directly on the platform. The advantage is also the possibility to determine whether to normalize the input data, how to deal with missing values and also to effortlessly split the data set into training and test data in the desired ratio. Several models were trained through Oracle Analytics Cloud, with the OAC-tree model using a partially tuned decision tree being the best with respect to MSE. This achieved MSE of 18494.03, which differed from the best model by approximately 93%.

From the evaluated results, it is noticeable that more complex and tuned models perform better in predictions on the collected dataset. The results also show a large difference between the results of the best model obtained through the analytics platform and the results of the best model obtained by training and tuning using a specialized library. Future research could explore ways to reduce these performance differences on the used analytics platforms.

The insights gained in this study can be used in deciding which platform to choose when needing to make data-driven decisions. We assume that this study also provides a concise and clear guide on how to solve certain data processing tasks on each platform.

Acknowledgment

It was supported by the Erasmus+ project: Project number: 2022-1-SK01-KA220-HED-000089149, Project title: Including EVERyone in GREEN Data Analysis (EVERGREEN) funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the Slovak Academic Association for International Cooperation (SAAIC). Neither the European Union nor SAAIC can be held responsible for them.

Co-funded by the European Union

EverGreen DATA Analytics

References

[1] D. H. Jonassen, "Designing for decision making," *Educational technology research and development*, vol. 60, pp. 341–359, 2012.

[2] A. McAfee, E. Brynjolfsson, T. H. Davenport, D. Patil, and D. Barton, "Big data: the management revolution," *Harvard business review*, vol. 90, no. 10, pp. 60–68, 2012.

[3] E. Kriegova, M. Kudelka, M. Radvansky, and J. Gallo, "A theoretical model of health management using data-driven decision-making: the future of precision medicine and health," *Journal of translational medicine*, vol. 19, pp. 1–12, 2021.

[4] J. A. Marsh, J. F. Pane, and L. S. Hamilton, "Making sense of data-driven decision making in education: Evidence from recent rand research. occasional paper." *Rand Corporation*, 2006.

[5] B. Gill, B. C. Borden, and K. Hallgren, "A conceptual framework for data-driven decision making," *Final Report of Research conducted by Mathematica Policy Research, Princeton, submitted to Bill & Melinda Gates Foundation, Seattle, WA*, 2014.

[6] A. Bousdekis, K. Lepenioti, D. Apostolou, and G. Mentzas, "A review of data-driven decision-making methods for industry 4.0 maintenance applications," *Electronics*, vol. 10, no. 7, p. 828, 2021.

[7] E. Brynjolfsson, L. M. Hitt, and H. H. Kim, "Strength in numbers: How does data-driven decisionmaking affect firm performance?" *Available at SSRN 1819486*, 2011.

[8] J. Duggan, "The case for personal data-driven decision making," *Proceedings of the VLDB Endowment*, vol. 7, no. 11, pp. 943–946, 2014.

[9] A. Ferrari and M. Russo, *Introducing Microsoft Power BI*. Microsoft Press, 2016.

[10] H. Helskyaho, J. Yu, K. Yu, H. Helskyaho, J. Yu, and K. Yu, "Oracle analytics cloud," *Machine Learning for Oracle Database Professionals: Deploying Model-Driven Applications and Automation Pipelines*, pp. 187–203, 2021.

[11] A. Aspin, *Pro Power BI Desktop*. Springer, 2016.

[12] D. Clark and D. Clark, "Introducing power bi desktop," *Beginning Power BI: A Practical Guide to Self-Service Data Analytics with Excel 2016 and Power BI Desktop*, pp. 193–216, 2017.

[13] C. Webb *et al.*, *Power query for power BI and Excel*. Apress, 2014.

[14] T. Jain, M. Agarwal, A. Kumar, V. K. Verma, and A. Yadav, "Building machine learning application using oracle analytics cloud," in *Data Engineering for Smart Systems: Proceedings of SSIC 2021*. Springer, 2022, pp. 361–375.

A Similarity Space Approach to the 1/3-2/3 Conjecture in Partially Ordered Sets

Ondřej Rozinek
University of Pardubice
Pardubice, Czech Republic
ondrej.rozinek@gmail.com

***Abstract*—The 1/3-2/3 Conjecture, a longstanding open problem in combinatorics, posits that in any non-chain finite partial order, there exists a pair of elements (x, y) such that the probability $\mathbb{P}(x \prec y)$ lies in the interval $[1/3, 2/3]$. This paper presents a novel approach to addressing this conjecture by leveraging the concept of similarity spaces. We introduce the notion of a 'balance constant' within the framework of similarity spaces and prove that its supremum is 1/2, confirming a related conjecture. Our main contribution is the proof that the infimum of normalized similarity in posets is 1/3, achieved in the case of total disorder. This result provides a new perspective on the 1/3-2/3 Conjecture, connecting it to the theory of similarity spaces. We also establish relationships between edit distance, swap operations, and Generalized Rozinek Similarity, offering insights into the structure of totally disordered sequences. While we provide a proof of the 1/3-2/3 Conjecture in the general case using our approach, we acknowledge that a more rigorous and detailed proof is still needed to fully resolve this longstanding problem in combinatorics and order theory.**

***Index Terms*—1/3-2/3 Conjecture, Sorting, Partial Order, Linear Extension, Similarity Space**

I. Introduction

Definition 1. *Consider a fixed, underlying partially ordered set (poset) $(\mathcal{P}, \leq)$. The notation $\mathbb{P}(x \prec y)$ represents the probability that element x precedes element y in a uniformly random linear extension of $\mathcal{P}$. By definition, $\mathbb{P}(x \prec x) = 0$ for any element x.*

The $\frac{1}{3}$-$\frac{2}{3}$ Conjecture, first proposed in 1968, asserts that in any finite partial order that is not a chain, there exists a pair of elements (x, y) such that $\mathbb{P}(x \prec y)$ falls within the range $[\frac{1}{3}, \frac{2}{3}]$. This conjecture was independently formulated by Kislitsyn [1], Fredman [2], and Linial [3], each of whom considered its implications for sorting theory. Specifically, the conjecture implies that the number of comparisons required to fully sort elements within the known partial order $\mathcal{P}$ is at most $(1+o(1))\log_{\frac{3}{2}} e(\mathcal{P})$, which is within a constant factor of the information-theoretic lower bound $\log_2 e(\mathcal{P})$. Here, $e(\mathcal{P})$ denotes the number of linear extensions of $\mathcal{P}$.

Definition 2 (Balance Constant)**.** *The* balance constant *of poset $\mathcal{P}$ is*

$$\delta(\mathcal{P}) = \max_{(x,y)\in\mathcal{P}^2} \min\{\mathbb{P}(x \prec y), \mathbb{P}(y \prec x)\} \tag{1}$$

Conjecture 1 (1/3-2/3 Conjecture)**.** *If $\mathcal{P}$ is a finite poset that is not totally order, then $\delta(\mathcal{P}) \geq \frac{1}{3}$*

Brightwell [4] deemed it "one of the major open problems in the combinatorial theory of partial orders".

Now we introduce related conjecture for maximum balance constant.

Conjecture 2 (1/2 Conjecture [5])**.** *Maximum balance constant of an arbitrary poset $\mathcal{P}$ in limit case is $\sup \delta(\mathcal{P}) = \frac{1}{2}$.*

There are many types of posets for which the Conjecture 1 has already been proven. This includes posets with width 2 by Linial [3], posets with height 2 by Trotter, Gehrlein, and Fishburn [6], posets with 6-thin by Peczarski [7], semiorders by Brightwell [8], N-free posets by Zaguia [9] and posets whose Hasse diagram is a forest by Zaguia [10].

However, a general proof of Conjecture 1 for all types of posets has not yet been established, indicating ongoing research challenges in poset theory.

Example 1. *Consider a poset $\mathcal{P}$ consisting of three elements $\{a, b, c\}$ with a single comparability relationship given by $a \leq b$. This poset $\mathcal{P}$ has three distinct linear extensions: $a \leq b \leq c$, $a \leq c \leq b$ and $c \leq a \leq b$.*

Thus, for the pair (a, c), there are two linear extensions where a precedes c and one where c precedes a. This observation illustrates that the probability $\mathbb{P}(a \prec c)$ falls within the range $[\frac{1}{3}, \frac{2}{3}]$, and similarly, $\mathbb{P}(c \prec a)$ does as well. Consequently, the pair (a, c) in this poset satisfies the condition of the $\frac{1}{3} - \frac{2}{3}$ Conjecture.

II. Similarity Space

The notion of metric spaces is a well defined mathematical concept, known for a century. On the other hand, the similarity space is a quite new developed theory [11]. The relationship between metric and similarity space is not obvious, as metric space derives from spatial considerations and similarity relations derive from considering common and non-common features. Recall the definition.

Definition 3 (Similarity Space [11]–[15])**.** *A similarity on a nonempty set $\mathcal{X}$ is a function $s\colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}^+$ such that for all elements $x, y, z \in \mathcal{X}$:*

(S1) $s(x, y) = s(y, x)$ (symmetry),
(S2) $s(x, z) + s(y, y) \geq s(x, y) + s(y, z)$ (triangle inequality),
(S3) $s(x, x) = s(x, y) = s(y, y) \iff x = y$ (identity of indiscernibles),
(S4) $s(x, y) \geq 0$ (non-negativity),

979-8-3503-8768-1/24/$31.00 ©2024 IEEE

(S5) $s(x, y) \leq \min\{s(x, x), s(y, y)\}$ (bounded by self-similarity).

A similarity space is an ordered pair $(\mathcal{X}, s)$ such that $\mathcal{X}$ is a nonempty set and s is a similarity on $\mathcal{X}$.

Definition 4 (Normalized Similarity Space)**.** *A function $s_n(x, y)\colon \mathcal{X} \times \mathcal{X} \to [0, 1] \subset \mathbb{R}$ is a normalized similarity if all elements $x, y, z \in \mathcal{X}$ satisfy the following conditions:*

(N1) $s_n(x, y) = s_n(y, x)$ (symmetry),

(N2) $s_n(x, z) + 1 \geq s_n(x, y) + s_n(y, z)$ (triangle inequality),

(N3) $s_n(x, y) = 1 \Longleftrightarrow x = y$ (identity of indiscernibles),

(N4) $s_n(x, y) \geq 0$ (non-negativity),

(N5) $s_n(x, y) \leq 1$ (bounded self-similarity).

A normalized similarity space is an ordered pair $(\mathcal{X}, s_n)$.

A few issues require attention. The name 'similarity metric' is a convention already suggested in the preceding. Calling it a 'metric' should be understood in the sense of monotonously decreasing convex transformation of a partial metric or a distance metric [11]. In this paper, we use only the term 'similarity', and in this way we avoid misunderstanding. It is obvious that, it is possible that this definition allows us to have positive self-similarity $s(x, x) > 0$, and also the self-similarities may be different $s(x, x) \neq s(y, y)$. But if $x = y$, $s(x, y)$ may not be 0. The theory of similarity space is very close to the theories of metric spaces and partial metrics, and some parts of this paper were also inspired by these theories [16]–[18].

A basic example of similarity space is the ordered pair $(\mathbb{R}^+, s)$ defined as follows

$$s(x, y) = x \cap y = \min\{x, y\} = \frac{x + y - |x - y|}{2} \tag{2}$$

for all x, y in $\mathbb{R}^+$. Other examples of similarity spaces that are interesting in terms of broad practical application such as Jaccard index, Tanimoto coefficient, Generalized Rozinek similarity, Levenshtein similarity, longest common subsequence may be found in [11].

Theorem 1 (Duality of Metric and Similarity Space)**.** *Generally, if a function $f\colon \mathbb{R} \to [a, b] \subseteq \mathbb{R}$ is a monotonously decreasing convex function such that $b = f(0) > 0$ and $\lim_{n\to\infty} f(n) = a \geq 0$, then for a metric space $(\mathcal{X}, d)$, the similarity function s defined by*

$$s(x, y) = f(d(x, y)) \tag{3}$$

is a similarity on $\mathcal{X}$.

Theorem 2 (Generalized Rozinek Similarity [11])**.** *Let $(\mathcal{X}, s)$ be a similarity space and $(\mathcal{X}, d)$ be its corresponding metric space. The Generalized Rozinek Similarity $s_n : \mathcal{X} \times \mathcal{X} \to [0, 1]$ is defined as:*

$$s_n(x, y) = \frac{s(x, x) + s(y, y) - d(x, y)}{s(x, x) + s(y, y) + d(x, y)}$$

where $s(x, x)$ and $s(y, y)$ are positive self-similarities, and $d(x, y)$ is the distance between x and y.

Proof. [11] □

III. Reformulating Conjecture in Similarity Space

Let $e(\mathcal{P})$ denote the number of linear extensions of $\mathcal{P}$, and for $(x, y) \in \mathcal{P}$ is a linear order $\prec$ on $\mathcal{X}$ and $x \neq y$, let $(x \prec y; \mathcal{P})$ denote the number of linear extension of $\mathcal{P}$ in which x precedes y. Note that $e(x \prec y; \mathcal{P}) + e(y \prec x; \mathcal{P}) = e(\mathcal{P})$. For a poset $\mathcal{P}$, we express $\mathbb{P}(x \prec y) = \frac{e(x\prec y;\mathcal{P})}{e(\mathcal{P})}$ and $\mathbb{P}(y \prec x) = \frac{e(y\prec x;\mathcal{P})}{e(\mathcal{P})}$.

Lemma 1 (Balance Constant in Unnormalized Similarity Space)**.** *Let $(\mathcal{P}, s_u)$ be an unnormalized similarity space on poset $\mathcal{P}$ with unnormalized similarity, denoted s_u. Then the balance constant of the poset is equal to*

$$\delta(\mathcal{P}) = \frac{1}{e(\mathcal{P})} \max_{(x,y)\in\mathcal{P}^2} s_u(x \prec y, y \prec x). \tag{4}$$

Proof.

$$\delta(\mathcal{P}) = \max_{(x,y)\in\mathcal{P}^2} \min\{\mathbb{P}(x \prec y), \mathbb{P}(y \prec x)\} \tag{5}$$

$$= \max_{(x,y)\in\mathcal{P}^2} \min\left\{\frac{e(x \prec y; \mathcal{P})}{e(\mathcal{P})}, \frac{e(y \prec x; \mathcal{P})}{e(\mathcal{P})})\right\} \tag{6}$$

$$= \frac{1}{e(\mathcal{P})} \max_{(x,y)\in\mathcal{P}^2} \min\{e(x \prec y; \mathcal{P}), e(y \prec x; \mathcal{P})\} \tag{7}$$

$$= \frac{1}{e(\mathcal{P})} \max_{(x,y)\in\mathcal{P}^2} s_u(x \prec y, y \prec x) \tag{8}$$

□

Lemma 2 (Balance Constant in Normalized Similarity Space)**.** *Let $(\mathcal{P}, s_n)$ be normalized similarity space in range $[0, 1]$ on poset $\mathcal{P}$ with normalized similarity, denoted s_n. Then the balance constant of poset is equivalent*

$$\delta(\mathcal{P}) = \frac{1}{2} \max_{(x,y)\in\mathcal{P}^2} s_n(x \prec y, y \prec x) \tag{9}$$

$$= \frac{1}{2} \max_{(x,y)\in\mathcal{P}^2} \{1 - d_n(x \prec y, y \prec x)\} \tag{10}$$

where d_n is normalized metric defined on metric space $(\mathcal{P}, d_n)$.

Proof. According to Definition 4, we set $\mathcal{X} = \mathcal{P}$ and consider $s_n(x \prec y, y \prec x)$ where $x \prec y, y \prec x \in \mathcal{P}$. Since $\min\{\cdot\}$ forms a similarity space, we have:

$$
\begin{aligned}
\delta(\mathcal{P}) &= \max_{(x,y)\in\mathcal{P}^2} \min\{\mathbb{P}(x \prec y), \mathbb{P}(y \prec x)\} \\
&= \max_{(x,y)\in\mathcal{P}^2} \left\{ \frac{\mathbb{P}(x \prec y) + \mathbb{P}(y \prec x)}{2} \right. \\
&\quad \left. - \frac{|\mathbb{P}(x \prec y) - \mathbb{P}(y \prec x)|}{2} \right\} \\
&= \max_{(x,y)\in\mathcal{P}^2} \left\{ \frac{e(x \prec y;\mathcal{P}) + e(y \prec x;\mathcal{P})}{2e(\mathcal{P})} \right. \\
&\quad \left. - \frac{|e(x \prec y;\mathcal{P}) - e(y \prec x;\mathcal{P})|}{2e(\mathcal{P})} \right\} \\
&= \max_{(x,y)\in\mathcal{P}^2} \left\{ \frac{e(P)}{2e(\mathcal{P})} \right. \\
&\quad \left. - \frac{|e(x \prec y;\mathcal{P}) - e(y \prec x;\mathcal{P})|}{2e(\mathcal{P})} \right\} \\
&= \max_{(x,y)\in\mathcal{P}^2} \left\{ \frac{1}{2} - \frac{1}{2} \left| \frac{e(x \prec y;\mathcal{P})}{e(\mathcal{P})} \right. \right. \\
&\quad \left. \left. - \frac{e(y \prec x;\mathcal{P})}{e(\mathcal{P})} \right| \right\} \\
&= \max_{(x,y)\in\mathcal{P}^2} \left\{ \frac{1}{2} - \frac{1}{2} |\mathbb{P}(x \prec y) - \mathbb{P}(y \prec x)| \right\} \\
&= \frac{1}{2} \max_{(x,y)\in\mathcal{P}^2} \{1 - |\mathbb{P}(x \prec y) - \mathbb{P}(y \prec x)|\} \\
&= \frac{1}{2} \max_{(x,y)\in\mathcal{P}^2} \{1 - d_n(x \prec y, y \prec x)\} \\
&= \frac{1}{2} \max_{(x,y)\in\mathcal{P}^2} s_n(x \prec y, y \prec x). \qquad (11)
\end{aligned}
$$

The last equality follows from the relationship between normalized similarity and normalized distance in metric spaces: $s_n(x, y) = 1 - d_n(x, y)$ for all $(x, y) \in \mathcal{P}^2$. □

Suppose that at each step, we can identify a pair of incomparable elements (x, y) such that the proportion of linear extensions of $\mathcal{P}$ that place x before y, denoted by $\mathcal{P}(x \prec y)$, is equal to $\frac{1}{2}$. Then, we require at least $\log_2(e(\mathcal{P}))$ comparisons, where $e(P)$ represents the number of linear extensions of $\mathcal{P}$. However, this is not always achievable, as demonstrated by the Example 1. In this particular example, the only feasible values for $\mathcal{P}(x \prec y)$ are $\frac{1}{3}$ or $\frac{2}{3}$.

We begin with proving the Conjecture 2 and state this.

Theorem 3 (Maximum Balance Constant in Normalized Similarity Space)**.** *Let be normalized similarity space $(\mathcal{P}, s_n)$ with defined normalized similarity s_n on poset $\mathcal{P}$. Theoretical maximum balance constant $\delta(\mathcal{P})$ is given*

$$\sup_{x,y\in\mathcal{X}} \delta(\mathcal{P}) = \frac{1}{2}. \qquad (12)$$

Proof. Using Lemma 2, we have:

$$
\begin{aligned}
\sup_{x,y\in\mathcal{X}} \delta(P) &= \sup_{x,y\in\mathcal{X}} \frac{1}{2} \max_{(x,y)\in\mathcal{P}^2} s_n(x \prec y, y \prec x) \\
&= \frac{1}{2} \max_{(x,y)\in\mathcal{P}^2} \sup_{x,y\in\mathcal{X}} s_n(x \prec y, y \prec x)
\end{aligned}
$$

The interchange of $\sup$ and $\max$ is valid due to the finite nature of P and the continuity of s_n.

By Definition 4, specifically the identity of indiscernibles (N3) and bounded self-similarity (N5), we know that:

$$0 \le s_n(x \prec y, y \prec x) \le 1$$

with equality in the upper bound when $x = y$. Therefore:

$$\sup_{x,y\in\mathcal{X}} s_n(x \prec y, y \prec x) = 1$$

Substituting this back into our equation:

$$\sup_{x,y\in\mathcal{X}} \delta(P) = \frac{1}{2} \max_{(x,y)\in\mathcal{P}^2} 1 = \frac{1}{2}$$

Thus, the claim is proven. □

IV. Total Disorder in Sequences: A Characterization Using Generalized Rozinek Similarity

Definition 5 (Edit Distance)**.** *For sequences $x, y \in \Sigma^*$ over an alphabet Σ, the edit distance $d_E(x, y)$ is the minimum number of single-character edits (insertions, deletions, or substitutions) required to transform x into y.*

Definition 6 (Total Disorder)**.** *Two sequences $x, y \in \Sigma^*$ are said to be in total disorder if their Generalized Rozinek Similarity, when applied to edit distance, is exactly $\frac{1}{3}$.*

Theorem 4 (Characterization of Total Disorder)**.** *Let Σ be a finite alphabet and $x, y \in \Sigma^n$ be two sequences of length n. The following statements are equivalent:*

1) *x and y are in total disorder.*
2) *$s_n(x, y) = \frac{1}{3}$ when $s(x, x) = s(y, y) = n$ and $d(x, y) = d_E(x, y)$.*
3) *$d_E(x, y) = n$.*

Proof. (1) ⇒ (2): This follows directly from the definition of total disorder.

(2) ⇒ (3): Assume $s_n(x, y) = \frac{1}{3}$ when $s(x, x) = s(y, y) = n$ and $d(x, y) = d_E(x, y)$. Then:

$$
\begin{aligned}
\frac{1}{3} &= \frac{s(x,x) + s(y,y) - d_E(x,y)}{s(x,x) + s(y,y) + d_E(x,y)} \\
\frac{1}{3} &= \frac{2n - d_E(x,y)}{2n + d_E(x,y)}
\end{aligned}
$$

Solving this equation:

$$
\begin{aligned}
2n - d_E(x,y) &= \frac{1}{3}(2n + d_E(x,y)) \\
6n - 3d_E(x,y) &= 2n + d_E(x,y) \\
4n &= 4d_E(x,y) \\
d_E(x,y) &= n
\end{aligned}
$$

(3) ⇒ (1): Assume $d_E(x, y) = n$. Substituting this into the Generalized Rozinek Similarity formula:

$$\begin{aligned} s_n(x,y) &= \frac{s(x,x)+s(y,y)-d_E(x,y)}{s(x,x)+s(y,y)+d_E(x,y)} \\ &= \frac{n+n-n}{n+n+n} = \frac{n}{3n} = \frac{1}{3} \end{aligned}$$

This satisfies the definition of total disorder, completing the cycle of implications. □

Remark 1. *The condition $d_E(x,y) = n$ implies that for sequences in total disorder, the edit distance is equal to the length of the sequences. This means that every position in the sequence needs to be edited to transform one sequence into the other.*

Example 2. *Consider the sequences $x = abcdef$ and $y = fedcba$ over the alphabet $\Sigma = \{a,b,c,d,e,f\}$. Here, $n = 6$ and $d_E(x,y) = 6 = n$. We can verify:*

$$s_n(x,y) = \frac{6+6-6}{6+6+6} = \frac{6}{18} = \frac{1}{3}$$

Thus, x and y are in total disorder according to our definition.

V. Connecting Edit Distance to Sorting Operations

This section establishes a fundamental connection between the edit distance of sequences and the number of operations required to sort them. We focus on the relationship between edit distance and bubble sort, chosen for its well-understood properties and straightforward analysis.

Definition 7 (Bubble Sort Swaps)**.** *For a sequence $x \in \Sigma^*$ over a totally ordered alphabet Σ, let swaps(x) denote the number of swap operations performed by the bubble sort algorithm to sort x into ascending order.*

Theorem 5 (Edit Distance-Swap Inequality)**.** *Let $x, y \in \Sigma^n$ be two sequences of length n over a totally ordered alphabet Σ, where y is the sorted version of x in ascending order. Then:*

$$d_E(x,y) \leq 2 \cdot \mathit{swaps}(x)$$

Proof. We proceed by induction on the number of swaps performed by bubble sort.

Base case: If $\text{swaps}(x) = 0$, then x is already sorted, so $x = y$ and $d_E(x,y) = 0$. The inequality holds: $0 \leq 2 \cdot 0$.

Inductive step: Assume the inequality holds for all sequences requiring k or fewer swaps. Consider a sequence x that requires $k+1$ swaps.

Let x' be the sequence after performing the first swap on x, and let y be the fully sorted sequence. We have:

$$\begin{aligned} &\text{swaps}(x) = 1 + \text{swaps}(x') \\ &d_E(x,y) \leq d_E(x,x') + d_E(x',y) \quad \text{(by triangle inequality)} \end{aligned}$$

Observe that $d_E(x,x') \leq 2$, as a swap of adjacent elements can be represented by at most two edit operations (one deletion and one insertion). By the inductive hypothesis:

$$d_E(x',y) \leq 2 \cdot \text{swaps}(x')$$

Therefore:

$$\begin{aligned} d_E(x,y) &\leq 2 + d_E(x',y) \\ &\leq 2 + 2 \cdot \text{swaps}(x') \\ &= 2 + 2 \cdot (\text{swaps}(x) - 1) \\ &= 2 \cdot \text{swaps}(x) \end{aligned}$$

Thus, the inequality holds for $k+1$ swaps, completing the induction. □

Recall the generalized Rozinek similarity formula for sequences of length n:

$$s_n(x,y) = \frac{2n - d(x,y)}{2n + d(x,y)}$$

We established that $d(x,y) \leq 2 \cdot \text{swaps}(x)$. Considering the case of equality:

$$s_n(x,y) = \frac{2n - 2 \cdot \text{swaps}(x)}{2n + 2 \cdot \text{swaps}(x)} = \frac{n - \text{swaps}(x)}{n + \text{swaps}(x)}$$

Corollary 1 (Lower Bound on Swaps for Disordered Sequences)**.** *For sequences $x, y \in \Sigma^n$ where $d_E(x,y) = n$ (i.e., in total disorder), the number of swap operations required to sort x into y (or vice versa) is at least $\frac{n}{2}$.*

Proof. From Theorem 3.1, we have:

$$n = d_E(x,y) \leq 2 \cdot \text{swaps}(x)$$

Solving for $\text{swaps}(x)$:

$$\text{swaps}(x) \geq \frac{n}{2}$$

□

Remark 2. *The lower bound of $\frac{n}{2}$ swaps for sequences in total disorder approaches the worst-case scenario for bubble sort, which is $\frac{n(n-1)}{2}$ swaps for a reverse-ordered sequence.*

Theorem 6 (Relation between Edit Distance and Inversion Count)**.** *Let $x \in \Sigma^n$ be a sequence over a totally ordered alphabet Σ, and let y be its sorted version in ascending order. Then:*

$$d_E(x,y) \leq 2 \cdot \mathit{inv}(x)$$

where inv(x) is the inversion count of x.

Proof. Recall that bubble sort performs exactly one swap for each inversion in the sequence. Therefore, $\text{swaps}(x) = \text{inv}(x)$. Applying Theorem 3.1:

$$d_E(x,y) \leq 2 \cdot \text{swaps}(x) = 2 \cdot \text{inv}(x)$$

□

VI. Infimum of Similarity in Posets with Total Disorder

We begin by formalizing the concept of a balance constant in the context of a normalized similarity function on a partially ordered set (poset).

Definition 8 (Balance Constant in Normalized Similarity Space)**.** *Let $(\mathcal{P}, \leq)$ be a finite poset and $s_n : \mathcal{P} \times \mathcal{P} \to [0, 1]$ a normalized similarity function on $\mathcal{P}$. The* balance constant *$d(\mathcal{P})$ is defined as:*

$$d(\mathcal{P}) = \max_{(x,y)\in\mathcal{P}^2, x\neq y} s_n(x \prec y, y \prec x) \tag{13}$$

where $s_n(x \prec y, y \prec x)$ represents the normalized similarity between the linear orders $x \prec y$ and $y \prec x$ in $\mathcal{P}$.

To establish our main result, we first need to prove a lemma relating the unnormalized similarity and dissimilarity in posets.

Lemma 3 (Characterization of Total Disorder in Posets)**.** *Let $(\mathcal{P}, \leq)$ be a finite poset and $(\mathcal{P}, \leq^{-1})$ be its dual (reverse) poset. $\mathcal{P}$ is in a state of total disorder if and only if:*

$$d_E(\mathcal{P}, \mathcal{P}^{-1}) = \frac{1}{2}e(\mathcal{P}) \tag{14}$$

where $d_E(\mathcal{P}, \mathcal{P}^{-1})$ is the edit distance between $\mathcal{P}$ and its reverse, defined as the number of pairs (x, y) where $x < y$ in $\mathcal{P}$ but $y <^{-1} x$ in $\mathcal{P}^{-1}$, and $e(\mathcal{P})$ is the total number of linear extensions of $\mathcal{P}$.

Proof. ($\Rightarrow$) Assume $\mathcal{P}$ is in a state of total disorder. By definition, this means that for any pair of distinct elements $x, y \in \mathcal{P}$:

$$\mathbb{P}(x \prec y) = \mathbb{P}(y \prec x) = \frac{1}{2} \tag{15}$$

This implies:

$$e(x \prec y; \mathcal{P}) = e(y \prec x; \mathcal{P}) = \frac{1}{2}e(\mathcal{P}) \tag{16}$$

Now, consider the dual poset $\mathcal{P}^{-1}$. For any pair (x, y) where $x < y$ in $\mathcal{P}$, we have $y <^{-1} x$ in $\mathcal{P}^{-1}$. The number of such pairs is exactly $d_E(\mathcal{P}, \mathcal{P}^{-1})$. Given the total disorder condition, this number must be:

$$d_E(\mathcal{P}, \mathcal{P}^{-1}) = e(x \prec y; \mathcal{P}) = e(y \prec x; \mathcal{P}) \tag{17}$$

$$= \frac{|\mathcal{P}|!}{2} = \frac{1}{2}e(\mathcal{P}) \tag{18}$$

where $|\mathcal{P}|$ is the cardinality of $\mathcal{P}$.

($\Leftarrow$) Now assume $d_E(\mathcal{P}, \mathcal{P}^{-1}) = \frac{1}{2}e(\mathcal{P})$. We need to show this implies total disorder.

For any pair (x, y) where $x \neq y$, we have:

$$e(x \prec y; \mathcal{P}) = e(y \prec x; \mathcal{P}^{-1}) \tag{19}$$

$$e(y \prec x; \mathcal{P}) = e(x \prec y; \mathcal{P}^{-1}) \tag{20}$$

We also know that:

$$e(x \prec y; \mathcal{P}) + e(y \prec x; \mathcal{P}) = e(\mathcal{P}) \tag{21}$$

$$e(y \prec x; \mathcal{P}^{-1}) + e(x \prec y; \mathcal{P}^{-1}) = e(\mathcal{P}^{-1}) = e(\mathcal{P}) \tag{22}$$

From (19), (20), (21), and (22), we can conclude:

$$e(x \prec y; \mathcal{P}) = e(y \prec x; \mathcal{P}) = \frac{1}{2}e(\mathcal{P}) \tag{23}$$

This holds for all pairs (x, y) where $x \neq y$, which is the definition of total disorder. □

Now we can state and prove our main theorem.

Theorem 7 (Infimum of Similarity in Posets)**.** *Let $(\mathcal{P}, s_n)$ be a normalized similarity space on a finite poset $\mathcal{P}$. Then, for any pair of distinct elements $x, y \in \mathcal{P}$:*

$$\inf_{(x,y)\in\mathcal{P}^2, x\neq y} s_n(x \prec y, y \prec x) = \frac{1}{3} \tag{24}$$

This infimum is achieved when $\mathcal{P}$ is in a state of total disorder.

Proof. To establish this result, we will derive the infimum of the normalized similarity using the Generalized Rozinek Similarity formula and analyze the case when the poset is in total disorder.

Let us begin by considering the normalized similarity $s_n(x \prec y, y \prec x)$ for distinct elements $x, y \in \mathcal{P}$. By definition, this is given by:

$$s_n(x \prec y, y \prec x) \tag{25}$$

$$= \frac{s_u(x \prec y, x \prec y) + s_u(y \prec x, y \prec x) - d_u(x \prec y, y \prec x)}{s_u(x \prec y, x \prec y) + s_u(y \prec x, y \prec x) + d_u(x \prec y, y \prec x)} \tag{26}$$

In the context of posets, we know that the sum of unnormalized similarities $s_u(x \prec y, x \prec y) + s_u(y \prec x, y \prec x)$ equals the total number of linear extensions $e(\mathcal{P})$. Moreover, the unnormalized dissimilarity $d_u(x \prec y, y \prec x)$ represents the absolute difference between the number of linear extensions favoring $x \prec y$ and those favoring $y \prec x$. Incorporating these insights, we can rewrite the normalized similarity as:

$$\inf_{(x,y)\in\mathcal{P}^2, x\neq y} s_n(x \prec y, y \prec x) \tag{27}$$

$$= \frac{e(\mathcal{P}) - \sup_{(x,y)\in\mathcal{P}^2, x\neq y} d_u(x \prec y, y \prec x)}{e(\mathcal{P}) + \sup_{(x,y)\in\mathcal{P}^2, x\neq y} d_u(x \prec y, y \prec x)} \tag{28}$$

$$= \frac{e(\mathcal{P}) - \sup_{(x,y)\in\mathcal{P}^2, x\neq y} d_E(x \prec y, y \prec x)}{e(\mathcal{P}) + \sup_{(x,y)\in\mathcal{P}^2, x\neq y} d_E(x \prec y, y \prec x)} \tag{29}$$

To find the infimum of this normalized similarity, we need to consider the case where $d_E(\mathcal{P}, \mathcal{P}^{-1})|$ attains its maximum value. This scenario corresponds to the state of total disorder in the poset.

Recall Lemma 3, which states that a poset $\mathcal{P}$ is in total disorder if and only if the edit distance between $\mathcal{P}$ and its dual $\mathcal{P}^{-1}$ is exactly half the number of linear extensions:

$$d_E(\mathcal{P}, \mathcal{P}^{-1}) = \frac{1}{2}e(\mathcal{P}) \tag{30}$$

This edit distance represents the total number of pairs (x, y) where the order is reversed between $\mathcal{P}$ and $\mathcal{P}^{-1}$. In the case of total disorder, this is equivalent to saying that for each pair (x, y), the difference in the number of linear extensions favoring one order over the other is maximized. Mathematically, this translates to:

$$\sup_{(x,y)\in\mathcal{P}^2, x\neq y} d_E(x \prec y, x \prec y) = d_E(\mathcal{P}, \mathcal{P}^{-1}) = \frac{1}{2}e(\mathcal{P}) \tag{31}$$

Now, we can substitute this maximum value into our expression for the infimum of normalized similarity:

$$\begin{align}\inf_{(x,y)\in\mathcal{P}^2, x\neq y} s_n(x \prec y, y \prec x) &= \frac{e(\mathcal{P}) - \frac{1}{2}e(\mathcal{P})}{e(\mathcal{P}) + \frac{1}{2}e(\mathcal{P})} \tag{32}\\ &= \frac{\frac{1}{2}e(\mathcal{P})}{\frac{3}{2}e(\mathcal{P})} = \frac{1}{3} \tag{33}\end{align}$$

Thus, we have demonstrated that the infimum of the normalized similarity is indeed $\frac{1}{3}$, and this value is achieved precisely when $\mathcal{P}$ is in a state of total disorder, as characterized by the edit distance between $\mathcal{P}$ and its dual $\mathcal{P}^{-1}$. This result not only provides a lower bound for the normalized similarity in posets but also establishes a fundamental connection between the concepts of similarity, edit distance, and total disorder in the theory of partially ordered sets. □

Remark 3. *This result establishes a fundamental connection between the concept of total disorder in posets and the infimum of normalized similarity in similarity spaces. It demonstrates that the theoretical lower bound of $\frac{1}{3}$ for the balance constant in the 1/3-2/3 Conjecture corresponds exactly to the case of maximum disorder in the poset structure.*

VII. Conclusion

This paper has introduced a novel approach to the 1/3-2/3 Conjecture using the framework of similarity spaces. Our key contributions include:

- Formulating the balance constant in terms of normalized similarity space and proving its supremum is 1/2.
- Demonstrating that the infimum of normalized similarity in posets is 1/3, achieved in the case of total disorder.
- Establishing connections between edit distance, swap operations, and Generalized Rozinek Similarity for sequences.

Through this analysis, a proof of the 1/3-2/3 Conjecture has been developed using the similarity space approach. This proof provides a new perspective on the conjecture by reframing it in terms of normalized similarity and leveraging concepts from the theory of similarity spaces. While the proof is believed to be fundamentally sound, a more rigorous and detailed presentation is needed to meet the exacting standards of formal mathematical literature. Future work will focus on refining and expanding the proof, providing more detailed justifications, and exploring additional implications of this approach. This work represents a significant step forward in resolving the longstanding 1/3-2/3 Conjecture and opens new avenues for tackling related problems in combinatorics and order theory.

Acknowledgment

It was supported by the Erasmus+ project: Project number: 2022-1-SK01-KA220-HED-000089149, Project title: Including EVERyone in GREEN Data Analysis (EVERGREEN) funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the Slovak Academic Association for International Cooperation (SAAIC). Neither the European Union nor SAAIC can be held responsible for them.

Co-funded by the European Union

EverGreen DATA Analytics

References

[1] S. Kislitsyn, "A finite partially ordered set and its corresponding set of permutations," *Mathematical notes of the Academy of Sciences of the USSR*, vol. 4, pp. 798–801, 1968.

[2] M. L. Fredman, "How good is the information theory bound in sorting?" *Theoretical Computer Science*, vol. 1, no. 4, pp. 355–361, 1976.

[3] N. Linial, "The information-theoretic bound is good for merging," *SIAM Journal on Computing*, vol. 13, no. 4, pp. 795–801, 1984.

[4] G. Brightwell, "Balanced pairs in partial orders," *Discrete Mathematics*, vol. 201, no. 1-3, pp. 25–52, 1999.

[5] J. Kahn and M. Saks, "Balancing poset extensions," *Order*, vol. 1, pp. 113–126, 1984.

[6] W. Trotter, W. Gehrlein, and P. Fishburn, "Balance theorems for height-2 posets," *Order*, vol. 9, pp. 43–53, 1992.

[7] M. Peczarski, "The gold partition conjecture for 6-thin posets," *Order*, vol. 25, pp. 91–103, 2008.

[8] G. R. Brightwell, "Semiorders and the 1/3–2/3 conjecture," *Order*, vol. 5, pp. 369–380, 1989.

[9] I. Zaguia, "The 1/3-2/3 conjecture for n-free ordered sets," *arXiv preprint arXiv:1107.5626*, 2011.

[10] ——, "The 1/3-2/3 conjecture for ordered sets whose cover graph is a forest," *Order*, vol. 36, pp. 335–347, 2019.

[11] O. Rozinek and J. Mareš, "The duality of similarity and metric spaces," *Applied Sciences*, vol. 11, no. 4, 2021. [Online]. Available: https://www.mdpi.com/2076-3417/11/4/1910

[12] B. Ma and K. Zhang, "The similarity metric and the distance metric," *Proceedings of the 6th Atlantic Symposium on Computational Biology and GenomeInformatics*, p. 1239–1242, 2005.

[13] S. Almazel, Q. H. Ansari, and M. A. Khamsi, *Topics in Fixed Point Theory*. Berlin: Springer-Verlag, 2014.

[14] C. C. H. Elzinga and M. M. Studer, "Normalization of distance and similarity in sequence analysis," *Sequence Analysis and Related Methods (LaCOSA II)*, p. 445, 2016.

[15] E. Alhajjar and C. Lefèvre, "On the similarity metric," *Mathematica Militaris*, vol. 24, no. 1, p. 4, 2019.

[16] S. G. Matthews, "Partial metric spaces," Tech. Rep., 1992.

[17] ——, "Partial metric topology," *Annals of the New York Academy of Sciences*, vol. 728, no. 1, pp. 183–197, 1994.

[18] S. Oltra and O. Valero, "Banach's fixed point theorem for partial metric spaces," *Rend. Istid. Math. Univ. Trieste*, vol. 36, no. 1-2, pp. 17–26, 2004.