ECG analysis software library based on NLP and ML methods

Yurii Oliinyk, Mykhailo Yazenok, Oleksandr Ocheretianyi, Igor Baklan, Kateryna Lishchuk, Elisa Beraudo

National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", 37, Prosp. Peremohy, Kyiv, 03056, Ukraine

Abstract

This article discusses the implementation of a software library for the analysis of the electrocardiogram signal. A feature of this library is to improve the functionality and simplify the interaction with existing machine learning software and tools for loading, processing and storing ECG signal datasets by using the Word2Vec model. The library increases development speed of a new software, which involves various ECG analysis. Therefore, scientists could more easily implement their ideas related to NLP and ML.

Keywords [1]

ECG signal, machine learning, WFDB, software library, NLP,Word2Vec

1. Introduction

Today, the application of machine learning algorithms in various fields is a key element in the study of various nature data in order to receive impulse for the scientific progress or to automate the conclusion generating system, which normally requires a human solution. Machine learning software libraries that exist today provide a wide range of tools for developers, making it easier to write products. That is why the improvement of these software tools is the key to accelerating the stage of scientific ideas implementation.

The process of studying the electrocardiogram (ECG) signal includes the use of software tools: for obtaining, formatting and storing datasets, analysis by machine learning methods and intermediate stages of processing. The most popular libraries that can meet these needs of a developer are: the Scikit Learn library [1], which contains machine learning algorithms and models for vectorized data; the Gensim library [2], which is an NLP tool; the WFDB library [3] providing access to a variety of datasets of electrocardiogram signals.

The listed software tools have a number of drawbacks in the context of ECG processing, some of them are: too low an abstraction level, the absence of an automatic caching system, the absence of additional tools for processing ECG signals. To improve these tools, it is necessary to perform software development to correct the listed drawbacks, reduce the amount of code to solve simple problems and allow a developer to focus on key research.

2. Related work

There are quite a few software libraries for implementing ECG signal processing. Scikit Learn or sklearn is a Python-based library that provides basic mechanisms for creating models that are then used to predict data. Also, this library provides a large number of additional tools and algorithms for pre-

[1]IDDM-2022: 5th International Conference on Informatics & Data-Driven Medicine, November 18–20, 2022, Lyon, France
EMAIL: oliyura@gmail.com (Y. Oliinyk); mihailyazenok@gmail.com (M. Yazenok); s.ocheretyany@gmail.com (O. Ocheretianyi); iaa@ukr.net (I. Baklan); lishchuk_kpi@ukr.net (K. Lishchuk); elisa.beraudolive@gmail.com (E. Beraudo);
ORCID: 0000-0002-7408-4927 (Y. Oliinyk); 0000-0002-0929-3626 (M. Yazenok); 0000-0001-9455-4781(O. Ocheretianyi) 0000-0002-5274-5261 (I. Baklan); 0000-0002-9902-0065 (K. Lishchuk); 000-0001-7550-3620 (E. Beraudo)

processing, post-processing, storing and transforming data [1]. The main types of data processing algorithms in this library are model dimension reduction, model selection, regression, classification, and cluster analysis. Also, this library provides several basic data sets for testing and verifying algorithms.

Gensim is an open source library written in Python. This library is the core for the implementation of algorithms for representing text documents in the form of semantic vectors that can be used in data analysis algorithms, the input values of which can be exclusively vectorized data [2].

The main algorithms implemented in Gensim have a common foundation of functionality, which consists in determining the semantic structure of the provided text documents by analyzing statistical repetitions of similar text sentence schemes. The input data of such algorithms does not have to be texts that a person can understand, thus, all these methods can be used to find statistical patterns of any sets of words that are a combination of absolutely different characters.

The principles of the Gensim library emphasize that they support a more practical point of view of using algorithms to solve real-world problems. That is, the set of algorithms is more focused on production-ready applications than on academic ones.

The considered libraries have a number of drawbacks when programmers use them to solve a wide range of problems. The proposed level of abstraction and the proposed architecture in these libraries significantly reduces the speed of writing code, requiring the programmer to write elementary parts, which subsequently accumulate, increasing the amount of code, thereby reducing readability. The specified list of drawbacks has a significant impact on the final assessment of the received software, increasing the number of man-hours spent on its development.

We can state a fact that there are a few software libraries for implementing ECG signal processing with using NLP and ML methods. In the article [12] authors found only 31 applications during the literature search for ECG analysis in paper between 1 April 2015 and 15 May 2020 which used ML methods in ECG analyses. In article [4] a linguistic approach to data analysis is proposed that include transformation ECG signals to linguistic chain.

In the article [11] proposed automatic ECG analysis system. The main idea of algorithm is based on using existing Support Vector Machine classifier and optimizing some of their parameters. Proposed hybrid optimization algorithm was developed using Particle Swarm Optimization and Migration Modified Biogeography Based Optimization algorithm.

Development of algorithms for ECG analysis and detection of various diseases based in them is a rather complex process. Detailed overview of models, datasets, and their accuracy in diagnosis of heart related diseases described authors in [9]. In article [10] authors presented a fully automatic and fast ECG arrhythmia classifier based on a simple brain-inspired machine learning approach known as Echo State Networks.

All the above-mentioned methods are based on the classical machine learning methods, but in the article [13] authors proposed a new technique to analyze ECG named ECG language processing that processes the ECG signal in a way a text document is treated in natural language processing (NLP) framework. This approach was extended by using Word2Vec model in our study [5]. But there are still unexplored tasks of assessing the quality of the classification methods application for ECG signal analysis, accuracy evaluation of the clustered representation accuracy of the ECG signal, evaluating software code reduction and accelerating development process

2.1. Researches tasks

Main aim: to expand the capabilities of the electrocardiogram automatic analysis by creating a Word2Vec model based on selected waves in the ECG. The following tasks should be solved within the framework of this research:

- selection of a data set for processing;
- development a software library to facilitate the development of data analysis software using the Word2Vec model;
- researching clustered representation accuracy of the ECG signal;
- researching classification algorithms efficiency;
- evaluating software code reduction and accelerating development process.

.

3. THE SOFTWARE LIBRARY DEVELOPMENT

The following describes the steps for the development of software library for analyzing the ECG signal.

3.1. ECG signal dataset preparation module

One of the modules of the developed library is the module for preparing electrocardiogram signal datasets. This module provides functionality for loading the most common datasets: MIT-BIH AFIB Dataset and PhysioNet MIT-BIH [3]. The amount of data in these datasets ensures high accuracy in machine learning.

This module provides a wide range of tools for processing the ECG signals themselves, in particular, we can distinguish: convenient dataset conversion to provide a standardized view of the entire dataset and cut off unnecessary and invalid heartbeats in the read electrocardiogram signals, thus reducing the amount of memory when storing the date without losing key information, which is necessary when solving problems using machine learning and data analysis algorithms. Also, the module provides functionality for convenient storing, reloading, rebuilding dataset data.

3.2. Caching System for Machine Learning Algorithms Data

Data caching is provided at all levels of advanced algorithms, so the programmer can reliably and, most importantly, quickly form a data cache after each stage of information processing at each stage.

The caching mechanism also makes it possible to reshape the data, so if a stage was designed incorrectly (which is often the case when developing using machine learning algorithms), then the data cached on it can be generated again.

Data caching also provides the programmer with a tool to reproduce the experiment. Since some machine learning algorithms have a certain moment of randomization of the steps of the algorithm, some of the generated data can lead to a system failure. Such situations are very difficult to reproduce, especially if the value of the basis for generating data, has not been passed to the algorithm. Implicit caching makes it possible to reproduce the experiment with the original data, which allows the programmer to make adjustments to the algorithm to ensure that it works in some edge cases that may possibly occur in the future.

It can also be noted that each extension of the main machine learning models implements a common interface, thus facilitating the work of the programmer. The implementation scheme is shown in Figure 1.

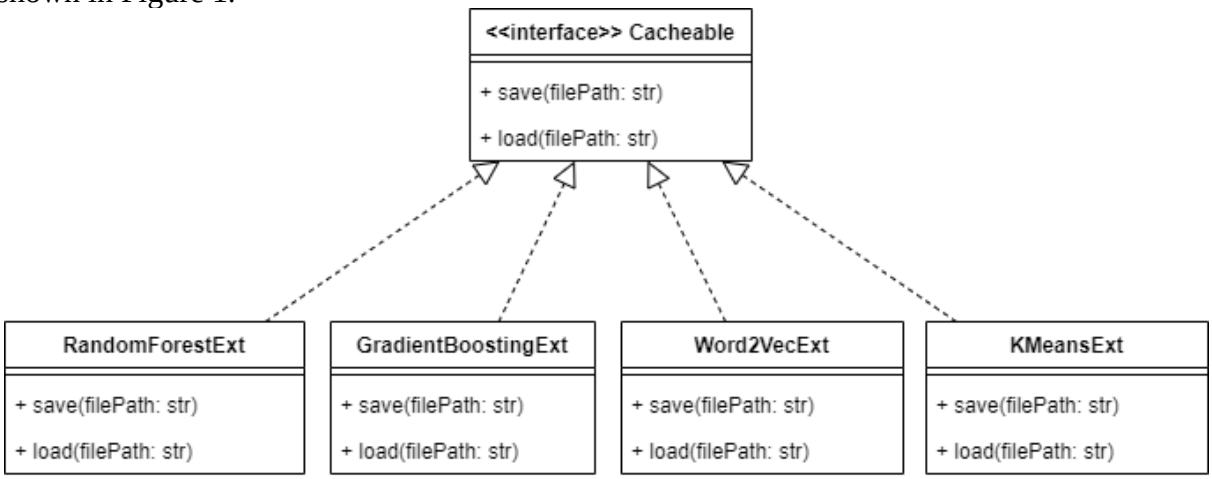

Figure 1: UML class diagram of the Cacheable interface implementation using the main models

3.3. Program Library Extension Classes

The extension class for generating datasets of electrocardiogram signals performs the following functions:

- Loading and caching of unformatted datasets.
- Standardization of dataset data to increase accuracy of machine learning.
- Splitting the complete electrocardiogram into individual heartbeats.
- Reformatting data with cutting off invalid heartbeats.
- Converting the arrhythmia class of each heartbeat to the corresponding numeric value.
- Preparing training and validation sample according to the assessment of the degree of arrhythmia of each electrocardiogram.
- Data caching before reformatting, after reformatting, and after creating training and validation sets.
- Optional sample limit for test runs.

The extension class will be implemented as a wrapper for a standard tool for obtaining datasets - the waveform-database (WFDB) package. The class supports standard electrocardiogram datasets, for example: MIT-BIH Arrhythmia Database and MIT-BIH Atrial Fibrillation Database [3].

The extension class for the Random Forest and Gradient Boosting models is designed to implement machine learning methods for ECG signal processing. The standard RandomForest and Gradient Boosting classes provided by the sklearn library have a similar functionality, namely parameterized machine learning based on an input sample of the x - y match type, however, these algorithms have specific parameters that are unique to a specific implementation, so combining extensions for them is not possible. For each of these two machine learning models, separate extensions have been created that support caching and parameterization of algorithms through interaction with the model base.

The Word2Vec [5] model extension class provides additional functionality that is useful both for the electrocardiogram and for any other input data.

In particular, the advanced caching functionality is extremely useful, which, immediately after preparing and training the model based on the specified parameters, creates a cache file for the finished model, which can be used explicitly and implicitly in the process of writing software for data analysis.

Also, specifically for the Word2Vec model, a separate functionality was created to convert the full array of words into the corresponding vector representation, omitting invalid words for which no vector transformation was created inside the trained Word2Vec model.

A separate electrocardiogram signal analysis module provides basic functions for obtaining a numeric arrhythmia class value and converting from a numeric value to an arrhythmia class, which are defined and described in the MIT database.

Also, this module provides functionality for splitting a continuous signal into separate heartbeats and highlighting individual components of the QRS complex [6] of each heartbeat. Isolation of the QRS complex can be useful when analyzing the results obtained and for forming the main data block for training models.

Separately, inside the module, the BeatsToWordsConverter class is described for converting an electrocardiogram signal into a linguistic form using the K-Means clustering algorithm [7]. This class also supports a caching system and a combined model training and caching operation for easier use by software developers.

The extension class will be implemented as a wrapper for a standard tool for obtaining datasets - the waveform-database (WFDB) package. The class supports standard electrocardiogram datasets, for example: MIT-BIH Arrhythmia Database and MIT-BIH Atrial Fibrillation Database [3].

The extension class for the Random Forest and Gradient Boosting models is designed to implement machine learning methods for ECG signal processing. The standard Random Forest and Gradient Boosting classes provided by the sklearn library have a similar functionality, namely parameterized machine learning based on an input sample of the x - y match type, however, these algorithms have specific parameters that are unique to a specific implementation, so combining extensions for them is not possible. For each of these two machine learning models, separate extensions have been created that support caching and parameterization of algorithms through interaction with the model base.

The Word2Vec [5] model extension class provides additional functionality that is useful both for the electrocardiogram and for any other input data.

In particular, the advanced caching functionality is extremely useful, which, immediately after preparing and training the model based on the specified parameters, creates a cache file for the finished model, which can be used explicitly and implicitly in the process of writing software for data analysis. Also, specifically for the Word2Vec model, a separate functionality was created to convert the full array of words into the corresponding vector representation, omitting invalid words for which no vector transformation was created inside the trained Word2Vec model.

A separate electrocardiogram signal analysis module provides basic functions for obtaining a numeric arrhythmia class value and converting from a numeric value to an arrhythmia class, which are defined and described in the MIT database.

Also, this module provides functionality for splitting a continuous signal into separate heartbeats and highlighting individual components of the QRS complex [6] of each heartbeat. Isolation of the QRS complex can be useful when analyzing the results obtained and for forming the main data block for training models.

Separately, inside the module, the BeatsToWordsConverter class is implemented for converting an electrocardiogram signal into a linguistic form using the K-Means clustering algorithm [7]. This class also supports a caching system and a combined model training and caching operation for easier use by software developers.

3.4. Using of Software library

To solve the arrhythmia classification problem based on the electrocardiogram signal, the algorithm shown in Figure 2 was developed.

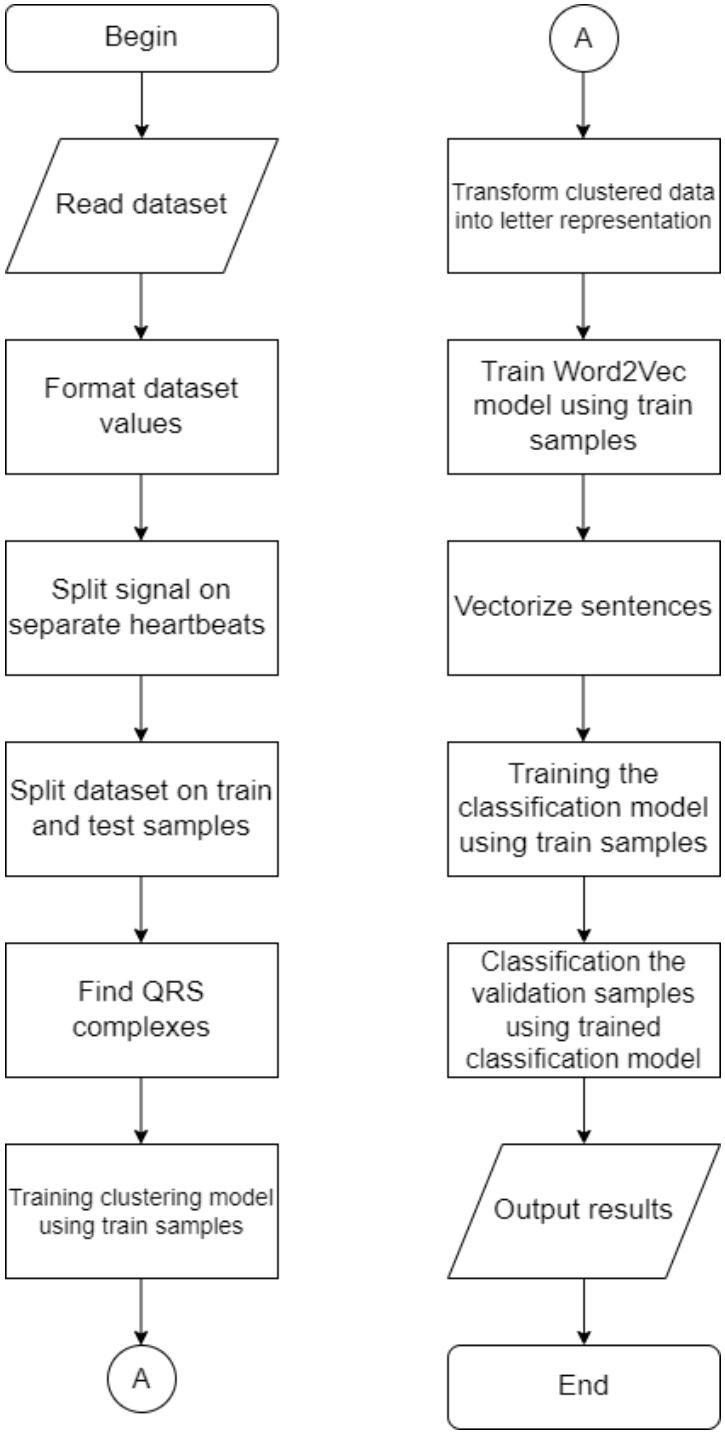

Figure 2: ECG Signal Classification Algorithm

The developed algorithm supports a simple change in the classifier model to obtain and compare results when using various machine learning algorithms, in particular Random Forest and Gradient Boosting.

The task of determining the connection between serial signals using Word2Vec. Word2Vec using the Skip Gram algorithm [8] can be used to search for context words surrounding the key. This algorithm is applicable to the sequence of electrocardiogram signals to determine whether the connection between successive signals is inherent in the context of the linguistic representation of the electrocardiogram.

The developed algorithm is shown in Figure 3.

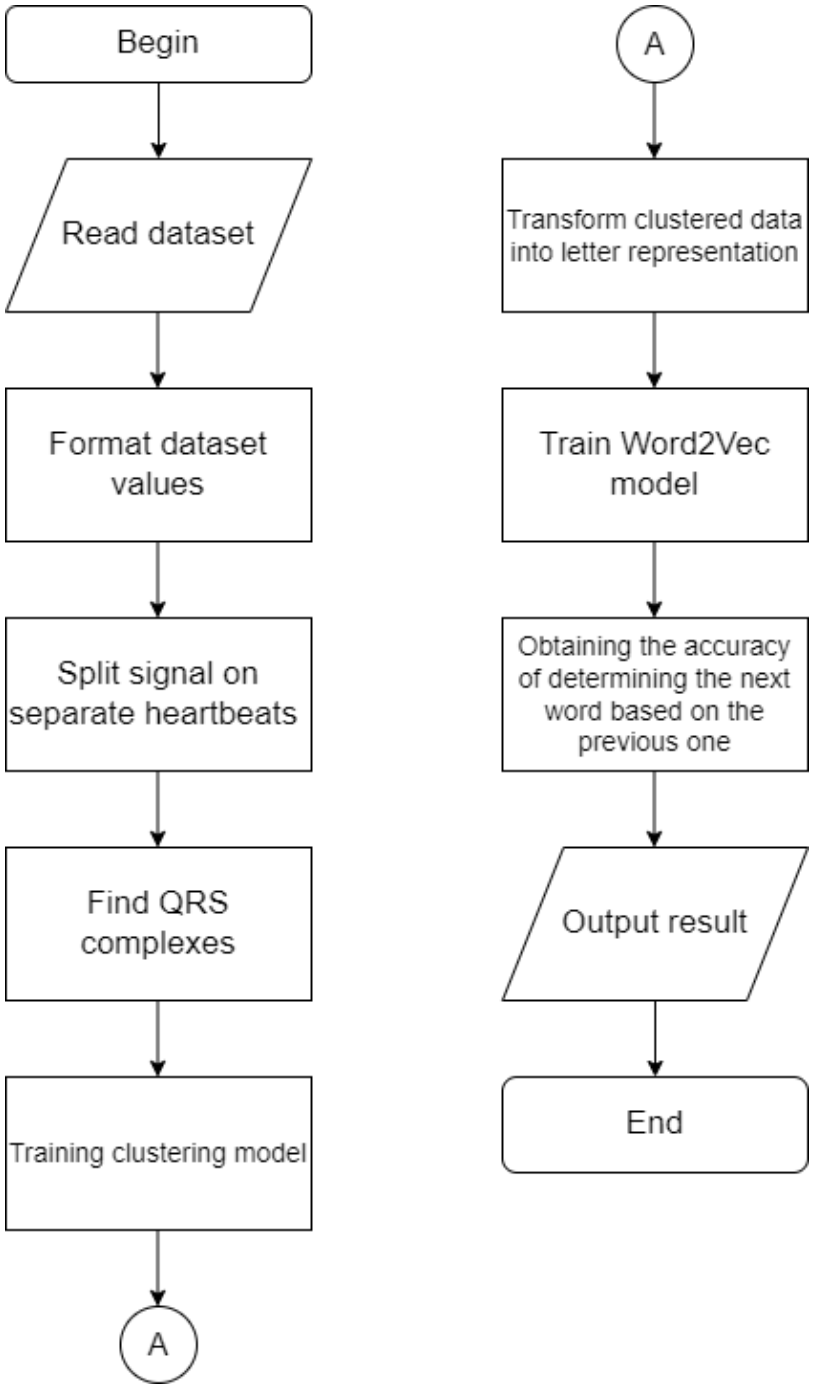

Figure 3: Algorithm for calculating connections between consecutive ECG signals

4. THE SOFTWARE LIBRARY EFFICIENCY

The following describes the efficiency of software library for analyzing the ECG signal. To study the efficiency, accuracy and conciseness of the developed and computer software, it is necessary to conduct a number of experiments:

- comparing the compactness of the program code with and without the library when solving problems: preparing an electrocardiogram dataset, splitting the dataset into training and validation samples, caching intermediate stage data, the general task of arrhythmia classification based on the ECG signal;

- calculating the speed of the machine learning algorithm when using the linguistic representation of the electrocardiogram signal;
- determination of the accuracy of the clustered representation of the electrocardiogram signal for different sizes of clusters;
- application of the library for the task of determining the presence of a connection between successive ECG signals in a linguistic representation;
- application of the library for the analysis of data presented by the TextRank method when applied to the linguistic representation of the ECG signal.

For research, we used hardware with the following characteristics: Intel Core i5-6200U CPU 2.3-2.4 GHz, 12 GB RAM, Samsung 870 Evo-Series 1TB SATA III, AMD Radeon R5 M330. The experiments were carried out on the operating system Windows 10 Corporate 2016.

4.1. Researching clustered representation accuracy of the ECG signal

The formation of a clustered representation consists of the formation of QRS complexes for individual heartbeats of the electrocardiogram signal, followed by clustering of the formed parts using two K-means models.

The first model is responsible for clustering the intervals of R-peaks of heartbeats - high points in the ECG signal.

The second model is responsible for clustering the PR intervals that are in the corresponding R-peak intervals and the ST intervals that are after the corresponding R-peak intervals.

Thus, the entire ECG signal is subject to clustering.

After clustering, we replace the corresponding signal intervals with cluster indices. The result is a sequence of indices, where every three indices represent one heartbeat. The first index is the PR interval, the second is the R-peak interval, the third index is the ST interval.

To reverse transform a clustered sequence, you must replace each index with the value of the center of the corresponding cluster. Thus, we get a copy of the electrocardiogram signal close to the original. The accuracy of such a transformation depends entirely on the number of clusters specified as parameters in both K-means models.

An example of inverse transformation is shown in Figure 4. The orange line is the original signal, the blue line is the restored signal from the clustered representation.

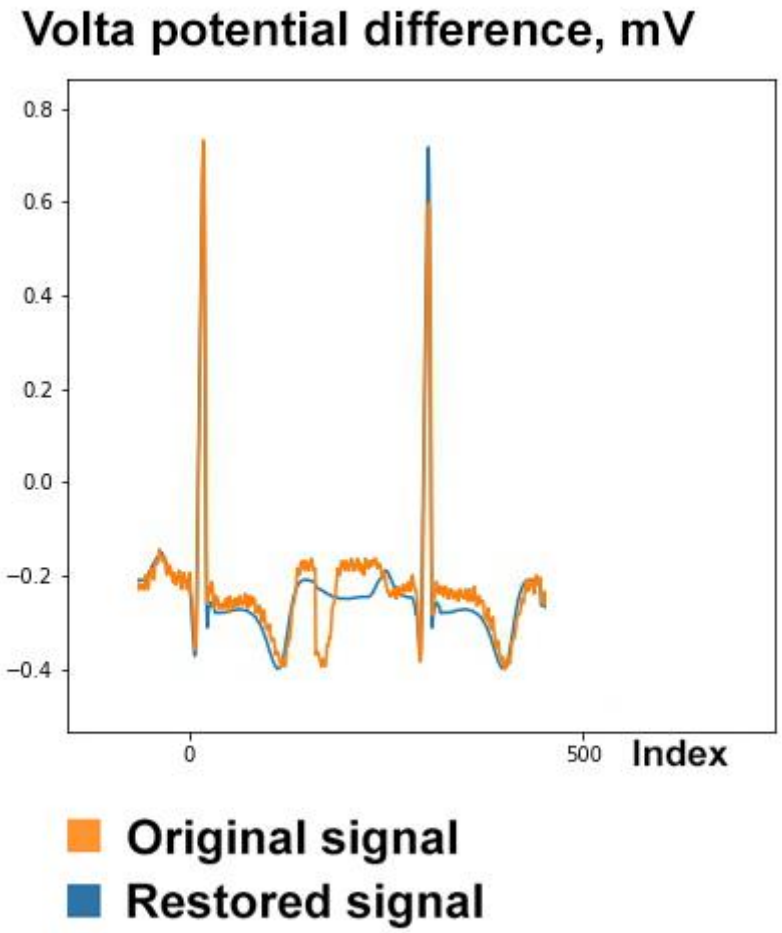

Figure 4: Comparison of the original and inverse transformation signals

The rms error was used to estimate the accuracy of the clustered transformation.

A graph of the change in the root-mean-square error depending on the total number of clusters in both models is shown in Figure 5. Specific values are given in Table 1.

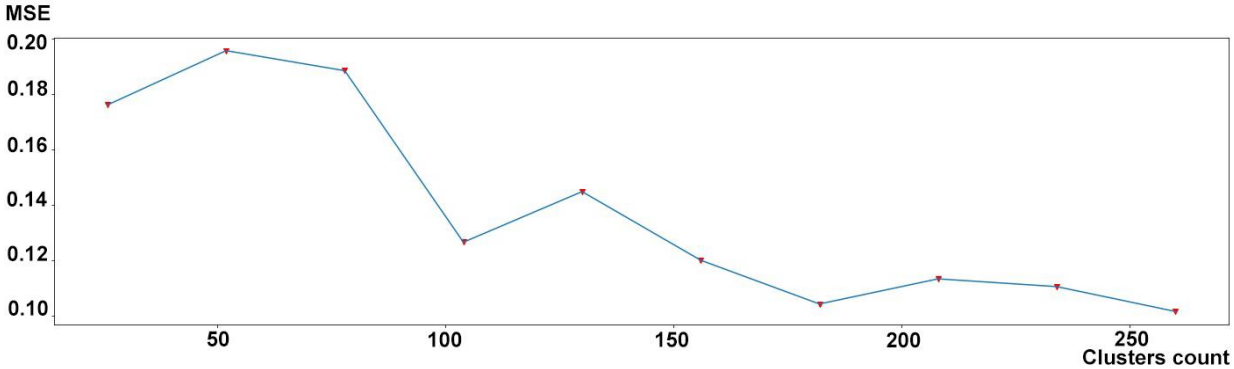

Figure 5: RMS errors versus total number of clusters

Table 1
RMS errors depending on the number of clusters

The number of clusters in the first model	Number of clusters in the second model	MSE
6	20	0.173
12	40	0.192
18	60	0.19
24	80	0.124
30	100	0.143
36	120	0.12
42	140	0.105
48	160	0.112
54	180	0.111
60	200	0.101

As can be seen from the graph, the error in increasing the number of clusters decreases. So, when using 60 clusters for the first model and 200 for the second model, the error is 10%.

A graphical comparison of the signal with maximum accuracy is shown in Figure 6.

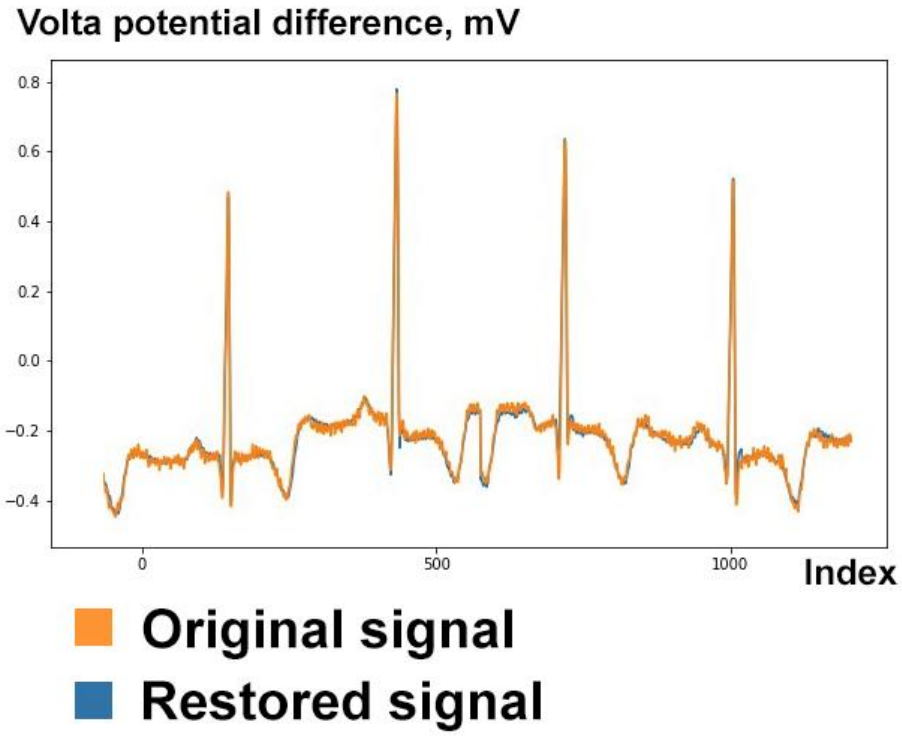

Figure 6: Comparison of the original signal and the signal reproduced after clustering using models with 60 and 200 clusters

4.2. Researching classification algorithms efficiency

When using a linguistic representation of an electrocardiogram followed by vectorization using Word2Vec, the amount of input data for machine learning algorithms is reduced, thereby accelerating the learning rate of classifier models. The developed algorithm is able to reduce the amount of input data by several times by initial clustering and then vectorization using the Word2Vec model.

To determine the improvement in speed, a number of experiments were carried out with the measurement of the time spent on training classifier algorithms. The results are presented in table 2.

Table 2

Model learning rate with and without using the Word2Vec model

Algorithm	Initial dataset volume (heart bit thousands)	Training duration, s		F- measure	
		Without Word2Vec model	With using Word2Vec model	Without using Word2Vec model	With using Word2Vec model
Random Forest	10.5	33,2447	27,796	0.95	0.96
Gradient Boosting		32,1669	27,877	0.96	0.98
Random Forest	13.2	495,244	45,063	0.95	0.97
Gradient Boosting		479,433	40,708	0.93	0.97

As can be seen from the table, the learning time really increases when using the text representation of the electrocardiogram signal vectorized using the Word2Vec model. Especially significant changes are noticeable when increasing the dataset volume.

4.3. Evaluating software code reduction and accelerating development process

To compare the compactness of the code, we will use the count of the number of words and characters to write the same functional block using the developed library and without it.

Table 3 compares the number of words and symbols when solving the problem of preparing an electrocardiogram signal dataset and dividing it into a training and validation set. Table 4 compares data caching implementations. Example of using software library code can be found in source [14].

Table 3

Counting the number of software code words and characters for preparing a dataset

Used decision	Number of the words for solving	Number of the symbols for solving
Using the library	37	545
Without using the library	813	5577

An example of using the library to prepare a dataset:

```
ecgdataset = ecgdatasetsholder.EcgDatasetsHolder.cache_from_mit(
    sets_count_limit=5,
    database_name="mitdb",
    mit_records_path=rsc_dir + "/mit_records",
    dataframe_path=rsc_dir + "/ecgdataset",
    annotator_type="symbol",
    reload=False
)

train, test = ecgdataset.split_train_test(test_size=0.25, random_state=42)
```

```
train_ready = train.concatenate_datasets()
test_ready = test.concatenate_datasets()
```

Table 4
Counting the number of words and characters for solving caching problem

The task of caching, or learning and caching	Using the library		Without using the library	
	Words	Symbols	Words	Symbols
Word2vec	37	336	169	1117
KMeans	20	237	121	736
WFDB Dataset	19	199	609	3932

An example of using the library for the task of learning and caching Word2Vec:

```
num_features = 300
word2vecExtModel = \
    word2vecext.Word2VecExt.load_or_fit_words_and_save(train_words,

train_ready.train_start_indices["start_indices"].tolist(),
                                                    rsc_dir + "/word2vec",
                                                    vector_size=num_features,
                                                    reset=False)
train_data = word2vecExtModel.vectorize_valid_with_labels(train_words,
train_ready.dataframe["labels"].tolist())

validation_data = word2vecExtModel.vectorize_valid_with_labels(validation_words,
test_ready.dataframe["labels"].tolist())

(train_x, train_y), (validation_x, validation_y) = (train_data, validation_data)
```

5. Discussion and Conclusion

Today among modern researches use of machine learning algorithm in medical data assesment becomes inevitable. Therefore we have developed software library for analyzing the ECG signal by using Word2Vec model , which includes ECG signal dataset preparation module, caching system for machine learning algorithms data, program library extension classes. With the help of the developed library, the amount of software code necessary for data preparation, caching or analysis is reduced several times.

Clustered representation accuracy of the ECG signal was investigated. RMS errors of restored data significantly decreases when using 100 or more clusters and is approximately 10%. Therefore, the developed method [5] of presenting the ECG signal by using Word2Vec model, which reduces the original ECG signal more than 100 times, can be effectively applied to significantly reduce the stored signal and its further analysis without loss of quality.

Use of Word2Vec model increases F-measure of Random Forest method from 0.95 to 0.97 and from 0.96 to 0.98 for Gradient Boosting. Learning time significantly increases when using the text representation of the electrocardiogram signal vectorized using the Word2Vec model, especially in case of increasing the dataset volume.

The developed library increases the level of abstraction, which allows researchers to use it with less programming experience in their fields. An important direction of the library's development will be its expansion and addition of support for new algorithms, so that scientists can more efficiently solve the tasks, without spending time on low-level adjustment of algorithms.

References

[1] Scikit-learn, a Python module for machine learning, 2020. URL:https://scikit-learn.org.

[2] Gensim library,2022. URL: https://radimrehurek.com/gensim.

[3] The WFDB Software Package, 2018. URL: https://archive.physionet.org/phys-iotools/wfdb.shtml.

[4] Igor Baklan, ECG Signal Processing Based on Linguistic Chain Fuzzy Sets, in: Alina Oliinyk, Iryna Mukha, Kateryna Lishchuk, Olena Gavrilenko, Svitlana Reutska, Anna Tsytsyliuk, Yurii Oliinyk, Proceedings of the 5th International Conference on Computational Linguistics and Intelligent Systems, COLINS'2021, volume I of Main Conference, CEUR-WS, volume 2870, pp. 1731-1741. URL: http://ceur-ws.org/Vol-2870/paper125.pdf

[5] Yurii Oliinyk, Andrii Tereschenko, Igor Baklan, Elisa Beraudo, ECG Analysis based on Word2Vec Model in Proceedings of the 4th International Conference on Informatics & Data-Driven Medicine, IDDM 2021 Valencia, Spain, CEUR-WS, volume 3038, pp. 203-232. URL: http://ceur-ws.org/Vol-3038/short9.pdf

[6] J. Pan and W. J. Tompkins, "A Real-Time QRS Detection Algorithm," in IEEE Transactions on Biomedical Engineering, vol. BME-32, no. 3, pp. 230-236, March 1985, doi: 10.1109/TBME.1985.325532.

[7] Tryon, Robert C. Cluster Analysis: Correlation Profile and Orthometric (factor) Analysis for the Isolation of Unities in Mind and Personality, Ann Arbor, Mich., Edwards Brothers,1939.

[8] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., Dean, J. , Distributed representations of words and phrases and their compositionality, in: Proceeding of the 27th Annual Conference on Neural Information Processing System, Advances in neural information processing systems, volume 26, pp.3111-1119,2013.

[9] Mishra A, Dharahas G, Gite S, Kotecha K, Koundal D, Zaguia A, Kaur M, Lee HN. ECG Data Analysis with Denoising Approach and Customized CNNs. Sensors (Basel). 2022 Mar 1;22(5):1928. doi: 10.3390/s22051928.

[10] Alfaras M, Soriano MC and Ortín S (2019) A Fast Machine Learning Model for ECG-Based Heartbeat Classification and Arrhythmia Detection. Front. Phys. 7:103. doi: 10.3389/fphy.2019.00103

[11] Manikandan Kaliappan, Sumithra Manimegalai Govindan, and Mohana Sundaram Kuppusamy. 2022. Automatic ECG analysis system with hybrid optimization algorithm based feature selection and classifier. J. Intell. Fuzzy Syst. 43, 1 (2022), 627–642. https://doi.org/10.3233/JIFS-212373

[12] Sulaiman Somani, Adam J Russak, Felix Richter, Shan Zhao, Akhil Vaid, Fayzan Chaudhry, Jessica K De Freitas, Nidhi Naik, Riccardo Miotto, Girish N Nadkarni, Jagat Narula, Edgar Argulian, Benjamin S Glicksberg, Deep learning and the electrocardiogram: review of the current state-of-the-art, EP Europace, Volume 23, Issue 8, August 2021, Pages 1179–1191, https://doi.org/10.1093/europace/euaa377

[13] Sajad Mousavi, Fatemeh Afghah, Fatemeh Khadem, U. Rajendra Acharya, ECG Language processing (ELP): A new technique to analyze ECG signals, Computer Methods and Programs in Biomedicine, Volume 202, 2021, 105959, ISSN 0169-2607, https://doi.org/10.1016/j.cmpb.2021.105959.

[14] Example of using software library based on NLP and ML methods, 2022. URL: https://colab.research.google.com/drive/1L46s8gcXfOzAsp5Lenoem9NFrNfKKWOm