

Система интеллектуального анализа ЭКГ на базе нейросетевой модели машинного обучения

Краткое руководство пользователя

2022 г.

Данный документ или его копии не может распространяться (полностью или частично) в любом формате без письменного разрешения ИСП РАН.

Содержание

| | |
|--|----|
| 1. Описание функциональных возможностей | 3 |
| 2. Лицензирование и установка | 4 |
| 2.1 Системные требования | 4 |
| 2.2 Установка | 4 |
| 3. Сценарий работы с системой анализа ЭКГ | 5 |
| 3.1 Запуск интерпретатора python 3 | 5 |
| 3.2 Загрузка набора данных PTB-XL и создание класса PTBXLDataset с целевой меткой AFIB (можно выбрать любую из представленных: AFIB, 1AVB, STACH, SBRAD, RBBB, LBBB, PVC, LVH) | 5 |
| 3.3 Загрузка предобученной модели | 5 |
| 3.4 Предсказание вероятности патологии для выбранной ЭКГ (можно выбрать любую из представленных: AFIB, 1AVB, STACH, SBRAD, RBBB, LBBB, PVC, LVH) | 6 |
| 4. Описание функций системы анализа ЭКГ | 7 |
| 4.1 Модуль Datasets | 7 |
| 4.2 Модуль Models | 8 |
| 4.3 Модуль Preprocessing | 9 |
| 5. Описание процессов, обеспечивающих поддержание жизненного цикла ПО | 10 |
| 5.1 Процессы разработки и совершенствования ПО | 10 |
| 5.2 Поддержка пользователей ПО | 10 |
| 5.3 Необходимый персонал для разработки и поддержки | 10 |
| 6. Перечень терминов и сокращений | 12 |

1. Описание функциональных возможностей

Система интеллектуального анализа ЭКГ на базе нейросетевой модели машинного обучения (далее – Система, Система анализа ЭКГ, альтернативное название – «EcgLib») применяется для автоматизации преобразования, хранения и анализа ЭКГ. Настоящий документ является кратким руководством пользователя Системы.

Система позволяет исследователям использовать архитектуры моделей, предварительно обученные на более чем 500 000 записей ЭКГ, для их точной настройки на собственных наборах данных. Система анализа ЭКГ является проектом с открытым исходным кодом (<https://github.com/ispras/EcgLib>).

Основными функциями Системы являются:

- предобработка сигналов ЭКГ;
- загрузка наборов данных;
- создание классов набора данных для обучения моделей по обнаружению патологий ЭКГ.

Минимальные системные требования: для функционирования Системы требуется ОС Ubuntu 18.04/20.04, не менее 4-х процессорных ядер x86-64, 16 ГБ оперативной памяти, 100 ГБ свободного дискового пространства.

Система анализа ЭКГ разрабатывается на языке программирования Python (версия ≥ 3.6).

2. Лицензирование и установка

Ссылка для скачивания:

<https://github.com/ispras/EcgLib/archive/refs/heads/main.zip>

По указанной ссылке Вы сможете загрузить систему интеллектуального анализа ЭКГ на базе нейросетевой модели машинного обучения. Предоставлена полная версия инструмента.

Система разрабатывается на языке программирования Python и распространяется под свободной лицензией Apache License 2.0 и представляет собой библиотеку для интеллектуального анализа ЭКГ.

По вопросам определения стоимости, приобретения и использования, обращайтесь по адресу ecglib@ispras.ru.

2.1 Системные требования

Минимальные системные требования: Для функционирования Системы требуется персональный компьютер с архитектурой x86-64, не менее 16 ГБ оперативной памяти, и не менее 100 ГБ свободного места на диске. Поддерживаются следующие 64-разрядные дистрибутивы операционной системы Linux: Ubuntu 18 и Ubuntu 20.

2.2 Установка

Для установки Системы в ОС Ubuntu 20.04 с python версии > 3.6 необходимо выполнить следующие команды установки:

```
#установка EcgLib
sudo apt install python3-pip
sudo apt update
sudo apt install python3-pip
pip3 install ecglib
```

3. Сценарий работы с системой анализа ЭКГ

Для ознакомления и работы с Системой анализа ЭКГ («EcgLib») в операционной системе Ubuntu 20.04 необходимо выполнить следующие действия:

1. Установить python3-pip и с его помощью Систему анализа ЭКГ
2. Перейти в интерпретатор питона
3. Загрузить набор данных PTB-XL и создать класс PTBXLDataset
4. Загрузить предобученную модель
5. Предсказать вероятность патологии для выбранной ЭКГ

3.1 Запуск интерпретатора python 3

```
python3
```

3.2 Загрузка набора данных PTB-XL и создание класса PTBXLDataset с целевой меткой AFIB (можно выбрать любую из представленных: AFIB, 1AVB, STACH, SBRAD, RBBB, LBBB, PVC, LVH)

```
# download PTB-XL ECG dataset

from ecglib.datasets import load_ptb_xl
ptb_xl_info = load_ptb_xl(download=True)

# create PTBXLDataset class from PTB-XL map file
# fit targets for 'AFIB' binary classification

from ecglib.datasets import PTBXLDataset
targets = [[0.0] if 'AFIB' in eval(ptb_xl_info.iloc[i]['scp_codes']).keys() else
[1.0] for i in range(ptb_xl_info.shape[0])]
ecg_data = PTBXLDataset(ecg_data=ptb_xl_info, target=targets)
```

3.3 Загрузка предобученной модели

```
# load pretrained 'densenet1d121' model from 'weights' folder for binary
classification 12-lead experiment
from ecglib.models import create_model
model = create_model(model_name='densenet1d121', pathology='AFIB',
pretrained=True, leads_num=12)
```

3.4 Предсказание вероятности патологии для выбранной ЭКГ (можно выбрать любую из представленных: AFIB, 1AVB, STACH, SBRAD, RBBB, LBBB, PVC, LVH)

```
import torch

# берем любую, например, первую, запись из датасета
y = ecg_data.__getitem__(1)[1][0]
z = y.unsqueeze(0)

with torch.no_grad():
    # get result as logits tensor
    predict = model(z)
    sigmoid = torch.nn.Sigmoid()
    percents = sigmoid(predict).numpy()
    print(percents)

#           Пример           вывода           (вероятность           AFIB) :
# [[0.01344207]]
```

4. Описание функций системы анализа ЭКГ

Основными функциями Системы анализа ЭКГ являются:

- предобработка сигналов ЭКГ
- загрузка наборов данных
- создание классов набора данных для обучения моделей по обнаружению патологий ЭКГ

Реализацию данных функций обеспечивают три основных модуля Системы анализа ЭКГ:

1. Datasets
2. Models
3. Preprocessing

4.1 Модуль Datasets

Данный модуль позволяет пользователю загружать и хранить наборы данных ЭКГ в различных форматах и извлекать метаданные о каждом отдельном сигнале ЭКГ (например: частоту, полный путь к файлу, scr-коды, возраст пациента и т. д.)

Через `load_datasets.py` можно загрузить базу данных ЭКГ PTB-XL (<https://physionet.org/content/ptb-xl/1.0.2/>) в исходном формате `wfdb` и сохранить информацию о каждой записи в файле `csv`.

```
# download PTB-XL ECG dataset

from ecglib.datasets import load_ptb_xl

ptb_xl_info = load_ptb_xl(download=True)
```

Скрипт `datasets.py` содержит классы для хранения наборов данных ЭКГ:

- `ECGDataset` — это общий класс для хранения основных характеристик набора данных ЭКГ, таких как количество отведений, количество классов для прогнозирования, аугментация и т. д.
- `PTBXLDataset` является дочерним классом по отношению к `ECGDataset`; можно загрузить каждую запись из формата `wfdb` или `prz` и предварительно

обработать ее перед дальнейшим использованием. Также возможно создать изображение png для каждой записи, используя ecg-plot.

```
# create PTBXLDataset class from PTB-XL map file
# fit targets for 'AFIB' binary classification

from ecglib.datasets import PTBXLDataset

targets = [[0.0] if 'AFIB' in eval(ptb_xl_info.iloc[i]['scp_codes']).keys() else
[1.0]
            for i in range(ptb_xl_info.shape[0])]
ecg_data = PTBXLDataset(ecg_data=ptb_xl_info, target=targets)
```

4.2 Модуль Models

Данный модуль включает компоненты архитектур моделей и открытые веса для моделей, полученных в результате экспериментов по обучению нейросетевой модели для задачи бинарной классификации при нескольких патологиях.

Функция `create_model` позволяет пользователю создавать модель с нуля (поддерживаемые в настоящее время архитектуры включают в себя `densenet1d121`, `densenet1d201`), а также загружать предварительно обученную контрольную точку модели из папки весов (поддерживаемые в настоящее время архитектуры включают в себя `densenet1d121`).

```
# create 'densenet1d121' model from scratch for binary classification 12-lead
experiment

from ecglib.models import create_model

model = create_model(model_name='densenet1d121', pathology='1AVB',
pretrained=False, leads_num=12)

# load pretrained 'densenet1d121' model from 'weights' folder for binary
classification 12-lead experiment

from ecglib.models import create_model

model = create_model(model_name='densenet1d121', pathology='AFIB',
pretrained=True, leads_num=12)
```


Папка “architectures” включает модели архитектур.

Папка “config” содержит классы данных параметров по умолчанию для построения модели.

В папке “weights” находится файл с путями к моделям, полученным из следующих экспериментов бинарной классификации с 12 отведениями. Имеющиеся в настоящее время патологии (scr-коды): AFIB, 1AVB, STACH, SBRAD, RBBB, LBBB, PVC, LVH.

4.3 Модуль Preprocessing

Данный модуль включает в себя структуру, вдохновленную библиотекой языка Python Alumentations (<https://alumentations.ai/>) для предварительной обработки и дополнения данных ЭКГ.

Скрипт composition.py содержит структуры SomeOf, OneOf и Compose для создания собственного конвейера предварительной обработки и аугментации.

Скрипты preprocess.py и functions.py содержат классы и функции, описывающие различные методы предварительной обработки и аугментации.

```
# augmentation example
import torch
from ecglib.preprocessing.preprocess import *
from ecglib.preprocessing.composition import *

# provide an ecg-record in `numpy.ndarray` form
ecg_record = read_any_ECG_ndarray_type

augmented_record = Compose(transforms=[
    SumAug(leads=[0, 6, 11]),
    RandomConvexAug(n=4),
    OneOf(transforms=[ButterworthFilter(), IIRNotchFilter()],
transform_prob=[0.8, 0.2])
], p=0.5)(ecg_record)
```

5. Описание процессов, обеспечивающих поддержание жизненного цикла ПО

5.1 Процессы разработки и совершенствования ПО

Разработка системы интеллектуального анализа ЭКГ на базе нейросетевой модели машинного обучения ведется по методологии Agile с привлечением современных средств повышения качества кода.

1. Для хранения кода используется система контроля версий git, и изменения в основной ветке проходят инспекцию кода (code review) другими разработчиками.
2. При разработке используется практика непрерывной интеграции (continuous integration): при помощи сервера тестирования Jenkins происходят регулярные автоматические сборки с последующим запуском тестов, упомянутых в предыдущем пункте.
3. При написании кода разработчики должны придерживаться строгих правил оформления кода. Нарушение этих правил не позволит пройти этап инспекции кода.

5.2 Поддержка пользователей ПО

Установка на рабочей станции системы интеллектуального анализа ЭКГ на базе нейросетевой модели машинного обучения не требует настройки ОС, установки внешнего ПО или каких-либо других дополнительных действий.

Со своей стороны ИСП РАН постоянно совершенствует Систему анализа ЭКГ, применяя в жизненном цикле разработки передовые методики. Добавление новых и улучшение существующих алгоритмов ведется в инициативном порядке. Обновления Системы анализа ЭКГ передаются пользователям через согласованные с ними каналы распространения обновлений.

5.3 Необходимый персонал для разработки и поддержки

Для разработки и поддержки программного продукта необходима соответствующая квалификация разработчиков. Это вызвано следующими причинами.

1. Специфическая предметная область, требующая глубоких знаний одновременно в нескольких областях: анализ медицинских данных,

кардиология, технологии разработки ПО с использованием технологий искусственного интеллекта.

2. Требования к производительности системы, из-за чего необходимо применять эффективные алгоритмы, в т. ч. хорошо масштабируемые по нескольким вычислительным ядрам.

В силу приведенных причин коллектив разработчиков Системы анализа ЭКГ («EcgLib») формируется из специалистов, получивших высшее профильное образование.

Для гарантийного обслуживания задействовано три научных сотрудника, для Технической поддержки задействован один научный сотрудник, для модернизации программного обеспечения задействованы три научных сотрудника.

Адрес электронной почты, по которому можно обратиться по вопросам, связанным с системой анализа ЭКГ — ecglib@ispras.ru.

6. Перечень терминов и сокращений

| | |
|---------|--|
| Система | Система интеллектуального анализа ЭКГ на базе нейросетевой модели машинного обучения |
| ЭКГ | Электрокардиограмма |
| AFIB | Мерцательная аритмия |
| 1AVB | Атриовентрикулярная блокада первой степени |
| STACH | Синусовая тахикардия |
| SBRAD | Синусовая брадикардия |
| RBBB | Блокада правой ножки пучка Гиса |
| LBBB | Блокада левой ножки пучка Гиса |
| PVC | Преждевременный желудочковый комплекс |
| LVH | Гипертрофия левого желудочка |