Eye-tracker Signals Processing in System Identification of Human Oculomotor Apparatus using Cloud Technologies

VITALIY PAVLENKO, ANDRII ILUTSA, YEVHEN KRAVCHENKO
Department of Computerized Systems and Software Technologies,
Odesa Polytechnic National University,
Odesa, Shevchenko ave., 1,
UKRAINE

Abstract: - The objective of this work is to design and develop a cloud-based web platform based on a new proposed concept of cloud computing organization, that extends the diagnostic capabilities of model-based information technology for assessing neurophysiological states using eye tracking data. Platform is intended to assess human neuro-physiological states using nonlinear dynamic methods for identifying the oculomotor system using eye tracking data. The proposed solution integrates advanced signal processing techniques and combines PaaS and SaaS services, which not only optimizes signal processing workflows but also improves the productivity and efficiency of scientific research. The developed web platform provides integration between eye tracking hardware and server-side architecture, which make possible real-time data collection and processing. The server safely processes large data sets generated by the eye tracking device, which are transmitted for further signal processing and analysis. The main feature of the platform is its ability to process large volumes of neurophysiological data with minimal hardware requirements on the client side, which is made possible by the use of cloud computing technologies. The modular structure allows the platform to be easily scaled to solve signal processing tasks, and also provides secure and isolated execution of scripts in a cloud environment. Compared to other similar services, the platform offers several advantages: it supports efficient work in research and education, supports Python and JavaScript programming languages, and allows the use of software-based signal processing via specially developed GUI interfaces. The inclusion of social features and a high level of abstraction further facilitates collaboration and data sharing, making this platform an innovative tool for research and education.

Key-Words: - signal processing, eye-tracking technology, identification, model-based intelligent information technology, web platform, cloud services, cloud computing, PaaS, SaaS, neuro-physiological research.

Received: April 5, 2024. Revised: October 17, 2024. Accepted: November 19, 2024. Published: December 31, 2024.

1 Introduction

Investigating the link between eye movement functions and the central nervous system, along with assessing an individual's psychological and emotional state, provides valuable insights into brain activity, its potential disorders, and the dynamics of psychophysiological processes. This also sheds light on mechanisms like perception, imagination, decision-making, and personal motivations. Eye-tracking technology has become a critical tool in various fields, including diagnosing neurophysiological conditions [1], [2], [3], [4], [5], studying cognitive functions and memory [6] and analyzing student behavior and learning patterns [7]. Such research not only deepens our understanding of human behavior but also bridges the gap between conscious and unconscious mental processes. Knowledge about eye movement has practical applications across numerous industries, allowing for the optimization of workflows and the enhancement of professional efficiency. The growing reliance on eye-tracking hardware in experiments exploring neural activities necessitates the creation of specialized software for signal analysis and management of extensive datasets, [8], [9].

There is an increasing demand for reliable methods to assess mental health at both individual and population levels. Emotion recognition technologies can reveal the state of the nervous system, even in everyday scenarios or high-risk environments, by analyzing variations in eye movement patterns. These insights enable researchers to draw specific conclusions about the psychophysiological state of individuals. To integrate this technology more effectively within academic and research environments, leveraging cloud computing services has proven to be a practical solution.

In recent years, the high-tech innovation of eye-tracking has undergone further development and effective application in constructing a mathematical model of the human oculomotor system (OMS) to detect anomalies in eye movement tracking data for the quantitative assessment of a person's psychophysiological state, [5]. This process involves the use of nonlinear dynamic Volterra models, [5], [10], [11]. The identification of the OMS is based on the use of deterministic test visual stimuli, [5].

Cloud services are now central to digital transformation efforts, providing on-demand access to essential computing resources such as servers, databases, storage, software, and analytical tools via the Internet, [12], [13], [14], [15]. These resources can be accessed through various service models, which in turn offer users flexibility, cost savings, and optimized management of IT resources.

In general, cloud computing services can be divided into several models, each of which meets specific requirements and is more suitable for solving certain problems, [16], [17], [18]: software as a service (SaaS), platform as a service (PaaS), infrastructure as a service (IaaS), and function as a service (FaaS).

Cloud computing is one of the fastest growing technologies today, [19], [20] [21] having a transformative impact on a number of areas where it integrates various IT resources, allowing the sharing of servers, storage, networks, while providing dynamic scalability, cost efficiency, and optimization of many processes. They provide many important benefits such as scalability, flexibility for scientific research, and cost savings for businesses due to reduced IT infrastructure deployment costs and improved data management capabilities. Despite persistent challenges such as data privacy and security risks, [22] [23] [24] [25] [26] cloud computing is rapidly evolving and integrating with other advanced technologies, opening up new opportunities for innovation. Continued exploration of this area will be essential to fully realize its potential. As a result, cloud computing has become an important and necessary part of modern research and business, increasing efficiency and opening up new avenues for advancement in various fields.

The main solutions for cloud-based workflow are Jupyter [27], and Google Colab [28]. Jupyter and Google Colab are more interactive notebooks than platforms for conducting experiments and working with results. They perform well as editors and execution environments but do not offer comprehensive functionality for working with identification methods.

All these competing solutions lack a social component, which prevents effective project sharing and collaboration on other users' projects. They perform their narrowly focused tasks well but do not allow for comprehensive project work, result sharing, and the use of other projects/methods (referring to the ease of finding these methods and interaction) to improve research results.

Besides that, all the mentioned analogs work exclusively with a single programming language and do not provide the option to choose other languages that might be necessary when working with different identification methods, etc.

We see a need for a cloud computing solution with a higher level of abstraction, support for multiple programming languages, and the unification of multiple cloud computing concepts at the same time. This would streamline the research process and provide new opportunities for its users.

This highlights the need for a new concept to facilitate cloud computing in neurophysiological research based on eye-tracking data. A specialized platform that combines PaaS and SaaS capabilities could address this gap, allowing researchers to work with diverse programming languages such as Python and JavaScript, refine algorithms, and utilize existing GUI-based identification tools. Furthermore, given the massive volume and interconnected nature of data in these studies, the inclusion of a collaborative social component would enable data sharing between researchers, fostering scientific progress and addressing limitations in existing solutions.

2 Purpose of the Work

The goal of this study is to design the architecture and create a web-based software platform grounded in a new concept of cloud computing organization. This approach enhances the diagnostic potential of model-based information technology tools for evaluating human neurophysiological states by employing nonlinear dynamic identification methods for analyzing the oculomotor system through eye-tracking data.

3 Volterra Models and OMS Identification based on Eye-Tracking Data

The foundation for creating a mathematical (informational) model of the oculomotor system

(OMS) is based on measurements of its input and output variables. Solving the identification problem involves obtaining experimental data and processing it while accounting for measurement noise.

For describing objects with an unknown structure, it is advisable to use the most universal nonparametric nonlinear dynamic models—Volterra models, [10]. These models provide an unambiguous description of the nonlinear and dynamic properties of the studied object through a sequence of multi-dimensional weight functions—Volterra kernels—that remain invariant to the type of input signal.

The "input-output" relationship for a nonlinear dynamic system with an unknown structure (a "black box" type system) with a single input and a single output can be represented by a discrete Volterra series in the form [11]:

$$y[m]=\sum_{n=1}^{\infty}y_n[m]=\sum_{k_1=0}^{m}w_1[k_1]x[m-k_1]+ \\ +\sum_{k_1,k_2=0}^{m}w_2[k_1,k_2]x[m-k_1]x[m-k_2]+ \\ +\sum_{k_1,k_2,k_3=0}^{m}w_3[k_1,k_2,k_3]x[m-k_1]x[m-k_2]x[m-k_3]+..., \quad (1)$$

where $w_1[k_1]$, $w_2[k_1,k_2]$, $w_3[k_1,k_2,k_3]$ – the discrete weighting functions (Volterra kernels) of the first, second, and third orders, respectively; $x[m]$, $y[m]$ – the system's input (stimulus) and output (response) functions (signals), respectively; $y_n[m]$ – are the partial response components (convolutions of order n); m – the discrete-time variable.

The identification problem consists of selecting test inputs $x[m]$ and developing an algorithm that allows extracting the partial response components $y_n[m]$, (n = 1, 2, 3, …), from the measured responses $y[m]$ and determining the corresponding kernels $w_1[k_1]$, $w_2[k_1,k_2]$, $w_3[k_1,k_2,k_3]$, etc. [10], [11]. In practical modeling, the expansion is typically limited to the first three orders of the series (1).

Considering the specific nature of the studied system, identification is performed using multi-step test signals [5], [10], with different amplitudes a_j (j=1,2,…,L; L – is the number of experiments, L>=N) $x_i(t)=a_i\theta(t)$, $\theta(t)$ – is the Heaviside step function. The measured OMS responses are denoted as $y_1[m]$, $y_2[m]$,…, $y_L[m]$. If the partial response components of the model $\hat{y}_1[m], \hat{y}_2[m], \hat{y}_3[m]$ are determined, this will lead to an estimation of the first-order transition functions $\hat{h}_1[m]$ and the diagonal cross-sections of the transition functions of order n (n=2,3) $\hat{h}_2[m,m]$, $\hat{h}_3[m,m,m]$:

$$y_1[m]=\sum_{k_1=0}^{m}w_1[m-k_1], \\ y_2[m]=\sum_{k_1,k_2=0}^{m}w_2[m-k_1,m-k_2], \quad (2) \\ y_3[m]=\sum_{k_1,k_1,k_3=0}^{m}w_3[m-k_1,m-k_2,m-k_3].$$

At the same time $y_1[m]=h_1[m]$, $y_2[m]=h_2[m,m]$, $y_3[m]=h_3[m,m,m]$.

The responses of the Volterra polynomial model is equal to:

$$\tilde{y}_i[m]=a_iy_1[m]+a_i^2y_2[m]+a_i^3y_3[m], \; i=1,2,\ldots,L. \quad (3)$$

To determine the transition functions $h_1[m], h_2[m,m], h_3[m,m,m]$ to determine the transition functions, used the least squares method (LSM), which provides the minimum mean square error of the deviation of the model responses from the OMS responses to the same stimulus.

To enhance the noise immunity of the deterministic identification method, a noise suppression (smoothing) procedure is applied to the obtained estimates of multidimensional transition functions, based on wavelet filtering [10]. Noise suppression is typically achieved by removing high-frequency components from the signal spectrum, which represents an additive mixture of the informational component—obtained from response processing—and noise caused by measurement equipment errors and the object's own imperfections.

In the context of wavelet decompositions, this can be implemented by directly removing the detail coefficients at high-frequency levels. By setting a certain threshold for these coefficients and truncating them accordingly, it is possible to reduce the noise level effectively.

For the widespread application of nonlinear dynamic identification of the oculomotor system (OMS) in various fields, it is advisable to implement identification software on a web-oriented cloud computing platform based on the novel cloud computing concept proposed here.

4 Concept of Cloud Computing Organization

A new concept for organizing cloud computing has been introduced, which integrates two modes of platform-user interaction: PaaS and SaaS. This innovative approach facilitates cloud computing by providing a built-in interface and code editor, enabling simultaneous research at both the code level and the interface level, with the ability to configure research parameters directly.

This concept supports both research and educational activities, allowing users to modify projects at the code level within a unified software platform. Streamlining workflows, it optimizes information exchange and simplifies the sharing of research outcomes.

5 Software Architecture

A concept, architecture, and web-based platform have been developed to provide an interface and functionality for cloud computing, operating on two main principles: Platform as a Service (PaaS) and Software as a Service (SaaS).

The platform facilitates research by automating the identification of nonlinear dynamic systems and incorporating new methods via a built-in code editor (PaaS functionality). It supports editing and running identification method scripts (provided the code adheres to platform documentation), adding experimental parameters to the scripts, executing calculations on server side, and collecting their results on the server-side and showing them on the client-side browser.

Additionally, the platform offers tools for specific stages of specialized preparation of experimental data from the oculomotor system (OMS), acquired through eye-tracking technology, for subsequent processing in nonlinear dynamic identification procedures (SaaS functionality).

The data preparation process involves the following stages:

Stage 1: Signal data cleaning

- Reading data provided by the eye tracker;
- Identifying and isolating individual research cycles from the data stream for each respondent;
- Removing artifacts;
- Extracting numerical arrays of responses to test visual stimuli;
- Isolating fragments—arrays of responses to individual stimuli;
- Removing fixations of various durations in the transient process.

Stage 2: Signal data preprocessing

- Transforming the sequence of fragments into a parallel structure and aligning the responses within the research cycle to a common starting point;
- Normalizing the data.

Stage 3: Identification procedure

- Conducting the identification procedure of the OMS based on the signal data obtained after preprocessing.

The developed software platform also enables researchers to upload study data in the form of pre-generated tables provided by the eye-tracker. This functionality streamlines the data import process, making signal processing more efficient and minimizing the need for manual data preparation.

The platform comprises multiple interconnected modules and nodes that work together seamlessly. Below is a schematic diagram illustrating the platform's structure (Figure 1, Appendix).

Administrator Panel of the Program. A user interface designed for core platform management tasks, including:

- managing user accounts;
- overseeing general service information.

This component integrates with the Firebase API to manage data storage and retrieval, ensuring that changes are saved efficiently and relevant information is displayed seamlessly.

User Interface (Main Program Interface). The platform’s primary interface enables users to interact with its features, offering functionalities such as:

- Adding new identification methods;
- Editing existing methods based on assigned access and editing permissions;
- Updating account information;
- Adding collaborators and sharing identification methods or experiment results.

It also includes a built-in code editor that facilitates efficient development and modification of identification methods. It integrates with the Firebase API to ensure changes are saved and displayed.

Firebase API. This service component processes all client requests, interfacing with the database or database service (Firebase). Its modularity ensures seamless integration with clients on diverse platforms, including Windows, MacOS, Linux, Android, and iOS.

Code Execution Environment. An isolated module that serves as the execution engine for Python and JavaScript code, running scripts for all identification methods. The results of executed scripts are sent to the client side for analysis and can be stored in the database if necessary. Code interpretation is handled by virtual machines for Python and JavaScript.

Firebase Library. A comprehensive library enabling interaction with the Firebase Firestore NoSQL database. It manages:

- User data storage;
- Authentication via Firebase Auth;
- File storage through Firebase Storage.

Firebase Auth ensures user authentication, supporting social media platforms (e.g., Facebook, GitHub, Twitter, Google, Google Play Games) and enabling secure access via email and password. Firebase Storage provides reliable file upload/download capabilities regardless of network quality. It supports storing user-generated content such as images, audio, and video, leveraging Google Cloud Storage for reliability.

System architecture. The software platform is a modular system consisting of:

- Server-side: Handles all computations, core logic, and data processing. Each computational module operates independently, ensuring scalability.
- Client-side: A cross-platform web interface, built as a Single-Page Application (SPA), serving as the user interface and client logic.

From a functional point of view, it consists of four main components:

- Server: Oversees technical processes, including request management, task distribution, and database operations.
- Web Interface: Provides user access to the platform and project management capabilities.
- Agents: Computational units responsible for task execution and code interpretation.
- Database: Stores all system and project-related data.

This modular structure ensures flexibility, scalability, and efficiency, making the platform suitable for a wide range of research and educational activities.

This architecture supports the independent development and scaling of the system's components. This flexibility paves the way for the future integration of mobile applications and other proprietary software for various platforms into the system.

Moreover, since all core logic and data processing are centralized on the server side, an API interface has been implemented. This interface enables external applications to leverage the functionality of the developed platform. As a result, the project can be expanded and scaled further, ensuring adaptability to evolving needs and compatibility with a broader range of technologies.

6 Description of the Interface and System Workflow

The process of interacting with the software system to obtain results involves several key steps, as shown in the diagram (Figure 2, Appendix).

The system's modular structure ensures that each module operates independently, allowing for effective scaling components. This architecture also enables the seamless integration of alternative clients tailored for specific platforms or tasks. Additionally, new server modules can be added without requiring modifications to the existing client-side applications.

On the login page, users can enter their credentials (username and password) to access the system's features. During the initial phase of deployment, user registration will not be available, and all accounts will be created exclusively by the system administrator (super admin). This measure ensures stricter control over the user base during closed testing.

For enhanced convenience, users can also log in using Google authentication. This eliminates the need for manual registration while providing better oversight of student accounts, as corporate accounts, particularly those linked to educational institutions, are often Google-based.

The system’s main page serves as a centralized information hub, providing users with a snapshot of their recent project activities, account details, and a summary of selected users and projects. From here, users can easily navigate to a complete list of selected users and projects through clickable links or buttons.

To view details about a specific user, users can click on their name, which will redirect them to a personalized dashboard displaying data related to the selected account (subject to access permissions).

The user profile page allows users to view and edit personal details, including profile photo (avatar), name, phone number, email, educational institution, and other relevant account information.

The projects page serves as a gateway to both public and personal projects, providing users with tools to manage, create, and explore projects. Each project card on this page displays essential details, including the title, author, last edited date, and a brief overview.

Users can create new projects, either public or private. By default, projects are private and accessible only to their creators but can be made public through the settings.

Users can view and execute scripts of public projects. However, public projects cannot be directly edited. If a user wishes to work with a public project as their own, they can copy it to their profile, granting full editing capabilities for the code, data, and results.

Clicking on a project redirects users to its dedicated page, where they can:

- View the project description, script code, and results of prior research (if available).
- Edit the project code (subject to access rights).
- Change the project's access level between public and private.
- Select the programming language for the script.
- Execute the script code and analyze the results.
- Share the project (code and results) with selected users.

The interface has been designed with a focus on modern UI/UX principles and is fully optimized for mobile devices like smartphones and tablets. Users benefit from an intuitive experience that adapts seamlessly to smaller screens, as illustrated by the mobile-optimized project page (Figure 3, Appendix). To further improve accessibility, the platform offers an alternative dark theme, catering to modern user preferences and enhancing usability for individuals with visual impairments or sensitivity to bright screens. This combination of adaptability and accessibility ensures an inclusive experience for all users.

To ensure transparency and system reliability, a logging service has been implemented. Accessible only to system administrators, the service provides detailed information about script executions, such as:

- Execution start time.
- User initiating the execution.
- Project and task execution IDs.
- Log ID and any errors (with error codes) that occurred during script execution.
- This system helps monitor usage patterns and diagnose potential issues.
- Modern Interface and Accessibility:

This structure and workflow enhance the usability and flexibility of the software platform while maintaining strict control over user access during the testing phase.

The executing results storage system, as shown in the diagram (Figure 4, Appendix), enables efficient organization and preservation of data from previous experiments without the need to rerun computations on the server. This solution ensures convenience, resource optimization, and improved researcher productivity, allowing for quick analysis of past results, verification of the parameters used during computations, and comparison of execution variants.

Each experiment execution is automatically registered in the system along with the corresponding launch parameters. This enables researchers to return to any result at any time for additional analysis or validation. Additionally, the platform allows saving all experiment parameters and their values. These data make much easier to repeat the experiment or change it, if necessary. Also, the system interface allows visualizing the results in the form of graphs, which allows users to easily evaluate the dependence of the parameters and monitor the dynamics of their changes.

The list of saved results is a set of cards, each of which contains a unique identifier, date and time of execution, key calculation parameters and buttons for viewing or deleting the results. Users can quickly acquaint themselves with the details of the experiment or delete the results if necessary. The use of this system provides such advantages as minimizing the time spent on repeated experiments, reducing the load on the server by saving intermediate and final results, and increasing the accuracy of the analysis due to the ability to track parameters and settings for each execution.

The results storage system is an integral part of the modern approach to automating research processes, especially when working with large data volumes and cloud technologies.

7 Isolation of User Code Execution using Docker in Cloud Environments

Ensuring the secure and efficient execution of user-submitted code is a critical challenge in modern cloud computing. This becomes particularly significant in multi-language environments where users expect real-time code execution via web-based platforms. Since user-submitted code is inherently untrusted, addressing concerns related to security, isolation, and resource management is essential. Docker containerization technology, [29] emerges as a robust solution to meet these demands, offering streamlined setup and management of diverse execution environments (Figure 5, Appendix).

Isolation. In multi-tenant cloud environments, strong isolation mechanisms are essential for executing user code. To do this, Docker containerization technology uses Linux kernel features such as namespaces and groups. Namespaces give containers isolated views of system resources such as process trees, file systems, and network interfaces, while groups manage and limit resource usage. This isolation ensures that code running in one container cannot interfere with other containers or the host system. Multiple user scripts can run simultaneously in separate Docker containers without the risk of one script affecting another. If a script behaves unpredictably, its effects are confined to its intended container, protecting the wider system and maintaining reliability for all users.

Security. Execution of untrusted user code poses significant security risks, especially within shared infrastructure. One of the main concerns is the possibility of malicious code accessing sensitive system or user data. Docker containers mitigate this risk by isolating each execution environment, effectively creating near-sandbox instances. Each container runs on its own file system, processes, and networking stack, preventing unauthorized interaction with the host system or other containers. By limiting the capabilities of a container, Docker reduces the attack surface, ensuring that even if an exploit is attempted, any impact remains confined to the isolated environment.

Resource Management. Effective resource management is vital when running user code in a cloud environment. Docker containers allow administrators to precisely control resource allocation, including CPU usage, memory limits, and disk I/O. This ensures that no container monopolizes resources, preventing performance degradation for other users. By setting specific limits on a container’s resource consumption, the overall stability and performance of the system is maintained.

Support for multiple programming languages. One of Docker’s outstanding advantages is its flexibility in supporting multiple programming languages. Containers can be tailored to specific languages, each containing the necessary dependencies, libraries, and runtime configurations. This ensures consistent execution of user code across environments. Docker also simplifies the process of introducing new languages. Once a container template for a specific language is created, extending support for additional languages requires minimal effort. This modularity allows the system to quickly adapt to changing user needs while maintaining a consistent interface for code execution.

Docker containerization technology offers significant benefits for the safe and efficient execution of user code in cloud computing environments. By providing robust isolation, granular resource management, and flexible support for multiple programming languages, Docker addresses key challenges associated with running untrusted code. Its ability to streamline the setup and management of multiple runtime environments makes it an ideal solution for modern cloud applications. Together, these features improve the resiliency, scalability, and performance of systems designed to meet the demands of modern cloud computing.

8 Scientific Novelty

A new approach to human neurophysiological diagnostics using intelligent information technology in combination with experimental eye-tracking data was presented for the first time. This approach uses cloud computing technology to improve the efficiency and scalability of scientific research, allowing researchers around the world to join participate in the diagnostic process regardless of their device type or location.

The process of organizing cloud computing was improved by introducing a new concept of organizing cloud computing, that combines the principles of both platform as a service (PaaS) and software as a service (SaaS). This presented concept offers significant advantages over existing platforms by optimizing research and educational workflows, increasing the level of abstraction for users, supporting multiple programming languages, reducing hardware dependency, and ensuring seamless cross-platform compatibility.

9 Conclusions

The developed software platform, based on a new concept of cloud computing, significantly improves the diagnostic capabilities of intelligent information technologies based on models for assessing the neurophysiological state of a person. The platform enables cross-platform scientific research, increasing both productivity and accuracy, due to the integration of non-parametric identification methods for oculomotor system analysis using eye tracking data. This is especially important for signal processing tasks, where advanced algorithms filter and analyze oculomotor response data, resulting in highly accurate and reliable neurophysiological assessments.

The developed platform allows users to seamlessly work with software code in several programming languages, including Python and JavaScript, which greatly simplifies research and educational processes, and facilitates scaling of research tasks. Thanks to intuitive graphical user interfaces (GUI), the developed complex provides easy access to pre-implemented identification methods. In turn, the adaptive interface of the platform ensures ease of use on devices of different sizes, ensuring effective interaction with the system, regardless of whether research is carried out from desktop computers, tablets or smartphones. As a cloud service, the developed platform provides constant functionality regardless of the device used and its computing power.

A key feature of the platform is the implementation of a new proposed concept of cloud computing, which combines such principles as Platform as a Service (PaaS) and Software as a Service (SaaS), as well as its low requirements for computing power of the equipment on the client side. The modular architecture ensures easy scalability, which makes the platform easily adaptable to the expanding needs of users. Compared to such platforms as Jupyter Notebook and Google Colab, the platform offers clear advantages in the form of integrated graphical interfaces for pre-configured research methods, support for several programming languages, and advanced collaboration features. These capabilities increase the level of abstraction and socialization in the research process, opening new opportunities for users.

The platform supports the execution of scripts in Python and JavaScript, providing robust tools for research and development. Its implementation leverages cutting-edge technologies, including JavaScript as the core programming language, HTML and CSS for interface design, and the Vue.js framework, [30]. Python is used for implementing nonlinear dynamic identification methods, while Node.js handles client-server interactions, ensuring smooth communication across the system.

Declaration of Generative AI and AI-assisted Technologies in the Writing Process

During the preparation of this work the authors used ChatGPT in order to improve the readability and language of manuscript. After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

References:

[1] Opwonya J., Doan D.N.T., Kim S.G., Kim J., Ku B., Kim S., Park S., Kim J.U., Saccadic Eye Movement in Mild Cognitive Impairment and Alzheimer's Disease: A Systematic Review and Meta-Analysis, *Neuropsychology Review*, Vol.32, 2022, pp. 193–227. https://doi.org/10.1007/s11065-021-09495-3.

[2] Jansson D., Rosén O., and Medvedev A., Non-parametric analysis of eye-tracking data by anomaly detection, *IEEE Transaction control system technology*, Vol.23, 2015, pp. 1578–1586. https://doi.org/10.23919/ECC.2013.6669561.

[3] Bro V., Medvedev A., Continuous and Discrete Volterra-Laguerre Models with Delay for Modeling of Smooth Pursuit Eye Movements, *IEEE Transactions on Biomedical Engineering*, Vol.70, No.1, 2023, pp. 97–104. https://doi.org/10.1109/TBME.2022.3185669.

[4] Lanata L., Sebastian L., Di Gruttola F., Di Modica S., Scilingo E.P., and Greco1 A., Nonlinear Analysis of Eye-Tracking Information for Motor Imagery Assessments, *Frontiers in Neuroscience*, Vol.13, 2020, pp. 1431. https://doi.org/10.3389/fnins.2019.01431.

[5] Pavlenko V., Milosz M., Dzienkowski M., Identification of the oculo-motor system based on the Volterra model using eye tracking technology, *4th Int. Conf. on Applied Physics, Simulation and Computing (APSAC'2020) 2020, 23-25 May, Rome, Italy. Journal of Physics: Conference Series 1603*, pp. 1–8. https://doi.org/10.1088/1742-6596/1603/1/012011.

[6] Keehn B., Monahan P., Enneking B., Ryan T., Swigonski N., Keehn R.M., Eye-Tracking Biomarkers and Autism Diagnosis in Primary Care, *JAMA Netw Open*, Vol.7, No.5, 2024, pp.1-14. https://doi.org/10.1001/jamanetworkopen.2024.11190.

[7] Weiss K., Kolbe M., Lohmeyer Q., and Meboldt M., Measuring teamwork for training in healthcare using eye tracking and pose estimation, *Front. Psychol. Sec. Organizational Psychology*, Vol.14, 2023, pp. 1–12. https://doi.org/10.3389/fpsyg.2023.1169940.

[8] Pavlenko V., Shamanina T., Chori V., Eyetracking Technology and its Application in Neuroscience, *Proceedings of the 12th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS-2023), 7-9 September, Dortmund, Germany*, Vol. 1, 2023, pp. 187-193. https://doi.org/10.1109/IDAACS58523.2023.10348754.

[9] Pavlenko V.D., Shamanina T.V., Chori V.V., Nonlinear Dynamics Identification of the Oculo-Motor System based on Eye Tracking Data, *International Journal of Circuits, Systems and Signal Processing*, Vol.15, 2021, pp. 569-577. (E-ISSN: 1998-4464). https://doi.org/10.46300/9106.2021.15.63.

[10] Solodusha, S., Kokonova, Y., and Dudareva, O., Integral Models in the Form of Volterra Polynomials and Continued Fractions in the Problem of Identifying Input Signals, *Mathematics*, Vol.11, No.23, 4724, 2023. https://doi.org/10.3390/math11234724.

[11] Doyle, F. J., Pearson, R. K., and Ogunnaike, B. A., *Identification and Control Using Volterra Models*, Communications and Control Engineering, Springer, London, 2001.

[12] Saraswat M., Tripathi R. C., Cloud computing: Analysis of top 5 CSPs in SaaS, PaaS and IaaS platforms, *9th International Conference System Modeling and Advancement in Research Trends (SMART)*, 2020, pp. 300-305. https://doi.org/10.1109/SMART50582.2020.9337157.

[13] Kollipara P., An Overview on Cloud Computing, *International Journal of Research in Engineering, Science and Management*, Vol.4, No.8, 2021, pp. 35-37, [Online]. https://journal.ijresm.com/index.php/ijresm/article/view/1147 (Accessed Date: September 30, 2024).

[14] King T. S., Reviews of Cloud Computing For Education: Services And Benefits, *PEOPLE: International Journal of Social Sciences*, Vol.1, No.1, 2015, pp. 1299-1305. https://doi.org/10.20319/pijss.2015.s21.12991305.

[15] Singh A., Sharma S., Kumar S. R., Yadav S. A., Overview of PaaS and SaaS and its application in cloud computing, *International Conference on Innovation and Challenges in Cyber Security (ICICCS-INBUSH)*, 2016, pp. 172-176. https://doi.org/10.1109/ICICCS.2016.7542322

[16] Mohammed C. M., Zeebaree, S. R., Sufficient comparison among cloud computing services: IaaS, PaaS, and SaaS, *International Journal of Science and Business*, Vol.5, No.2, 2021, pp. 17-30, [Online]. https://ijsab.com/wp-content/uploads/667.pdf (Accessed Date: September 30, 2024).

[17] Wulf F., Lindner T., Strahringer S., Westner M., IaaS, PaaS, or SaaS? The why of cloud computing delivery model selection. Vignettes on the post-adoption of cloud computing, *Proceedings of the 54th Hawaii International Conference on System Sciences (HICSS-54), 4-9 January*, *Lahaina, Hawaii, USA,* 2021, pp. 6285-6294, [Online]. https://aisel.aisnet.org/hicss-54/os/practice-based_research/5/ (Accessed Date: September 30, 2024).

[18] Nadeem F., Evaluating and ranking cloud IaaS, PaaS and SaaS models based on functional and non-functional key performance indicators, *IEEE Access*, Vol.10, 2022, pp. 63245-63257 https://doi.org/10.1109/ACCESS.2022.3182688.

[19] Yoo S. K., Kim B. Y., The effective factors of cloud computing adoption success in organization, *The Journal of Asian Finance, Economics and Business*, Vol.6, No.1, 2019, pp. 217-229. https://doi.org/10.13106/jafeb.2019.vol6.no1.217.

[20] Faridi F., Sarwar H., Ahtisham M., Jamal K., Cloud computing approaches in health care, *Materials Today: Proceedings*, Vol.51, No.1, 2022, pp. 1217-1223. https://doi.org/10.1016/j.matpr.2021.07.210.

[21] Larcher T., Gritsch P., Nastic S., Ristov S., BaaSLess: Backend-as-a-Service (BaaS)-Enabled Workflows in Federated Serverless Infrastructures, *IEEE Transactions on Cloud Computing*, Vol.12, No.4, 2024, pp. 1088-1102. https://doi.org/10.1109/TCC.2024.3439268.

[22] Parast F. K., Sindhav C., Nikam S., Yekta H. I., Kent K. B., Hakak S., Cloud computing security: A survey of service-based models, *Computers & Security*, Vol.114, 2022, pp.

102580. https://doi.org/10.1016/j.cose.2021.102580.

[23] Fatima E., Sumra I. A., Naveed R., A Comprehensive Survey on Security Threats and Challenges in Cloud Computing Models (SaaS, PaaS and IaaS), *Journal of Computing & Biomedical Informatics*, Vol.7, No.1, 2024, pp. 537-544, [Online]. https://jcbi.org/index.php/Main/article/view/403/428 (Accessed Date: September 30, 2024).

[24] Mehboob T., Sumra I. A., Shahzadi I., Detection of DDoS/DoS Attack Methodologies in Cloud Computing Network: A Survey, *Journal of Computing & Biomedical Informatics*, Vol.8, No.1, 2024, pp. 2710-1606, [Online]. https://jcbi.org/index.php/Main/article/view/191/564 (Accessed Date: September 30, 2024).

[25] Arunkumar M., Ashokkumar K., A review on cloud computing security challenges, attacks and its countermeasures, *AIP Conference Proceedings of International Conference on Recent Innovations in Science and Technology (RIST2022), 8–9 July, Malappuram, India*, Vol.3037, No.1, 2024. https://doi.org/10.1063/5.0196063.

[26] Damaraju A., Cloud Security Challenges and Solutions in the Era of Digital Transformation, *International Journal of Advanced Engineering Technologies and Innovations*, Vol.1, No.3, 2024, pp. 387-413, [Online]. https://ijaeti.com/index.php/Journal/article/download/348/359 (Accessed Date: September 30, 2024).

[27] Project Jupyte, [Online]. https://jupyter.org (Accessed Date: September 30, 2024).

[28] Google Colub, [Online]. https://colab.research.google.com (Accessed Date: September 30, 2024).

[29] Docker, [Online]. https://www.docker.com (Accessed Date: September 30, 2024).

[30] Vue.js framework, [Online]. https://vuejs.org (Accessed Date: September 30, 2024).

Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)

The authors equally contributed in the present research, at all stages from the formulation of the problem to the final findings and solution.

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself

No funding was received for conducting this study.

Conflict of Interest

The authors have no conflicts of interest to declare.

APPENDIX

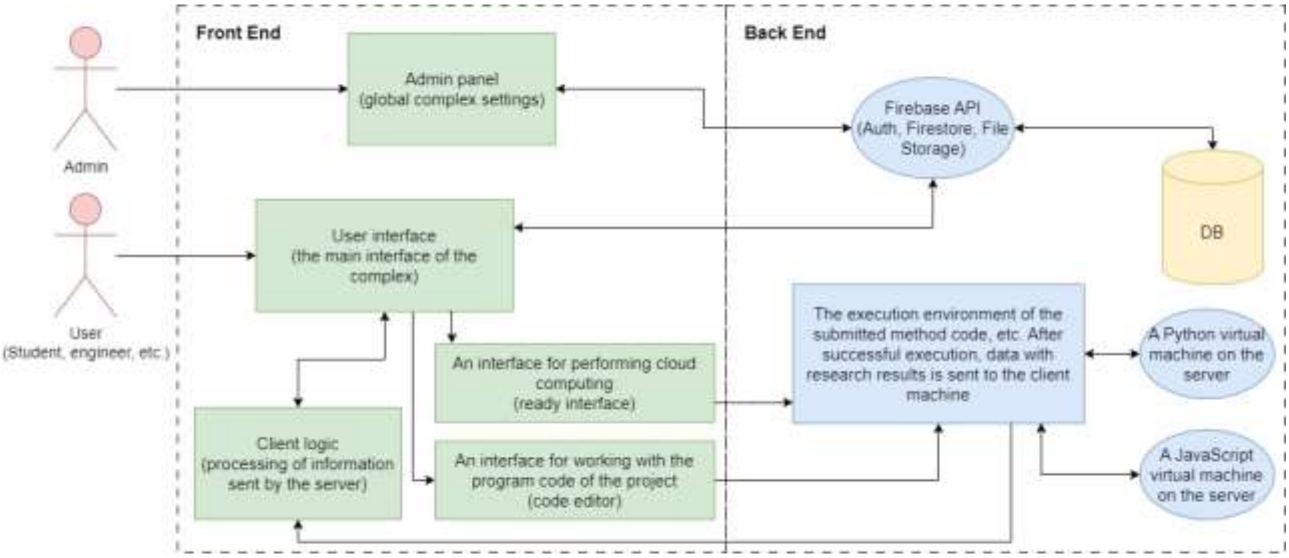

Fig. 1: The structure of the software complex

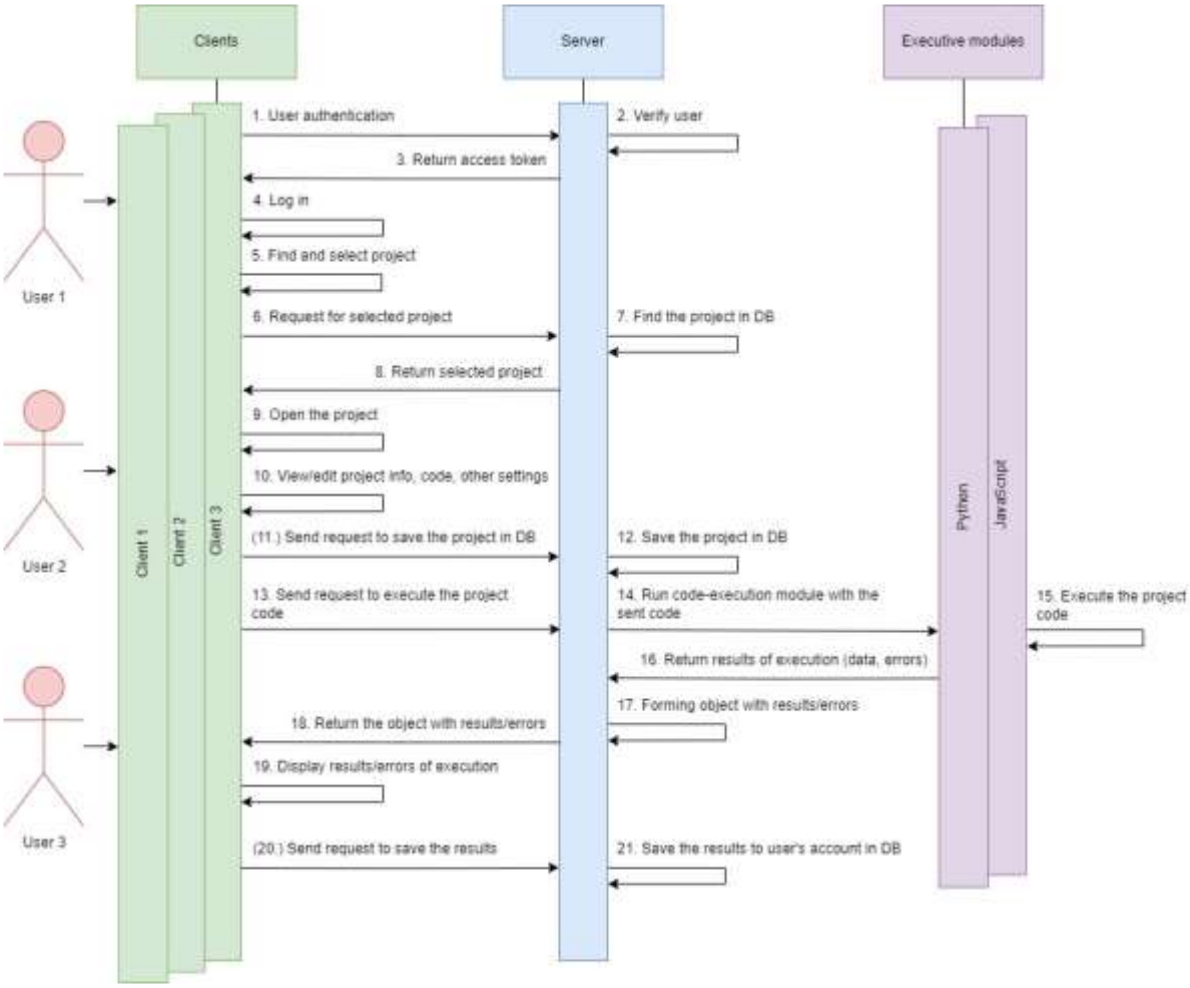

Fig. 2: The process of working with the software complex

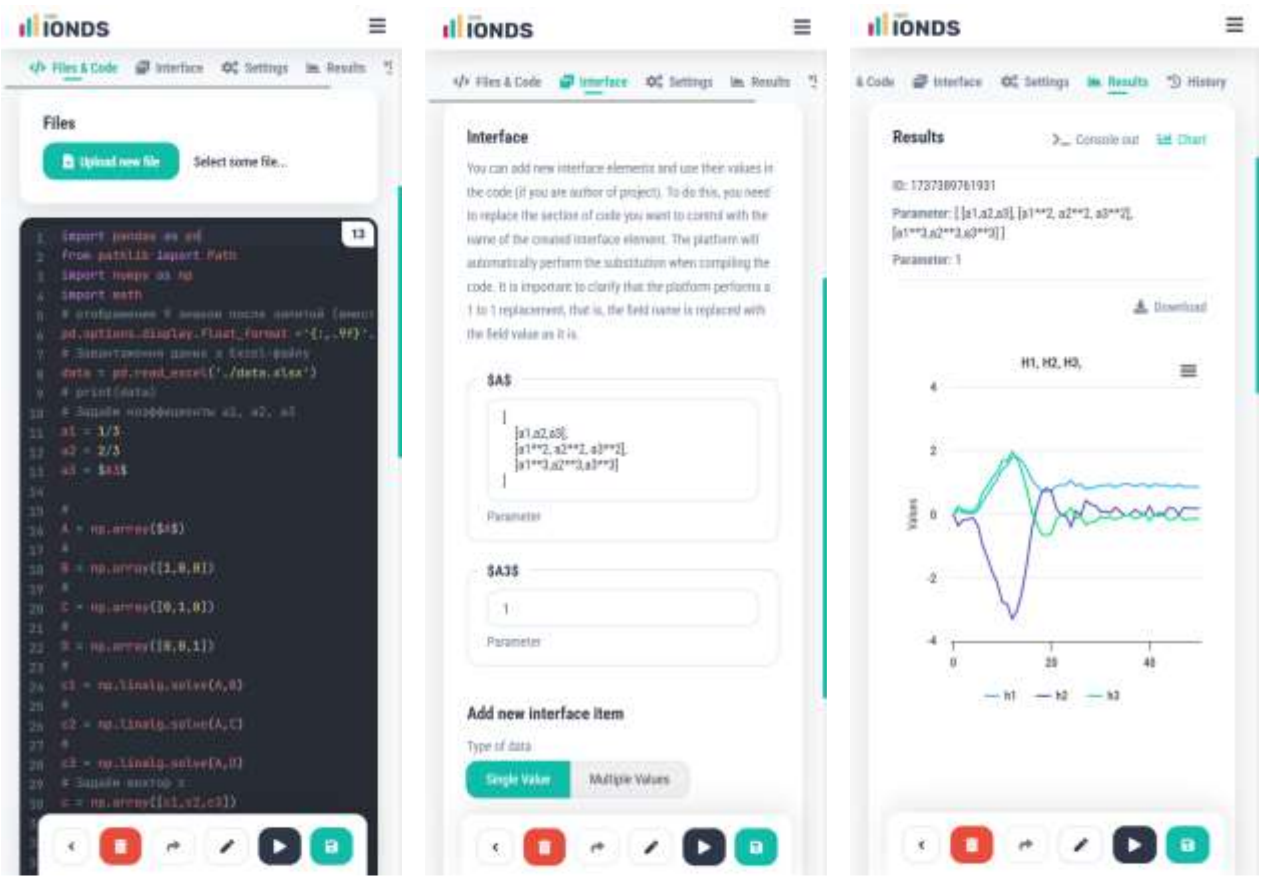

Fig. 3: Components of the project page on a mobile device

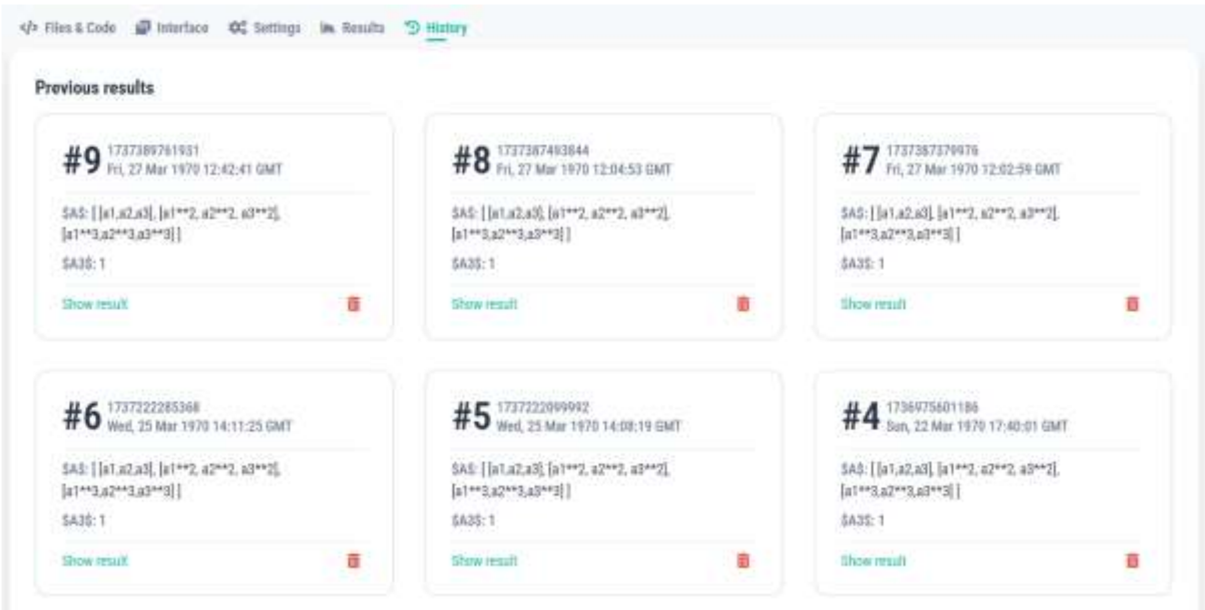

Fig. 4: View of the results storage system

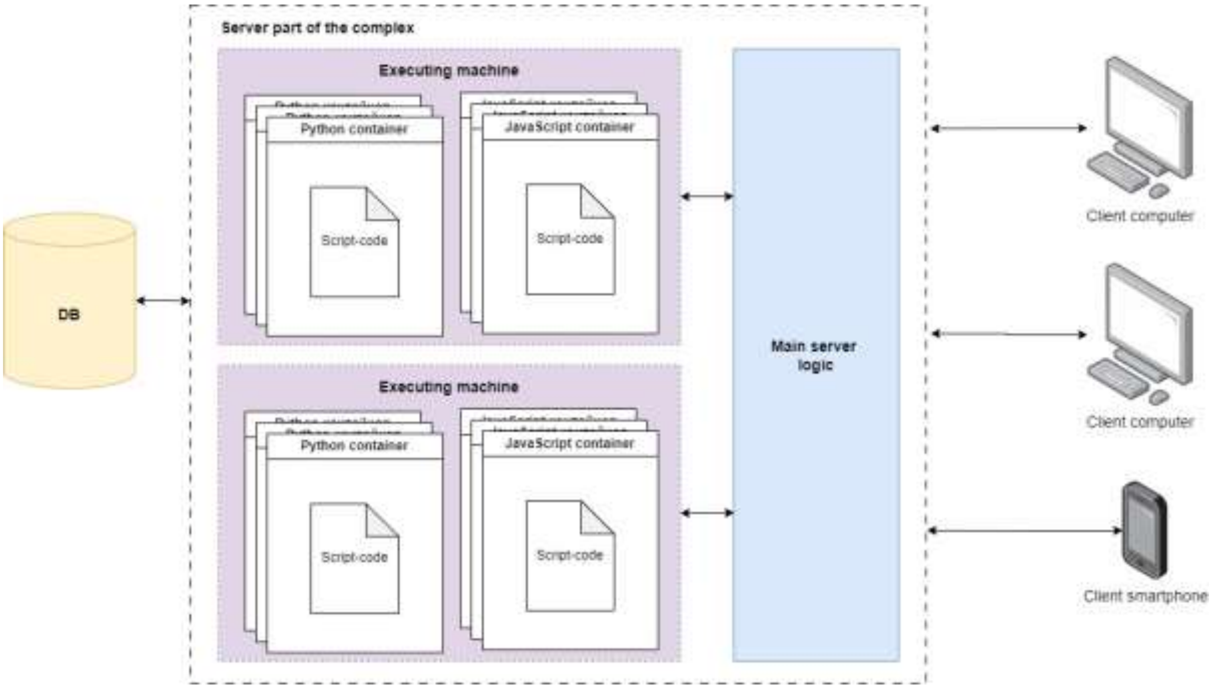

Fig. 5: The principle of isolation of script code execution