

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
Імені ІГОРЯ СІКОРСЬКОГО»
ФАКУЛЬТЕТ БІОМЕДИЧНОЇ ІНЖЕНЕРІЇ
(повна назва інституту/факультету)

Кафедра БІОМЕДИЧНОЇ ІНЖЕНЕРІЇ
(повна назва кафедри)

**Застосування методів частотного аналізу та машинного
навчання для оцінки ефективності та персоніфікації
реабілітаційних методик у хворих**

Аспірант: Федорюк Дмитро Олександрович

Науковий керівник: проф., д.м.н. Худецький І.Ю.

Мета досліджень

Метою роботи є створення системи раннього діагностування аритмії в реальному часі на портативних пристроях для фітнес-трекінгу.

Об'єктом дослідження є програмна система раннього діагностування аритмії.

Предметом дослідження є фізичні характеристики ЕКГ та програмні моделі, створені в середовищі Python 3.6.

Робота присвячена створенню достатньо точної системи раннього діагностування аритмії, що здатна працювати в реальному часі. Розроблену систему можливо застосовувати у пристроях для фітнес-трекінгу, що є значною перевагою перед аналогами.

Ключові слова: аритмія, нейронна мережа, конволюційна нейронна мережа, рекурентна нейронна мережа, фітнес-трекер.

Наукове обґрунтування

Серцева аритмія - група умов, при яких серцебиття є нерегулярним, занадто швидким або занадто повільним. Частота серцевих скорочень, яка надмірно швидка - понад 100 ударів на хвилину у дорослих - називається тахікардією, а пульс, який є надто повільним - нижче 60 ударів на хвилину, - називається брадикардія. Багато видів аритмії не мають симптомів. При наявності симптомів вони можуть включати серцебиття або відчуття паузи між серцебиттями. У більш серйозних випадках може бути легенева голова, виснаження, задишка або біль у грудях. Хоча більшість типів аритмій не є серйозними, деякі схильні до таких ускладнень, як інсульт або серцева недостатність. Інші можуть призвести до зупинки серця. Тому важливою є можливість раннього діагностування аритмії.

Технічні можливості фітнес-трекерів дозволили реєструвати ЕКГ на зап'ясті людини в реальному часі. Це дало можливість більш точно аналізувати стан здоров'я носія трекеру, зокрема, розпізнавати ранні симптоми аритмії. Оскільки внаслідок фізичної активності сигнал ЕКГ піддається зашумленню сигналом з м'язів, то стає важко визначити зубці кардіограми класичними методами. З цією задачею чудово справляється рекурентна нейронна мережа, а саме її тип – двонаправлена LSTM, що і було показано у цій роботі.

АЛГОРИТМИ КЛАСИФІКАЦІЇ ЕЛЕКТРОКАРДІОГРАМ

Ритми, які можливо класифікувати на ЕКГ, - це нормальний (синусовий) ритм, шум / артефакт (чи інше неідентифікована аритмія), шлуночкова тахікардія, фібриляція передсердь, бігемінія, і передчасне скорочення шлуночків. Приклад кожного класу показано на рисунках 1.2 - 1.6; для чіткості побудовано лише один канал.

На рисунку 1.2 зображено нормальний ритм:

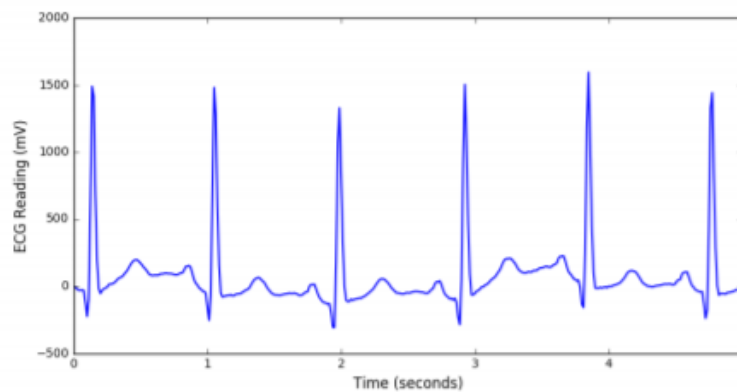


Рисунок 1.2 – Нормальний ритм

На рисунку 1.3 зображено артефакт:

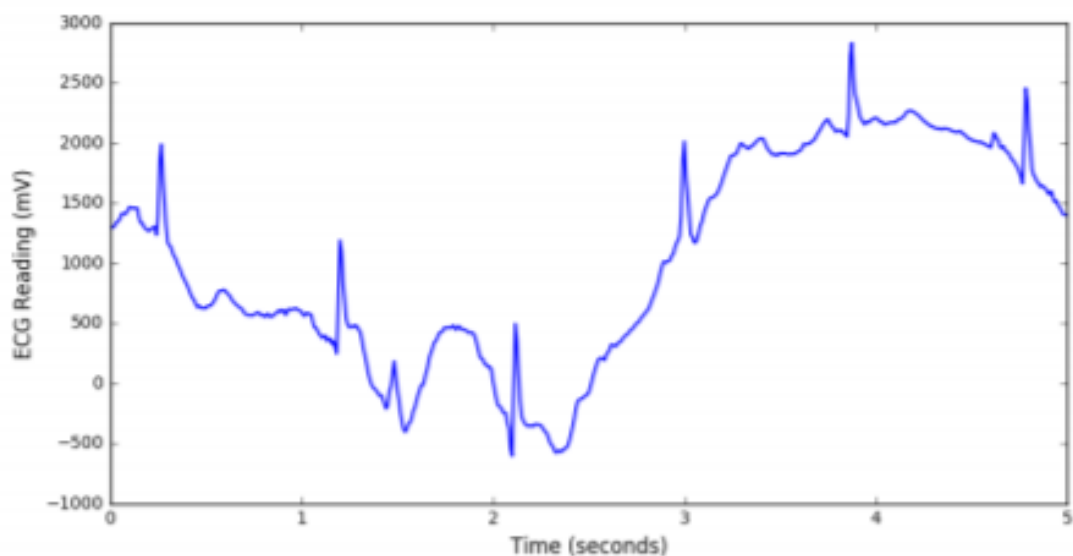


Рисунок 1.3 – Артефакт серцебиття

На рисунку 1.4 наведено вентрикулярну тахікардію:

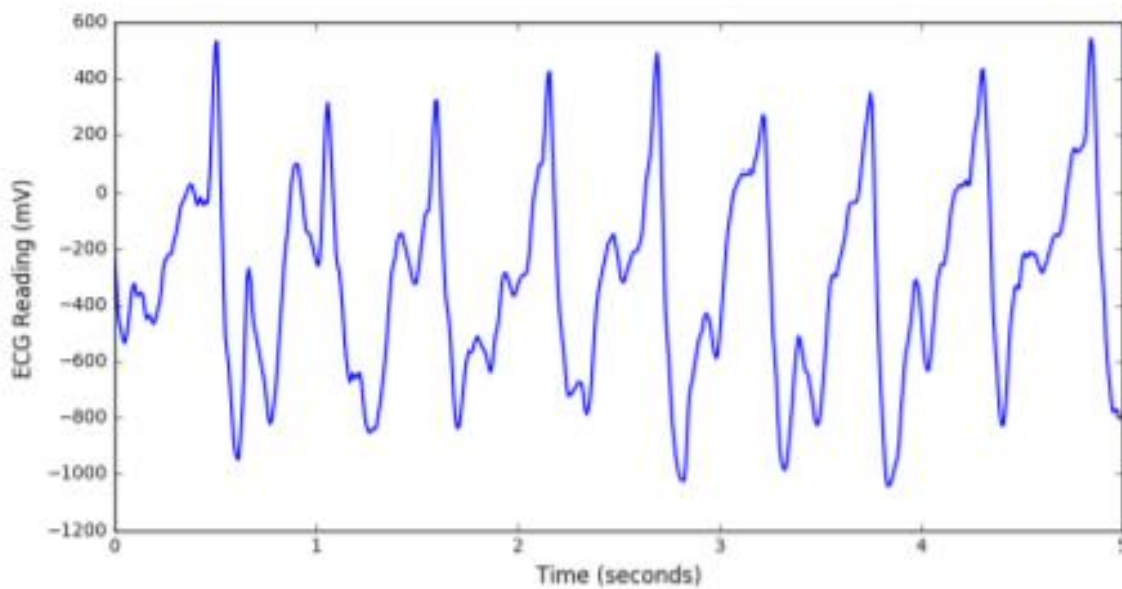


Рисунок 1.4 – Вентрикулярна тахікардія

На рисунку 1.5 наведено фібриляцію передсердь:

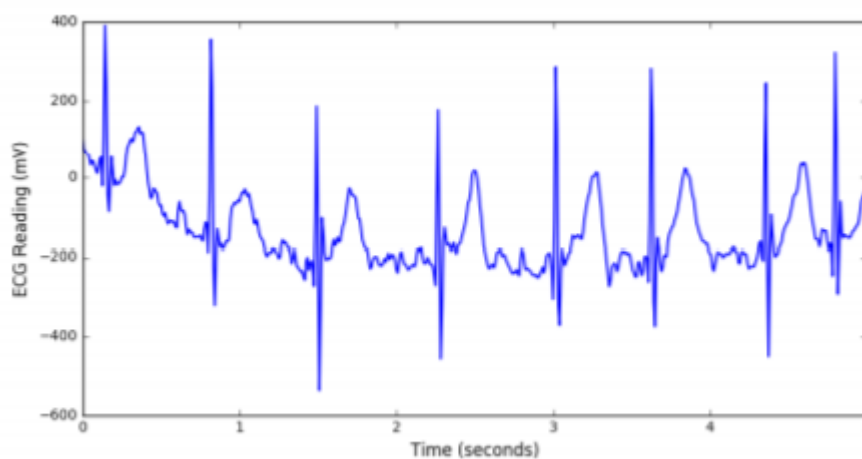


Рисунок 1.5 – Фібриляція передсердь

На рисунку 1.6 наведено бігемінію шлуночків

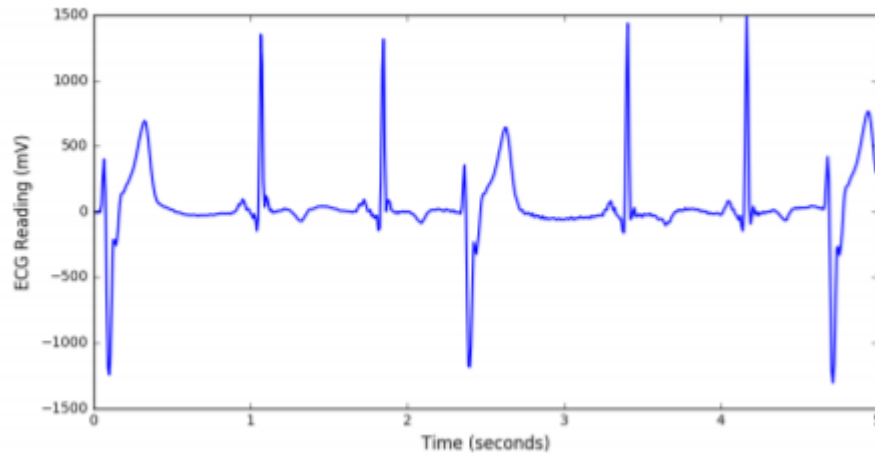


Рисунок 1.6 – Бігемінія шлуночків

На ЕКГ порушення ритму серця відображається патологією зубців та інтервалів. Для того, щоб програмно ідентифікувати ранні симптоми аритмії, необхідно використати алгоритм, що здатний класифікувати сигнали ЕКГ.

Для класифікації ЕКГ використовується:

1. Аналіз вейвлетів на основі алгоритму Маллата.
2. Алгоритм подвійного порогу для визначення втрачених ударів.
3. Методи опорних векторів, такі як комбінація нечіткої теорії для уточнення класифікації SVM у поєднанні з ансамблем класифікаторів, генетичними алгоритмами у поєднанні з обмеженими нечіткими SVM та найменшими квадратами SVM .
4. Штучні нейронні мережі на основі архітектури ANN, яка найчастіше використовується для класифікації аритмії, - багатошарові перцептрони (MLP) та ймовірнісні нейронні мережі (PNN).
5. Лінійні дискримінанти – це класифікатори, які більш широко використовуються в методах, що слідують за схемою, запропонованою де Шазалом та ін. і рекомендований ААМІ.
6. Резервуарні обчислення з логістичною регресією (RC).

Висновки щодо класифікації ЕКГ

1. Архітектури нейронних мереж доволі легко оптимізувати. Методи глибокого навчання показують доволі оптимістичні результати в задачах класифікації електрокардіограми. Вони чудово працюють з зашумленими сигналами, не є необхідною попередня обробка сигналу. Також, на сьогодні, нейронні мережі доволі легко розобляти, впроваджувати в програмні застосунки та підтримувати.

2. Для розробки нейронних мереж доцільно застосувати Keras. Keras – це високорівневий API нейронних мереж, написаний на Python і здатний працювати на вершині TensorFlow, CNTK або Theano. Він був розроблений з урахуванням можливості швидкого експерименту, що дає можливість перейти від ідеї до результату з мінімальною затримкою.

3. Для класифікації ЕКГ обрано рекурентні нейронні мережі – це клас штучних нейронних мереж, у якому з'єднання між вузлами утворюють орієнтований цикл. Це створює внутрішній стан мережі, що дозволяє їй проявляти динамічну поведінку в часі. На відміну від нейронних мереж прямого поширення, РНМ можуть використовувати свою внутрішню пам'ять для обробки довільних послідовностей входів.

МАТЕРІАЛИ ТА МЕТОДИ

1. База даних фізіологічних сигналів WFDB

База **PhysioBank** пропонує набір інструментів для розвідки та експорту даних, але відносно повільний дисплей сигналу (він генерує растрові зображення на сервері PhysioNet, які можна переглядати навіть у дуже старих браузерях) у порівнянні з LightWAVE (який використовує Javascript і SVG у будь-якому сучасному браузері для швидка та якісна векторна графіка).

Якщо необхідним є завантаження всіх файлів в будь-яку з баз даних PhysioBank без вибірки кожного окремо, необхідно скористатись утилітою для пакетних HTTP-передач, таких як `wget`, доступна тут у вихідній формі для всіх версій UNIX і як попередньо скомпільована двійкова для MS Windows.

У цьому контексті база даних - це просто збірка записів (записів), доступних як набір файлів. На відміну від типових реляційних баз даних, бази даних PhysioBank складаються з відносно невеликої кількості (від десятків до тисяч) записів, кожен може бути досить великим (в деяких базах даних PhysioBank розмір запису може бути гігабайт або більше, хоча типові розміри записів кілька Mb).

2. Python та NumPy

Python – це інтерпретована мова високого рівня програмування для загального використання. Вона забезпечує конструкції, які дозволяють чітко програмування на малих і великих масштабах.

NumPy – це фундаментальний пакет для наукових обчислень з Python. Вона містить, серед іншого:

- потужний об'єкт N-розмірного масиву
- складні (трансляція) функції
- інструменти для інтеграції C / C++ і Fortran коду
- корисна лінійна алгебра, перетворення Фур'є і можливості випадкових чисел

Крім очевидних наукових застосувань, NumPy також може бути використаний як ефективний багатовимірний контейнер загальних даних. Можна визначити довільні типи даних. Це дає змогу без проблем і швидкої інтеграції NumPy з різноманітними базами даних.

Бібліотека Python – Keras використовується для розробки нейронних мереж. Keras – це високорівневий API нейронних мереж, написаний на Python і здатний працювати на вершині TensorFlow, CNTK або Theano. Він був розроблений з урахуванням можливості швидкого експерименту, що дає можливість перейти від ідеї до результату з мінімальною затримкою.

3. Рекурентні нейронні мережі (LSTM)

Рекурентні нейронні мережі – це клас штучних нейронних мереж, у якому з'єднання між вузлами утворюють орієнтований цикл. Це створює внутрішній стан мережі, що дозволяє їй проявляти динамічну поведінку в часі. На відміну від нейронних мереж прямого поширення, РНМ можуть використовувати свою внутрішню пам'ять для обробки довільних послідовностей входів. Це робить їх застосовними до таких задач, як розпізнавання несегментованого неперервного рукописного тексту та розпізнавання мовлення.

Результат навчання РНМ – функція, яка максимізує якість класифікації на відкладеній вибірці. Мірою якості зазвичай розглядають точність – відсоток правильно класифікованих об'єктів:

$$Acc = \frac{1}{K} \sum_i^K [f(X_i = y_i)]$$

де об'єкти вибірки беруться з відкладеної вибірки розміру K .

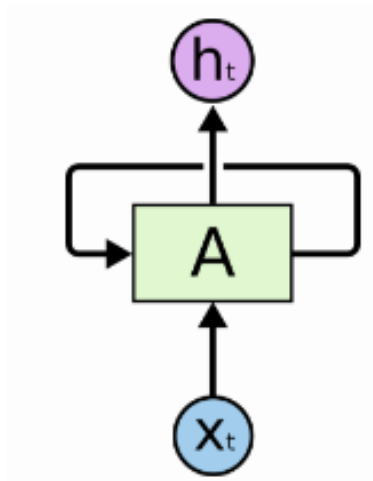


Рисунок 2.1 – Діаграма РНМ

Розгорнута структура РНМ

1. Розгорнута структура РНМ

РНМ можна репрезентувати як безліч копій однієї і тієї ж мережі, при цьому кожна копія передає повідомлення наступній копії, як зображено на рисунку 2.2

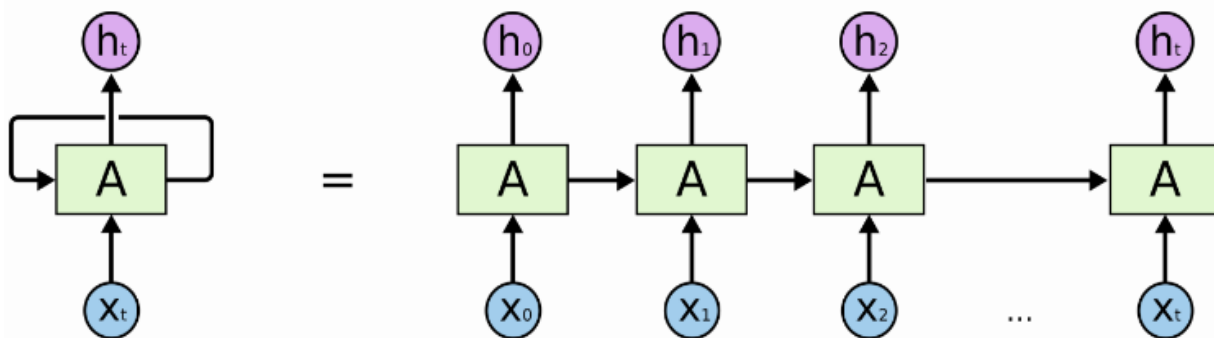


Рисунок 2.2 – Розгорнута структура РНМ

Як і більшість РНМ, **мережа LSTM** є універсальною в тому сенсі, що за достатньої кількості вузлів мережі вона може обчислювати будь-що, що може обчислювати звичайний комп'ютер, за умови, що вона має належну матрицю вагових коефіцієнтів, що може розглядатися як її програма.

На відміну від традиційних РНМ, мережа LSTM добре підходить для навчання з досвіду з метою класифікації, обробки або передбачення часових рядів в умовах, коли між важливими подіями існують часові затримки невідомої тривалості. Відносна нечутливість до довжини прогалин дає LSTM перевагу в численних застосуваннях над альтернативними РНМ, прихованими марковськими моделями та іншими методами навчання послідовностей.

Всі рекурентні нейронні мережі мають форму ланцюга повторюваних модулів нейронної мережі.

2. Розрахунок коефіцієнтів LSTM

На рисунку 2.5 подані рівняння розрахунку коефіцієнтів LSTM:

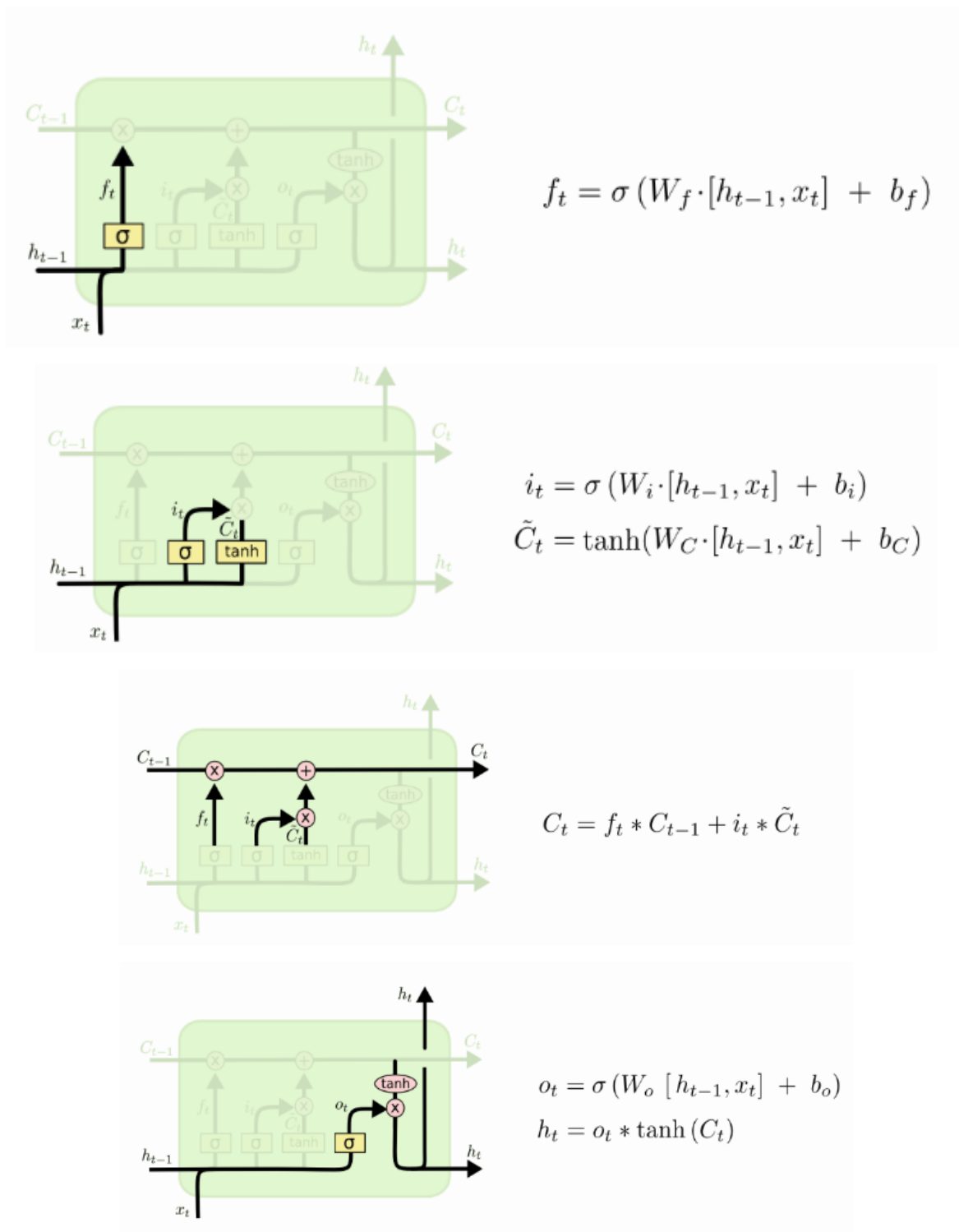


Рисунок 2.5 – Рівняння розрахунку коефіцієнтів LSTM

РОЗРОБКА СИСТЕМИ РАНЬОГО ДІАГНОСТУВАННЯ АРИТМІЇ

На рисунку 3.1 наведено запропоновану функціональну схему системи раннього діагностування аритмії.

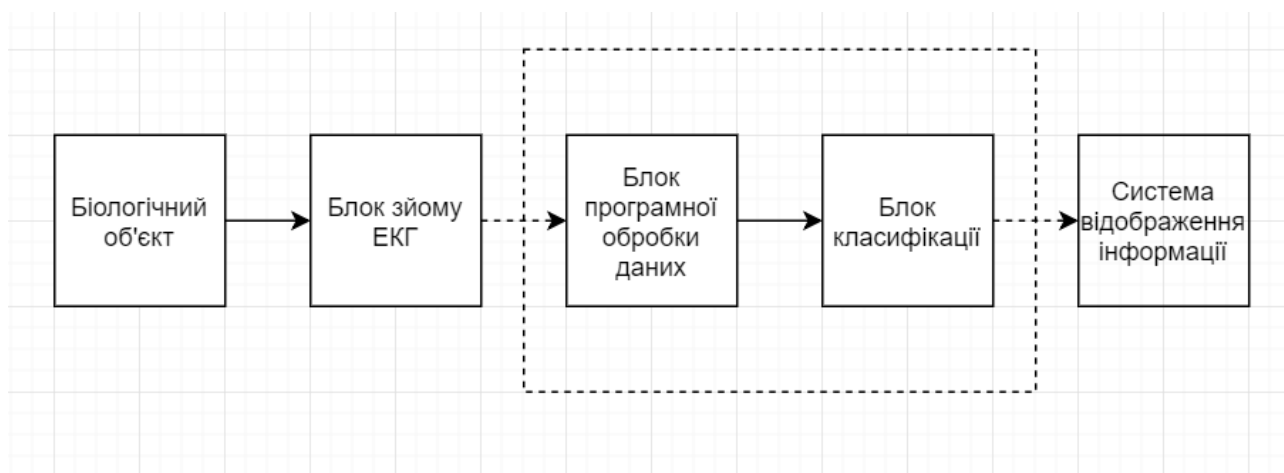


Рисунок 3.1 – Загальна функціональна схема

Біологічним об'єктом виступає людина, що носить фітнес-трекер. Блок зйому ЕКГ розташований у фітнес-трекері. Наразі є можливим зняття ЕКГ із зап'ястя, що широко застосовується в сучасних системах фітнес-трекінгу. Отриманий ЕКГ сигнал передається бездротовим зв'язком на хмару, де відбувається попередня обробка даних та класифікація аритмії. Тим же бездротовим зв'язком на систему відображення інформації (дисплей фітнес трекеру) передається попередження про загрозу аритмії у разі, якщо така є.

В даній роботі розроблено блоки програмної обробки сигналу та класифікації. В якості обробки сигналу виступає скрипт наведений в додатку 1 та описаний в підглаві 3.1.1. В якості класифікатора виступає нейронна мережа, що розпізнає зубці ЕКГ та класифікує їх як патологічні чи ні. Архітектура потенційно розширювана та може бути перетренована не лише на виявлення аритмії (рисунок 3.3).

Блок класифікації та розпізнавання зубців ЕКГ

На рисунку 3.1 наведено Блок класифікації та розпізнавання зубців ЕКГ



Рисунок 3.3 – Блок класифікації

1. Було здійснено спробу зменшити розмірність вхідних даних за рахунок подачі на вхід системи лише зубців ЕКГ. Для цього було створено підсистему розпізнавання зубців ЕКГ. Точність більше 90% є прийнятною. Оскільки система має працювати в реальному часі, архітектура мережі обиралася простою.

2. Було перевірено конволюційні та рекурентні нейронні мережі з різними конфігураціями прихованих шарів та оптимізаторів. Модель, що показала себе найкраще було серіалізовано. Розробка приладу не велася, адже це вже є роботою розробника, що інтегрує нейронну мережу в свій сервіс.

1. Запис ЕКГ для формування тренувального та тестувальних наборів

Записи 32 сигналів ЕКГ для формування тренувального та тестувальних наборів було отримано з бази фізіологічних сигналів PhysioBank. Вивантаження сигналів проводилося за допомогою Native Python WFDB package.

Доступ до файлів формату .dat із сигналами та .qlc з анотаціями здійснювався по URL physionet.org/physiobank/database/qtdb/. Сигнали у файлах розмічені анотаціями по патерну (p) (N) t).

На рисунку 3.4 зображено сигнал ЕКГ, розмічений анотаціями:

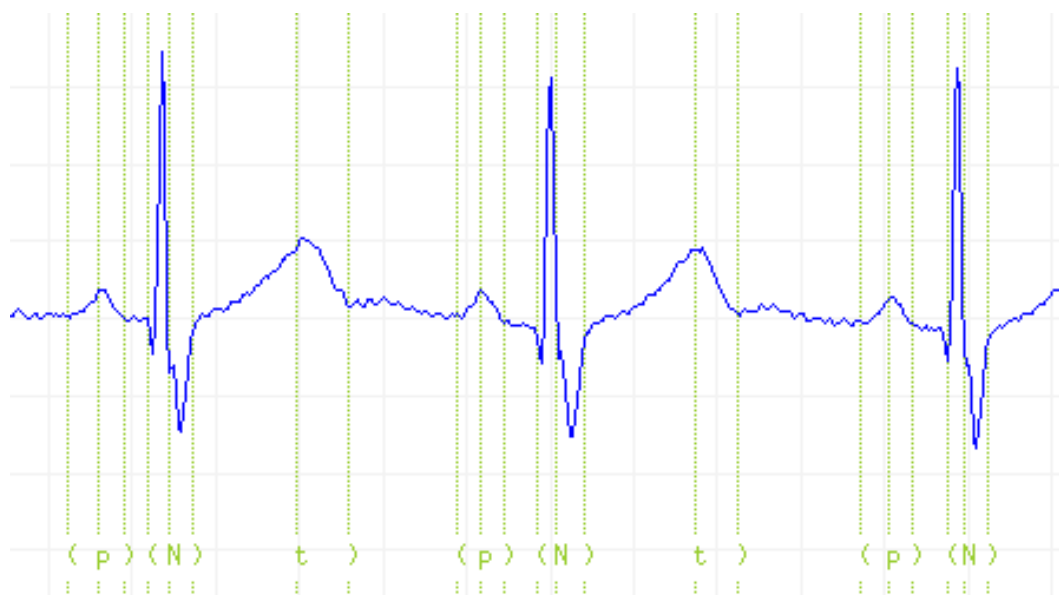


Рисунок 3.4 – Сигнал ЕКГ, розмічений анотаціями

Як видно на рисунку 3.5 на отриманих сигналах ЕКГ не всі ділянки розмічені анотаціями.

2. Очищення сигналу та формування наборів даних

На рисунку 3.5 зображено ділянки сигналів, що не були розмічені анотаціями, тому їх було видалено.

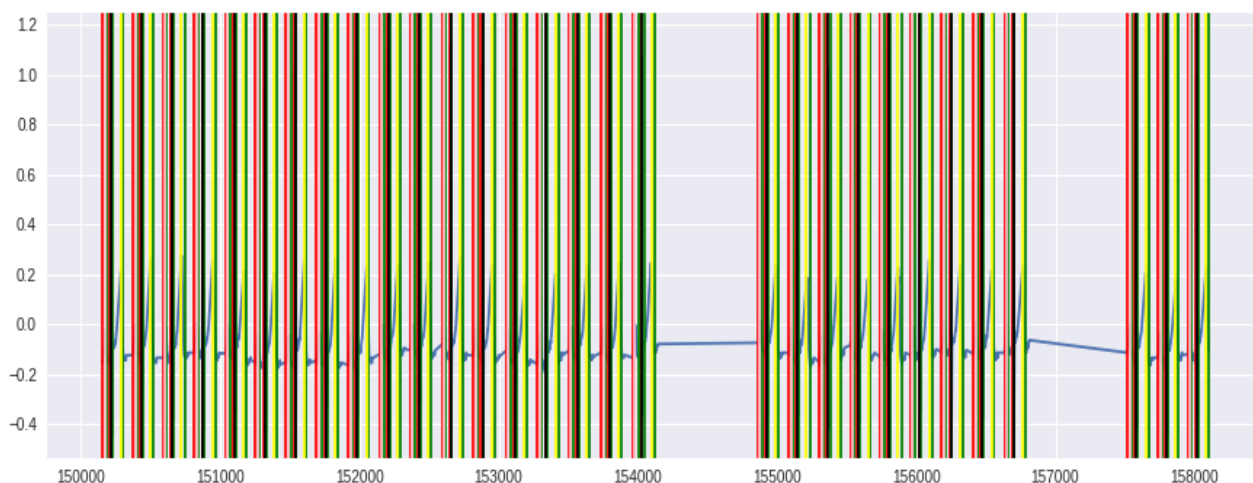


Рисунок 3.5 – Сигнал, що позбавлений нерозмічених областей

Сигнали всіх записів ЕКГ було конкатеновано та нарізано з перетином вікном шириною в 1000 та з кроком в 10 відліків.

На вхід мережі LSTM необхідно подавати тривимірні дані у вигляді $(batch_size, timesteps, input_dim)$, тому дані було приведені до такого вигляду: $X - (17531, 1000, 1)$ та $y - (17531, 1000, 4)$ відповідно. Вектор y було закодовано методом one-hot encoding. Весь набір даних стандартизовано.

Всього 20% записаних даних було залишено на тестовий набір. Вибір точки розділу набору даних на тренувальний та тестовий обирається випадковим чином.

3. Побудова LSTM моделі та підбір параметрів архітектури

Модель було побудовано у відкритій неймережевій бібліотеці Keras на Python 3. Модель складається: з одного шару двонаправленої LSTM та одного `time_distributed` шару. Вихід `time_distributed` подається на вхід `softmax`-функції. В якості функції втрат було обрано бінарну крос-ентропію, оптимізатор – RMSprop.

Тренування моделі було здійснено на 100 епохах, ваги, що давали найбільшу точність збережено у файлі. На рисунку 3.7 зображено процес тренування моделі на чистому сигналі ЕКГ:

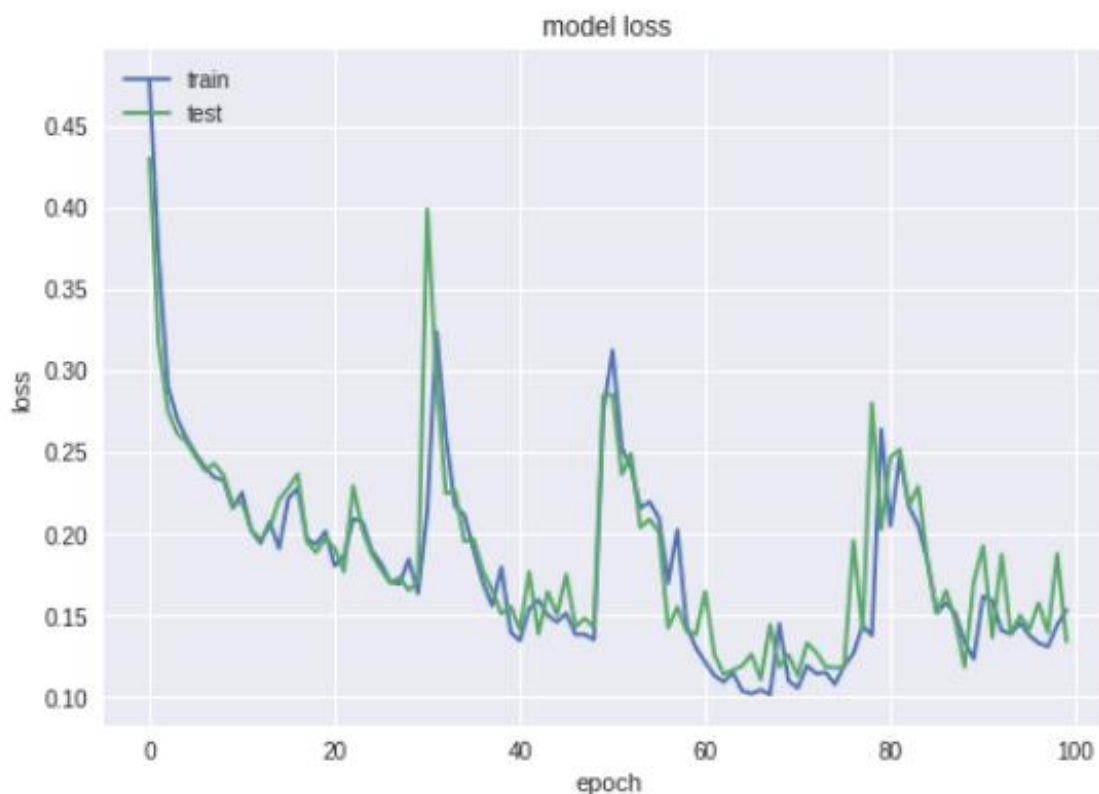


Рисунок 3.7 – Графік тренування моделі. Залежність значення функції втрат від номеру епохи.

На рисунку 3.8 зображено залежність точності (подано у відсотках) від номеру епохи:

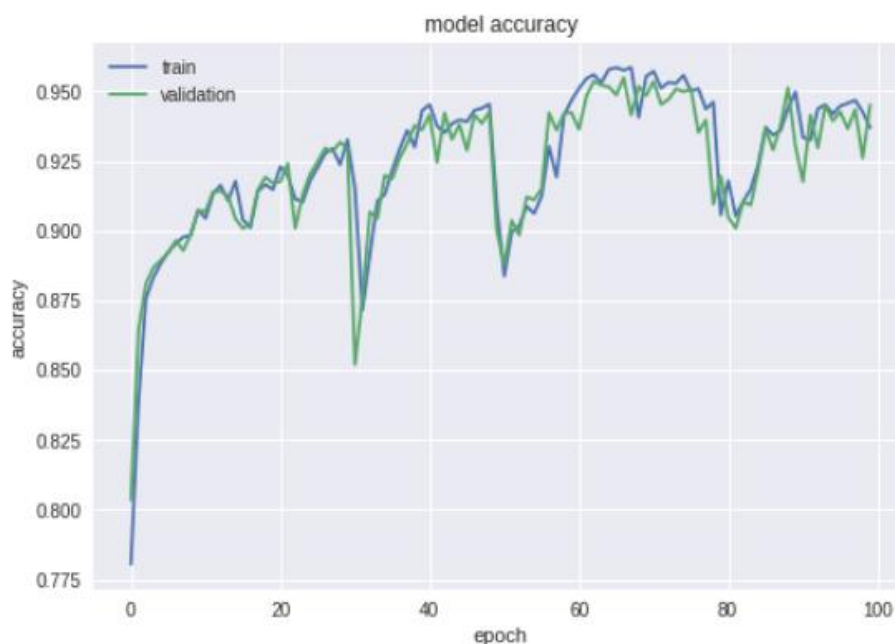


Рисунок 3.8 – Графік тренування моделі. Залежність точності (подано у відсотках) від номеру епохи

Найкращий результат – **95,26%** точності, значення loss-функції – 0,11.

Для тренування моделі на зашумленому сигналі, сигнал ЕКГ було зашумлено адитивно сигналом ЕМГ. На рисунку 3.10 зображено сигнал ЕКГ, зашумлений сигналом ЕМГ.

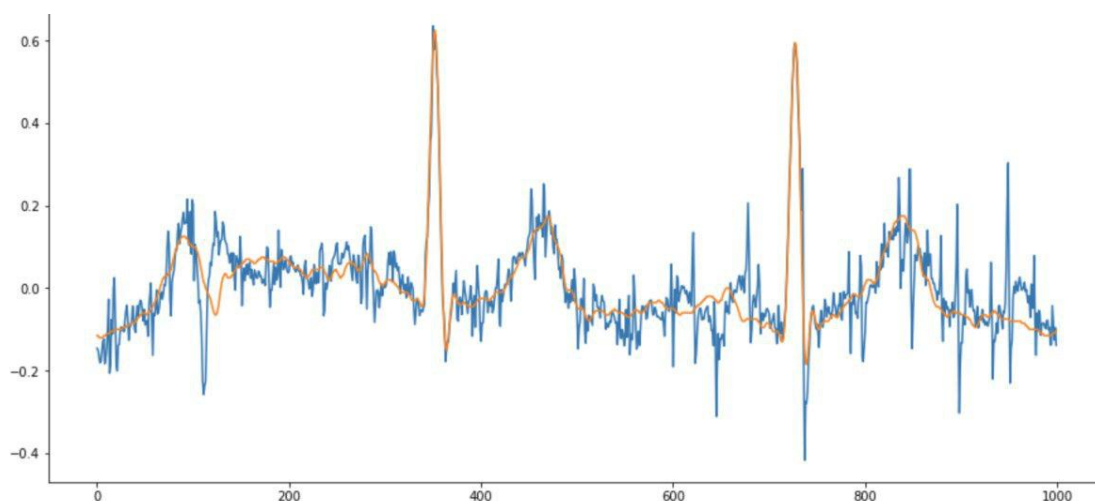


Рисунок 3.10 – сигнал ЕКГ, зашумлений сигналом ЕМГ

Модель було навчено на зашумлених даних та отримано точність 95,5%.

4. Побудова CNN та підбір архітектури мережі

Конволюційні нейронні мережі застосовуються в задачах обробки сигналів. Було перевірено базову модель з 7 конволюційними шарами. При проектуванні архітектури мережі застосовувалася batch-нормалізація, що може вплинути на час роботи в реальному часі, оскільки вимагає попередньої обробки ЕКГ сигналу. На рисунку 3.13 наведено CNN модель у Keras.

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 1000, 1)	0
conv1d_1 (Conv1D)	(None, 1000, 256)	1024
batch_normalization_1 (Batch Normalization)	(None, 1000, 256)	4000
max_pooling1d_1 (MaxPooling1D)	(None, 500, 256)	0
conv1d_2 (Conv1D)	(None, 500, 128)	98432
batch_normalization_2 (Batch Normalization)	(None, 500, 128)	2000
max_pooling1d_2 (MaxPooling1D)	(None, 250, 128)	0
conv1d_3 (Conv1D)	(None, 250, 128)	49280
batch_normalization_3 (Batch Normalization)	(None, 250, 128)	1000
max_pooling1d_3 (MaxPooling1D)	(None, 125, 128)	0
conv1d_4 (Conv1D)	(None, 125, 128)	49280
batch_normalization_4 (Batch Normalization)	(None, 125, 128)	500
up_sampling1d_1 (UpSampling1D)	(None, 250, 128)	0
conv1d_5 (Conv1D)	(None, 250, 128)	49280
batch_normalization_5 (Batch Normalization)	(None, 250, 128)	1000
up_sampling1d_2 (UpSampling1D)	(None, 500, 128)	0
conv1d_6 (Conv1D)	(None, 500, 256)	98560
batch_normalization_6 (Batch Normalization)	(None, 500, 256)	2000
up_sampling1d_3 (UpSampling1D)	(None, 1000, 256)	0
conv1d_7 (Conv1D)	(None, 1000, 4)	3076

Total params: 359,432		
Trainable params: 354,102		
Non-trainable params: 5,250		

Рисунок 3.13 – Скомпільована у Keras модель

Розробка підсистеми класифікації аритмії

1. Розробка базових моделей

Розроблено дві базові моделі: одношарова, односпрямована LSTM, і згортова нейронна мережа без залишкових з'єднань. Потім було здійснено тренування моделей з трьох категорій (Лише LSTM, залишкові мережі та комбіновані мережі LSTM-CNN) загалом 6 моделей, що не включають базові.

В якості базових моделей було натреновано 4 блоки (кожен блок, що містить послідовність згортки, maxpool, нормалізацію батчів та відсівні шари) CNN з залишковими з'єднаннями maxpool, а також одношарову, односпрямовану LSTM з прихованим шаром на 100 (розмірність прихованого / вихідного простору). В таблиці 3.2 зазначено точності базових моделей

Таблиця 3.2 – Точність базових моделей

Модель	Точність на тренувальному наборі	Точність на валідаційному наборі	Точність на тестовому наборі
Однонаправлена LSTM	66.9%	63.2%	62.2%
Стекова двонаправлена LSTM (2 шари)	83%	82.1%	81%
Стекова однонаправлена LSTM (5 шарів)	80.1%	79.9%	80.4%
Глибока CNN з залишками (16 блоків)	85.1%	84.6%	79.9%
Комбінована однонаправлена LSTM-CNN	84.6%	81.3%	80%
Комбінована двонаправлена LSTM-CNN	93.2%	90%	89.9%

2. Загальні точність всіх LSTM-CNN моделей

Всі мережі тренуються за допомогою мініатюри розміром 64 і 100 епох навчання. Мінімізовано категоричну втрату крос-ентропії у всіх випадках, і використано оптимізатор Адама, розширення класичного оптимального стохастичного градієнтного спуску.

У всіх мережах LSTM використовується прихована розмірність 100, а також невеликий параметр відсіву ($p = 0,1$) для всіх воріт. Усі відсівні шари в моделях CNN використовуються з $p = 0,5$ для максимізації регуляризації.

Таблиця 3.3 – Експериментальні результати.

Клас ритму	BDLSTM	Залишки	LSTM-CNN
Синусовий	0.820	0.690	0.702
Артефакти (шум)	0.843	0.802	0.820
Вентрикулярна тахікардія	0.221	0.07	0.432
Фібриляція передсердь	0.830	0.784	0.777
Бігемінія	0.679	0.110	0.565
Передчасне скорочення шлуночків	0.780	0.854	0.765
Загальна	0.80	0.714	0.750

Таблиця 3.4 – F1 порівняння за класами для тестового набору,

Клас	BDLSTM	LSTM-CNN
S	0.83	0.78
A/N	0.845	0.82
VT	0.213	0.396
AF	0.674	0.712
B	0.67	0.555
PVC	0.74	0.688

3. Розробка фінальної моделі

На рисунку 3.21 зображено пропоновані моделі

Layers	Types	Activation function	Output Shapes	Kernel Size	No. of Filters	Stride	No. of trainable parameters	Scheme A	Scheme B
0	Input	-	1000 x 1	-	-	-	0	-	-
1	1D Full convolution without bias	ReLU	1019 x 3	20 x 1	3	1	60	-	-
2	1D Max-pooling	-	509 x 3	2 x 1	3	2	0	-	-
3	1D Full convolution without bias	ReLU	518 x 6	10 x 1	6	1	180	-	-
4	1D Max-pooling	-	259 x 6	2 x 1	6	2	0	-	-
5	1D Full convolution without bias	ReLU	263 x 6	5 x 1	6	1	180	-	-
6	1D Max-pooling	-	131 x 6	2 x 1	6	2	0	-	-
7	LSTM		20	-	-	-	2160	Recurrent dropout (20%) Dropout (20%)	Recurrent dropout (20%)
8	Fully-connected	ReLU	20	-	-	-	420	-	Dropout (20%)
9	Fully-connected	ReLU	10	-	-	-	210	-	Dropout (20%)
10	Fully-connected	Softmax	5	-	-	-	55	-	-
Total							3265		

Рисунок 3.21 – Пропоновані мережеві моделі

Пропонована модель складається з трьох згорткових шарів зі схемою 1. Кожна операція згортки здійснюється шляхом зміщення ядра на вхідний вектор одного зразка в той час, коли накладаються матриці множуться і підсумовуються. Під час тренувального процесу ваги ядра k постійно налаштовуються мережею для отримання значущої просторової інформації, наявної в даних.

Перша схема (без dropout) – це тренувана мережа без будь-якого відсічення та введена в якості посилення для вивчення ефекту між регулярною мережею та випускною мережею. На рисунку 3.22 наведено критерії точності тренування та перевірки на кожну з цієї схемою.

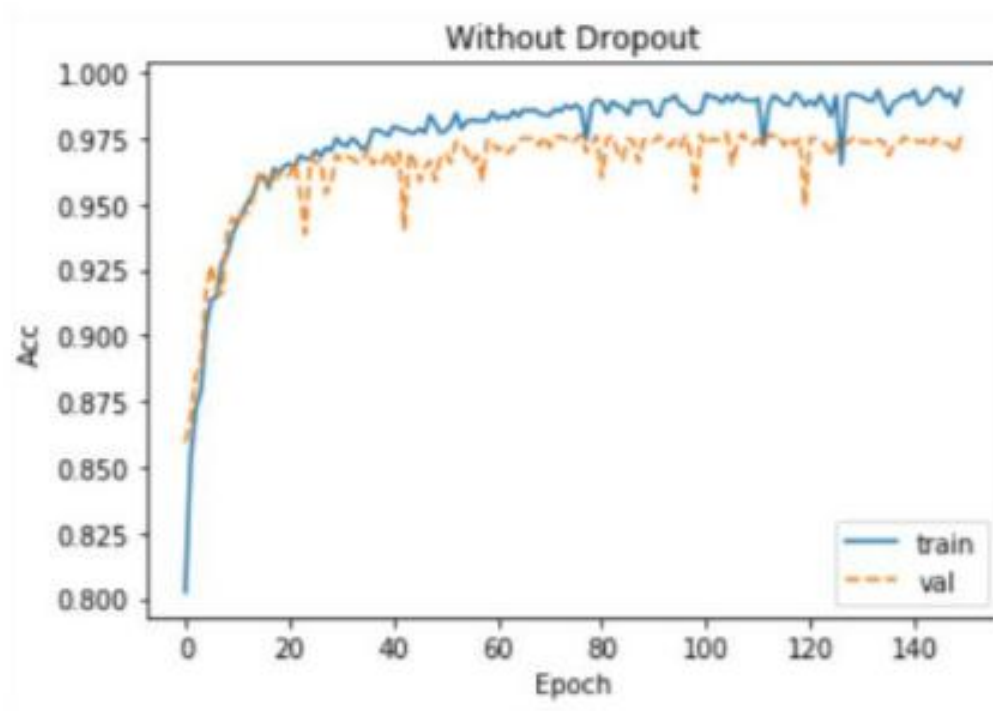
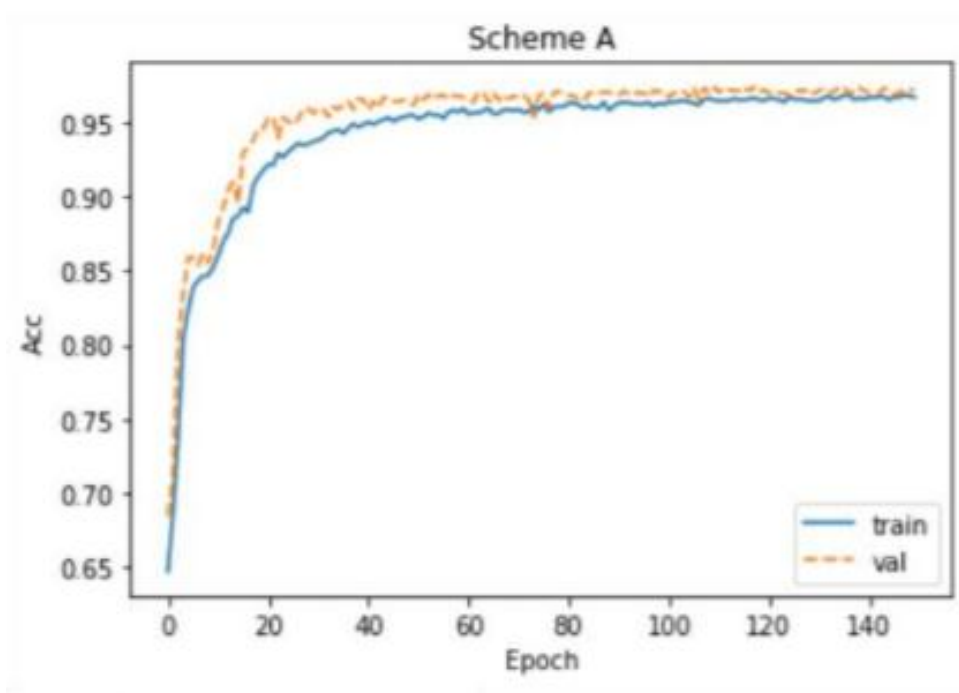


Рисунок 3.22 – Тренування без dropout

Інші схеми (Схема А і Схема Б)

Для Схеми А, скинуто 20% рекурентних і вхідних з'єднань рівня LSTM, а в Схемі В, 20% повторюваних з'єднань LSTM, а потім 20% з двох повністю підключених шарів випадають. Критерії точності тренування та перевірки на кожну з цих схем представлено на рисунках 3.23 - 3.24.

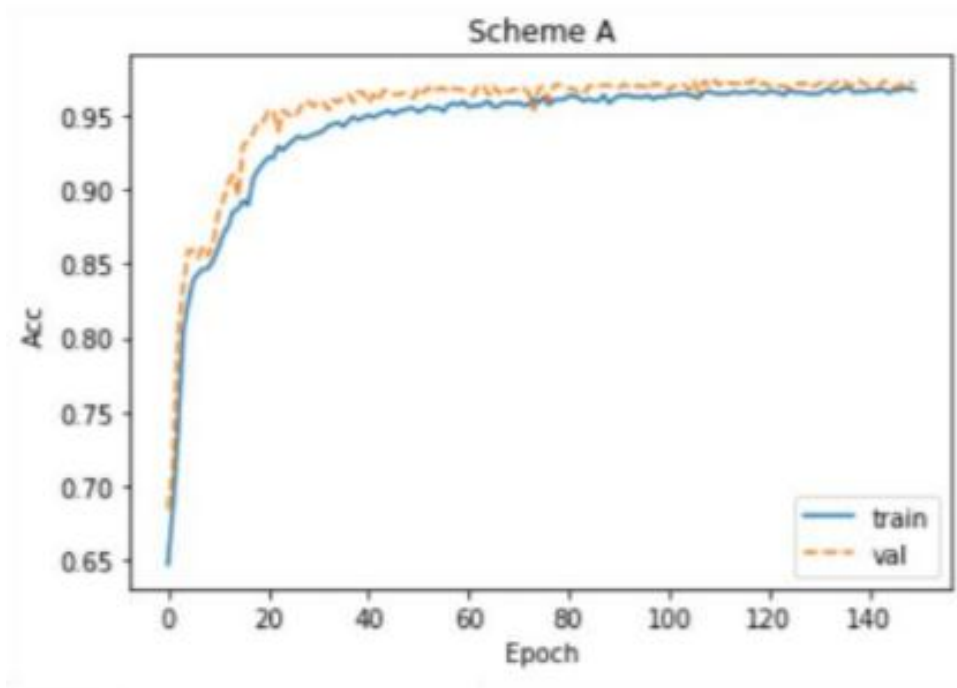


Рисунок 3.23 – Тренування схеми А

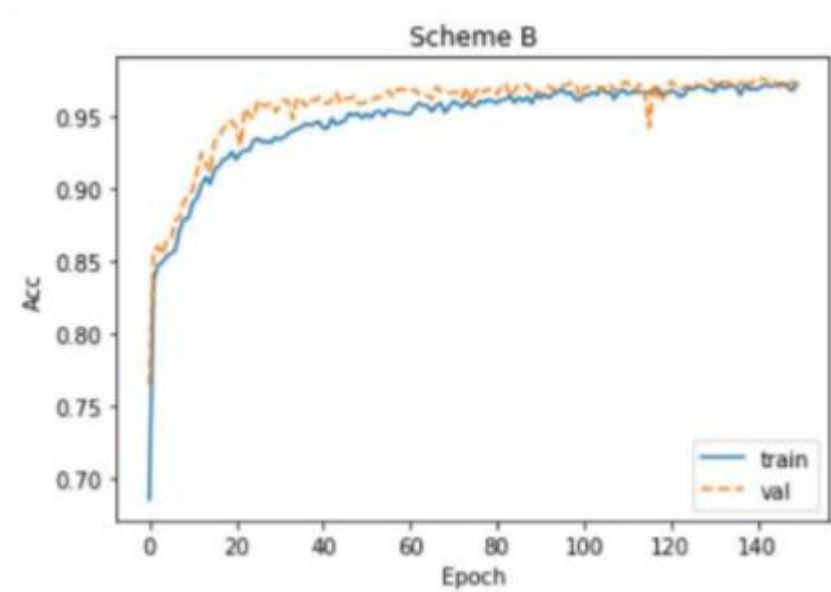


Рисунок 3.24 – Тренування схеми Б

4. Загальні показники перехресної валідації

Загальні показники перехресної валідації для всіх трьох схем узагальнені у Таблиці 3.5. Найвища точність, чутливість, специфічність 98.42%, 98.07%, 98.76% відповідно.

Таблиця 3.5 – Метрики точності, чутливості та специфічності для схем

Схема	Точність	Чутливість	Специфічність
Без dropout	95.6	92	96.2
А	96.3	93.2	92.1
В	98.1	94.1	94.1

Висновки щодо фінальної моделі

1. Було створено базові моделі для порівняння точності класифікації аритмії, після чого було розроблено більш точну архітектуру. Базові моделі перенавчалися, що створювало проблему в класифікації на тестовій вибірці. Для того, щоб подолати проблему перенавчання було застосовано техніки dropout та регуляризації.

2. Розроблену модель серіалізовано. Найкраще показала себе модель схеми Б. Розробка велася на Keras, серіалізована модель може бути застосована для запуску на хмарі.

ВИСНОВОКИ

1. З розвитком систем фітнес-трекінгу з'явилася можливість використання програмних систем раннього діагностування серцевих хвороб, зокрема, аритмії. Вихід в Інтернет даних пристроїв дозволяє аналізувати стан людини за допомогою хмарних обчислень.

2. Глибинне навчання дозволяє якісно вирішити проблему класифікації часових послідовностей, що також було продемонстровано в даній роботі. Після порівняння існуючих алгоритмів класифікації аритмії, було виявлено, що нейронні мережі найбільш точно вирішують задачу.

3. Розроблена система раннього діагностування дозволить попереджати ускладнення аритмії та своєчасно сповіщувати носія фітнес-трекеру про небезпеку. Модель працює в реальному часі, здатна до розширення та інтеграції в програмні застосунки.

4. Можливість розширити систему робить її гнучкою. Перетренування моделі на виявлення інших серцевих хвороб займає мало часу та зусиль зі сторони розробника.

5. Точність в 98% є прийнятною. Сповіщення про загрозу враховує дану похибку.

Публікації:

1. Худецький І. Ю., Антонова-Рафі Ю. В., Федорюк Д. О., «Метрологічні аспекти системи підтримки прийняття рішень при протезуванні» Дев'ятнадцята міжнародна науково-технічна конференція «ВИМІРЮВАЛЬНА ТА ОБЧИСЛЮВАЛЬНА ТЕХНІКА В ТЕХНОЛОГІЧНИХ ПРОЦЕСАХ» (ВОТТП-19, 2019). – Одеса. — 2019, — С.16-17.