

ПОБУДОВА ДІАГНОСТИЧНОЇ ЕКСПЕРТНО-МЕДИЧНОЇ СИСТЕМИ З ВИКОРИСТАННЯМ НЕЙРОННИХ МЕРЕЖ

М.М. Глибовець, К.В. Салата, Н.А. Ткач

У статті розглянуто основні методи (дерева прийняття рішень, глибинне навчання, метод k-найближчих сусідів, нейронні мережі) створення діагностичних експертно-медичних систем. Для практичної частини створення API було відібрано декілька класифікаторів та проведено дослідження їх роботи. А саме було порівняно класифікатори засновані на нейронних мережах, деревах прийняття рішень та методі k-найближчих сусідів. Були оптимізовані параметри для обраних класифікаторів. В результаті були відібрані параметри на яких досліджувались дані. Також досліджено збірку даних пацієнтів, які мали серцевий напад, для розробки діагностичної системи визначення наявності хвороб серця. Описано діагностичний API для виявлення хвороб серця у пацієнта.

Ключові слова: діагностичні системи, медичні експертні системи, нейронні мережі, дерева прийняття рішень, діагностичне API.

В статье рассмотрено методы (деревья принятия решений, глубинное обучение, метод k-ближайших соседей, нейронные сети) создания диагностических экспертно-медицинских систем. Для практической части создания API были выбраны классификаторы и проведено исследование их работы. А именно были сравнены классификаторы, базирующиеся на нейронных сетях, деревьях принятия решений и методе k-ближайших соседей. Были оптимизированы параметры для этих классификаторов. В результате были отобраны параметры на которых исследовались данные. Также был исследован датасет информации о пациентах, у которых был сердечный приступ, для разработки программной системы диагностирования сердечных болезней. Описано диагностическое API определения сердечных болезней пациентов.

Ключевые слова: диагностические системы, медицинские системы, нейронные сети, деревья принятия решений, диагностическое API.

In the article was discussed the methods (decision trees, deep learning algorithms, k-nearest neighbors, neural networks) to create diagnostic expert medical systems. For practice part were developed diagnostic API based on chosen classifiers that implement the algorithms and a study of their work was conducted. Namely, classifiers based on neural networks, decision trees and k-nearest neighbors method were compared. The parameters for the selected classifier were optimized. As a result, were selected parameters on which the data were researched. In addition, the dataset of information of patients who had heart attack was researched to develop a diagnostic system for revealing heart diseases. The diagnostic API for revealing patients' heart diseases is described. Keywords: diagnostic systems, medical systems, neural networks, decision trees, diagnostic API.

Key words: diagnostic systems, medical systems, neural networks, decision trees, diagnostic API.

Вступ

Об'єктом дослідження цієї роботи є діагностичні експертно-медичні системи (ЕМС), які можна використовувати для діагностування або обрахування ймовірності виникнення певної хвороби у пацієнта. У ході роботи було проаналізовано підходи до створення таких систем, які базуються на деревах прийняття рішень, нечіткій математиці, глибинному навчанні, методі k-найближчих сусідів, нейронних мережах та на основі останніх розробити програмну систему.

Одним з головних факторів якісного лікування є точне та швидке визначення хвороби. Для вирішення цієї проблеми можна використовувати діагностичні ЕМС. Наприклад, IBM Watson Google та DeepMind активно використовують нейронні мережі для діагностування, прогнозування та моніторингу стану пацієнтів у ряді лікарень – США та Англії.

ЕМС – це заснована на певній медичній базі знань система, що має аналітичні здібності та може діагностувати хвороби пацієнта або прогнозувати можливість виникнення певної хвороби без участі лікаря. Такі системи моделюють поведінку лікаря, найчастіше базуючись на вже відомі їм знання декількох експертів-лікарів. Також медичні експертні системи можуть зменшити хибні діагностування хвороб. За допомогою однієї з перших експертних систем [1], яка була розроблена на нейронній мережі, знизили відсоток хибно-позитивних діагнозів до 4%.

Загалом на теперішнім рівні розвитку медичні експертні системи вирішують достатньо прості задачі. Системи засновані на алгоритмах машинного навчання можуть аналізувати рентгенівські знімки, електрокардіограми та інші медичні аналізи, при цьому точність аналізу на певні патології більше ніж 90%. Такі системи потребують великої кількості даних аби навчитися діагностувати та прогнозувати хвороби.

Перші медичні системи були розроблені ще у 1970х роках – MYCIN. У 1990х роках були створені перші нейронні мережі а також мережі з використанням алгоритмів глибинного навчання. До сучасних медичних систем, які використовують алгоритми машинного навчання належать: IBM Watson Health, Google DeepMind Health, мобільні додатки для діагностування хвороб та моніторингу лікування (Sense.ly – симулює медсестру, BabylonHealth – зв'язок з лікарем та попереднє визначення хвороб).

Одною з основних проблем створення ЕМС є потреба отримання знань від лікарів та аналіз великої кількості даних. В останній час значний розвиток дістали дослідження побудови ЕМС з використанням нейронних мереж (НМ). Традиційно виділяють декілька методів створення НМ: метод опорних векторів,

© М.М. Глибовець, К.В. Салата, Н.А. Ткач, 2020

кластеризація, самоорганізаційна карта. Головними перевагами використання НМ є можливість порівнювати велику кількість карток пацієнтів, що забезпечує більшу точність постановки діагнозу, та швидкість використання.

За допомогою медичних систем які базуються на алгоритмах глибинного навчання можливо аналізувати комп'ютерні знімки. Наразі близько 70 відсотків усіх медичних досліджень базуються на таких алгоритмах, проте лише декілька з них були впроваджені у клініки [5].

Аналіз підходів

Глибинне навчання має декілька шарів з нелінійними обробляючими одиницями. На кожному шарі аналізуються певні характеристики з масиву даних утворюючи ієрархію від характеристик низького рівня до високого, використовуючи алгоритми навчання з вчителем або без.

На відміну від НМ, глибинне навчання використовує значно більшу кількість шарів, здебільшого це від 10 до 100 шарів у той час як у НМ 2–3 шари. Як наслідок такої особливості, НМ не можуть аналізувати складні параметри. Глибинне навчання вимагає більшої кількості даних для тренування та більшої кількості ресурсів для виконання. Шари мають більше нейронів, що надає можливість опрацювання більшої кількості параметрів. У медицині НМ краще діагностують хвороби пацієнтів, глибинне навчання краще може аналізувати рентгенівські знімки, МРТ та інші аналізи пацієнтів.

ЕМС також можуть бути розроблені за допомогою інших алгоритмів машинного навчання. Наприклад на деревах прийняття рішень та методу k-найближчих сусідів.

Дерева прийняття рішень, аналізуючи медичні картки пацієнтів, визначають головний симптом або ознаку після чого створюють дерево, яке в залежності від ознак приводить по гілках до листків, ознак які вже не можуть бути розділені. Процес розгалуження триватиме до досягнення всіх листків [1]. Аналіз тренувальних даних відбувається за принципом «розділяй та володарюй». Наприклад, якщо у пацієнта є кашель то дерево може привести до такої хвороби як грип, або якщо кашля немає то воно, при певних інших ознаках, може привести й до діагностування у пацієнта інсульту.

У діагностуванні за допомогою дерева прийняття рішень пацієнт отримає серію запитань. Аналізуючи відповіді на них приймаються рішення щодо ознак, які алгоритм виділив. Запитання побудовані від більш важливих до менш.

Метод k-найближчих сусідів використовується для класифікації медичних карток у діагностуванні хвороби. В ході аналізу даних, медичні картки розбиваються на групи за схожими між собою картками. Такі групи репрезентують класи випадків. При розбитті вказується параметр k визначення кількості груп. Головними перевагами цього методу є те, що аргументація віднесення окремого випадку до певної групи є зрозуміла експертам (максимально схожий випадок з вибірки визначає належність випадку розгляду до групи). Окрім, цей підхід стійкий до аномальних випадків.

Головною перевагою алгоритмів машинного навчання є можливість виявлення прихованих зв'язків (факторів) компонент аналізу та виявлення найважливіших параметрів діагностики. НМ мають переваги по точності постановки діагнозу. Це підтверджують дослідження зроблені спеціалістами з Китайського Університету Електричних Наук та Технологій [1]. Вони порівняли використання штучних нейронних мереж на основі методу опорних векторів, дерева прийняття рішень, метод k-найближчих сусідів, глибинного навчання та систем нечіткої логіки, правила для якої були побудовані на основі декількох алгоритмів серед яких алгоритм Мамдані та алгоритм Сугено. Тести проводились на даних МІТ-ВІН.

Аналізуючи ці результати ми можемо стверджувати, що найточнішими є НМ, проте потенційно глибинне навчання може розвинути таку ж точність. За швидкістю НМ є найгіршими, а от метод k-найближчих сусідів та дерева прийняття рішень показали гарну швидкість. Системи засновані на нечіткій логіці показали середні показники по швидкості і точності відносно інших алгоритмів. Звичайно неможливо визначити наскільки людський фактор (якість експертних висновків) впливав на цю систему.

Якщо задачі потребують великої кількості шарів, наприклад аналіз рентгенів, МРТ, то краще використовувати системи з глибинним навчанням.

Підготовка датасету

Для дослідження ми обрали базу даних Клівлендського університету, яка містить 14 атрибутів та призначена для визначення хвороби серця людини. Загалом база містить 303 записи. Атрибути бази даних такі: вік пацієнта, стать пацієнта, sr – тип болю у грудині, trestbps – кров'яний тиск, chol – рівень холестерину у крові, fbs – рівень цукру у крові, restecg – результати електрокардіограми, thalach - максимальна частота серцевих скорочень, exang – стенокардія, oldpeak – ST депресія викликана вправами, slope – нахил ST сегменту на кардіограмі, ca – кількість великих судин пофарбованих на флюороскопії, thal – дефект серця, num – діагноз.

За допомогою бібліотек Pandas та matplotlib було опрацьовано цю базу даних. У результаті аналізу ми визначили, що атрибут визначення рівня цукру в крові є розрідженим. Тому у подальших дослідженнях ми його не використовували.

Оскільки, датасет має пропущені дані, порожні місця були заповнені нулями. Такий підхід знижує важливість параметру. Порожні місця можна було заповнити середнім значенням, або медіаною значень

стовпчика, проте таке заповнення може вплинути на точність роботи алгоритму та хибно збільшити важливість параметру.

Дані були розподілені на дві частини: для тренування класифікаторів – 75 % та для перевірки точності роботи – 25 %.

Опис класифікаторів

Random Forest Classifier – класифікатор з бібліотеки sklearn [8], що генерує певну кількість дерев прийняття рішень, яка тренується на датасеті та за допомогою усередненого значення усіх дерев приймає рішення. Такий підхід знижує можливість перенавчання та збільшує точність за допомогою усереднення результату. Головний параметр підбирання – `n_estimators`, кількість дерев, які створюються для прийняття рішення.

MLPClassifier – багатoshаровий перцептрон, з бібліотеки sklearn. Головні параметри, які підбирались:

- Activation – активуюча функція для прихованого шару:
 - Identity – $f(x) = x$;
 - Logistic – $f(x) = \frac{1}{1+e^{-x}}$;
 - Tanh – $f(x) = \text{th}(x)$;
 - Relu – $f(x) = \max(0, x)$.
- Solver – вирішальна функція для оптимізації ваг:
 - Lbfgs – оптимізатор заснований на квазиньютоновських методах;
 - Sgd – оптимізатор заснований на методі стохастичного градієнта;
 - Adam – покращений метод стохастичного градієнта.
- Hidden_layer_sizes – вектор, що задає кількість нейронів на кожному рівні перцептрону.

KNeighborsClassifier – класифікатор з бібліотеки sklearn заснований на методі k-найближчих сусідів. Для дослідження було обрано параметр `n_neighbors` – кількість сусідів.

Метрики порівняння класифікаторів

Для порівняння класифікаторів, а також порівняння їх роботи з різними параметрами ми використовували декілька метрик: точність, повнота, f1-міра.

Точність – це відсоток правильно визначених результатів. У ході дослідження використовувався метод визначення `accuracy_score` з бібліотеки sklearn.metrics.

Повнота – важливий параметр для визначення якості роботи ЕМС, визначає кількість знайдених хворих відносно усіх хворих пацієнтів. Для визначення повноти ми застосовували метод `recall_score` з бібліотеки sklearn.metrics.

F1-міра – зважене середнє точності та повноти. Цю міру ми знаходили за допомоги методу `f1_score` з бібліотеки sklearn.metrics.

У ході тестування класифікатори отримували однакові дані в однаковій кількості. Оскільки в нас було 5 можливих варіантів результату метод k-найближчих сусідів відповідно мав 5 сусідів. Класифікатор KNeighborsClassifier отримав однакові результати точності, повноти та f1-міри – 0,539.

Для класифікатору RandomForestClassifier ми вирішили зробити перший прогін на 10 деревах. Метрики набули однакових показників, а саме – 0,539.

Для першого прогону класифікатор MLPClassifier запускався без параметрів аби оцінити класифікатор на параметрах по замовченню. Метрики набули однакових показників, а саме – 0,486.

Перші прогони класифікаторів RandomForest та K-NN дали однакові результати метрик. Прогін MLP класифікатора показав гірші результати метрик ніж інші алгоритми. Також у всіх класифікаторів метрики приблизно співпадали. Ми припустили, що це є результатом наявності у датасет декількох класів. Тому для повноти та f1-міри було обрано варіант обрахування «місго», що є глобальним обрахуванням показників і для подальших прогонів датасету на декількох класах вимірювалась точність, як головна метрика.

Результати MLP класифікатору можна виправдати тим, що параметри по замовченню, які він отримує, є 1 прихований шар з 100 нейронів, та вирішальна функція «adam», що більше підходить для великих датасетів.

У подальшому аналізі важливим стало визначення кількості дерев для RF класифікатора та кількості нейронів на кожному прихованому шарі MLP класифікатора. Очевидно, що важливо було проводити дослідження вирішальної та активуючих функцій вже на гарному для датасету векторі прихованих шарів.

Ми перевіряли точність роботи RF класифікатора для різної кількості дерев від 10 до 100. Під час порівняння результатів було визначено, що найкраща точність досягалася при 40 деревах – 0,618. На рис. 1 можна побачити графік, який відображає точність при різній кількості дерев.

Для аналізу MLP класифікатора було обрано 2 приховані шари, оскільки в цьому випадку отримувалася більша точність у порівнянні з одним прихованим шаром. Тестування показало, що при 15 нейронах на першому прихованому шарі було досягнуто найбільшу точність – 0,605. Вона була досягнута при 14 нейронах на другому прихованому шарі.

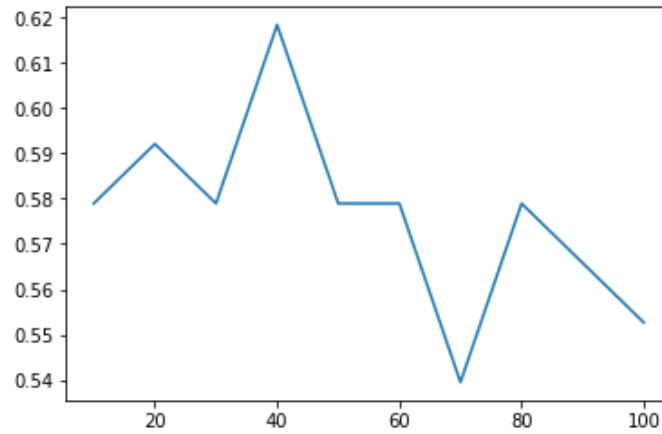


Рис. 1. Точність роботи RF класифікатору для різної кількості дерев

Важливим параметром для MLP класифікатору виявився вектор шарів. Змінивши цей вектор, точність класифікатору значно зросла та випередила інші класифікатори. З метою покращення точності, а також дослідження інших параметрів MLP класифікатору було зроблено тестування, у якому використовувались різні активуючі та вирішальні функції.

Під час дослідження вирішальної функції була отримана така точність: Adam – 0,5, Sgd – 0,539, Lbfgs – 0,592. На цьому датасеті у функції «lbfgs» точність більша ніж у інших.

Аналізувалися чотири активуючі функції. Загалом результати активуючих функцій є приблизно однаковими, проте краще за інших були результати функції «relu». Функції «logistic» та «tanh» показали однаковий результат.

Тому в подальших дослідженнях будуть використовуватись вирішальна функція «lbfgs» та функція-активатор «relu».

Максимальна отримана точність у подальшому була близько 63 %. Оскільки, така точність є незадовільною для користувачів медичної системи, адже ризик того що вона визначить є занадто великим та лікар-терапевт в такому випадку має кращі шанси виявити хворобу, то для медичної системи потрібно було шукати краще рішення.

Найважливішим фактором, окрім параметрів, для успішної роботи класифікаторів є дані опрацювання. Тому важливо було дослідити роботу класифікаторів, змінюючи дані.

Дослідження даних

Важливим фактором успішної роботи класифікаторів є велика кількість даних, тому перш за все потрібно було її збільшити. Для цього ми знайшли ще два додаткові датасети зі схожими ознаками. Ці датасети були зібрані угорським та швейцарським університетами. Датасет угорського університету містить у собі 294 записи про пацієнтів, швейцарський містить 123 записи. Датасети були поєднані у один та оброблено пропущені значення. Потім ми використали класифікатори на параметрах які були визначені у попередньому розділі. Загальна кількість записів у датасеті становила 720.

Результати першого прогону класифікаторів після розширення даних можна побачити на рис. 2.

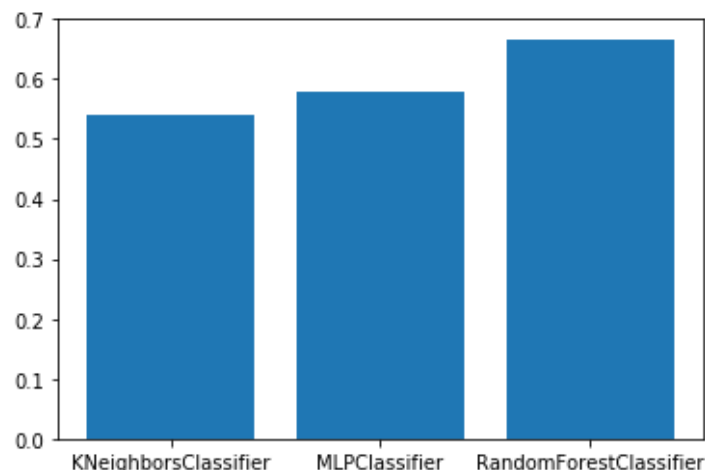


Рис. 2. Результати першого прогону класифікаторів після розширення даних

Внаслідок розширення датасету було значно покращено точність роботи RandomForestClassifier. RF класифікатор відпрацював з точністю близько 67 %. MLP близько 58 %, порівняно з попередніми результатами це гірше ніж було, проте такий результат можна списати на нестабільність роботи класифікатора, адже на однакових даних класифікатор при різних прогонах має різну точність. Використання KNN становить приблизно 51 %, що є краще ніж попередні результати.

У ході дослідження було вирішено, що наступним кроком для покращення точності є нормалізація даних – заміна значень ознак на значення від 0 до 1. Аналізуючи тестування ми побачили, що RF класифікатор покращив результати приблизно на один відсоток. Результати роботи MLP класифікатора також покращились, та збільшились приблизно на 3 %. Найкраще покращення точності в даному експерименті має KNN класифікатор – близько 15 %. Такі результати демонструють кращу роботу класифікаторів з нормалізованими даними, особливо це відноситься до KNN класифікатора.

У зв'язку з тим що в цьому датасеті результуюча ознака може набирати 5 можливих варіантів, при тому що ці варіанти не є описані та невідомо, що саме кожен з них означає, а також через те що у доданих датасетах результуюча ознака може набувати лише двох варіантів ми вирішили перейти до бінарної класифікації: хвора людина, здорова людина. Класифікатори тестувалися на нормалізованих та ненормалізованих даних.

Ненормалізовані дані дали значний приріст точності класифікаторів. Для RF класифікатора точність піднялась до 82.7 %, тобто порівняно з попереднім результатом піднялась приблизно на 15 %. Точність роботи MLP класифікатора піднялась з 58 до 85 %, що є найкращим результатом. Точність роботи KNN збільшилась до 69,9 % приблизно на 18 відсотків. Усі класифікатори збільшили точність та показали кращий результат порівняно з класифікацією на 5 груп.

На нормалізованих даних класифікатори покращили такі результати. RF класифікатор показав точність приблизно 85,5%, що значно більше ніж було до бінарної класифікації та приблизно на 2,8% більше ніж на ненормалізованих даних. Таким чином для подальшого використання краще використовувати RF класифікатор з нормалізованими даними. MLP класифікатор показав точність 85 %, відносно класифікації на 5 груп точність значно збільшилась проте відносно класифікації на ненормалізованих даних точність залишилась такою ж як і була. Тому, для майбутнього використання потрібно перевірити на ненормалізованих та нормалізованих даних результати більшої кількості прогонів та порівняти точність а також стабільність використання, адже MLP класифікатор не є стабільним та має великий розкид точності відносно різних прогонів. KNN класифікатор показав точність приблизно 80 %, що значно більше попередніх результатів, приблизно на 10 % більше ніж на ненормалізованих даних. Цей класифікатор краще використовувати на нормалізованих даних.

Для порівняння роботи MLP класифікатора на нормалізованих та ненормалізованих даних було вирішено зробити по десять прогонів на кожному варіанті.

Побудовані графіки показали, що на обох варіантах даних максимальна точність під час класифікації є приблизно 85 %. Найменша точність на нормалізованих даних приблизно 78%, на ненормалізованих даних трохи більше ніж 81 %. Такі результати показують нам, що під час класифікації на нормалізованих даних розкид по точності під час різних прогонів є більший ніж під час класифікації на ненормалізованих даних. Також у середньому точність на ненормалізованих даних є більшою.

Враховуючи ці результати наступні дослідження для MLP класифікаторі проводились на ненормалізованих даних для зменшення розкиду.

Порівняння ознак датасету

Оскільки дані є найважливішим фактором для ефективної роботи класифікатора важливо знати які саме ознаки з датасету мають найбільший вплив на класифікацію, а які навпаки зовсім не мають впливу. Маючи таку інформацію можна маніпулювати тим, які дані подавати на вхід. Таким чином можна збільшити цінність системи адже на даному етапі після отримання результатів все рівно потрібна консультація лікаря та більш точне діагностування. В цьому датасеті є певні ознаки, які користувач не зможе сам дізнатись проте є ймовірність, що цінність системи впаде для нього якщо він спочатку дізнається ці ознаки у лікаря.

Для дослідження ознак було зроблено експеримент в якому кожну ознаку по чергово вилучали та проводили класифікацію. Класифікація проводилась на RF класифікаторі.

Випробування показали, що такі поля як нахил ST сегменту на кардіограмі, кров'яний тиск та максимальна частота серцевих скорочень майже не впливають на точність класифікації. Найважливішими ознаками для класифікації є тип болю у грудині та рівень холестерину у крові. Інші ознаки втрачають приблизно по 1–2 % точності, що в деяких випадках може бути не дуже важливо. Враховуючи такі результати для медичної системи потрібно провести експеримент в якому буде скорочено кількість ознак. Залишаться такі ознаки як стать, вік, тип болю у грудині, рівень холестерину, наявність стенокардії під час навантажень. Такий набір ознак дозволить без консультації лікаря швидко встановити наявність хвороби.

Аналіз експерименту показав, що загалом для класифікаторів точність впала. Найбільше вона знизилась у KNN класифікатора – 71 %. RF класифікатор – 78 % точності. MLP класифікатор показав точність 82,2 %.

Базуючись на результатах експериментів, для головного модуля EMC ми зупинилися на застосуванні MLP класифікатора з 5 ознаками, аби користувачі могли без участі лікаря перевіряти чи хворі вони, а також RF класифікатор який буде базуватись на всіх ознаках, для більш точного визначення чи хвора людина.

Опис діагностичної системи

Медичні системи можуть виконувати багато задач, головні серед них це діагностування та моніторинг стану пацієнтів. У ході роботи ми вирішили створити Heart Diagnostic API, головною метою якого є діагностування хвороб у пацієнтів.

Heart Diagnostic API можна використовувати як частину медичної системи. На базі API можна буде створювати Web, Mobile, Desktop застосування.

Heart Diagnostic API складається з чотирьох головних складових (рис.3): база даних; модуль, який діагностує за допомогою RandomForest; модуль, який діагностує за допомогою MLP; контролер, який приймає запити.

Модуль MLP приймає від контролеру 5 ознак за допомогою яких проводить діагностування та вирішує чи хвора людина. Також цей модуль зберігає дані та результат в базі даних, а також може навчатись за допомогою тих даних що вже є в базі даних.

Модуль RF приймає від контролеру 13 ознак та проводить діагностування. Він може навчатись та відправляти у базу даних нові дані.

База даних зберігає інформацію яку отримує система та може від контролера отримувати інформацію про результати огляду користувачів у лікаря аби покращити роботу системи.

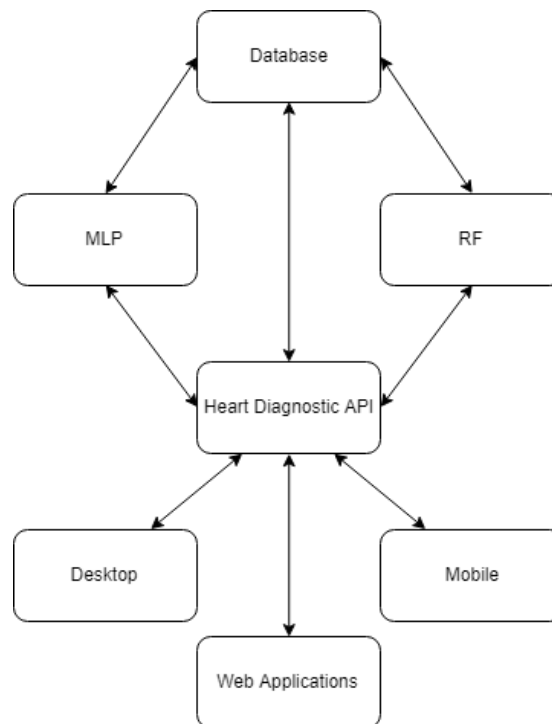


Рис. 3. Структура Heart Diagnostic API

База даних складається з чотирьох сутностей: Medical System, Patient, Card, Extended Card. Кожна сутність має свої атрибути серед яких атрибут за яким екземпляри ідентифікуються.

Medical System – сутність відповідає за ті системи, які можуть використовувати HDA, зберігаються ім'я системи та ідентифікатор.

Patient – сутність відповідає за інформацію про пацієнта, ідентифікатором є пара ідентифікатор системи, з якої отримали пацієнта, та ідентифікатор картки, яка зберігається у базі даних. Також можливе зберігання інформації про ім'я та прізвище клієнта, проте ця інформація не є обов'язковою.

Card – сутність відповідає за збереження інформації про пацієнтів, має ідентифікатор, результат роботи та 5 ознак по яким діагностується хвороба, використовується для модулів RF та MLP.

Extended Card – сутність відповідає за збереження інформації про пацієнтів, зберігає інформацію про 8 ознак, які не зберігаються у Card, використовується для модуля RF поєднуючи дані з Card.

Для розробки діагностичного модулю було використано такі технології: Django, Django REST Framework, Sklearn, Pandas.

Django – безкоштовний фреймворк для розробки веб-застосувань на мові Python. Він використовує MVC шаблон проектування веб-застосувань, а також має власну ORM для зв'язування бази даних з веб-застосуванням. За допомогою цього в Django сутності бази даних описуються Python-класами, з яких генерується схема БД [9].

Django REST Framework – бібліотека, яка допомагає створювати API на базі серверу розробленого за допомогою Django.

Sklearn – бібліотека, у якій реалізовані методи машинного навчання мовою Python [8].

Pandas – бібліотека, яка створює масиви даних та інструменти керування цими даними реалізована мовою Python [8].

Для використання Heart Diagnostic API потрібно зареєструвати медичну систему, яка буде використовувати HDA, аби отримати доступ. Після реєстрації потрібно буде оновити медичну систему та додати необхідні запити до неї.

Загалом HAD підтримує 3 типи запитів: до модулю MLP; до модулю RF; на підтвердження або спростування результатів.

Аби користувачу медичної системи отримати діагностування за меншою кількістю ознак потрібно відправити запит в якому відповідно позначити ці ознаки, після чого HDA опрацює запит та після відпрацювання модулю MLP поверне результат до медичної системи. Інформацію про користувача модуль збереже в базі даних.

Для використання цього запиту потрібно в медичній системі направити запит на heartdiagnostic.herokuapp.com/hda/mlp.

У випадку коли користувач має повну інформацію по усім вищеописаним 13 ознакам потрібно направити запит, з цими ознаками після чого контролер підключить модуль RF та поверне результат. Також після відпрацювання модуль розділить ознаки та збереже їх до бази даних у різні сутності.

Для використання цього запиту потрібно в медичній системі направити запит на heartdiagnostic.herokuapp.com/hda/rl.

Для підтвердження або спростування результатів роботи системи потрібно направити запит на heartdiagnostic.herokuapp.com/hda/res, вказавши номер пацієнта та результат після обстеження.

Висновки

У ході роботи було досліджено алгоритми та методи створення медичних систем, існуючі медичні системи. Було проведено порівняльний аналіз головних методів та алгоритмів на яких базуються діагностичні системи. Для практичної частини створення API було відібрано декілька класифікаторів які реалізують алгоритми та проведено дослідження їх роботи. А саме було порівняно класифікатори засновані на нейронних мережах, деревах прийняття рішень та методи k-найближчих сусідів. Були оптимізовані параметри для обраних класифікаторів. В результаті відібрані параметри на яких досліджувались дані.

Під час дослідження датасету були виявлено, що для деяких класифікаторів важливу роль на якість роботи грає нормалізація даних. При цьому для MLP класифікатору нормалізація даних мала гірший результат ніж на ненормалізованих даних. Також було проведено роботу по зведенню до бінарної класифікації, що дало великий приріст точності роботи класифікації. Після чого було порівняно ознаки датасету. Для кожної ознаки зроблено прогін без неї у датасеті та порівняно те як це вплинуло на точність роботи класифікатора. Результат тестування дав змогу знизити кількість ознак необхідних для класифікації.

Використовуючи результати цих досліджень було розроблено Heart Diagnostic API.

Розроблений API може слугувати базисом для подальшого створення багатьох систем на різних платформах, а також може використовуватись в дослідженнях роботи класифікаторів. Важливою цінністю є можливість накопичення даних, що в подальшому можуть покращити діагностування хвороб серця.

Література

1. Lin Di., Vasilakos A., Tang Yu, Yao Yuanzhe. Neural Networks for Computer-Aided Diagnosis in Medicine: A Review. *Neurocomputing*. 2016. 216. 10.1016/j.neucom.2016.08.039.
2. Al-Shayea Qeethara. Artificial Neural Networks in Medical Diagnosis. *Int J Comput Sci*. 2011. Issues. 8. P.150–154.
3. Madkour M., Roushdy M. Methodology for Medical Diagnosis based on Fuzzy Logic. *Egyptian Computer Science Journal*. 2004. 26.
4. Podgorelec V., Kokol P., Stiglic B., Rozman I. Decision trees: an overview and their use in medicine. *Journal of Medical Systems*, Kluwer Academic/Plenum Press. 2002. V. 26. N. 5. P. 445–463.
5. Startups to industry giants are leaning into AI to transform healthcare. November 20, 2018 by KIMBERLY POWELL URL: <https://blogs.nvidia.com/blog/2018/11/20/healthcare-industry-uses-nvidia-ai/> (дата звернення: 24.04.2020).
6. Change of Heart: How AI Can Predict Cardiac Failure Before It's Diagnosed April 11, 2016 by JAMIE BECKETT URL: <https://blogs.nvidia.com/blog/2016/04/11/predict-heart-failure/> (дата звернення: 24.04.2020).
7. Козін М.: Штучний інтелект у медицині URL: <https://blog.mednote.life/articles/technology/iskusstvennyy-intellekt-v-medicine> (дата звернення: 24.04.2020).
8. Scikit-learn URL: <https://en.wikipedia.org/wiki/Scikit-learn> (дата звернення: 24.04.2020).
9. Django URL: <https://ru.wikipedia.org/wiki/Django> (дата звернення: 24.04.2020).

References

1. Lin Di., Vasilakos A., Tang Yu, Yao Yuanzhe. Neural Networks for Computer-Aided Diagnosis in Medicine: A Review. *Neurocomputing*. 2016. 216. 10.1016/j.neucom.2016.08.039.
2. Al-Shayea Qeethara. Artificial Neural Networks in Medical Diagnosis. *Int J Comput Sci*. 2011. Issues. 8. P.150–154.
3. Madkour M., Roushdy M. Methodology for Medical Diagnosis based on Fuzzy Logic. *Egyptian Computer Science Journal*. 2004. 26.

4. Podgorelec V., Kokol P., Stiglic B., Rozman I. Decision trees: an overview and their use in medicine. *Journal of Medical Systems*, Kluwer Academic/Plenum Press. 2002. V. 26. N. 5. P. 445–463.
5. Startups to industry giants are leaning into AI to transform healthcare. November 20, 2018 by KIMBERLY POWELL URL: <https://blogs.nvidia.com/blog/2018/11/20/healthcare-industry-uses-nvidia-ai/> (дата звернення: 24.04.2020).
6. Change of Heart: How AI Can Predict Cardiac Failure Before It's Diagnosed April 11, 2016 by JAMIE BECKETT URL: <https://blogs.nvidia.com/blog/2016/04/11/predict-heart-failure/> (дата звернення: 24.04.2020).
7. Kozin M. Artificial Intelligence in medicine URL: <https://blog.mednote.life/articles/technology/iskusstvennyy-intellekt-v-medicine> (Accessed: 24.04.2020).
8. Article about Sklearn URL: <https://en.wikipedia.org/wiki/Scikit-learn> (Accessed: 24.04.2020).
9. Article about Django framework URL: <https://en.wikipedia.org/wiki/Django> (Accessed: 24.04.2020).

Одержано 10.03.2020

Про авторів:

¹*Глибовець Микола Миколайович*,
доктор фізико-математичних наук, професор.
Кількість наукових публікацій в українських виданнях – понад 170.
Кількість наукових публікацій в іноземних виданнях – понад 20,

¹*Салата Кирило Володимирович*,
асистент кафедри інформатики,

²*Ткач Наталія Альбертівна*,
кандидат медичних наук, старший науковий співробітник.
Кількість наукових публікацій в українських виданнях – 55.
Кількість наукових публікацій в іноземних виданнях – 8.

Місце роботи авторів:

¹Національний університет «Києво-Могилянська академія»,
вул. Григорія Сковороди, 2, Київ, 04070.
Тел.: 425-60-59.
E-mail: salata.kirill@gmail.com.

²ДУ ННЦ «Інститут кардіології імені академіка М.Д. Стражеска» НАМН України.,
03151, м. Київ, вул. Народного ополчення, 5.
Тел.: (044) 275- 86-77.
E-mail: natalia.tkach@ukr.net