Raport z realizacji projektu

Wersja 1.3

Temat projektu: **Symulator biosygnałów na potrzeby systemów IoT Healthcare**

Autorzy projektu: Jowita Lalewicz, Marta Mąka, Michał Szwalec

PM: Michał Szwalec

Abstrakt

Celem projektu było zaprojektowanie i stworzenie modelu, który w sposób realistyczny symuluje sygnały EKG o wybranej charakterystyce. Metody realizacji projektu opierają się na języku programowania Python, wraz z bibliotekami NumPy, Random, Pandas i Matplotlib. Sygnał EKG o określonej charakterystyce jest generowany z wykorzystaniem równania matematycznego, które poddano pseudolosowej parametryzacji, umożliwiającej generowanie sygnałów z zachowaniem ogólnej charakterystyki, lecz o zróżnicowanej wartości. Zakładane funkcjonalności obejmują wizualizację przebiegu czasowego sygnału oraz eksport wygenerowanego biosygnału w postaci pliku csv. Mierzalnym wynikiem projektu jest korelacja pomiędzy generowanymi sygnałami a sygnałem wzorcowym. Wygenerowane sygnały oraz ich charakterystyczne wartości zostaną porównane z sygnałami EKG pobranymi z bazy danych PhysioNet w celu oceny generowanego sygnału, sprawdzone zostanie czy wygenerowany sygnał jest realistyczny i odzwierciedla wybrane schorzenie. Projekt może znaleźć zastosowanie w dziedzinie IoT Healthcare jako narzędzie wspomagające procesy diagnostyczne i terapeutyczne. Dzięki realistycznemu symulowaniu wybranych schorzeń serca za pomocą generowania sygnałów EKG, projekt może pomóc w szkoleniach personelu medycznego, a także służyć jako narzędzie diagnostyczne do wykrywania nieprawidłowości w sygnałach EKG pacjentów.

Wstęp i badania literaturowe

a) Cele i założenia projektu.

Symulatory biosygnałów są niezwykle ważnym narzędziem w projektowaniu i testowaniu systemów IoT Healthcare, które obejmują monitoring stanu zdrowia pacjentów, diagnozowanie chorób i przeprowadzanie terapii. Symulator biosygnałów może działać w czasie rzeczywistym i umożliwiać generowanie sygnałów z różnych źródeł, takich jak fizyczne symulatory, bazy danych sygnałów lub modele matematyczne. Może także umożliwić użytkownikowi symulowanie różnych scenariuszy klinicznych i warunków, takich jak choroby

serca. Celem projektu jest zaprojektowanie i stworzenie modelu, który będzie w stanie w sposób realistyczny symulować sygnały EKG o wybranej charakterystyce, a następnie ich ocena.

b) Zarys ogólny proponowanego rozwiązania.

Sygnał EKG o określonej charakterystyce zostanie wygenerowany z wykorzystaniem równania matematycznego, które będzie poddane pseudolosowej parametryzacji, umożliwiającej generowanie sygnałów z zachowaniem ogólnej charakterystyki, lecz o zróżnicowanej wartości. Mierzalny wynikiem projektu będzie korelacja pomiędzy generowanymi sygnałami, a sygnałem wzorcowym. Wygenerowane sygnały oraz ich charakterystyczne wartości zostaną porównane z sygnałami EKG pobranymi z bazy danych PhysioNet w celu oceny generowanego sygnału, sprawdzone zostanie czy wygenerowany sygnał jest realistyczny i odzwierciedla wybrane schorzenie.

Alternatywne rozwiązanie na podstawie przeglądu literatury zostało przedstawiono w poniższej tabeli:

Tytuł artykułu	Simulation Of Pathological ECG Signal Using Transform Method
Link	https://doi.org/10.1016/j.procs.2020.04.229
Cytowanie	*Manju B.R., Akshaya B., Simulation Of Pathological ECG Signal Using Transform Method, Procedia Computer Science, Volume 171, 2020, Pages 2121-2127, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2020.04.229.*
Cel artykułu, hipoteza badawcza	Opracowanie modelu generującego realistyczny sygnał EKG patologii serca
Wykorzystane technologie	MATLAB

Zastosowane metody przetwarzania, analizy i klasyfikacji sygnałów, klasyfikatory	Aproksymacja szeregu Fouriera: • funkcja paraboliczna - załamek P i T • funkcja o trójkątnym kształcie - zespół QRS
Skrócony protokół przeprowadzonego eksperymentu (w jaki sposób przeprowadzono eksperymenty)	1. Zebranie danych (parametrów) z biologicznych sygnałów EKG 2. Matematyczne modele poszczególnych fragmentów charakterystycznych sygnału EKG 3. Aproksymacja szeregu Fouriera każdego zamodelowanego fragmentu sygnału 4. Otrzymanie pełnego sygnału EKG i ocena specjalistów
Rodzaje/ patologie sygnału EKG	• Normalny • Hipokaliemia • Hiperkaliemia • Dekstrokardia • Dusznica bolesna
Uzyskane wyniki (skuteczność, dokładność pomiaru, wielkość błędu)	Pozytywna ocena lekarzy specjalistów

Koncepcja proponowanego rozwiązania

Wykorzystane technologie

Projekt został napisany w języku Python, który oferuje bilbioteki takie jak NumPy, Random, Pandas i Matplotlib, które również zostały wykorzystane. Dane zostały zaimportowane z bazy danych PhysioNet w postacji pliku csv.

Funkcjonalność

Wizualizacja przebiegu czasowego sygnału EKG o określonej charaketrystyce na podstawie wybranych sygnałów z bazy danych PhysioNet. Wybór stopnia wygładzenia jak również wartości korelacji. Możliwość oceny sygnału przez porównanie go z sygnałem wzorcowym. Eksport wygenerowanego biosygnału w postaci pliku csv.

Opis algorytmu

W ramach projektu zdefiniowane zostało pięć różnych rodzajów sygnałów EKG, a dla każdego z nich zostały wyodrębnione odpowiednie próbki ze zbioru danych. Następnie przeprowadzono generowanie sygnałów na podstawie pseudolosowej parametryzacji, które były modyfikowane w oparciu o wybraną wartość korelacji. W celu oczyszczenia danych usunięto wartości skrajne maksimum i minimum (metoda eliminacji szumów). Zastosowano również wygładzanie sygnału. Otrzymane sygnały zostały przedstawione na wykresach wraz z sygnałem wzorcowym.

Kod źródłowy

Główna pętla programu:

```
# usuwanie z zbioru granicznych wartosci max i min w celu "oczyszczenia" danych


Normal = pd.DataFrame(Normal, columns=column_names)
Normal = Normal.iloc[:, :-1]
print(f' shape before max/min drop: {Normal.shape}')
for i in range(Normal.shape[1]):
 if Normal.shape[0] > 40:
 if 184 > i > 5:
 max_value = (Normal[column_names[i]]).max()
 min_value = (Normal[column_names[i]]).min()
 max_value_indexes = Normal.index[Normal[column_names[i]] == max_value].tolist()
 Normal.drop(max_value_indexes, inplace=True)
 min_value_indexes = Normal.index[Normal[column_names[i]] == min_value].tolist()
 Normal.drop(min_value_indexes, inplace=True)
print(f'shape after max/min drop: {Normal.shape}')

# utworzenie zbioru max i min


minimal = Normal.min()
maximal = Normal.max()


#Generowanie sygnału
```

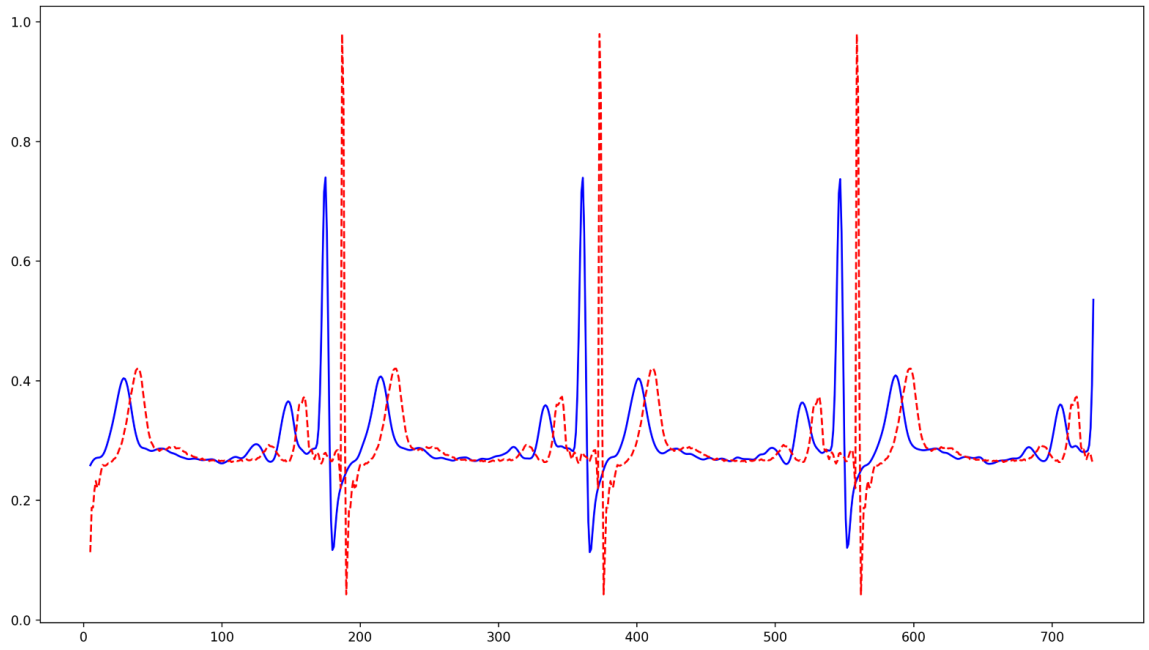
```
for n in range(1, 5):
 mean = Normal.mean()
 Generated_signal = mean
 index = Generated_signal.index
 for i in range(186):
 if i < 2:
 Generated_signal = Generated_signal.replace(float(Generated_signal[i]), -0.7)
 elif 184 > i >= 2:
 minimal_num = float(minimal[i])
 maximal_num = float(maximal[i])
 mean_num = float(mean[i])
 minimal_num = minimal_num + ((mean_num - minimal_num) * input_num)
 maximal_num = maximal_num - ((maximal_num - mean_num) * input_num)
 Generated_signal_number = ((mean_num + random.uniform(minimal_num, maximal_num)) / 2)
 Generated_signal = Generated_signal.replace(Generated_signal[index[i]],
Generated_signal_number)
 else:
 Generated_signal = Generated_signal.replace(Generated_signal[index[i]], 2.2)

 Generated_signal = Generated_signal.set_axis(list(range(start, end)))
 mean = mean.set_axis(list(range(start, end)))
 start = start + 186
 end = start + 186
 Generated_signals = Generated_signals._append(Generated_signal)
 mean_signal = mean_signal._append(mean)
for x in range(smooth):
 for i in range(3, (186 * n)):
 Generated_signals[i] = (Generated_signals[i] + Generated_signals[i + 1]) / 2

# ocena


dif = 0
for i in range(1, Generated_signals.shape[0]):
 dif = dif + (abs(Generated_signals[i] - mean_signal[i])) / mean_signal[i]
ocena = (dif / Generated_signals.shape[0]) * 100
print(
 f'-------------\n procentowa różnica pomiędzy wygenerowanym sygnałem \n a sygnałem
referencyjnym wynosi: {ocena}\n -------------------------')


#wykres wygenerowany sygnał / sygnał wzorcowy


plt.figure()
Generated_signals[4:730].plot(color='blue')
mean_signal[4:730].plot(color='red', linestyle='dashed')
plt.show()


#Zapis do pliku csv

Generated_signals.to_csv('generated_signal.csv')
```

Rezultaty i wnioski

Projekt spełnia założone funkcjonalności, ponieważ służy do generowania sygnałów EKG z możliwością ich modyfikacji i porównania z oryginalnymi sygnałami.

Zdefiniowane zostały opcje zmiany generowanego sygnału w celu dopasowania go do oryginalnego, takie jak korelacja i wygładzanie. Generowany sygnał jest tworzony na podstawie średnich wartości zebranej w każdej klasie, a następnie modyfikowany zgodnie z wybranymi opcjami. Możliwy jest również eksport wygenerowanego biosygnału w postaci pliku csv.

Rezultaty można zaobserwować na rysunku 1, gdzie przedstawiono przebieg wygenerowanego sygnału EKG (oznaczony kolorem niebieskim) wraz z sygnałem wzorcowym pochodzącym z bazy PhysioNet (oznaczony kolorem czerwonym).



Rys. 1. Zestawienie przebiegu wygenerowanego sygnału EKG wraz z sygnałem wzorcowym pochodzącym z bazy PhysioNet.

Podział zadań

Zadania podział	Osoba odpowiedzialna	Planowana data wykonania	Potwierdzenie wykonania przez PM (data wykonania)	Ocena realizacji przez PM (0 - niewykonane, 100% - wykonane w pełni)

Wybór i opisanie 1 artykułu z bazy IEEE Explore wg. wytycznych/ wybór rzeczywistego sygnału EKG	Lalewicz	28.03.2023	11.04.2023	100%
Przegląd oraz wybór bazy danych sygnałów, przygotowanie danych do algorytmu	Mąka	04.04.2023	11.04.2023	100%
Implementacja algorytmu symulacji sygnału EKG	Szwalec, Mąka, Lalewicz	18.04.2023	18.04.2023	100%
Testowanie algorytmu symulacji sygnału EKG oraz ocena wygenerowanego sygnału	Szwalec	25.04.2023	08.05.2023	100%