

ЕЛЕКТРОНІКА

УДК 004.056.53

Крилов А.В.

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

Ямненко Ю.С.

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

ЗАСТОСУВАННЯ БІБЛІОТЕКИ MACHINE LEARNING ДЛЯ СИСТЕМИ РОЗПОДІЛЕНОЇ ГЕНЕРАЦІЇ MICROGRID

У статті викладено принципи побудови системи розподіленої генерації MicroGrid із застосування бібліотеки Machine Learning (машинного навчання), оскільки система MicroGrid з IoT дає велику кількість даних, яку треба обробляти для реалізації єдиної стратегії інтелектуального й енергоефективного керування, а також наявність «людського фактору» зумовлюють застосування методів машинного навчання, а саме методу Support Vector Machine (метод прикладних векторів)

Ключові слова: MicroGrid, Machine Learning, Support Vector Machine, метод, бібліотека, штучний інтелект, вибірка, вектор.

Постановка проблеми. Задача розробки сучасних автоматизованих систем керування та прийняття рішень для систем розподіленої генерації MicroGrid у наш час є досить актуальною. При цьому виникає питання майже повної автоматизації системи MicroGrid із мінімальним втручанням людини. Це призведе до мінімізації фізичних й енергетичних затрат на функціонування будинку. Тож введення до системи MicroGrid з концепцією Internet of Things бібліотек і методів «Machine learning» дасть змогу відкрити нові способи енергозбереження чи ефективного керування усього дому.

Такий підхід забезпечує не тільки «інтелектуалізацію», а й подальше навчання дому нових функцій без втручання інженерів чи програмістів. Такий метод контролю дає змогу впровадити «штучний інтелект» у повсякденне життя. Важливим є залучення алгоритмів машинного навчання, а саме – метод опорних векторів. Завдяки цьому методу машина дасть змогу виконувати не тільки примітивні задачі, а й замінить людину в побутових проблемах чи навіть допоможе в екстрених ситуаціях.

Виклад основного матеріалу дослідження.

Система MicroGrid

MicroGrid є інноваційною концепцією малої розподіленої енергетики, що передбачає створення локальних мережевих енергетичних струк-

тур. Уніфікована структура системи MicroGrid включає акумуляторні батареї, перетворювальні пристрої напруги, контролери заряду/розряду, різномісні генератори (альтернативні та відновлювальні джерела енергії), а також навантаження. Наявність альтернативних джерел дає змогу забезпечити безперебійність роботи при відключенні централізованої мережі живлення або паралельно з нею. При тривалій відсутності напруги мережі та недостатності енергії від альтернативних джерел відбувається відключення електротехнічних пристроїв з урахуванням їх пріоритетів [1].

Machine Learning

Машинне навчання – це підмножина штучного інтелекту в області інформатики, яка часто використовує статистичні методи, щоб дати комп'ютерам можливість, поступово покращувати продуктивність у конкретними операціями з даними, без явного запрограмування.

Машинне навчання тісно пов'язане з обчислювальною статистикою, яка також зосереджується на прогнозуванні шляхом застосування комп'ютерів. Воно має тісні зв'язки з математичною оптимізацією, яка забезпечує цю галузь методами, теорією та прикладними областями. Машинне навчання іноді об'єднують із добуванням даних, де друга підгалузь фокусується більше на розвідувальному аналізі даних.

Том Мітчелл запровадив широко цитоване формальніше визначення алгоритмів, які досліджують у галузі машинного навчання: «Кажуть, що комп'ютерна програма вчиться з досвіду E щодо якогось класу задач T та міри продуктивності P , якщо її продуктивність у задачах із T , вимірювана за допомогою P , покращується з досвідом E ». Це визначення задач, якими займається машинне навчання, пропонує фундаментально операційне визначення замість того, щоб визначати цю галузь у когнітивних термінах [2].

Для використання такої технології для MicroGrid треба обрати напрям розрахунку ймовірностей зміни поведінки машини в залежності від потреб користувача. Наприклад, щоб машина сама знала, коли людині самій треба підняти температуру повітря чи вимкнути освітлення тощо.

Незважаючи на активний розвиток таких технологій, система MicroGrid передбачає жорстку взаємодію з командами людини та невелику частку автоматизації дій. Однак така система має вміння підлаштовуватися під людину і надавати їй доступ до реалізації її потреб практично без її участі.

Для вирішення цієї проблеми використовується впровадження технологій машинного навчання в керування системою.

Першою дією, яку має робити ця система, є визначення наявності власника будинку [1].

Можливими способами відстеження є наведені нижче.

1. Явний вхід за допомогою будь-якої RFID технології, при цьому вхідні двері обладнані електронним замком або входом за паролем.

2. Аутентифікація через смартфон, який завжди з собою

2.1. Bluetooth завдяки малому радіусу дії підходить дуже добре. Достатньо виявити смартфон із необхідною міткою в радіусі дії і можна зробити висновок, що хазяїн близько.

2.2. Wi-Fi. Кожен смартфон обладнаний унікальним MAC-адресою, типовий сценарій використання означає активність модуля Wi-Fi при приході до дому. Наявність у локальній мережі необхідного пристрою дає інформацію про місцезнаходження будинку.

2.3. Додаток на смартфоні. Можна налаштувати смартфон виконати певні дії при знаходженні в зоні з потрібною Wi-Fi-міткою, необхідною комбінацією доступних GSM вишок тощо.

Для всіх цих типів відстеження використовується метод опорних векторів. На основі отриманих даних можна будувати функції системи, які

залежать від результатів першочергового алгоритму. Наприклад, підключивши систему безпеки, можна дізнатися, чи не зайшла до будинку стороння людина. Також при знаходженні в приміщенні кількох людей можна навчити машину визначати оптимальний клімат для присутніх.

У разі, коли система максимально наближена до ідеальної, тобто в 9 з 10 прогнозувань всі прогнози сходяться, MicroGrid здатна прогнозувати аномальну поведінку людини, викликати допомогу чи якщо можна, надати допомогу самій.

Будь-яка програма являє собою певний алгоритм дій, який детально, послідовно роз'яснює, що треба виконати машині. Але в галузі машинного навчання все не зовсім так. Алгоритми машинного навчання вгадують все самі, зробивши висновок на основі отриманих даних. Чим більше алгоритм отримує такі дані, тим кращим він стає і адаптується. По суті, ці алгоритми програмують самі себе. Це дає змогу порівняно невеликій програмі (приблизно кілька сотень строк коду) генерувати складні і величезні алгоритми дії залежно від різних умов.

У наш час вони отримали значне поширення і вже ввійшли в наше життя. Хоча, звичайно, вони можуть бути невидимими з боку, і людина навіть не здогадується про їх вплив, але більшість інформаційних потоків, з якими взаємодіє людина, залежить від таких алгоритмів. В ідеальному вигляді машинне навчання дає змогу створити єдиний загальний алгоритм, який може описувати абсолютно все. Але в житті будь-якого алгоритму стикається з надто складними для нього задачами і в якийсь момент перестає ефективно діяти. І, може, знайдеться єдиний алгоритм, який зможе ознайомити всіх, але він, напевно, являтиме собою об'єднання основних типів алгоритмів, відомих нині.

Поняття MicroGrid з Machine learning

Наявна модель MicroGrid, обладнана пакетом Machine Learning, являє собою машину, яка працює за методом «навчання з вчителем». Навчання з учителем (англ. *Supervised learning*) – один зі способів машинного навчання, у процесі якого випробувана система примусово навчається за допомогою наявної множини прикладів «стимул-реакція» з метою визначення «реакції» (реакція системи на дію людини під час навчання або реакція системи на передбачену дію людини після навчання) для «стимулів» (дія людини під час навчання або передбачена дія людини), які не належать наявній множині прикладів. З точки зору кібернетики, є одним із видів кібернетичного експерименту.

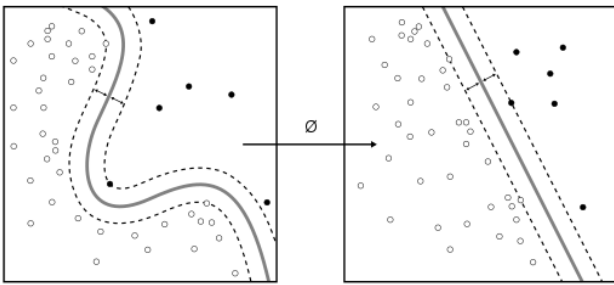


Рис. 1. Ядро методу

Принцип цього методу зумовлений тим, що між входами та еталонними виходами (стимул-реакція) може існувати деяка залежність, але вона невідома. Відома лише кінцева сукупність *прецедентів* – пар «стимул-реакція», звана *навчальною вибіркою*. На основі цих даних треба відновити залежність (побудувати модель відносин стимул-реакція, придатних для прогнозування), тобто побудувати алгоритм, здатний для будь-якого об'єкта видати досить точну відповідь. Для вимірювання точності відповідей так само, як і в навчанні на прикладах, може вводитися *функціонал якості* [2].

Формально задача навчання може бути сформульована таким чином: існує деяка сукупність «стимулів» – x_i , «реакцій на стимули» – y_i необхідно визначити залежність між y та x таку, що в межах припустимої помилки e буде справедливим $f(x) - e \leq y \leq f(x) + e$ [3].

Алгоритм «Support vector machine» для MicroGrid

Метод опорних векторів (Support vector machine) – контролююча модель навчання типу «навчання з вчителем», який аналізує дані, що використовуються для класифікації та регресійного аналізу. Враховуючи набір навчальних прикладів, кожен з яких позначається як такий, що належить до тієї чи іншої з двох категорій, алгоритм навчання SVM створює модель, яка призначає нові приклади однієї категорії або іншої, що робить його наймовірнішим бінарним лінійним класифікатором (хоча методи наприклад, масштабування Platt існують для використання SVM в імовірнісній класифікації). SVM-модель являє собою представлення прикладів як точки у просторі, маповані таким чином, що приклади окремих категорій діляться явним розривом, який є максимально можливим (рис. 2).

Нові приклади потім вказуються в тому самому просторі і передбачають, що вони належать до категорії, на підставі якої сторони розриву вони падають [3].

Крім виконання лінійної класифікації, SVM можуть ефективно виконувати нелінійну класифікацію, використовуючи те, що називається ядром, що неявно відображає їхні входи у високорозмірні простори.

Коли дані маркуються, контрольоване навчання неможливе, і необхідний підхід до некерованого навчання, який намагається знайти природне групування цих групам, а потім картувати нові дані до цих сформованих груп. Алгоритм векторної кластеризації, створений Хавою Сігельманом та Володимиром Вапніком, застосовує статистику векторів підтримки, розроблених в алгоритмі векторних машин підтримки, для класифікації непомічених даних і є одним із найбільш широко використовуваних алгоритмів кластеризації в промисловому застосуванні.

Вирішення алгоритму «Support vector machine» для MicroGrid

Алгоритм побудований на вирішенні трьох частин:

- формальна постановка завдання;
- випадок лінійної роздільності;
- випадок лінійної нероздільності.

Формальна постановка завдання

Точки припускають під собою певний випадок, а саме вектор із кількох значень за датчиками у вигляді «так» або «ні» (наприклад, датчик детекції руху та датчик спрацювання вимикача).

Тому вважаємо, що точки мають вигляд:

$$\{(X_1, C_1), (X_2, C_2), \dots, (X_m, C_m)\},$$

де C_i приймає значення 1 або -1, в залежності від того, якого класу належить точка X_i . Кожне X_i – це p -мірний речовий вектор, зазвичай нормалізований значеннями $[0,1]$ або $[-1,1]$. Якщо точки не будуть нормалізовані, то точка з великими відхиленнями від середніх значень координат точок занадто сильно вплине на класифікатор. Ми можемо розглядати це як навчальну колекцію, в якій для кожного елемента вже заданий клас, до якого він належить. Потрібно, щоб алгоритм методу опорних векторів класифікував випадок таким же чином. Для цього ми будемо розділяючу гіперплощину, яка має вигляд:

$$w \cdot x - b = 0$$

Вектор w – перпендикуляр, який розділяє гіперплощину і ділить її на допустимі й недопустимі випадки. Параметр $\frac{b}{\|w\|}$ дорівнює за модулем відстані від гіперплощини до початку координат. Якщо параметр b дорівнює нулю, гіперплощина проходить через початок координат, що обмежує рішення [3].

Головним є оптимальний розподіл між точками (випадками), тому потрібні опорні вектори і

гіперплощини, паралельні до оптимальної і найближчої, опорних векторів, двох класів. Можна сказати, що ці паралельні гіперплощини можуть бути описані такими рівняннями (з точністю до нормування):

$$\begin{aligned} w \cdot x - b &= 1 \\ w \cdot x - b &= -1 \end{aligned}$$

Якщо навчальна вибірка лінійно роздільна, то можна вибрати гіперплощини таким чином, щоб між ними не лежала жодна точка навчальної вибірки і потім максимізувати відстань між гіперплощинами. Ширину смуги між ними легко знайти з міркувань геометрії, вона дорівнює $\frac{2}{\|w\|}$; таким чином, головним завданням є мінімізувати $\|w\|$. Щоб виключити всі зайві точки зі смуги, треба, щоб для всіх i виконувалося:

$$\begin{cases} w \cdot x - b \geq 1, c_i = 1 \\ w \cdot x - b \leq -1, c_i = -1 \end{cases}$$

Це може бути також записано у вигляді:

$$C_i(w \cdot x_i - b) \geq 1, 1 \leq i \leq n, \quad (1)$$

Випадок лінійної роздільності

Проблема побудови оптимальної роздільної лінії гіперплощини зводиться до мінімізації $\|w\|$ за умови (1). Це завдання квадратичної оптимізації, яка має вигляд:

$$\begin{cases} \|w\|^2 \rightarrow \min \\ c_i(w \cdot x_i - b) \geq 1, 1 \leq i \leq n \end{cases}$$

За теоремою Куна-Таккера, ця задача еквівалентна двоїстій задачі пошуку точки функції Лагранжа:

$$\begin{cases} L(w, b, \gamma) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^n \gamma_i (c_i(w \cdot x_i - b) - 1) \rightarrow \min_{w,b} \max_{\gamma} \\ \gamma_i \geq 0, 1 \leq i \leq n \end{cases} \quad (2)$$

де $\gamma = (\gamma_1, \dots, \gamma_n)$ – вектор двоїстих змінних.

Зведемо цю задачу до еквівалентної задачі квадратичного програмування, що містить тільки двоїсті змінні:

$$\begin{cases} -L(\gamma) = -\sum_{i=1}^n \gamma_i + \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \gamma_i \gamma_j c_i c_j (x_i \cdot x_j) \rightarrow \min_{\gamma} \\ \gamma_i \geq 0, 1 \leq i \leq n \\ \sum_{i=1}^n \gamma_i c_i = 0 \end{cases}, \quad (1)$$

Припустимо, ми вирішили це завдання, тоді w і b можна знайти за формулами:

$$w = \sum_{i=1}^n \gamma_i c_i \cdot x_i \quad b = w \cdot x_i - c_i \gamma_i > 0$$

В результаті алгоритм класифікації може бути записаний у вигляді:

$$a(x) = \text{sign} \left(\sum_{i=1}^n \gamma_i c_i x_i \cdot x - b \right)$$

При цьому підсумовування йде не за всією вибіркою, а тільки за опорними векторами, для яких $\gamma_i \neq 0$ [2].

Випадок лінійної нероздільності

Для того щоб алгоритм міг працювати, якщо класи лінійно нероздільні (точки знаходяться на межі розділяючої смуги), дозволимо йому припускатися помилок у навчальній вибірці. Введемо набір додаткових змінних $\varepsilon_i \geq 0$, що характеризують величину помилки на об'єктах x_i , $1 \leq i \leq n$. Візьмемо за відправну точку (2), пом'якшимо обмеження нерівності, так само введемо в мінімізуючий функціонал обмеження за сумарну помилку:

$$\begin{cases} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n \varepsilon_i \rightarrow \min_{w,b,\varepsilon_i} \\ c_i(w \cdot x_i - b) \geq 1 - \varepsilon_i, 1 \leq i \leq n \\ \varepsilon_i \geq 0, 1 \leq i \leq n \end{cases}$$

Коефіцієнт C – параметр налаштувань методу, який дає змогу регулювати відношення між максимізацією ширини розділяючої смуги і мінімізацією сумарної помилки.

Аналогічно, за теоремою Куна-Таккера, зводимо задачу до пошуку сідлової точки функції Лагранжа:

$$\begin{cases} L(w, b, \varepsilon; \gamma, \vartheta) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^n \gamma_i (c_i(w \cdot x_i - b) - 1) - \\ - \sum_{i=1}^n \varepsilon_i (\gamma_i + \vartheta_i - C) \rightarrow \min_{w,b,\varepsilon} \max_{\gamma,\vartheta} \\ \varepsilon_i \geq 0, \gamma_i \geq 0, \vartheta_i \geq 0, 1 \leq i \leq n \\ \gamma_i = 0 \\ c_i(w \cdot x_i - b) = 1 - \varepsilon_i, 1 \leq i \leq n \\ \vartheta_i = 0 \\ \varepsilon_i = 0, 1 \leq i \leq n \end{cases}$$

За аналогією, зведемо цю задачу до еквівалентної:

$$\begin{cases} -L(\gamma) = -\sum_{i=1}^n \gamma_i + \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \gamma_i \gamma_j c_i c_j (x_i \cdot x_j) \rightarrow \min_{\gamma} \\ 0 \leq \gamma_i \leq C, 1 \leq i \leq n \\ \sum_{i=1}^n \gamma_i c_i = 0 \end{cases}$$

На практиці для побудови машини опорних векторів вирішують саме це завдання, а не (3), адже гарантувати лінійну роздільність точок на два класи в загальному випадку не представляється можливим. Цей варіант алгоритму називають алгоритмом із м'яким проміжком (soft-margin SVM), тоді як у лінійно роздільному випадку говорять про жорсткий проміжок (hard-margin SVM) [3].

Для алгоритму класифікації зберігається формула (4), з тією лише різницею, що тепер ненульовими γ_i володіють не тільки опорні об'єкти, але і об'єкти, які не підходять до правил навчальної вибірки. У певному сенсі це недолік, оскільки

об'єкти, які порушують навчальні правила, часто виявляються шумовими викидами і побудоване на них вирішальне правило спирається на шум.

Константу C зазвичай вибирають за критерієм змінного контролю. Це трудомісткий спосіб, адже завдання доводиться вирішувати заново при кожному значенні C .

Якщо є підстави вважати, що вибірка майже лінійно роздільна, і лише об'єкти-викиди класифікуються невірною, то можна застосувати фільтрацію викидів. Спочатку завдання вирішується при деякому C , і з вибірки видаляється невелика частка об'єктів, що мають найбільшу величину помилки ϵ . Після цього завдання вирішується заново по усеченій вибірці. Можливо, доведеться виконати кілька таких ітерацій, поки ті об'єкти, що залишилися, не виявляються лінійно нероздільними.

Отже, використання Machine Learning є досить актуальною темою, зокрема використання такого ядра. Алгоритм побудови оптимальної розділяючої гіперплощини, запропонований в 1963 р. Володимиром Вапніком, – алгоритм лінійної класифікації. Однак в 1992 р. був запропонований спосіб створення нелінійного класифікатора, в основі якого лежить перехід від скалярних творів до довільних ядер, так званий kernel trick, що дає змогу будувати

нелінійні роздільники. Результируючий алгоритм вкрай схожий на алгоритм лінійної класифікації, з тією лише різницею, що кожен скалярний твір у наведених вище формулах замінюється нелінійною функцією ядра (скалярним утворенням у просторі з більшою розмірністю). У цьому просторі вже може існувати оптимальна роздільна гіперплощина. Оскільки розмірність одержуваного простору може бути більшою розмірністю, ніж вихідного, то перетворення, що зіставляють скалярні утворення, буде нелінійним, але це свідчить про те, що відповідна в вихідному просторі оптимальна розділяюча гіперплощина буде також нелінійною [2].

Висновки. Впровадження алгоритмів Machine Learning до системи розподіленої генерації енергії MicroGrid дасть змогу вирішити питання швидкості оброблення даних із датчиків: система дасть змогу, пропустивши етап обробки даних, на пряму передбачати дію користувача, що значно збільшує швидкодію системи і дає більше апаратних ресурсів на інші процеси.

В залежності від функціонального призначення до функціоналу системи можна додати методи передбачення погоди, стану здоров'я хазяїна тощо за допомогою внутрішніх та зовнішніх датчиків.

Список літератури:

1. Крилов А.В. Система розподіленої генерації microgrid із застосуванням технології іот. Електроніка-2018: XI міжнародна науково-технічна конференція молодих вчених. Збірник статей. Київ: «Кафедра», 2018. С. 312–315.
2. Understanding Machine Learning: From Theory to Algorithms. New York: Cambridge University Press, 2014. –449 p.
3. Vapnik V. Statistical Learning Theory. Wiley, New York, 1998.

ПРИМЕНЕНИЕ БИБЛИОТЕКИ MACHINE LEARNING ДЛЯ СИСТЕМЫ РАСПРЕДЕЛЕННОЙ ГЕНЕРАЦИИ MICROGRID

В статье изложены принципы построения системы распределенной генерации MicroGrid с применением библиотеки Machine Learning (машинного обучения), так как система MicroGrid с IoT дает большое количество данных, которые нужно обрабатывать для реализации единой стратегии интеллектуального и энергоэффективного управления, а наличие «человеческого фактора» обуславливает применение методов машинного обучения, а именно метода Support Vector Machine (метод прикладных векторов).

Ключевые слова: MicroGrid, Machine Learning, Support Vector Machine и метод библиотека, искусственный интеллект, выборка, вектор.

USAGE OF THE MACHINE LEARNING LIBRARY FOR THE DISTRIBUTED GENERATION OF MICROGRID

The article describes the principles of building a distributed generation of MicroGrid from the use of the Machine Learning library (machine learning). Since the MicroGrid system with IoT provides a large amount of data that needs to be processed to implement a single strategy for smart and energy efficient management, as well as the presence of a “human factor”, requires the use of machine learning methods, namely the Support Vector Machine (Applied Vectors Method)

Key words: MicroGrid, Machine Learning, Support Vector Machine, method, library, artificial intelligence, sampling, vector.