УДК 004.9 http://doi.org/10.15407/pp2022.03-04.485

CLOUD SERVICE FOR AUTHENTICATION OF A PERSON BASED ON THEIR ELECTROCARDIOGRAM

Yurii Luhovskyi

Телемедицина з кожним роком пришвидшує свій розвиток. Відбувається це за рахунок винайдення нових технологій, що розвивають комунікацію та роботу з даними. Одним із напрямків, в телемедицині є дистанційний моніторинг, задачею якого є постійно контролювати стан пацієнта. Це здійснюється перевіркою різних показників, в тому числі, йі електрокардіограми. Важливо, щоб надіслана кардіограма належала саме пацієнтові на моніторингу. Для цього розробляється технологія автентифікації по електрокардіограмі. Дослідники з різних країн працюють над цим питанням використовуючи підходи, що можуть сильно відрізнятись.

В цій роботі описується прототип хмарного сервісу по автентифікації. Описуються алгоритми, що використовувались для побудови технології автентифікації та додатків, що реалізують цю технологію. Показуються послідовні розробки додатків та описуються їх задачі. Останній розроблений додаток - це хмарний сервіс, побудований на мікросервісах, що виконує реєстрацію електрокардіограм. Це є найбільш витратна щодо часу операція автентифікації. Приводиться архітектура сервісу та результати експериментів. По результатах помітно, що є проблеми з машинним навчанням, а саме бібліотекою ML.NET. Коли виділяється багато ядер для одного екземпляра мікросервіса, що навчає нейромережі, виникають накладні витрати на розпаралелювання, що сильно впливає на продуктивність. Це підтверджує закон Амдала. За наявної архітектури була знайдена конфігурація, при якій час на проведення експериментів був найменшим. Враховуючи проблему з накладними витратами нова архітектура була запропонована для подальших досліджень.

Розглядається, як розроблена архітектура повинна допомогти в розвитку технології. Ідея системи полягає у використанні одного і того ж сервісу для досліджень і в промисловому використанні. Це дасть гарантію, що промисловий сервіс буде поводитись так само як і при дослідженні.

Ключові слова: електрокардіограма, автентифікація, класифікація, хмарний сервіс

Telemedicine grows faster with each year. In scope of it new technologies have been created to solve information communication problems. One of them is distance monitoring which requires electrocardiogram analysis. In this case it is important to confirm that the transferred electrocardiogram through information channels belongs to the patient. Researchers from different countries work on this problem. They suggest different methods of authentication by electrocardiogram. The goal of this work is to suggest a prototype of a service that could be used to authenticate electrocardiograms.

The paper describes which algorithms have been used to build authentication technology and how it was implemented. There is a short history of built applications. It shows their structures and the purpose. The recently developed system is a prototype to authentication service. It performs registration of new electrocardiograms which is the most time consuming process in authentication. The paper describes the architecture of the system and shows the result of executed experiments. The results show that there is a performance issue with the machine learning library ML.NET. When a lot of cores are allocated to one machine learning instance the overheads highly decrease the overall experiment time. These experiments confirmed Amdahl's law. Nevertheless, an architecture was found where experiments took the least time for execution. Knowing the issue with the machine learning library, a new architecture setup was suggested and will be implemented in future works.

Besides, attention is paid on how developed service should help researchers to improve the technology. The main idea of the system is using one service for developing and testing the technology. That means we can be sure that the cloud service that runs on production would behave the same as in the research phase.

Keywords: electrocardiogram, authentication, classification, cloud infrastructure

Information systems

Introduction. Nowadays telemedicine has become much more popular. After the epidemic of COVID19 the need for distance consultations rapidly grew. That forced creation of new technologies and new equipment in this field. The distance monitoring that is a part of telemedicine is also improving. The part of it is biometric authentication by electrocardiogram (ECG). It is created to prove that sending ECG from a patient to a doctor through information channels really belongs to that person.

Researchers from different countries suggest using authentication by ECG not only in telemedicine but in other systems where access can be provided with ECG. It can be used on its own or as an additional security to other authentication methods. To solve authentication problems researchers use different approaches [1, 2, 3, 4, 5, 6]. Some of them use naked ECG curves and with help of deep learning create neural models which are used to classify other naked ECG curves [2]. Some other approaches use algorithms that can compare two different curves which are used to distinguish ECG curves [3]. And others that process ECG, get features and use them in a couple with machine learning to classify ECGs.

The paper described developed programs that were based on research presented in the articles [7, 8, 9]. There is a patent for this technology [10]. The intention is not to show the efficiency of technology but describe chosen program decisions required to implement this technology. First a desktop program was developed which could register a new ECG and make authentication based on it. When the efficiency of technology was confirmed, a new application was developed. It was able to register a big set of ECGs. After running a couple of experiments it became clear that developed algorithms take too much time to register one ECG. It is a big problem since nobody wants to use this technology on live production. To solve the issue a new information system has been created that is based on microservices and deployed to a cloud. It was shown the revealed issues with the system and how the calculation was improved using existing resources. The developed system became a prototype of cloud service for authentication.

Besides, the developed system saves researchers time usually spent for manual data manipulation or other manual work. It was built in the way to organize work with experiments.

Results. Before looking into programs and information systems it is worth understanding the principles used for authentication by ECG. Input data for authentication is a file with ECG signal. The signal is loaded from the file then processed and presented in three dimensional phase space [3]. Then from the presented cardiac cycles selected one that is the most closest to the average cycle. Using parametric spline found 4 coordinates in three dimensional space that approximate the selected cardiac cycle the most [2]. Founded coordinates used as input for machine learning to learn a model and then to classify ECGs. The efficiency of this approach is described in the article [1].

The authentication approach can be divided logically into two main parts that are presented on Fig 1. The first is ECG processing and creating coordinates for machine learning. The second is learning neural networks and classifying ECGs by coordinates.

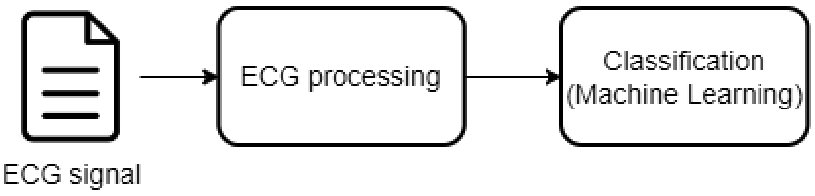

Fig. 1 Simplified schema of authentication technology

Having prommissing result for the technology described in the article [1] a new program has been developed using C# language. It implements the described approach so the functionality allows to register a new ECG and authenticate a loaded one. The goal of this program and other solutions is to authenticate only one registered ECG. The idea is to provide or not provide access to a registered user.

The functionality related to ECG processing was organized and put into a library. The library for machine learning was created as well. It uses open source framework ML.NET under the hood. The library consumes a list of coordinates and provides a learned model or classification results. The library made communication with machine learning much easier since it defines contracts that used to exchange data.

It is worth describing how the algorithm decides if ECG belongs to a registered one or not. The program contains coordinates of 10 fake ECGs that don't belong to anybody. They are all used when registering a new ECG. New ECG requires 10 learned models. The modal is learned based on the pair of coordinates from input ECG and one fake ECG. Then these 10 learned models are used to identify ECG by average classification results.

The next step in research was to repeat the experiments using a bigger amount of input data. For this purpose a new desktop application was created. It also included the ECG processing and machine learning libraries. The application developed to read the amount of input ECG files and save intermediate and output results with learned models into a specific folder in a file system.

This application can be described with the schema presented on Fig. 2. It realizes the idea of running isolated experiments. That means that for each experiment creates a folder with input ECGs. On execution all artifacts are saved to the same experiment folder. That approach allows to have isolated environment for an experiment where saved data related only to this experiment. With this approach it is good to track changes. Having intermediate results that are stored in the folders is very useful since it makes possible to validate a part of the application.

The application starts with a name of an experiment that corresponds to the name of a folder where input ECGs are stored. Then all files are read from a folder (1) and the first file loaded into memory (2). Next step is to process ECG signals and find the coordinates of ECG (3). After that the coordinates from 10 faked persons are loaded (4). Then loaded coordinates and the found one from the input ECG are combined together and saved into a file that machine learning will use. Created file loaded into a folder (5). Then the file with coordinates for machine learning download from the folder and the data are formatted to 10 datasets (6). In the next step one model is learned using prepared data (7). Learning model repeats 10 times. When all models are learned the application takes the next ECG file from the list and executes all that steps. At the end of experiments each ECG data will have corresponding 10 learned models.

Described application performs all calculations sequentially. Running the application on a local machine with 2 cores 2.3 GHz each using 30 ECGs took 12 min 41 sec. On average registration of one ECG took 25 sec. Most of the time spent learning 10 neural network models, it took 18 sec on average. The second costly part was to process ECG and find coordinates, it took 5 sec on average. It takes a lot of time to create a record for one ECG. If there is a need to run a lot of experiments or the amount of ECG data will rapidly increase, most of the time would be spent waiting for results. To decrease time spending for registration of one ECG heavy parts of application should be parallelised.

So the next step after the application that works with a big amount of data was to create an information system that would be more efficient. It was created and built on microservices which deployed into the university cloud. Shortly speaking it is a separated virtual machine on the university server.

The information system uses the same approaches that a desktop application does. The schema on Fig. 3 presents the way how the application was divided into microservices.

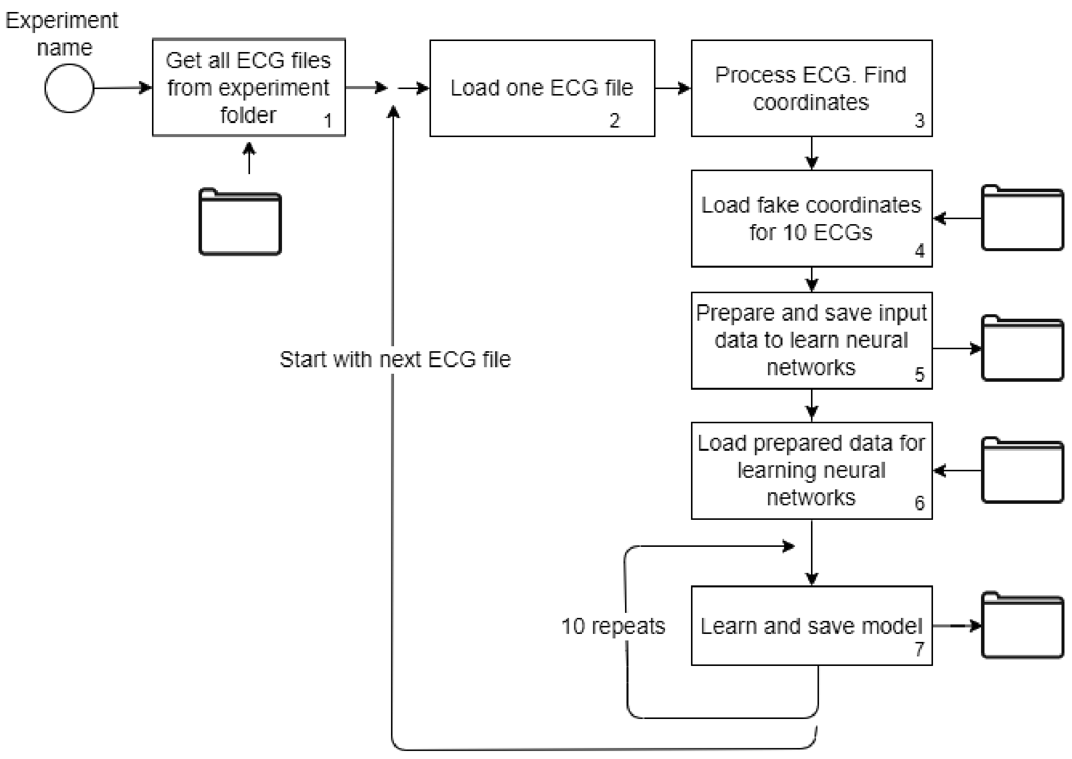
Experiment name
Get all ECG files from experiment folder 1
Load one ECG file 2
Process ECG. Find coordinates 3
Load fake coordinates for 10 ECGs 4
Prepare and save input data to learn neural networks 5
Load prepared data for learning neural networks 6
Learn and save model 7
10 repeats
Start with next ECG file

Fig. 2 Execution workflow for the application

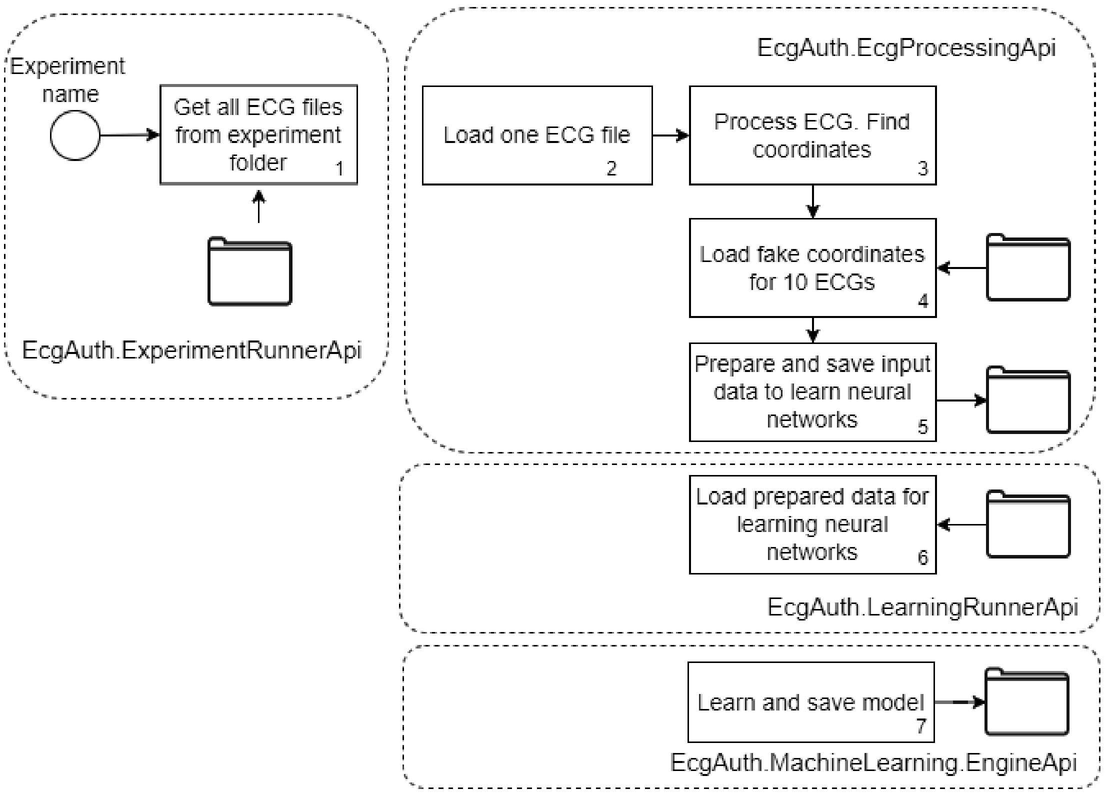
Experiment name
Get all ECG files from experiment folder 1
EcgAuth.ExperimentRunnerApi
EcgAuth.EcgProcessingApi
Load one ECG file 2
Process ECG. Find coordinates 3
Load fake coordinates for 10 ECGs 4
Prepare and save input data to learn neural networks 5
Load prepared data for learning neural networks 6
EcgAuth.LearningRunnerApi
Learn and save model 7
EcgAuth.MachineLearning.EngineApi

Fig. 3 Application breakdown to different microservices

Four microservices were created: ExperimentRunnerApi, EcgProcessingApi, LearningRunnerApi, MachineLearning.EngineApi. The idea of partitioning was to move the time consuming calculations to a separate microservices. The MachineLearning.EngineApi is responsible for learning models and EcgProcessingApi processes ECG and prepares coordinates for machine learning. ExperimentRunnerApi is an orchestrator that starts the process for selected ECG and coordinates microservices. LearningRunnerApi prepares data for machine learning and manages work with neural models.

The communication with ExperimentRunnerApi and between microservices themself happen using HTTP protocol. The information system uses high performance object storage MinIO to store input ECG files and all artifacts. The system saved the same principle of isolated experiment. The schema on Fig. 4 presents the flow of requests when the experiment is executed.

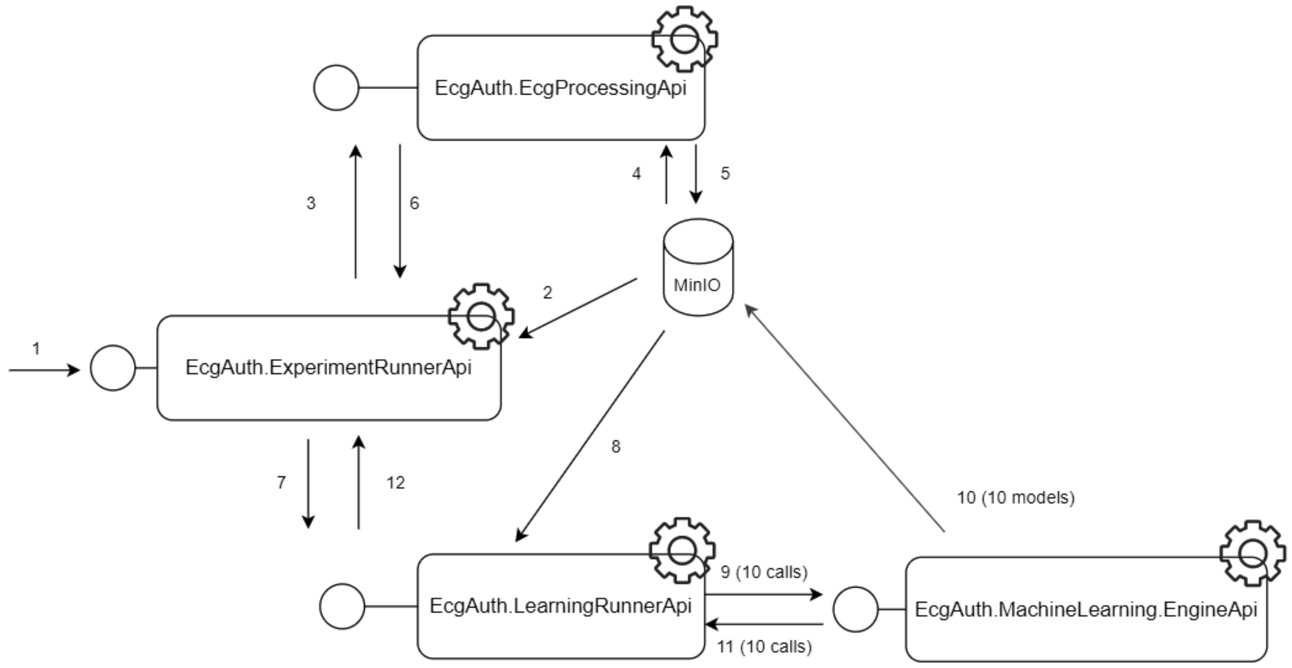
EcgAuth.EcgProcessingApi
EcgAuth.ExperimentRunnerApi
EcgAuth.LearningRunnerApi
EcgAuth.MachineLearning.EngineApi
MinIO
1
2
3
4
5
6
7
8
9 (10 calls)
10 (10 models)
11 (10 calls)
12

Fig. 4 Requests workflow between microservices and storage

The workflow is the following. Numbers present a sequence of requests or responses.

1) Start the experiment. The request contains the name of the experiment.

2) Read all file names with ECGs

3) Send request to process ECG and find coordinates

4) Download ECG file content

5) Save prepared data for machine learning model

6) Success response that data for machine learning model is created

7) Send request to learn models for previously selected ECG

8) Read prepared data for learning

9) Send request to learn model for one pair of data that consists from input ECG coordinates dataset and one fake ECG coordinates dataset.

10) Save learned model.

11) Success response that model has been learned

12) Success response that models for input ECG were learned

Microservices have the ability to process a couple of requests simultaneously so it should help to reduce the overall execution time. Microservices deployed to a virtual machine which consumes a dedicated number of server cores. A microservice has access to all virtual machine cores. Configuration and deployment of microservices are performed with Kubernetes. Each microservice presented in one instance and there was no horizontal scaling for loaded instances.

All communications between microservices and storage were sequential by default. Only the machine learning instance performs learning neural models with parallel computations since ML.NET does it by default. The machine learning instance will use all allocated cores to the virtual machine. To reduce total experiment time the requests to learn neural models were sent in the way to learn the specific amount of models simultaneously. The idea was to find the best configuration for a system that took minimum time for the experiment.

The experiments included registration of 30 ECGs. Table 1 presented the time results of experiments and related configurations to them.

Table 1. Experiment results

Allocated cores	Simultaneously learned models	Total experiment time for 30 ECGs	Average time to learn one model
2	1	12 min 8 sec	2.1 sec
2	2	10 min 31 sec	3.2 sec
2	3	9 min 2 sec	4 sec
2	4	8 min 59 sec	5.2 sec
2	5	8 min 47 sec	6.4 sec
4	1	12 min 31 sec	2.2 sec
4	2	8 min 15 sec	2.7 sec
4	3	7 min 51 sec	3.5 sec
4	4	7 min 19 sec	4.1 sec
4	5	6 min 59 sec	5 sec
6	1	15 min 28 sec	2.8 sec
6	2	9 min 39 sec	3.2 sec
6	3	8 min 52 sec	3.9 sec
6	4	8 min 48 sec	4.7 sec
8	1	20 min 3 sec	3.7 sec
8	2	12 min 18 sec	4.2 sec
8	3	10 min 29 sec	4.7 sec
8	4	9 min 33 sec	5.8 sec

Learning 5 and more neural models simultaneously didn't give valuable improvements so the experiments contained maximum 5 simultaneously learned models.

Analyzing the results we can see a weird pattern when with an increasing number of allocated cores the execution time increased. There is Amdahl's law that could be helpful here [11]. It says that performance for the application that has some parallelizable part of the program would be increasing with adding new cores to some specific amount of cores. This happen because execution time for all parallel work can't be less than longest sequential work that parralized. But in these experiments we see that efficiency dropping down with adding new cores. This is very visible in experiments when only one model is learning simultaneously. Amdahl's law also mentions this issue and says that it is related to overheads. All parallel work contains some sequential part which performs parallelization. So sometimes with adding a lot of cores a lot of sequential work is also added which can decrease the performance. Comparing experiment time when experiment runs on 2 and 8 cores where only one neural model learned simultaneously we see that efficiency dropped for 8 minutes. This is the issue with learning neural models since the average time to learn one increased by 1.6 seconds. Because learning neural models took 86% of all experiment time the parallelization delays influence a lot. The ML.NET didn't work efficiently in this case.

Better performance was achieved when learning a couple of models simultaneously. For this architecture where presented only one instance of microservice the best configuration was allocation 4 cores to virtual machine and learning 5 models simultaneously.

Taking into account the drawback of ML.NET, the better architecture should have up to 10 instances for machine learning microservice where each instance has its own core. This helps to mitigate the issue with overheads.

Future research. The next step in research is to deploy and test architecture with 10 instances created for machine learning microservice. Then configure horizontal scaling for all microservices and parallelize others.

Also expand functionality for the system by making it possible to measure the efficiency of authentication on a big set of data. It will help to track all changes in the technology and see how new changes influence the authentication result. It will help to improve technology iteratively.

Conclusions. While testing cloud authentication service revealed a problem with decreasing performance when increasing the number of allocated cores to the virtual machine. The issue was in learning neural models and it is easy to see on experiments results. The explanation to them is found in Amdahl's law. The performance dropped since learning models happened with overheads. It highly impacted overall experiment time since learning models take 86% of time for one ECG registering process. This forces the creation of another architecture where each instance of machine learning microservices has its own core.

Using this authentication service in research and testing provides us confidence that the service on live production will behave the same as on research.

References

1. Yogendra N. S., Gupta P. Biometrics Method for Human Identification Using Electrocardiogram. Advances in Biometrics: Third International Conference, ICB 2009. – Italy, 2019. – P. 1270-1279.
2. Pei-Lun Hong, Jyun-Ya Hsiao, Chi-Hsun Chung, Yao-Min Feng, and Shun-Chi Wu. Ecg biometric recognition: template-free approaches based on deep learning. In 2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pages 2633– 2636. IEEE, 2019.
3. Sairul I Safie, John J Soraghan, and Lykourgos Petropoulakis. Electrocardiogram (ECG) Biometric Authentication Using Pulse Active Ratio (PAR). IEEE Transactions on Information Forensics and Security. – 2011. – Vol. 6. – №4. – P. 1315 – 1322.
4. Adrian D. C. Chan, Mohyeldin M. Hamdy, Armin Badre, Vesal Badee Wavelet Distance Measure for Person Identification Using Electrocardiograms. IEEE Transactions on Instrumentation and Measurement. – 2008. – Vol.57. – P. 248-253.
5. Fainzilberg L.S., Potapova T.P. Computer Analysis and Recognition of Cognitive Phase Space Electrocardiographic Image. Proceeding of 6th International Conference on Computer analysis of Images and Patterns (CAIP-95). – Prague, 1995. – P. 668-673.
6. Shih-Chin Fang, Hsiao-Lung Chan. QRS detection-free electrocardiogram biometrics in the reconstructed phase space. Pattern Recognition Letters. – 2013. – Vol. 34. – P. 595-602.
7. Вишневский В.В., Романенко Т.Н., Кизуб Л.А. Биометрическая идентификация человека по его электрокардиограмме. Математичні машини і системи. 2018. № 2. С. 88–95.
8. Вишневский В.В., Калмыков В.Г., Романенко Т.Н. Аппроксимация одно-, дву- и трехмерных дуг кривых параметрическими сплайнами. Математичні машини і системи. 2015. № 4. С. 57–64.
9. Вишневський В.В., Романенко Т.М. Застосування метрики Хаусдорфа для визначення нетипових кардіоциклів у тривимірному фазовому просторі координат вектор-кардіограми. Медична інформатика і інженерія. 2019. № 3. С. 31–36.
10. Вишневський В.В. Спосіб автоматичної автентифікації людини за її електрокардіограмою: пат. України на винахід № 117713; заявл. 15.02.17; опубл. 10.09.18, Бюл. № 17. 3.66 с.
11. Gene Amdahl. Validity of the single processor approach to achieving large scale computing capabilities, reprinted from the afips conference proceedings, vol. 30 (atlantic city, n.j., apr. 18–20). Solid-State Circuits Newsletter, IEEE , 12:19 – 20, 02 2007.

Received 02.08.2022

About the author:

Luhovskyi Yurii Olexandrovich
postgraduate student,
Ukraine, Kyiv, Balzaka street 58, flat 43;
publications in domestic magazines - 3,
h-index 0,
https://orcid.org/0000-0002-0195-2770

Place of work:

The Institute Of Mathematical Machines and Systems Problems
National Academy Of Science Of Ukraine (IMMSP NASU).
42 Academician Glushkov Avenue,
Kyiv 03187, Ukraine. (044) 526-24-97

Прізвища та ініціали авторів і назва доповіді українською мовою:
Луговський Ю. О.
Хмарний сервіс ідентифікації людини за її електрокардіограмою

Прізвища та ініціали авторів і назва доповіді англійською мовою:
Yurii Luhovskyi
Cloud service for authentication of a person based on their electrocardiogram