ІНФОКОМУНІКАЦІЙНІ ТЕХНОЛОГІЇ ТА ЕЛЕКТРОННА ІНЖЕНЕРІЯ
INFORMATION AND COMMUNICATION TECHNOLOGIES, ELECTRONIC ENGINEERING

№ 4 (2), 2024

https://doi.org/10.23939/ictee2024.02.162

КОНСТРУЮВАННЯ ОЗНАК ДЛЯ ЗАСТОСУВАННЯ НАВЧАННЯ МАШИН ПРИ ОБРОБЦІ КЛІНІЧНИХ ДАНИХ

О. Кирсанов [ORCID: 0009-0001-5987-4728] , **С. Кривенко** [ORCID: 0000-0002-5244-1276]

Харківський національний університет радіоелектроніки, пр. Науки, 14, м. Харків, 61166, Україна

Відповідальний за рукопис: Олександр Кирсанов (e-mail: oleksandr.kyrsanov@nure.ua).

(Подано 19 Березня 2024)

У цій статті представлено дослідження створення ознак для застосування машинного навчання (ML) при обробці клінічних даних, зосереджуючись на бінарній класифікації даних часових рядів. Дослідження демонструє ефективність використання перетворення Хаара для підвищення значущості ознак і покращення ефективності класифікації. Перетворення Хаара дозволяє підвищити точність прогнозування за рахунок збільшення ваги важливих параметрів, що особливо важливо при обробці складних клінічних даних. Результати дослідження показують значне збільшення площі під кривою робочих характеристик приймача (AUC-ROC) з 0,44 для базової моделі до 0,82 для моделі з перетворенням Хаара, що вказує на значні покращення в точності прогнозування. Методологія, описана в статті, охоплює різні етапи, включаючи попередню обробку даних, навчання моделі за допомогою алгоритму XGBoost та оцінку продуктивності за допомогою кривих AUC-ROC. Попередня обробка включає очищення та нормалізацію даних, що є важливими кроками для забезпечення високої якості результатів машинного навчання. Особливу увагу приділено використанню даних Інтернету речей (IoT) у клінічних умовах, що відкриває нові можливості для прогнозної аналітики та прийняття рішень у сфері охорони здоров'я. Підходи, описані в статті, можуть бути застосовані для аналізу великої кількості інформації, зібраної з різних медичних пристроїв, підключених до мережі IoT. Це дозволяє отримати більш точні прогнози і приймати обґрунтовані рішення на основі реальних даних, що сприяє покращенню якості медичних послуг і підвищенню рівня догляду за пацієнтами. Результати дослідження підкреслюють потенціал методів машинного навчання в закладах охорони здоров'я для підвищення точності прогнозів і прийняття рішень. Майбутні напрямки досліджень можуть передбачати вивчення додаткових методів розробки ознак і використання передових алгоритмів машинного навчання для подальшого підвищення корисності клінічної аналітики даних IoT. Зокрема, вивчення можливостей глибокого навчання і нейронних мереж може відкрити нові горизонти для аналізу і обробки клінічних даних.

Ключові слова: *Машинне навчання (ML); клінічний Інтернет речей; перетворення Хаара; штучний інтелект; AUC-ROC*
УДК 004.85:61

1. Вступ

Поширення пристроїв Інтернету речей (IoT) у клінічних умовах створило величезні обсяги даних, пропонуючи безпрецедентні можливості для прийняття рішень у сфері охорони здоров’я на основі даних. Однак використання повного потенціалу клінічних даних IoT вимагає складних

аналітичних методів. У цій статті описано структуру для використання методів машинного навчання та розробки функцій для отримання корисної інформації з клінічних даних Інтернету речей, з особливим акцентом на завданнях бінарної класифікації.

Наразі розробляється фреймворк за принципами TIPPSS для валідації та інтероперабельності даних і пристроїв клінічного Інтернету речей IoT [1].

Створюється конвеєр даних для обробки та очищення даних, виявлення важливих прогностичних ознак за допомогою будь-яких методів машинного навчання ML, наприклад [2].

У цій статті розглядається навчання моделі машинного навчання для клінічних даних Інтернету речей для застосування в дослідженні результатів бінарної класифікації часових рядів.

2. Машинне навчання замість аналітики даних

Наука про дані, що лежить на основі баз даних, поділяється на дві основні категорії: аналіз даних і штучний інтелект (ШІ) або його підгалузь – машинне навчання ML. Основна відмінність між ними виникає в тому, як дослідники збирають дані для отримання своїх результатів.

Аналітика даних спирається на логіку програмування та інструменти для отримання прогнозів на основі даних. Цей підхід добре підходить для структурованих даних з обмеженою кількістю змінних. Існує певне програмне забезпечення з інструкціями для бінарної класифікації часових рядів, наприклад [3].

Функція детектора голосової активності VAD полягає в тому, щоб відрізнити шум з присутнім мовленням від шуму без мовлення. Це досягається шляхом порівняння енергії відфільтрованої версії вхідного сигналу з пороговим значенням. Присутність мовлення відображається щоразу, коли поріг перевищено.

Виявлення мови в мобільному середовищі є складним завданням через низьке співвідношення сигнал/шум, як зустрічається, зокрема, в транспортних засобах, що рухаються. Щоб підвищити ймовірність виявлення мови, вхідний сигнал фільтрується, щоб зменшити вміст шуму, перш ніж прийняти рішення про голосову активність.

Частотний спектр і рівень шуму можуть змінюватися в межах певного середовища, а також між зовнішніми середовищами. Тому необхідно адаптувати коефіцієнти вхідного фільтра та енергетичний поріг через регулярні проміжки часу.

Цей детектор VAD призначений лише для каналів розширеної повної швидкості передачі голосового трафіку.

Методом цього дослідження є вивчення даних для вирішення подібних проблем за допомогою машинного навчання.

3. Машинне навчання

Штучний інтелект та ML виводять процес прогнозування на рівень, який раніше був неможливим. За допомогою ML-моделей фахівець в галузі обробки даних може реалізувати здатність програми до навчання та покращити якість її прогнозів. Наприклад, інженер працює на постачальника медичних послуг – стоматологічну клініку [4], модель ML може використовувати дані про шум з наявністю мови і шум без присутності мови, для покращення виявлення голосової активності у пацієнтів [5]. Модель ML навчається, отримуючи все більше прикладів. ML-моделі також можуть знаходити взаємозв’язки між даними, які людині спостерігачеві було б набагато складніше припустити. Чим більше якісних даних ми можемо надати, тим кращі прогнози можна зробити ML. Кращі прогнози забезпечують більш точну основу для прийняття рішень.

Завдання вирішити цю проблему за допомогою машинного навчання ML. Є доступ до набору даних, який містить п'ять ознак і цільову величину, яка є нормальною або аномальною. Можна використовувати цей набір даних для навчання ML-моделі, щоб передбачити, чи буде у пацієнта відхилення від норми.

Цей набір біомедичних даних було створено на основі цифрових тестових компонентів для розширеного повно швидкісного GSM модуля кодування голосу [6]. Дані були організовані в

задачу класифікації. Завдання полягає у віднесенні пацієнтів до однієї з двох категорій: Нормальні (179 пацієнтів) або Аномальні (187 пацієнтів).

Передбачено одну вхідну послідовність для кодування (TEST11.INP) [7]. Це жіноча мова, шум автомобіля, рівень активної мови: 15,8 дБ, пласка амплітудно частотна характеристика. Для цілей цього документа застосовуються такі терміни та визначення: шум – це компонент сигналу, що є результатом акустичних перешкод навколишнього середовища (Аномальні пацієнти).

Кожен пацієнт представлений у наборі даних п’ятьма біоелектричними атрибутами, які виводяться з матриці залишкових векторів прогнозу, що розбивається на 5 підматриць розмірністю 2 x 2 (в такому порядку) [7].

1) *F1 – 7-бітний індекс першої підматриці.*
2) *F2 – 8-бітний індекс другої підматриці.*
3) *F3 – 9-бітний індекс третьої підматриці (8 біт індекс плюс 1 біт знаку).*
4) *F4 – 8-бітний індекс четвертої підматриці.*
5) *F5 – 6-бітний індекс п'ятої підматриці.*

Для позначення класів використання наступна конвенція: Нормальні (NO) – 0; Аномальні (AB) – 1.

Короткострокове прогнозування. або лінійне прогнозування LP, виконується двічі на кадр мови з використанням автокореляційного підходу з асиметричними вікнами по 30 мс для вхідної послідовності (TEST11.INP).

Pandas DataFrames це найпоширеніше представлення в пам'яті складних наборів даних у Python.

Атрибути, функції та методи цього пакету наведені в цій статті і виділені курсивом.

На першому етапі проаналізовано дані з набору даних. По-перше, функція *tail(3)* Pandas повертає останні 3 рядки від об'єкта на основі позиції. Її було використано для дослідження останніх трьох рядків фрейму даних (таблиця 1).

Таблиця 1

Клас F1-F5

	F1	F2	F3	F4	F5	клас
363	20.0	42.0	488.0	44.0	19.0	0
364	31.0	56.0	82.0	61.0	19.0	0
365	28.0	33.0	206.0	47.0	24.0	0

Ця функція повертає останні три рядки для фрейму даних на основі позиції. Це корисно для швидкої перевірки того, чи містить фрейм даних правильний тип даних. По-друге, атрибут *shape* Pandas було використано для перевірки кількості рядків і стовпців. Результат виявився (366, 6). Потім було отримано список стовпців Index ['F1', 'F2', 'F3', 'F4', 'F5', 'class']. Існує п'ять біоелектричних ознак, а цільова колонка називається клас.

На другому етапі йде підготовка даних для навчання моделі ML, їх необхідно розбити на три набори даних. Під час пошуку в інтернеті було знайдено багато способів розбиття наборів даних. Багато прикладів коду, які можна знайти, розділяють набір даних на цільовий та функціональний. Потім кожен з цих двох наборів даних розбивається на три піднабори, в результаті чого ми отримуємо загалом шість наборів даних для відстеження.

Для переміщення цільової позиції стовпця XGBoost вимагає, щоб дані для навчання були в одному файлі. Файл повинен містити цільове значення у першому стовпчику. Цільовий стовпець було переміщено на першу позицію. Після цього отримано список стовпців: Index(['class', 'F1', 'F2', 'F3', 'F4', 'F5'], dtype='object').

Набір даних було розділено на два набори даних. Один набір даних був використаний для навчання, а інший набір даних був знову розділений для перевірки та тестування. Функція *train_test_split* була використана з бібліотеки scikitlearn, яка є безкоштовною бібліотекою

машинного навчання для Python. Вона має багато алгоритмів і корисних функцій, таких як та, що була використана.

Оскільки досліджено не так багато даних, переконалися, що розділені набори даних містять репрезентативну кількість представників кожного класу. Таким чином, було використано перемикач стратифікації. Нарешті, було використано випадкове число, щоб розбиття можна було повторити. Далі, тестовий і перевірочний набори даних були розділені на дві рівні частини. Три набори даних були досліджені за допомогою методу *shape* Pandas: (296.6); (37,6); (37,6). Розподіл класів було перевірено функцією *value_counts()*: 1-149, 0-143; 1-19, 0-18; 1-19, 0-18.

На третьому етапі, завантаження даних до Amazon S3. Дані для навчання з Amazon Simple Storage Service (Amazon S3) завантажено за допомогою XGboost. Таким чином, дані були записані у файл у форматі CSV, а потім завантажено цей файл на Amazon S3. Було налаштовано деякі змінні в S3-бакеті, а потім створено функцію для завантаження CSV-файлу в Amazon S3. Цю функцію можна використовувати повторно. Функція складається з наступного рядка:

dataframe.to_csv(csv_buffer, header=False, index=False)

Цей рядок записує Pandas DataFrame (який було передано у функцію) у буфер вводу-виводу з назвою csv_buffer. Буфер використовується тому, що файл не потрібно записувати локально. Щоб не виводити заголовки стовпців, використовується header=False. Щоб не виводити індекс Pandas, використовується index=False.

Щоб записати csv buffer до Amazon S3 як об'єкт, використовується операція *put* над об'єктом, який є властивістю буфера. Для вказівки стандартного γ3- буфера, виділеного для нашого сеансу SageMaker, використовується стандартний код. префікс це шлях в межах буфера, де SageMaker зберігає дані для поточного навчального завдання. Створена функція використовується для завантаження трьох наборів даних. Перед навчанням моделі було протестовано останні три рядки фрейму даних (таблиця 2).

Таблиця 2

Клас F1-F5

	клас	F1	F2	F3	F4	F5
363	0	20.0	42.0	488.0	44.0	19.0
364	0	31.0	56.0	82.0	61.0	19.0
365	0	28.0	33.0	206.0	47.0	24.0

На четвертому етапі, коли дані знаходяться в Amazon S3, модель була навчена. Першим кроком було отримання URI контейнера XGBoost. Далі було встановлено деякі гіперпараметри для моделі. Оскільки модель була навчена вперше, були використані деякі значення для початку. Було встановлено деякі гіперпараметри для моделі {"num_round":"42", "eval_metric": "auc", "objective": "binary: logistic"} перший раз, коли модель навчалася.

Функція *estimator* використовується для налаштування моделі. Ось декілька параметрів, шо становлять інтерес: *instance_count* – визначає, скільки екземплярів буде використано для навчання, використовується один екземпляр; *instance_type* – визначає тип екземпляра для навчання. У цьому випадку це ml.m4.xlarge.

Оцінювачу потрібні канали для подачі даних у модель. Для навчання використовується train_channel та validate_channel. Отже, дані були імпортовані і готові до використання. Навчання моделі відбувалося за допомогою методу припасування. Цей процес займав до 5 хвилин.

4. Вивчення результатів

Результатом роботи моделі є ймовірність. Перш за все, ця ймовірність була перетворена в один з двох класів 0 або 1. Для цього було створено функцію, яка виконує перетворення. У функції використовується поріг 0,6.

Останні п'ять значень для прогнозу були виведені за допомогою функції Pandas *head*(5) (таблиця 3).

Таблиця 3

Цільовий прогнозований бінарний файл для F1-F5

Номер екземпляра з тестового набору даних	Клас
32	1
33	1
34	0
35	0
36	0

Останні п'ять значень для тестового набору даних було виведено за допомогою функції *head(5)* Pandas (таблиця 4).

Таблиця 4

Набір тестових даних для F1-F5

	клас	F1	F2	F3	F4	F5
344	0	23.0	54.0	78.0	74.0	22.0
166	1	13.0	14.0	89.0	56.0	9.0
360	0	33.0	47.0	54.0	67.0	15.0
235	0	7.0	14.0	59.0	138.0	29.0
164	1	13.0	29.0	124.0	79.0	20.0

Виходячи з цих результатів, стало зрозуміло, що початкова модель була не такою вже й гарною. Це важко сказати, порівнюючи кілька значень. Нижче розглядаються деякі метрики, щоб побачити, наскільки добре працює модель.

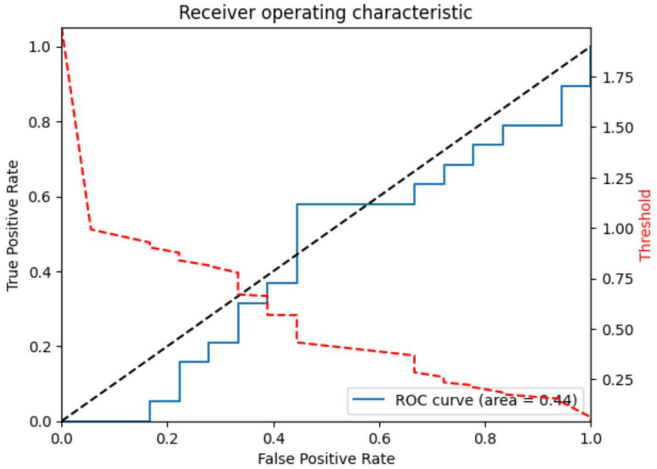

Рис. 1 Робоча характеристика приймача F1-F5

Бібліотека scikitlearn має функції, які можуть обчислювати площу під кривою робочої характеристики приймача AUC-ROC. ROC – це крива ймовірності. AUC показує, наскільки добре модель може розрізняти класи. AUC можна обчислити. Цю метрику можна використовувати для вимірювання продуктивності моделі. У цьому прикладі, чим вище AUC, тим краще модель розрізняє аномальних і нормальних пацієнтів. Залежно від значення порогу, AUC може змінюватися. AUC була побудована з використанням ймовірності замість перетвореного класу. Перевірочне значення AUC дорівнює 0.4415204678362573. Зазвичай ROC-крива будується з TPR

проти FPR, де TPR відкладається на осі *y*, а FPR на осі *x*. У scikitlearn є функція *roc_curve*, яка допомагає генерувати ці значення для побудови кривої (рис.1).

5. Дослідження даних перетворення Хаара

Кожен пацієнт представлений в наборі даних двадцятьма двома біоелектричними атрибутами, які отримані з матриці розщеплення залишкових векторів прогнозування. Набір даних з двадцяти двох біоелектричних ознак є коефіцієнтами перетворення Хаара n=20-точкового сигналу для двох залишкових векторів LSF [7].

Коефіцієнти перетворення Хаара для n=32-точкового сигналу *r32* були знайдені, наприклад:

$$h32 = H32 * r32 \tag{1}$$

Набір даних за двадцятьма двома біоелектричними ознаками є ненульовими коефіцієнтами вейвлет перетворення Хаара [8].

1) h1,..., h7 – перша група з семи коефіцієнтів перетворення Хаара.

2) h9,..., h13 – друга група з п'яти коефіцієнтів перетворення Хаара.

3) *h17,...,h26 третя група з десяти коефіцієнтів перетворення Хаара*.

Pandas DataFrames це найпоширеніше представлення в пам'яті складних наборів даних у Python.

Функція head(3) Pandas була використана для дослідження останніх трьох рядків даних (таблиця 5).

Таблиця 5

Клас H32

	h22	h23	h24	h25	h26	Клас
363	-53.	397.	426.	187.	206.	1
364	-1280	-37	-20.	187.	206.	1
365	-77	252.	254.	38.	-24.	1

Ця функція повертає останні 3 рядки для фрейму даних на основі позиції. Це корисно для швидкої перевірки того, чи містить фрейм даних правильний тип даних. По-друге, атрибут *shape* Pandas було використано для дослідження кількості рядків і стовпців. Результати були (366, 23). Потім отримано список стовпців Index(['h1', 'h2', 'h3', 'h4', 'h5', 'h6', 'h7', 'h', 'h10', '11', 'h12', 'h13', 'h17', 'h18', 'h19', 'h20',' h21',' h22',' h23', 'h24', ' h25', 'h26', ' class'], dtype='object'). Існує п'ять біоелектричних ознак, а цільовий стовпчик має назву клас.

На другому етапі підготовкb даних для навчання моделі ML, їх необхідно розбити на три набори даних.

Цільовий стовпець було переміщено на першу позицію. Після цього отримано список стовпців: Index(['class', 'h1', 'h2', '3', 'h4', '5', 'h6', 'h7', 'h', 'h10', '11', 'h12', 'h13', 'h17', '118', 'h19', 'h20', 'h21', 'h22', 'h23', 'h24', 'h25, 'h26'],dtype='object').

Набір даних було розділено на два набори даних. Один набір даних був використаний для навчання, а інший набір даних був знову розділений для перевірки та тестування.

Три набори даних було досліджено за допомогою методу *shape* Pandas: (296,23); (37.23); (37,23). Розподіл класів було перевірено за допомогою функції *value_counts*(): 1-147, 0-145; 1-19, 0-18; 1-19, 0- 18.

На третьому етапі (завантаження даних до Amazon S3) дані для навчання з Amazon Simple Storage Service (Amazon S3) були завантажені за допомогою XGboost. Таким чином, дані були записані у файл у форматі CSV, а потім цей файл було завантажено на Amazon S3. Було налаштовано деякі змінні в γ3-бакеті, а потім створено функцію для завантаження CSV-файлу в Amazon S3. Цю функцію можна використовувати повторно. Функція складається з наступного рядка:

dataframe.to_csv(csv_buffer, header=False, index=False)

Цей рядок записує Pandas DataFrame (який було передано у функцію) у буфер вводу-виводу з назвою csv_buffer. Буфер використовується тому, що файл не потрібно записувати локально. Щоб не виводити заголовки стовпців, використовується header=False. Щоб не виводити індекс Pandas, використовується index=False. Щоб записати csv_buffer до Amazon S3 як об'єкт, використовується операція *put* над об'єктом, який є властивістю буфера. Для вказівки стандартного γ3- буфера, виділеного для нашого сеансу SageMaker, використовується стандартний код *prefix* це шлях в межах буфера, де SageMaker зберігає дані для поточного навчального завдання. Створена функція використовується для завантаження трьох наборів даних. Перед навчанням моделі були протестовані останні три рядки фрейму даних (таблиця 6).

Таблиця 6

Клас H32

	Клас	h1	h2	h3	h4	h5
363	1	-2270.	-1470.	721.	-401.	410.
364	1	-39	763	3290.	-401.	-359.
365	1	-1400	-1600.	944	98.	1780.

На четвертому етапі навчання моделі, коли дані знаходяться в Amazon S3, модель була навчена.

Першим кроком отриманий ідентифікатор URI контейнера XGBoost. Далі встановлені деякі гіперпараметри для моделі. Оскільки модель навчено вперше, використані деякі значення для початку. Було встановлено деякі гіперпараметри для моделі {"num_round":"42", "eval_metric": "auc", "objective": "binary: logistic"} перший раз, коли модель навчалася. Функція *estimator* використовується для налаштування моделі. Ось декілька параметрів, що становлять інтерес: *instance_count* – визначає, скільки екземплярів буде використано для навчання. Використовується один екземпляр; *instance_type* – визначає тип екземпляра для навчання. У цьому випадку це ml.m4.xlarge.

Оцінювачу потрібні канали для подачі даних у модель. Для навчання використано train_channel та validate_channel. Отже, дані були імпортовані і готові до використання. Навчання моделі відбувалося за допомогою методу припасування. Цей процес займав до 5 хвилин.

6. Дослідження результатів перетворення Хаара

Результатом роботи моделі є ймовірність. Перш за все, ця ймовірність була перетворена в один з двох класів 0 або 1. Для цього було створено функцію, яка виконує перетворення. У функції використовується поріг 0,6. Останні п'ять значень для прогнозу було виведено за допомогою функції Pandas *head(5)* (таблиця 7).

Таблиця 7

Цільовий прогнозований бінарний файл для H32

Номер екземпляра з тестового набору даних	Значення
32	0
33	1
34	0
35	1
36	1

Останні п'ять значень для тестового набору даних було виведено за допомогою функції Pandas *tail(5)* (таблиця 8).

Таблиця 8

Набір тестових даних для h1-h5

	клас	h1	h2	h3	h4	h5
302	0	-5920.	-5120.	2080.	-401.	814.
205	1	-588.	974.	3910.	-781.	759.
315	0	-623.	-179.	-3630.	-401.	-44.
317	0	-3210.	-1320.	2150.	-948.	-313.
192	1	2620.	2660.	-1610.	-20.	-230.

Нижче ми розглянемо деякі метрики, щоб побачити, наскільки добре працює модель. Функція *roc_curve* згенерувала відповідні значення для побудови кривої (рис. 2).

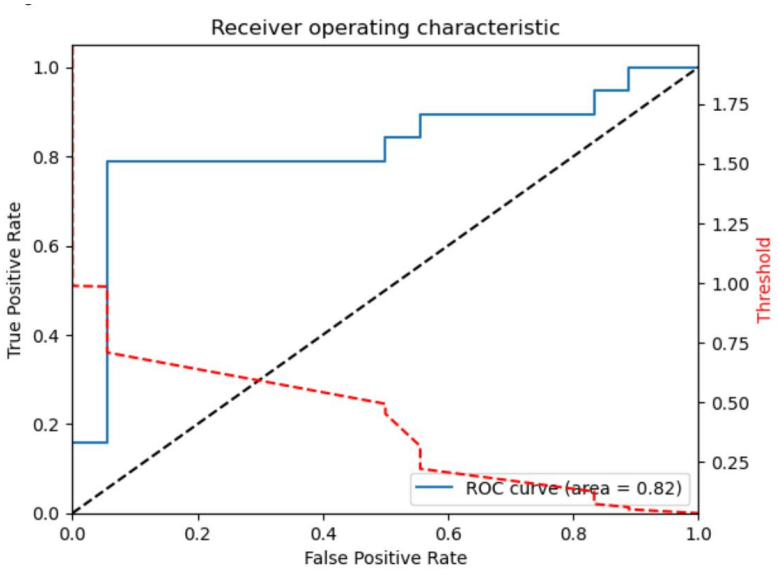

Рис. 2 Робоча характеристика приймача h1-h5

Щоб перевірити ефективність моделі, ми можемо порівняти прогнозовані значення з фактичними. Для кожної моделі можна розрахувати ще кілька метрик. Моделі класифікації повертають ймовірність для цілі. Ця ймовірність це значення між 0 і 1 вхідних даних, які належать до цільового класу. Щоб перетворити значення на клас, необхідно визначити поріг, який використовується. Можна вважати, що цей поріг дорівнює 50 відсоткам, але можна змінювати цю цифру на меншу або більшу, щоб покращити результати.

Графік робочої характеристики приймача ROC підсумовує всі матриці плутанини, які створює кожен поріг. Щоб побудувати такий графік, було обчислено чутливість (або частоту правильних спрацьовувань) у порівнянні з частотою хибних спрацьовувань для значення. Потім ці значення нанесені на графік, частоти хибно позитивних результатів розраховані, відніманням специфічності від 1. Коли ці точки нанесені на графік, була проведена лінія між ними. Пунктирна чорна лінія від (0,0) до (1.1), яка є лінією, що відображає відношення чутливості до TPR, дорівнює частоті хибно позитивних результатів. Точка на (1,1) означає, що всі аномалії правильно ідентифіковані, але також неправильно ідентифіковані всі норми. Такий результат не є добрим. Будь-яка точка на цій лінії означає, що частка правильно класифікованих зразків така ж, як і частка неправильно класифікованих зразків. Точка (0,0) означає нуль правильних спрацьовувань і нуль хибних спрацьовувань. Зазвичай метою є модель, яка має високу чутливість і низький відсоток помилкових спрацьовувань. Краще, коли лінія між пороговими записами знаходиться ближче до

лівого верхнього кута. Якщо є дані двох моделей, можна побудувати ROC-криву для кожної моделі і порівняти їх. Однак це може бути стомлюючим, тому замість цього була використана площа під кривою AUC-ROC. Як і у випадку з чутливістю та специфічністю, класифікація передбачає компроміс між правильною та неправильною ідентифікацією класів. Зміна порогового значення може вплинути на результат. Площа під кривою "крива-оператор" AUC-ROC – це ще одна метрика оцінки. Частина AUC це площа під побудованою лінією. Чим вище AUC, тим краще модель прогнозує аномалії як аномалії і норми як норми. Метрика AUC використана для швидкого порівняння моделей між собою.

Висновок

Площу під кривою робочої характеристики приймача було збільшено з 0,44 для базової моделі до 0.82 для моделі перетворення Хаара. Підсумовуючи, це дослідження демонструє ефективність інтеграції методів машинного навчання, зокрема алгоритму XGBoost, і методів розробки функцій, таких як перетворення Хаара, для покращення клінічного аналізу даних IoT. Підвищуючи точність прогнозування та сприяючи більш обґрунтованому прийняттю рішень у закладах охорони здоров'я, ці методології мають значні перспективи для покращення догляду за пацієнтами та оптимізації робочих процесів у сфері охорони здоров'я. Майбутні напрямки досліджень можуть передбачати вивчення додаткових методів розробки функцій і використання передових алгоритмів машинного навчання для подальшого підвищення корисності клінічної аналітики даних IoT.

Список використаних літературних джерел

[1] *T. Thompson, "P2933 – Standard for Clinical Internet of Things (IoT) Data and Device Interoperability with TIPPSS – Trust, Identity, Privacy, Protection, Safety, Security," IEEE EMBC, 21 05 2019. [Online]. Available: https://standards.ieee.org/project/2933.html.*

[2] *M. Saqib, Y. Sha and M. D. Wang, "Early Prediction of Sepsis in EMR Records Using Traditional ML Techniques and Deep Learning LSTM Networks," 2018 40th Annual International Conference of the IEEE EMBC, 2018, pp. 4038-4041.*

[3] *K. Järvinen, "Voice Activity Detector (VAD) for Enhanced Full Rate (EFR) speech traffic channels," 21 07 2020. [Online]. Available: https://portal.3gpp.org/...2753. [Accessed 16 12 2024].*

[4] *V. M. Bezruk, S. A. Krivenko, M. B. Samochernov, L. S. Kryvenko and S. S. Krivenko, "Model Discrete Wavelet Transform for Clinical IoT Data and Device Interoperability," 2022 IEEE 16th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET), Lviv-Slavske, Ukraine, 2022, pp. 64-69, doi:10.1109/TCSET55632.2022.9767044.*

[5] *[K. Järvinen, "Test sequences for the GSM Enhanced Full Rate (EFR) speech codec," 08 04 2022. [Online].*

[6] *[K. Järvinen, "Enhanced Full Rate (EFR) speech transcoding," 08 04 2022. [Online]. Available: https://portal.3gpp.org//....2748. [Accessed 14 02 2024].*

[7] *K. Järvinen, "ANSI-C code for the GSM Enhanced Full Rate (EFR) speech codec," 08 04 2022. [Online].*

[8] *Pulavskyi A., Krivenko S., Krivenko S. "The computation of line spectral frequencies using discrete wavelet transform for electrocardiograms processing", IEEE 36th International Conference on ELNANO, 2016, pp. 202-205.*

FEATURE ENGINEERING FOR THE IMPLEMENTATION OF MACHINE LEARNING IN CLINICAL DATA PROCESSING

O. Kyrsanov, S. Krivenko

Kharkiv National University of Radio Electronics, 14 Nauky Ave, Kharkiv, 61166, Ukraine

This paper presents a study of feature engineering for the application of machine learning (ML) in clinical data processing, focusing on binary classification of time series data. The study demonstrates the effectiveness of using the Haar transform to enhance feature importance and

improve classification performance. The Haar transform allows for increased predictive accuracy by augmenting the weight of significant features, which is especially crucial in handling complex clinical data. The research results show a substantial increase in the area under the receiver operating characteristic curve (AUC-ROC) from 0.44 for the baseline model to 0.82 for the Haar transform model, indicating significant improvements in predictive accuracy. The methodology described in the paper encompasses various stages, including data preprocessing, model training using the XGBoost algorithm, and performance evaluation via AUC-ROC curves. Data preprocessing involves cleaning and normalizing the data, critical steps to ensure high-quality machine learning outcomes. Special attention is given to using Internet of Things (IoT) data in clinical settings, which opens new possibilities for predictive analytics and decision-making in healthcare. The approaches described in the paper can be utilized to analyze large amounts of information collected from various medical devices connected to the IoT network. This allows for more accurate predictions and informed decisions based on real data, contributing improving of medical services and patient care quality. The research results underscore the potential of machine learning methods in healthcare institutions to enhance predictive accuracy and decision-making. Future research directions may include exploring additional feature engineering methods and using advanced machine learning algorithms to further increase the utility of clinical IoT data analytics. In particular, exploring the possibilities of deep learning and neural networks may open new horizons for clinical data analysis and processing.

Keywords: *Internet of Things (IoT); clinical IoT; Haar transform; artificial intelligence; machine learning (ML)*