https://doi.org/10.31649/1997-9266-2025-178-1-92-103

УДК 004.85

А. І. Топольський[1]
Є. А. Паламарчук[1]

ВИКОРИСТАННЯ МАШИННОГО НАВЧАННЯ ДЛЯ ВИЗНАЧЕННЯ МІСЦЕЗНАХОДЖЕННЯ ЛЮДЕЙ У ПРИМІЩЕННЯХ

[1]Вінницький національний технічний університет

Досліджено проблему автоматизованої обробки даних для фіксації присутності студентів на заняттях. Запропоновано використовувати методи машинного навчання, адже вони дозволяють спрогнозувати місцезнаходження студентів в приміщеннях навіть за умов аномалій у даних. Вирішення цієї проблеми сприятиме підвищенню ефективності освітнього процесу та зменшення залежності від традиційних способів фіксації присутності, які потребують витрат часу та людських ресурсів.

Проведено експерименти з використанням різних методів машинного навчання для задач регресії та класифікації. Мірою для порівняння різних методів використано точність прогнозування.

Досліджено застосування таких методів регресії як SVR, LinearSVR, NuSVR, PLSRegression, KernelRidge, RidgeCV, BayesianRidge, DecisionTreeRegressor та ExtraTreeRegressor. Найкращу точність прогнозування отримано методами DecisionTreeRegressor, KernelRidgeRegression та ExtraTreeRegressor — 92,5, 93,9 та 95,5 % відповідно. Проте для методів регресії необхідні неперервні дані, такі як координати користувача, що обмежує їхнє використання в умовах, де технічні засоби не дозволять отримати такі дані.

Як альтернатива розглянуто методи класифікації, а саме: SVC, KNeighborsClassifier, DecisionTreeClassifier та RandomForestClassifier. Первинні результати показали нижчу точність у порівнянні з методами регресії, що зумовлено недостатньою репрезентативністю тренувальних даних. Для вирішення цієї проблеми застосовано покроковий алгоритм, який поступово прогнозує будівлю, поверх та конкретне приміщення. Такий алгоритм забезпечив значне підвищення точності. Найкращий результат показав метод RandomForestClassifier — 94,3 %.

Підсумовуючи зазначимо, що вибір методу машинного навчання залежить від використовуваних технічних засобів. Якщо вони дозволяють отримувати неперервні дані, такі як координати, оптимально використовувати методи регресії ExtraTreeRegressor, DecisionTreeRegressor або KernelRidgeRegression. Якщо ж неперервні дані неможливо отримати, то оптимально використовувати метод класифікації RandomForestClassifier із запропонованим покроковим алгоритмом.

Ключові слова: автоматизовані системи відвідуваності, електронні навчальні системи, машинне навчання, методи класифікації, методи регресії, локалізація людей у приміщенні.

Вступ

Розвиток технологій дозволяє використовувати нові технології для оптимізації та автоматизації різноманітних процесів. Одним з таких процесів є фіксація присутності студентів на занятті. Найрозповсюдженішим методом фіксації присутності є заповнення паперового журналу викладачем за перекличкою студентів. Одним з недоліків цього способу є можливість помилки: студент може не почути своє прізвище у великій аудиторії, викладач може випадково поставити відмітку в іншу комірку тощо. Іншим недоліком є витрати часу на заповнення паперового журналу. Перекличка, зазвичай, починається на початку або через декілька хвилин після початку заняття. В залежності від кількості студентів, які знаходяться в аудиторії, цей процес може тривати 5—15 хвилин [1]. Автоматизація процесу фіксації студентів дозволить використати цей час на викладення матеріалу, що підвищить ефективність педагогічної роботи.

Задача автоматизації фіксації присутності студентів на занятті складається з двох основних етапів — збір даних та їхня обробка. Для збору даних можуть використовуватися різноманітні

технології, такі як Wi-Fi, Bluetooth, RFID, QR код, розпізнавання обличчя, сканування відбитку пальця та ін. Кожна технологія має свої переваги і недоліки, вибір технології залежить від поставлених задач та умов [2].

Одним з методів оброблення даних є машинне навчання. Методологія машинного навчання дозволяє спрогнозувати аудиторію, в якій знаходиться студент, навіть якщо в отриманих даних є непередбачувані аномалії. Наприклад, якщо дані отримані під час знаходження студента в коридорі, а не в аудиторії, студента можуть зафіксувати незвичні для найближчої аудиторії датчики. Таке прогнозування виконується шляхом навчання на основі великих даних. Таким чином, за можливості створення власної вибірки даних, машинне навчання можна розглядати як оптимальну методологію для визначення місцезнаходження студента [3].

Метою роботи є аналіз різних методів та алгоритмів машинного навчання для визначення оптимального способу прогнозування місцезнаходження студентів у межах вищого навчального закладу.

Постановка задачі і передумови

Для дослідження використано датасет «UjiIndoorLoc: An indoor localization dataset» [4]. Цей датасет зосереджений на технологіях і методологіях позиціонування «відбитків пальців» у бездротовій локальній мережі (також відомих як WiFi Fingerprinting). Ці дані офіційно використані у конкурсі IPIN2015 [5]. Опис даних, з яких складається датасет, наведено у табл. 1.

Таблиця 1

Дані з датасету UjiIndoorLoc: An indoor localization dataset

Назва	Опис даних	Формат даних
WAP001—WAP520	значення інтенсивності сигналу для WAP (Wireless Access Point)	Дискретні, –104…0, 100
Longitude	координати смартфону під час сканування, довгота	неперервні
Latitude	координати смартфону під час сканування, ширина	неперервні
Floor	поверх будівлі	дискретні, 0…4
BuildingID	ідентифікатор будівлі, вимірювання проводились у трьох різних будівлях	дискретні, 0…2
SpaceID	внутрішній ідентифікаційний номер для ідентифікації приміщення, в якому здійснено сканування	дискретні
RelativePosition	відносне положення відносно SpaceID	дискретні, 1 — в середині, 2 — зовні
UserID	ідентифікатор користувача	дискретні
PhoneID	ідентифікатор пристрою Android	дискретні
Timestamp	час UNIX, коли зроблено сканування	дискретні

В цьому дослідженні як вхідні дані використано дані з датчиків WAP001-WAP520 та координати Longitude та Latitude, а для прогнозування вибрано номер аудиторії SpaceID. Структура використаних даних, з яких складається датасет, подана у табл. 2.

Таблиця 2

Структура використаних даних датасету

№	WAP001	...	WAP520	LONGITUDE	LATITUDE	FLOOR	BUILDINGID	SPACEID
1	100	...	100	–7541,2643	4864921	2	1	106
2	100	...	100	–7536,6212	4864934	2	1	106
3	100	...	100	–7519,1524	4864950	2	1	103
4	100	...	100	–7524,5704	4864934	2	1	102
5	100	...	100	–7632,1436	4864982	0	0	122

Задачею дослідження є побудова методами машинного навчання такої моделі, яка, використовуючи дані з табл. 2, із задовільною точністю буде прогнозувати місцезнаходження людини. На прогнозування місцезнаходження людини впливають параметри Longitude, Latitude, Floor, BuildingID, SpaceID. Для застосування методів машинного навчання використано бібліотеку scikit-learn.

Розв'язання поставленої задачі

Одним з основних методів аналізу даних машинним навчанням є регресія. Задача методів регресії полягає у тому, щоб знайти залежність між вхідними даними та неперервними вихідними даними [6].

В залежності від типу використаної вибірки даних різні методи машинного навчання будуть мати різні показники точності прогнозування. В цьому дослідженні використано такі методи регресії: SVR, LinearSVR, NuSVR, PLSRegression, KernelRidge, RidgeCV, BayesianRidge, DecionTreeRegressor та ExtraTreeRegressor.

Метод регресії на основі опорних векторів (SVR, Support Vector Regression) полягає в побудові гіперплощини, яка найкраще представляє залежність між вхідними і вихідними змінними [7]. Цей метод використовує спеціальну функцію втрат, яка ігнорує помилки менші за встановлений параметр ε, створюючи зону нечутливості. Це дозволяє моделі бути стійкішою до певного рівня шуму. Задача методу SVR полягає у мінімізації такої функції [8]:

$$\min\left(\frac{1}{2}w_j^2 + C\sum_{i=1}^{n}\left(\xi_i + \xi_i^*\right)\right) \text{ за умов } y_i - wx_i - b \le \epsilon + \xi_i,\ wx_i + b - y_i \le \epsilon + \xi_i^*,\ \xi_i, \xi_i^* \ge 0,\ \forall i, \tag{1}$$

де C — коефіцієнт, що регулює компроміс між шириною зони нечутливості ε і допустимими помилками; ξ_i — величина, на яку точка (x_i, y_i) розташована вище верхньої межі зони ε; ξ_i^* — величина, на яку точка (x_i, y_i) розташована нижче нижньої межі зони ε; w — вектор коефіцієнтів або ваг моделі; b — зміщення гіперплощини.

LinearSVR — це спрощений варіант методу Support SVR, що використовується для задач лінійної регресії. Цей метод націлений на побудову лінійної моделі, через що він швидший та ефективніший для великих наборів даних, де залежності між змінними є лінійними або майже лінійними [9]. Модель описується такою лінійною функцією:

$$y = w^T x + b, \tag{2}$$

де w — вектор коефіцієнтів або ваг моделі; x — вектор вхідних ознак моделі.

Метою алгоритму є знаходження таких значень w та b, які мінімізують кількість точок за межами гіперплощини та мінімізують функцію (1).

Nu-SVR (Nu-Support Vector Regression) — ще один варіант регресії на основі SVR, але з іншою методикою контролю якості моделі. В інших SVR методах параметр ε фіксований, а в Nu-SVR цей параметр залежить від ν, що регулює параметр ε та автоматично підлаштовується під дані. Параметр ν контролює максимальну кількість точок, що можуть бути опорними векторами, та максимальну кількість помилок (точок, що виходять за межі зони нечутливості ε) [10]. Наприклад, якщо дані містять багато шуму або складну структуру, великий параметр ν може забезпечити більшу кількість опорних векторів та дозволити більше похибок, щоб забезпечити кращу узгодженість моделі з даними. Якщо ж дані передбачувані та не містять багато шуму, менший параметр ν забезпечить меншу кількість опорних векторів, що звузить гіперплощину та дозволить досягти високої точності. Кількість опорних векторів регулюється функцією обмеження

$$\sum_{i=0}^{n}\left(\xi_i + \xi_i^*\right) \le \nu n, \tag{3}$$

де: ν — параметр регулювання кількості опорних векторів; n — загальна кількість об'єктів у навчальній вибірці.

PLSRegression (Partial Least Squares Regression) — метод регресійного аналізу, суть якого полягає у знаходженні набору латентних векторів з максимальною коваріацією вхідних даних X та вихідних даних Y [11]. Таким чином метод «стискає» вхідні дані, залишаючи максимально корисну інформацію для прогнозування вихідних даних. PLS регресія декомпозує X як [12]:

$$X = \mathrm{TP}^T, \tag{4}$$

де T — матриця оцінок, яка містить значення нових латентних змінних; P — матриця навантажень, яка містить міру впливу початкових даних на створення нових латентних змінних.

Подібним чином Y прогнозується як

$$\hat{Y} = \mathrm{TBC}^T, \tag{5}$$

де B — діагональна матриця з коефіцієнтами регресії для латентних змінних; C — матриця ваг для залежних (цільових) змінних.

Метод Kernel Ridge Regression (KRR) — це поєднання методів Ridge Regression та Kernel Trick. Він використовується для задач регресії, де зв'язок між ознаками та цільовою змінною є складним і нелінійним. Метою методу Ridge Regression є знаходження ваг w, які мінімізують функцію вартості [13]:

$$C(w) = \frac{1}{2}\sum_{i=1}^{n}\left(y_i - w^T x_i\right)^2 + \frac{1}{2}\lambda \ \|w\ \|^2, \tag{6}$$

де y — реальне значення цільової змінної; x — передбачення моделі; λ — параметр регуляризіції. Регуляризація додає штраф за великі значення ваг w, через що модель стійкіша до перенавчання.

Метод Kernel Trick [14] використовується для відображення вхідних ознак X у новому просторі за допомогою функції ядра (kernel function). Функція ядра визначає подібність між парами точок у цьому новому просторі, не вимагаючи явного обчислення координат точок. Функція ядра $K(x_i, x_i)$ відповідає скалярному добутку точок x_i та x_j у просторі ознак. Типовим прикладом є функція поліноміального ядра:

$$K\left(x_i, x_j\right) = \left(x_i x_j + c\right)^d, \tag{7}$$

де x_i та x_j — два вектори ознак (вхідні значення), скалярний добуток яких є мірою схожості між двома векторами у просторі ознак; c — константа зміщення, яка лінійно впливає на ядро. За великих значень c модель стає менш чутливою до малих відмінностей між x_i та x_j; d — ступінь полінома, якою задається складність поліноміальної трансформації. Коли d = 1, ядро еквівалентно лінійному ядру, більші значення d дозволяють моделювати складніші залежності між змінними, але робить модель схильнішою до перенавчання.

У методі Kernel Ridge Regression замість мінімізації функції вартості (6) ваги w подаються у вигляді коефіцієнтів Лагранжа [15]. Таким чином, розв'язок задачі оптимізації моделі подається у вигляді такої функції:

$$\alpha_i^* = \left(\mathrm{K} + \lambda \mathrm{I}\right)^{-1} y, \tag{8}$$

де K — матриця ядерних функцій для всіх пар точок; I — одинична матриця; y — вектор цільових значень.

RidgeCV — це метод Ridge-регресії з автоматичним вибором оптимального значення λ за допомогою крос-валідації. Автоматичний вибір λ дає змогу знайти найкращий баланс між регуляризацією та продуктивністю моделі. Щоб оцінити, яке значення λ забезпечує найкраще узагальнення моделі на незалежних даних, RidgeCV використовує крос-валідацію k-fold cross-validation [16].

Для цього дані $D = \left\{\left(x_1, y_1\right), \left(x_2, y_2\right), \ldots, \left(x_n, y_n\right)\right\}$ розділяються на K підмножини (folds) $D_1, D_2, \ldots, D_K$. Для кожної підмножини D_K використовується як тестова вибірка, а всі інші підмножини $D_1, \ldots, D_{k-1}, D_{k+1}, \ldots, D_K$ використовуються як вибірка для тренування. Далі проводиться K тренувань моделі на вибраних даних. Для кожної підмножини обчислюється середньоквадратична помилка

$$\mathrm{MSE}_k = \frac{1}{|D_k|}\sum_{i=1}^{|D_k|}\left(y_i - \hat{y}_i\right)^2, \tag{9}$$

де $|D_k|$ — кількість елементів у k-й підмножині; y_i — фактичне значення; $\hat{y}_i$ — передбачене значення.

Ефективність моделі визначається значенням середньої помилки [17]

$$\mathrm{CV}\left(\hat{f}\right) = \frac{1}{K}\sum_{k=1}^{K}\mathrm{MSE}_k, \tag{10}$$

де K — кількість підмножин.

BayesianRidge — це метод регресії, який на відміну від стандартної лінійної регресії, де оцінювання параметрів моделі відбувається за допомогою методів оптимізації, використовує імовірнісний підхід, припускаючи, що як параметри моделі, так і відгуки є випадковими величинами, що підпорядковуються певним розподілам ймовірності [18]. У рамках моделі передбачається, що дані y залежні від ознак X через лінійну функцію з додаванням нормального шуму. Параметри моделі, такі як коефіцієнти регресії та параметри шуму, мають апріорні розподіли, які потім оновлюються на основі даних з використанням апостеріорного розподілу

$$p(\beta \mid \varepsilon) \propto p(\varepsilon \mid \beta) \cdot p(\beta), \tag{11}$$

де p(β|ε) — апостеріорна ймовірність параметрів β, яка показує ймовірність різних значень β після отримання даних ε; $p(\varepsilon|\beta)$ — функція правдоподібності, яка показує ймовірність отримання даних ε при параметрах β; $p(\beta)$ — апріорна ймовірність параметрів β, яка відображає початкові припущення або знання щодо параметрів до того, як дані були отримані.

DecisionTreeRegressor (DTR) — метод машинного навчання, який оснований на принципі побудови дерева рішень. Алгоритм будує дерево рішень, ітеративно розділяючи дані на вузли, використовуючи середнє квадратичне відхилення (MSE) прогнозування як критерій для порівняння [19]. Для цього обчислюється MSE кожного вузла та загальне MSE

$$\mathrm{MSE}_{\mathrm{left}} = \frac{1}{N_{\mathrm{left}}} \sum_{i=1}^{N_{\mathrm{left}}} \left(y_i - \hat{y}_{\mathrm{left}}\right); \tag{12}$$

$$\mathrm{MSE}_{\mathrm{right}} = \frac{1}{N_{\mathrm{right}}} \sum_{i=1}^{N_{\mathrm{right}}} \left(y_i - \hat{y}_{\mathrm{right}}\right); \tag{13}$$

$$\mathrm{MSE}_{\mathrm{split}} = \frac{N_{\mathrm{left}}}{N} \mathrm{MSE}_{\mathrm{left}} + \frac{N_{\mathrm{right}}}{N} \mathrm{MSE}_{\mathrm{right}}, \tag{14}$$

де N — кількість елементів у наборів даних.

Метою цих обчислень є знаходження вузла, що мінімізує $\mathrm{MSE}_{\mathrm{split}}$.

Цей процес триває, поки не виконаються умови зупинки, такі як досягнення максимальної глибини дерева, мінімальної кількості об’єктів у вузлі або відсутності значного поліпшення MSE.

ExtraTreeRegressor (ETR) — метод машинного навчання, який також базується на принципі побудови дерева рішень, але, на відміну від DTR, який вибирає оптимальні вузли для мінімізації MSE. ETR випадковим чином вибирає поріг для поділу даних на кожному вузлі [20]. Поріг t вибирається випадково в межах вхідної ознаки X:

$$t \sim U\left(\min\left(X_i\right), \max\left(X_i\right)\right). \tag{15}$$

Після вибору t дані розбиваються на лівий ($X_i \le t$) та правий ($X_i > t$) вузли. Для оцінки якості розбиття використовуються MSE як у формулах (12), (13) і (14). Цей підхід збільшує різноманітність дерев у моделі та допомагає знизити перенавчання, через що метод стійкіший до шуму в даних.

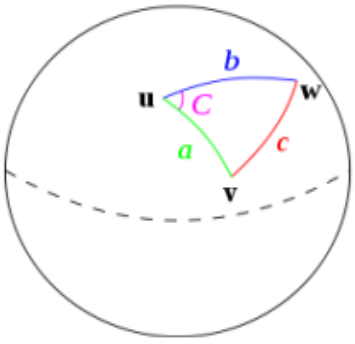

Рис. 1. Метод гаверсинусу, застосований для земної кулі

Задача методів регресії полягає у прогнозування неперервного числового значення. Як видно з табл. 1, до таких даних відносяться лише Longitude та Latitude. Проте за допомогою формули гаверсинуса ці координати можна об’єднати у нове значення дистанції. Метод гаверсинуса розраховує дистанцію між двома координатами на поверхні земної кулі по прямій лінії (рис. 1) [21].

На рис. 1 показано три координати u, v, та w. За допомогою методу гаверсинусу можливо обчислити дистанцію на поверхні сфери між цими точками. В загальному вигляді метод гаверсинусу записується так [21]:

$$\alpha = \sin^2\left(\frac{\Delta\varphi}{2}\right) + \cos\varphi_1 \cdot \cos\varphi_2 \cdot \sin^2\left(\frac{\Delta\lambda}{2}\right); \tag{16}$$

$$c = 2 \cdot \operatorname{atan}2\left(\sqrt{\alpha}, \sqrt{1-\alpha}\right); \tag{17}$$

$$d = R \cdot c, \tag{18}$$

де φ — широта; λ — довгота; R — радіус Землі (6371 км).

Проте за методом гаверсинусу обчислюється дистанція між двома точками, а в контексті рядка даних з датасету є координати лише однієї точки. Рішенням цієї проблеми буде використання однакової для всіх рядків другої координати, дистанція до якої буде вказувати на місцезнаходження людини. Якщо за таку фіксовану координату використати точку (0,0) та підставити ці значення у формулу (16), то вона спроститься та набуде такого вигляду:

$$\alpha = \sin^2\left(\frac{\varphi}{2}\right) + \cos\varphi \cdot \sin^2\left(\frac{\lambda}{2}\right). \tag{19}$$

Таким чином можна отримати дистанцію між координатами (x, y) та (0,0), тобто маючи точки лише однієї координати. За допомогою формул (19), (17) та (18) дані Longitude та Latitude перетворені у нове неперервне значення distance, яке буде використовуватися для прогнозування місцезнаходження людини. Структура оновлених даних подана у табл. 3.

Таблиця 3

Оновлена структура даних датасету для регресії

№	WAP001	WAP002	WAP003	...	WAP520	distance
1	100	100	100	...	100	13069,52
2	100	100	100	...	100	11614,32
3	100	100	100	...	100	10046,89
4	100	100	100	...	100	11442,45
5	100	100	100	...	100	8961,35

Використовуючи дані WAP001…520 як незалежні дані (x) та дані distance як залежні дані (y) побудовано моделі регресії методами SVR, LinearSVR, NuSVR, PLSRegression, KernelRidge, RidgeCV, BayesianRidge, DecionTreeRegressor та ExtraTreeRegressor. Для оцінки точності прогнозування моделей використано метод коефіцієнта детермінації [22]

$$r_y^2 = 1 - \frac{\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2}{\sum_{i=1}^{n} y_i^2}, \tag{20}$$

де r_y^2 — точність прогнозування; $\hat{y}_i$ — прогнозоване значення; y_i — фактичне значення.

Аргументи моделей регресії та точність прогнозування distance для цих аргументів подано у табл. 4.

Таблиця 4

Результати прогнозування distance різними методами регресії

Метод	Аргументи	Точність r_y^2, %
SVR	C = 1000, epsilon=1	90
LinearSVR	C = 1000, epsilon=1	85,8
NuSVR	Nu = 0,35, C=1000	90
PLSRegression	n_components = 10	86,8
KernelRidgeRegression	kernel = poly, degree = 7	93,9
RidgeCV	—	86,3
BayesianRidge	—	87
DecisionTreeRegressor	splitter = ‘random’	92,5
ExtraTreeRegressor	—	95,5

Як видно з табл. 4, найкращий результат отримано методами ExtraTreeRegressor, DecisionTree-Regressor та KernelRidgeRegression. Проте для використання методів регресії необхідне числове

значення distance, яке отримується з даних Longitude та Latitude, тобто для навчання моделям регресії необхідні координати місцезнаходження людини. Ця вимога накладає обмеження на вибір технічних засобів для збору даних. При цьому різні технічні засоби мають свої переваги і недоліки та доцільні у використанні в певних умовах [2].

Приклад технології, яка може надавати поточні координати — смартфон. На території університету розміщуються точки доступу Wi-Fi, з яких можна отримувати дані про активні пристрої, що їх оточують. В момент під'єднання до мережі смартфон надсилає свої ідентифікаційні дані та поточні координати у систему, після чого можна прогнозувати місцезнаходження студента.

Якщо ж розглядати інші технології, до прикладу, RFID, то до корисних даних, які можна використати для прогнозування місцезнаходження студента, відносяться лише RSSI (Received Signal Strength Indicator) [23]. Принцип роботи схожий на метод з Wi-Fi — на території університету розміщуються UHF RFID трансивери (ультрависокочастотні RFID зчитувачі), які сканують своє оточення в радіусі до 10 м [24]. Якщо під час сканування RFID мітка знаходиться в радіусі сканера, сканер зчитує ідентифікаційні дані з мітки.

В такому разі місцезнаходження студента можна було б встановлювати лише вираховуючи той факт, що він знаходиться у радіусі сканера, але тут виникає дві можливі проблеми: сканер може покривати декілька приміщень одночасно та студент може одночасно знаходитися в радіусі двох або більше сканерів. При цьому, в залежності від особливостей розташування датчиків, найбільша сила сигналу RSSI не обов'язково може свідчити про знаходження людини в одному приміщенні зі сканером (наприклад, людина в коридорі або близько до сканера з іншої сторони стіни).

Ці ж проблеми можуть виникнути і у разі використання Wi-Fi технологій, але в такому випадку смартфон генерує неперервні числові дані (координати), які можна використати для задач регресії. Щоб вирішити цю проблему в умовах, коли такі дані відсутні, можна використати машинне навчання методами багатокласової класифікації [25]. В цьому дослідженні використано такі методи класифікації: SVC, KNeighborsClassifier, RandomForestClassifier та DecisionTreeClassifier.

Support Vector Classification (SVC) — це метод машинного навчання, який оснований на концепції опорних векторів. У своїй роботі SVC будує гіперплощину, яка, на відміну від методу SVR, не апроксимує функцію, що описує залежність вихідних даних від вхідних, а відділяє один від одного різні класи даних. Задача оптимізації моделі полягає у максимізації маржі [26] (відстані між класами на гіперплощині), що виконується за рахунок мінімізації такої функції:

$$\min \frac{1}{2} w^T w + C \sum_{i=1}^{n} \xi_i \text{ за умов } y_i (w x_i + b) \geq 1, \ \ \xi_i \geq 0, \forall i. \tag{21}$$

KNeighborsClassifier — це метод машинного навчання, який базується на використанні алгоритму K-Nearest Neighbors. Алгоритм основується на ідеї, що об'єкти, розташовані близько один до одного в просторі ознак, найімовірніше належать до одного класу. Об'єкту призначається той клас, який є найпоширенішим серед сусідів цього елемента, класи яких уже відомі [27]. Типовою метрикою відстані, яку використовують для оцінки відстані між об'єктами, є евклідова відстань

$$d(x, y) = \sqrt{\sum_{i=1}^{n} \left(x_i - y_i \right)^2}, \tag{22}$$

де (x, y) — координати у просторі; n — кількість вимірювань.

DecisionTreeClassifier — метод машинного навчання, який оснований на принципі побудови дерева рішень, яке рекурсивно розділяє простір ознак, вибираючи на кожному кроці оптимальну ознаку для поділу [28]. Оптимальною ознакою вважається та, яка формує найменшу невизначеність моделі. Для оцінки невизначеності моделі використовується критерій поділу. Одним з типових критеріїв є Gini Impurity, який означає міру невпорядкованості у вузлі дерева, тобто міру ознак різних класів в одному вузлі:

$$\text{Gini} = 1 - \sum_{i=1}^{n} p_i^2, \tag{23}$$

де p_i — частка ознак, які належать до класу i; n — кількість класів, присутніх у вузлі.

Процес розбиття вузлів триває до досягнення певних умов: максимальної глибини дерева,

мінімальної кількості об'єктів у вузлі або досягнення максимальної впорядкованості вузла (коли всі ознаки в вузлі належать до одного класу). Після навчання модель може бути використана для класифікації нових даних, де дерево проводить серію перевірок ознак і наприкінці присвоює обє'кту клас, відповідний листу, до якого він потрапив.

RandomForestClassifier — алгоритм машинного навчання, заснований на методі ансамблів, який об'єднує кілька дерев рішень з більшою стійкістю та точністю прогнозування, ніж у кожного окремого дерева [29]. На кожному кроці розбиття вузлів у кожному дереві випадково вибирається підмножина ознак. Такий випадковий розподіл зменшує кореляцію між деревами, адже кожне дерево буде побудовано на різних даних. Після цього нові дані класифікуються поєднанням прогнозів усіх дерев. Для цього кожне дерево визначає належність об'єкта до певного класу, після чого вибирається прогноз, який отримав найбільшу кількість голосів:

$$\hat{y} = \arg\max_k \sum_{i=1}^{T} I\left(y_i = k\right), \tag{24}$$

де y_i — клас, який прогнозувало i-те дерево в ансамблі; T — кількість дерев в ансамблі; k — можливий клас у задачі класифікації; I — індикаторна функція, яка повертає 1, якщо $y_i = k$ правдиве та 0 — якщо неправдиве; $\hat{y}$ — кінцевий прогноз.

Задача методів класифікації полягає у прогнозуванні дискретного числового значення. Як видно з табл. 1, до таких даних відносяться BuildingID, Floor та SpaceID. Структура даних датасету після видалення надлишкових стовбців подана у табл. 5:

Таблиця 5

Оновлена структура даних датасету для класифікації

№	WAP001	...	WAP520	FLOOR	BUILDINGID	SPACEID
1	100	...	100	2	1	106
2	100	...	100	2	1	106
3	100	...	100	2	1	103
4	100	...	100	2	1	102
5	100	...	100	0	0	122

Використовуючи дані WAP001…520 як незалежні дані (x) та дані SpaceID як залежні дані (y), побудовано моделі класифікації методами: SVC, KNeighborsClassifier, RandomForestClassifier та DecisionTreeClassifier. Для оцінки точності прогнозування моделей використано метод Accuracy score [30]

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}, \tag{25}$$

де TP — true positives, кількість об'єктів, які правильно спрогнозовано як істині; TN — true negatives, кількість об'єктів, які правильно спрогнозовано як хибні; FP — false positives, кількість об'єктів, які помилково спрогнозовано як істині; FN — false negatives, кількість об'єктів, які помилково спрогнозовано як хибні.

Таблиця 6

Результати прогнозування SpaceID різними методами класифікації

Метод	Accuracy, %
SVC	75
KNeighborsClassifier	68,3
RandomForestClassifier	85
DecisionTreeClassifier	64,9

Результати прогнозування SpaceID різними методами класифікації подано у табл. 6.

Якщо порівняти результати з табл. 4 та 6, то видно, що методи класифікації мають значно меншу точність прогнозування. Можливою проблемою може бути недостатня репрезентативність даних, тобто модель, натренована на такій вибірці даних, буде недостатньо ефективною для нових даних [31].

Після аналізу вибірки даних дійшли висновку — під час навчання використовується багато даних, які не несуть корисної інформації. Повертаючись до табл. 1, вибірка складається з таких даних: WAP001-WAP520, Longitude, Latitude, Floor, BuildingID, SpaceID, RelativePosition, UserID, PhoneID, Timestamp. З них для класифікації використано лише WAP001-WAP520 (сила Wi-Fi сигналу) як вхідні ознаки та SpaceID (номер приміщення) як вихідна ознака. Проте точки

доступу Wi-Fi розташовані у різних будівлях.

Висунуто гіпотезу, що навчання моделі на даних певної будівлі деякою мірою спотворює прогнозування для інших будівель. Для перевірки гіпотези вирішено змінити підхід до навчання моделі. Замість того, щоб надавати моделі одразу всі дані, процес буде поділено на кроки та різні моделі.

Першим кроком буде тренування моделі на всіх даних WAP, але замість SpaceID буде прогнозуватися BuildingID. Далі з вибірки видаляються всі дані WAP, які не мають відношення до прогнозованої будівлі. На цих даних тренується наступна модель, яка буде прогнозувати Floor. Аналогічно далі будуть видалені всі дані, які не належать до прогнозованого поверху. Останнім кроком буде тренування моделі на даних WAP з конкретного поверху та прогнозування SpaceID на цьому поверху. Цей алгоритм показано на UML-діаграмі діяльності (рис. 2).

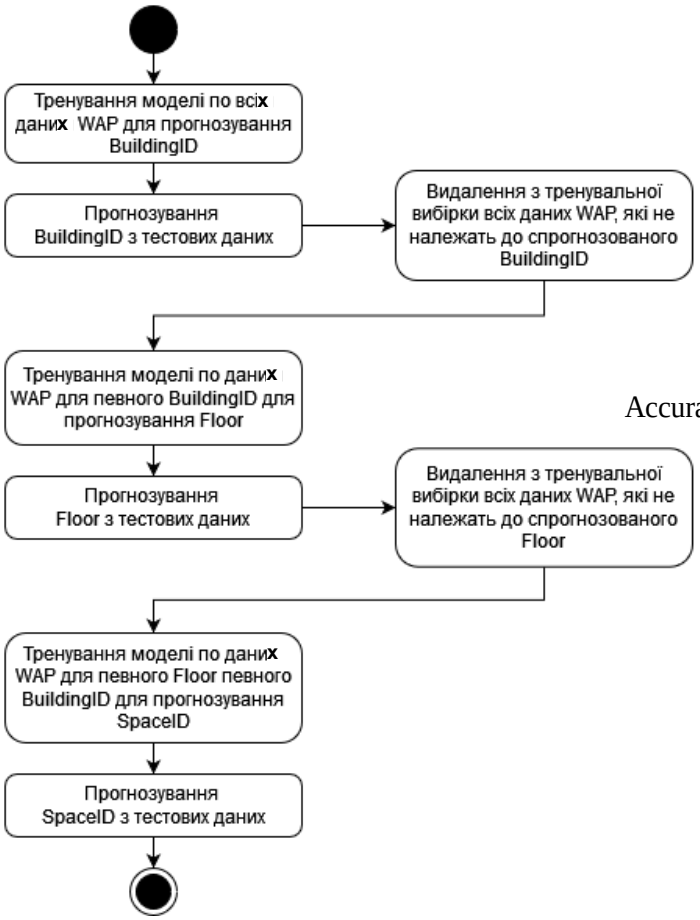

Рис. 2. UML-діаграма діяльності покрокового алгоритму класифікації машинним навчанням

Оскільки в цьому алгоритмі тренуються три різні моделі та проводяться три послідовних прогнозування, то загальна точність прогнозування розраховується як добуток трьох точностей:

$$\text{Accuracy} = \text{Accuracy}_b \cdot \text{Accuracy}_f \cdot \text{Accuracy}_s, \quad (26)$$

де Accuracy_b — точність прогнозування будівлі; Accuracy_f — точність прогнозування поверху; Accuracy_s — точність прогнозування приміщення.

Використовуючи дані WAP001…520 як незалежні дані (x) покроково спрогнозовано значення залежних даних (y) BuildingID, Floor та SpaceID методами класифікації SVC, KNeighborsClassifier, RandomForestClassifier та DecisionTreeClassifier. Результат роботи методів класифікації за новим алгоритмом подано у табл. 7.

Таблиця 7

Результати покрокового прогнозування BuildingID, Floor та SpaceID методами класифікації

Метод	Точність будівлі Accuracy_b, %	Точність поверху Accuracy_f, %	Точність приміщення, Accuracy_s, %	Складена точність Accuracy, %
SVC	99,7	99,3	79,9	79,1
KNeighborsClassifier	99,7	98,7	71,6	70,4
DecisionTreeClassifier	99,7	96,8	72,8	70,25
RandomForestClassifier	99,7	99,6	95	94,3

Результати дослідження

Серед розглянутих методів машинного навчання (SVR, LinearSVR, NuSVR, PLSRegression, KernelRidge, RidgeCV, BayesianRidge, DecionTreeRegressor та ExtraTreeRegressor) найкращі результати для прогнозування параметра distance отримано методами ExtraTreeRegressor (95,5 %), DecisionTreeRegressor (92,5 %) та KernelRidgeRegression (93,9 %). Точність прогнозування distance всіма розглянутими методами регресії показано на рис. 3:

Проте методи регресії орієнтовані на прогнозування неперервних числових значень (координ-

нат). Залежність моделі від неперервних даних накладає певні обмеження. Як альтернатива розглянуто такі методи класифікації для прогнозування SpaceID: SVC, KNeighborsClassifier, DecisionTreeClassifier і RandomForestClassifier.

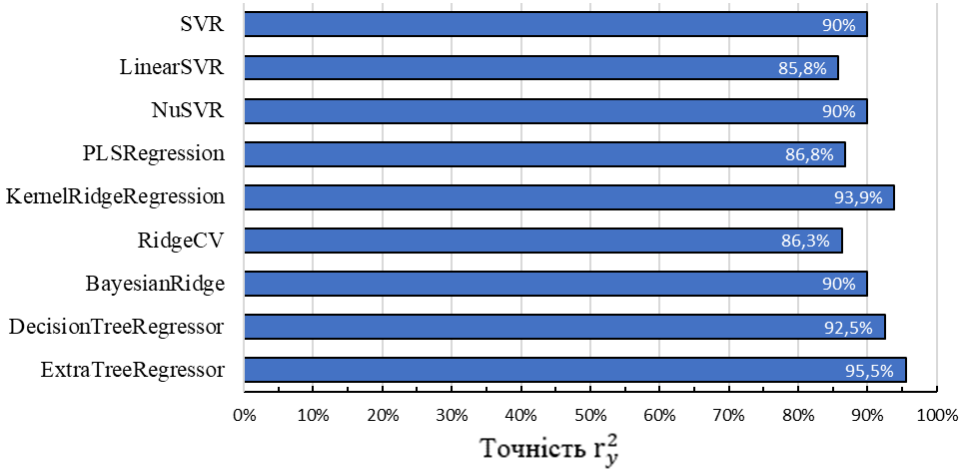

Рис. 3. Точність прогнозування distance методами регресії

Первинні результати показали незадовільну точність прогнозування — від 64,9 % до 85 %. Проте у результаті покрокової класифікації будівлі, поверху і приміщення вдалося поліпшити результат — від 70,25 % до 94,3 %. Найліпший результат отримано методом RandomForestClassifier (94,3 %). Точність прогнозування SpaceID всіма розглянутими методами класифікації показано на рис. 4.

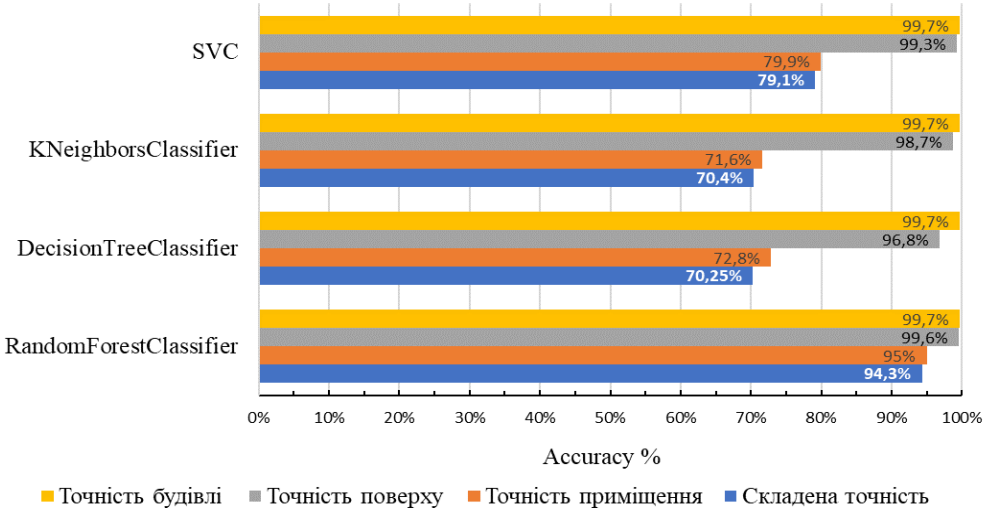

Рис. 4. Точність прогнозування BuildingID, Floor та SpaceID методами класифікації

Висновки

В дослідженні розглянуто задачу оброблення даних для автоматичної фіксації присутності студентів на заняттях. Як оптимальну методологію для розв'язання цієї задачі вибрано машинне навчання, оскільки воно дозволяє ефективно прогнозувати аудиторію, в якій перебуває студент, навіть за наявності непередбачуваних аномалій в отриманих даних. До того ж у процесі фіксації присутності студентів формуються списки, які можуть бути використані як вибірки даних для машинного навчання.

Досліджено принципи роботи таких методів машинного навчання для задач регресії: SVR, LinearSVR, NuSVR, PLSRegression, KernelRidge, RidgeCV, BayesianRidge, DecionTreeRegressor та ExtraTreeRegressor. Встановлено, що зазначені методи регресії забезпечують достатньо велику точність прогнозування, але вони орієнтовані на прогнозування неперервних числових значень (координат). Таким чином, застосування методів регресії накладає низку обмежень на вибір технічних засобів, оскільки координати можуть бути отримані тільки з використанням певних технологій.

Як альтернативний метод аналізу даних машинним навчанням вибрано класифікацію. Методи класифікації призначені для прогнозування дискретних значень, до яких у контексті задачі визначення місцезнаходження людей у приміщеннях відносяться номери будівель, поверхів і приміщень. Такі дані можуть бути отримані з використанням простіших технологій.

Досліджено принципи роботи таких методів класифікації: SVC, KNeighborsClassifier, DecisionTreeClassifier і RandomForestClassifier. Первинні результати моделювання показали незадовільну точність прогнозування. Припущено, що причиною є недостатня репрезентативність тренувальних даних. Для поліпшення репрезентативності вирішено класифікувати дані покроково — спочатку будівля, потім поверх і потім приміщення. На кожному кроці відсікалися дані WAP, які не належать до спрогнозованих значень. Використання покрокового алгоритму підвищило точність прогнозування. Підсумкова точність класифікації стала співмірною з точністю регресії.

Таким чином, дослідження показало, що вибір методів машинного навчання залежить від використовуваних технічних засобів. Якщо вони дають змогу отримувати координати людини, можна застосовувати методи регресії, з яких найліпші результати отримано методами ExtraTreeRegressor, DecisionTreeRegressor і KernelRidgeRegression. Якщо ж технічні засоби не дозволяють отримувати координати, альтернативним рішенням є використання методів класифікації з покроковим алгоритмом, серед яких найкращий результат показав RandomForestClassifier.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

[1] A. Ademola, T. E. Somefun, and A. Oluwabusola, “Web based fingerprint roll call attendance management system,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 9, no. 5, pp. 4364-4371, Oct. 2019. https://doi.org/10.11591/ijece.v9i5.pp4364-4371 .

[2] А. І. Топольський, і Є. А. Паламарчук, «Аналіз практичних реалізацій автоматизованих систем ідентифікації студентів в електронних навчальних системах,» *Вісник Вінницького політехнічного інституту*, № 2, с. 61-70, 2024. https://doi.org/10.31649/1997-9266-2024-173-2-61-70 .

[3] Ji-Hyun Yoo, “Study on Prediction of Attendance Using Machine Learning,” *Journal of IKEEE*, vol. 23, no. 4, pp. 1243-1249, 2019. https://doi.org/10.7471/ikeee.2019.23.4.1243 .

[4] *UjiIndoorLoc: An indoor localization dataset*. [Electronic resource]. Available: https://www.kaggle.com/datasets/giantuji/UjiIndoorLoc/data .

[5] *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, Banff, AB, Canada, 13-16 October 2015. https://doi.org/10.1109/ipin.2015.7346747 .

[6] E. Hossain, “Machine learning algorithms,” in *Machine Learning Crash Course for Engineers*, Cham: Springer Int. Publishing, 2023, pp. 117-140. https://doi.org/10.1007/978-3-031-46990-9_3 .

[7] K. Tewari, S. Vandita, and S. Jain, “Predictive analysis of absenteeism in MNCS using machine learning algorithm,” in *Lecture Notes in Electrical Engineering*, Cham: Springer Int. Publishing, 2019, pp. 3-14. https://doi.org/10.1007/978-3-030-29407-6_1 .

[8] D. Basak, S. Pal, and D. Patranabis, “Support Vector Regression,” *Neural Information Processing – Letters and Reviews*, vol. 11, no. 10, pp. 203-224, 2007.

[9] I. Ali, “Machine Learning models and feature relevance for student grade prediction,” *Journal of High School Science*, vol. 8, no. 1, pp. 180-197, 2024.

[10] I. Aqeel, “Location fingerprinting for IoT systems using machine learning.” dissert. Dr. Sc. (Eng.), Glasgow, Scotland, 2020. https://doi.org/10.48730/1jdb-6258 .

[11] C. Wu, Z. Yang, and C. Xiao, “Automatic Radio Map Adaptation for Indoor Localization Using Smartphones,” *IEEE Transactions on Mobile Computing*, vol. 17, no. 3, pp. 517-528, 2018. https://doi.org/10.1109/tmc.2017.2737004 .

[12] A. Höskuldsson, “PLS regression methods,” *Journal of Chemometrics*, vol. 2, no. 3, pp. 211-228, 1988. https://doi.org/10.1002/cem.1180020306 .

[13] M. Welling, *Kernel ridge regression: Notes for a lecture on machine learning*. Toronto: University of Toronto, Department of Computer Science, 2013, 3 p.

[14] P. Exterkate, et al., “Nonlinear forecasting with many predictors using kernel ridge regression,” *International Journal of Forecasting*, vol. 32, no. 3, pp. 736-753, 2016. https://doi.org/10.1016/j.ijforecast.2015.11.017 .

[15] V. Siahaan, *Data visualization, time-series forecasting, and prediction using machine learning with tkinter, Balige:* BALIGE Publishing Ltd., 2023, 266 p.

[16] Y. Jung, “Multiple predicting K-fold cross-validation for model selection,” *Journal of Nonparametric Statistics*, vol. 30, no. 1, pp. 197-215, 2017. https://doi.org/10.1080/10485252.2017.1404598 .

[17] A. Verma, and M. Jain, “Mediation Analysis of Diabetes and Heart Diseases Influenced by Obesity Using Machine Learning Classifiers,” *Austrian Journal of Statistics*, vol. 53, № 5, pp. 90-111, 2024.

[18] B.-W. Chen, et al., “Efficient multiple incremental computation for Kernel Ridge Regression with Bayesian uncertainty modeling,” *Future Generation Computer Systems*, vol. 82, pp. 679-688, 2018. https://doi.org/10.1016/j.future.2017.08.053 .

[19] Judit Kunné Tamás, “Classification based Symbolic Indoor Positioning.” Doctoral dissertation, Debrecen, 2021, 127 p.

[20] T. Aziz, and K. Insoo, “Enhancing Indoor Localization Accuracy through Multiple Access Point Deployment,” *Electronics*, vol. 13, no. 16, pp. 3307, 2024. https://doi.org/10.3390/electronics13163307 .

[21] Dwi Arman Prasetya, et al., “Resolving the Shortest Path Problem using the Haversine Algorithm,” *Journal of Critical Review*, vol. 7, no. 1, pp. 62-64, 2020.

[22] O. Renaud, and M.-P. Victoria-Feser, “A robust coefficient of determination for regression,” *Journal of Statistical Planning and Inference*, vol. 140, no. 7, pp. 1852-1862, 2010. https://doi.org/10.1016/j.jspi.2010.01.008 .

[23] N. Singh, S. Choe, and R. Punmiya, “Machine Learning Based Indoor Localization Using Wi-Fi RSSI Fingerprints: an Overview,” *IEEE Access*, vol. 9, pp. 127150-127174, 2021. https://doi.org/10.1109/access.2021.3111083 .

[24] J. Uddin, “UHF RFID antenna architectures and applications,” *Scientific Res. Essays*, vol. 5, no. 10, pp. 1033-1051, 2010.

[25] W. Charoenruengkit, et al., “Position Quantization Approach with Multi-class Classification for Wi-Fi Indoor Positioning System,” in *2018 International Conference on Information Technology (InCI*T), Khon Kaen, 24-26 October 2018. https://doi.org/10.23919/incit.2018.8584863 .

[26] Z. Chunhong, J. Licheng, and L. Yongzhao, “Support vector classifier based on principal component analysis,” *Journal of Systems Engineering and Electronics*, vol. 19, no. 1, pp. 184-190, 2008. https://doi.org/10.1016/s1004-4132(08)60065-1 .

[27] S. Dhanabal, Dr. S. Chandramathi, “A Review of various k-Nearest Neighbor Query Processing Techniques,” *International Journal of Computer Applications*, vol. 31, no. 7, pp. 14-22, 2011.

[28] S. Milinković, and M. Maksimović, “Using Decision Tree Classifier for Analyzing Students’ Activitie,” *JITA, Journal of Information Technology and Applications* (Banja Luka), *APEIRON*, vol. 6, no. 2, pp. 82-95, 2013. https://doi.org/10.7251/jit1302087m .

[29] Y. Wang, et al., “WiFi Indoor Localization with CSI Fingerprinting-Based Random Forest,” *Sensors*, vol. 18, no. 9, pp. 2869, 2018. https://doi.org/10.3390/s18092869 .

[30] M. Sokolova, N. Japkowicz, and S. Szpakowicz, “Beyond Accuracy, F-Score and ROC: A Family of Discriminant Measures for Performance Evaluation,” *AI 2006: Advances in Artificial Intelligence: Australian Joint Conference on Artificial Intelligence*, Hobart, 4-8 December 2006, Heidelberg, 2006, pp. 1015-1021.

[31] E. Schat, et al., “The data representativeness criterion: Predicting the performance of supervised classification based on data set similarity,” *PLOS ONE*, vol. 15, no. 8, pp. 1-16, 2020. https://doi.org/10.1371/journal.pone.0237009 .

Рекомендована кафедрою автоматизації та інтелектуальних інформаційних технологій ВНТУ

Стаття надійшла до редакції 8.01.2025

Топольський Андрій Іванович — аспірант кафедри автоматизації та інтелектуальних інформаційних технологій, e-mail: topolskiy.vntu@gmail.com ;

Паламарчук Євген Анатолійович — канд. техн. наук, доцент, професор кафедри автоматизації та інтелектуальних інформаційних технологій, e-mail: p@vntu.edu.ua .

Вінницький національний технічний університет, Вінниця

A. I. Topolskiy[1]
Ye. A. Palamarchuk[1]

Using Machine Learning to Locate People Indoors

[1]Vinnytsia National Technical University

The article investigates the problem of automated data processing to record the presence of students in classes. It is proposed to use machine learning methods, since they allow predicting the location of students in the premises even in the presence of anomalies in the data. The solution to this problem will help to increase the efficiency of the educational process and reduce dependence on traditional methods of recording presence, which require time and human resources.

Experiments were conducted using various machine learning methods for regression and classification tasks. Prediction accuracy was used as a measure to compare different methods.

Among the regression methods, the following were considered: SVR, LinearSVR, NuSVR, PLSRegression, KernelRidge, RidgeCV, BayesianRidge, DecisionTreeRegressor, and ExtraTreeRegressor. The best accuracy was obtained by DecisionTreeRegressor, KernelRidgeRegression and ExtraTreeRegressor methods — 92.5, 93.9 and 95.5 %, respectively. However, regression methods require continuous data, such as user coordinates, which limits their use in environments where technical means do not allow obtaining such data.

As an alternative, classification methods were considered, namely: SVC, KNeighborsClassifier, DecisionTreeClassifier and RandomForestClassifier. The initial results showed lower accuracy compared to regression methods, which was due to the lack of representativeness of the training data. To solve this problem, a step-by-step algorithm was applied, which gradually predicts the building, floor and specific room. This algorithm led to a significant improvement in accuracy, with the best result being achieved by the RandomForestClassifier method — 94.3 %.

It was concluded that the choice of a machine learning method depends on the technical means used. If they allow you to obtain continuous data, such as coordinates, it is optimal to use the ExtraTreeRegressor, DecisionTreeRegressor, or KernelRidgeRegression regression methods. If continuous data cannot be obtained, it is optimal to use the RandomForestClassifier classification method with the proposed step-by-step algorithm.

Keywords: automated attendance systems, e-learning systems, machine learning, classification methods, regression methods, indoor people localization.

Topolskiy Andriy I. — Post-Graduate Student of the Chair of Automation and Intelligent Information Technologies, e-mail: topolskiy.vntu@gmail.com ;

Palamarchuk Yevhen A. — Cand. Sc. (Eng.), Associate Professor, Professor of the Chair of Automation and Intelligent Information Technologies, e-mail: p@vntu.edu.ua