

Комплексний алгоритм створення керуючих автоматів на базі машинного навчання

(Представлено: Dr. Wiselin Mathuram)

У статті досліджено комплексний алгоритм створення керуючих автоматів на базі машинного навчання. Розкрито необхідність застосування принципів машинного навчання на керуючих автоматах. Описано генезис автоматів навчання та їх приналежність до штучного інтелекту. Визначено принципи побудови машини Цетліна та перспективні напрями її застосування щодо умов дослідження. Наголошено, що алгоритм навчання керуючого автомата використовує дискретні етапи оновлення, застосовуючи лінійну тактику, це, у свою чергу, дозволяє обмежити алгоритм кінцевою кількістю станів, визначених ансамблем автоматів Цетліна. Визначено умови, якими керується принцип оновлення автомата: значення літералів, параметри та поточні дії автоматів Цетліна з точки зору включення та виключення. Запропоновано основну архітектуру машинного класифікатора Цетліна з двома пунктами та наголошено, що ефективність машинного навчання з використанням автоматів навчання залежить від ряду гіперпараметрів, які необхідно ретельно налаштувати перед тренуванням. Процес налаштування точності та збіжності для цієї задачі називається пошуком гіперпараметрів. Обґрунтовано застосування принципу максимального паралелізму для машинної реалізації автоматів навчання Цетліна. Визначено метод бінаризації із замкнутим циклом зі зворотним зв'язком щодо точності, який ігнорує статистичну значущість даних, яка визначає, як класи виведення висновків співвідносяться з набором даних. Наведено діаграму потоку даних методу бінаризації, який орієнтовано на значущість. Описаний метод організований у три етапи: етап 1 (зменшення розмірності), етап 2 (класифікація), етап 3 (двійкове кодування). Запропоновано схематично логічну реалізацію програмування керуючого автомата. Визначено перспективи подальших досліджень, які ґрунтуються на розробці масштабованої апаратної архітектури.

Ключові слова: алгоритм; керуючий автомат; штучний інтелект; машинне навчання; логіка; ресурс; апаратне забезпечення.

Постановка проблеми у загальному вигляді та її зв'язок з важливими практичними завданнями.

Сенсорні пристрої щільно увійшли в життя сучасної людини. Великий обсяг даних, отриманих цими пристроями, просуває технологічний фронт нового покоління штучного інтелекту (ШІ) для додатків Інтернету речей (IoT). Очікується, що ці програми миттєво прийматимуть важливі рішення в реальному світі, а не завантажуватимуть дані на хмарні сервери. Така поступова зміна технологій вимагає значних досягнень в енергоефективності, що продовжує залишатися основною проблемою проєктування для дизайнерів обладнання IoT.

Автомати навчання, спочатку визначені Михайлом Цетліном у 1960-х роках, становлять клас алгоритму машинного навчання (МН), який підсилює поточні дії, використовуючи попередні дані. Кожна дія слідує траєкторії розподілу ймовірностей, яка оновлюється на основі реакції навколишнього середовища, яку автомат отримує, виконуючи певну дію. Оскільки кількість дій та їх траєкторії розподілу ймовірностей можуть мати дуже велику кількість комбінацій, перехід на автоматизацію навчання до апаратного забезпечення складний.

Машина Цетліна вважається перспективним алгоритмом машинного навчання на основі автоматів навчання. Машина Цетліна спрощує традиційні навчальні автомати за допомогою оновлень дискретних кроків завдяки автоматам Цетліна, що визначені як кінцеві автомати з лінійною тактикою. Для оновлення дій кожен автомат Цетліна використовує нагороди за посилення дії та покарання за послаблення впевненості автомата у виконанні дії. Ця дискретність з лінійними покрововими оновленнями дозволяє формулювати проблему навчання, використовуючи потужну логіку пропозицій; крім того, це спрощує механізм навчання, забезпечуючи ефективне навчання на мікросхемі. Вхідні дані в машині Цетліна кодуються у двійковій формі як набір змінних логіки, які називаються літералами. Ці літерали використовуються для побудови логічних виразів, що відповідають класам висновків, за допомогою ансамблів паралельних автоматів Цетліна, які називаються пунктами, під час навчання. Після завершення навчання результати висновку описуються бінарними класифікаціями.

Логічна структура машин Цетліна надає можливості для енергоефективного проектування апаратного забезпечення штучного інтелекту. Це вимагає вирішення основних проблем систематичного розподілу архітектури низькорівневих ресурсів, а також параметричного налаштування та бінаризації даних, що не може бути досягнуто за допомогою високорівневого синтезу або засобів апаратного прискорення.

Аналіз останніх досліджень та публікацій, на які спирається автор. Науковці різних сфер спрямовували свої дослідження на визначення принципів та механізмів застосування штучного інтелекту на базі автоматизованих систем. Подчашинський Ю.О., Шавурський Ю.О. та Лугових О.О. [1] розглянули загальні принципи проектування та конструювання пристроїв і систем управління, методи формалізації завдання на проектування та методи розробки алгоритмів управління виробничим обладнанням на промислових підприємствах. Формування принципів впровадження машинного навчання у складні системи дослідив О.С. Адамов [2]. Автором наведено детальне визначення механізмів впливу машинного навчання та дію штучного інтелекту у складних системах. У [3] викладено теоретичні основи автоматизації процесу проектування складних комп'ютерних систем, зокрема вбудованих на ПЛІС, а також програмних засобів за допомогою мови опису апаратури VHDL та VeryLog. Савченко-Синякова Євгенія та Степашко Володимир [4] виконали порівняльний аналіз термінів і задач метанавчання та метамодельовання. Навели приклади таких задач, методів і засобів їх розв'язання. Охарактеризували зв'язок понять «онтологія» та «метамодель». Посібник [5] присвячений питанням теоретичних і практичних основ комп'ютерної арифметики і алгебри логіки, вивчення методів подання чисел в ЕОМ, алгоритмів виконання основних арифметичних та логічних операцій з числами в різних системах числення, аналізу та синтезу цифрових операційних та керуючих автоматів. Косолапов А.А. та Жуковицький І.В. [6] розкрили моделі, методи й алгоритми концептуального проектування комп'ютерних систем реального часу. Серед зарубіжних авторів варто виокремити такі роботи, як: G.Balavignesh & Perumal, Kumaresan & R.Kavitha & G.Ramya & Subramani Venkatesan [7], Sharma Neelima & Chawla V. & Ram N. [8], Neelima & Chawla Vivek [9], P.R.L. Alves, L.G.S. Duarte and L.A.C.P. da Mota [10], Viktor Avrutin, Björn Schenke and Laura Gardini [11], Qamar Din [12], Dia Ghosh, Arindum Mukherjee, Nikhil Ranjan Das and Baidya Nath Biswas [13], S.Moe, A.M. Rustad and K.G. Hanssen [14], Somenath Mukherjee, Rajdeep Ray, Rajkumar Samanta, Mofazzal H. Khondekar and Goutam Sanyal [15], E.Scornet [16] та інші.

Проте, враховуючи описані наукові набутки за темою, питання розкриття комплексного алгоритму створення керуючих автоматів на базі машинного навчання залишається відкритим та потребує детального опрацювання.

Постановка завдання. Розкрити комплексний алгоритм створення керуючих автоматів на базі машинного навчання. Підтвердити принципи автоматичного навчання, що застосовуються при економічному проектуванні апаратних засобів штучного інтелекту.

Викладення основного матеріалу дослідження. Алгоритм навчання керуючого автомата використовує дискретні етапи оновлення, застосовуючи лінійну тактику. Це дозволяє обмежити алгоритм кінцевою кількістю станів, визначених ансамблем автоматів Цетліна. Вхідні дані визначаються як набір бінарних ознак та їх доповнень, які називаються літералами. Літерали подаються до структури автомата навчання через дві основні частини: одна відповідає за висновок (тобто класифікацію), а інша – за зміцнення та зворотний зв'язок для навчання. Основним компонентом висновку є сполучний пункт, який використовує логічні вирази пропозиції для класифікації виводу. Склад кожного пункту контролюється командою автоматів Цетліна, кожен з яких має заздалегідь визначену кількість станів, розділених між діями. Автомати вирішують, чи слід включити їх асоційований літерал у розділ, чи ні, виконуючи ряд кроків посилення. Кожен клас висновків має набір пов'язаних з ним пунктів. Кожен пункт здійснює голосування або не голосує за свій клас. Половина пунктів може проголосувати позитивно, а половина пунктів – проти. Кожен пункт сам собою нелінійний, оскільки вони є сполучниковими. Система голосування є лінійною (підсумовування), за якою слідує порогове значення – argmax (аргумент максимізації).

Підсумовуються голоси, щоб отримати колективний результат, який свідчить про максимальну значущість. Ця максимальність використовується для впливу на майбутні рішення автоматів. У задачі виводу одного класу вихідний рівень є простою пороговою функцією. Якщо голоси позитивні (або нульові), визначається, що вхідні дані належать до класу. Для від'ємної суми вхідні дані визначаються як відсутні у класі. Для задач із кількома класами замінюємо порогове значення на argmax для визначення класу виводу. У цьому випадку підсумовування класів стає свідченням впевненості, і argmax вибирає клас з найвищою впевненістю, тим самим уникаючи будь-яких двозначностей у класифікації. Основоположними для підкріплення є команда автоматів Цетліна. Такі автомати також відомі як автомати з лінійною тактикою, щоб підкреслити той факт, що вони дозволяють поступове сходження або підкріплення під час виконання певної дії, і однаково поступовий спуск від однієї дії до виконання іншої дії. У машинній реалізації Цетліна автомат Цетліна з двома діями описується діаграмою стану. Автомат

може отримати винагороду, змусивши його посилити поточне рішення про дію (наприклад, дію 1), віддалившись від середньої частини.

Оновлення автоматів вимагає посилення за рахунок штрафу, винагороди та бездіяльності. Рішення про те, чи потрібно оновлювати окремі автомати, контролюється рядом умов, що ґрунтуються на: значеннях літералів, параметрах з пунктів, описаних раніше, поточних діях автоматів Цетліна з точки зору включення та виключення.

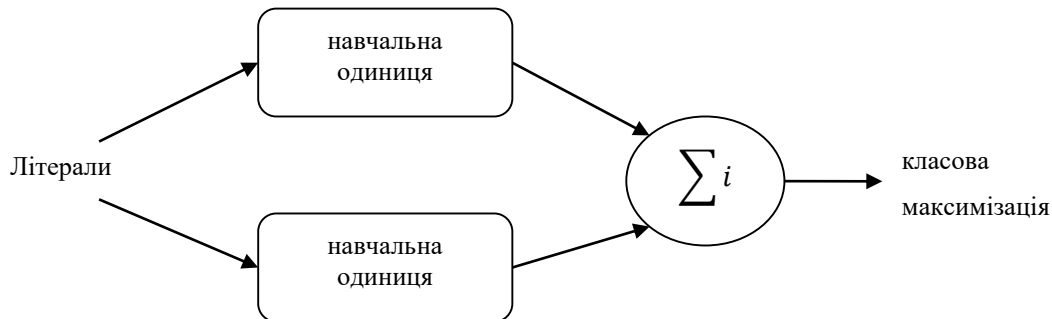


Рис. 1. Основна архітектура машинного класифікатора Цетліна з двома пунктами

Ефективність машинного навчання з використанням автоматів навчання залежить від ряду гіперпараметрів, які необхідно ретельно налаштувати перед тренуванням. Процес налаштування точності та збіжності для цієї задачі називається пошуком гіперпараметрів. Для вирішення проблем машинного навчання пошук на основі програмного забезпечення за допомогою гіперпараметрів може бути обчислювально складним, вимагаючи від кількох годин до тижнів ітераційного обчислення. Зазвичай пошук гіперпараметрів на основі програмного забезпечення спрямований на досягнення кращої точності та продуктивності.

Однак апаратні цілі незначно відрізняються, тому що для мінімальної енергоефективності при одночасному досягненні допустимої точності гіперпараметри потрібно застосовувати в низькорівневих проектних конфігураціях. Для досягнення цих цілей, а також прискореного пошуку, необхідно використовувати прототип апаратного забезпечення на основі програмованої логічної інтегральної схеми (ПЛІС) для вирішення проблем БД з різними розмірами набору даних. Цей прототип також буде використовуватися для автоматизації процесу пришвидшення розробки конструкції управління компромісами між потужністю, продуктивністю та ефективністю. Апаратна архітектура машинної реалізації автоматів навчання Цетліна розроблена з використанням принципу максимального паралелізму.

Оновлення всіх автоматів Цетліна та обчислення всіх пунктів виконуються паралельно. Це дозволяє обробляти одну точку даних (що є набором функцій введення, одночасно оновлюючи всі автомати в машині Цетліна) за один такт. На рисунку 2 показана основна архітектура машинного виведення (тобто класифікатора) Цетліна. Цей блок дублюється на кількість необхідних класів. Вхідними даними для запропонованої архітектури є набір бінарних літералів. Літерали організовані в пари закодованих двійкових файлів та їх обернених значень (тобто літерали = {входи, входи}). Блоки навчання є автономними і включають автомати Цетліна, генерацію зворотного зв'язку та випадкову генерацію, пов'язану з одним пунктом. Для виводу потрібні лише обчислення та стани, розраховані раніше автоматами; решту навчальної частини можна опустити або вимкнути. Для синтезу інтегральної схеми (ASIC) параметри машини Цетліна фіксуються під час компіляції, забезпечуючи найменшу можливу площу та потужність. Для прототипування ПЛІС гіперпараметри можна регулювати на ходу для швидкого дослідження та оптимізації.

Вхідні літерали кодуються у бінаризованій формі. Цей процес бінаризації впливає на точність системи; тому, щоб налаштувати точність на необхідний рівень, бінаризація включається в цикл зворотного зв'язку під час навчання, як показано на рисунку 3. Використовуючи належним чином обраний метод кодування, вихідні дані кодуються з підвищеною двійковою точністю до досягнення необхідної точності.

Існуючий метод кодування використовує заздалегідь визначені пороги та рівні точності для кодування вихідного набору даних. Цей метод ігнорує статистичну значущість даних, яка визначає, як класи виведення висновків співвідносяться з набором даних. Наприклад, у задачі машинного навчання з кількома класами можливо, що більшість класів висновків є статистично ортогональними та незалежними. Метод агностики значущості даних не використовуватиме цю ортогональність у напрямі зменшення бінарних кодувань, що призведе до надмірного забезпечення ресурсів (таких як кількість паралельних автоматів та кількість пунктів) в апаратній архітектурі.

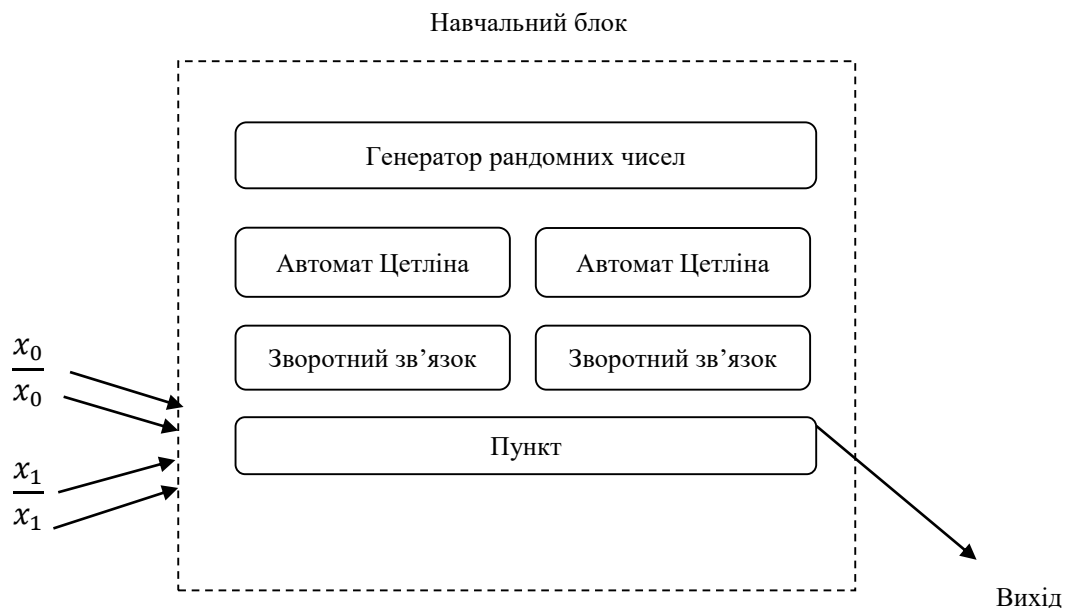


Рис. 2. Навчальний блок з двома входами

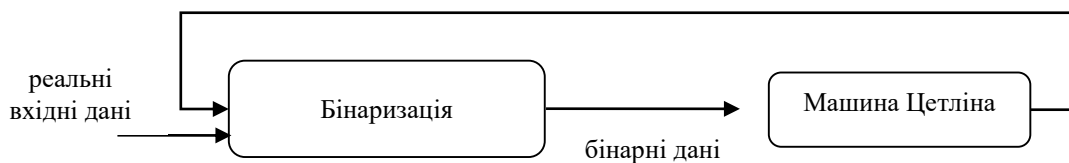


Рис. 3. Метод бінаризації із замкнутим циклом зі зворотним зв'язком щодо точності

На рисунку 4 наведено діаграму потоку даних методу бінаризації, орієнтованого на значущість, організований у три етапи.

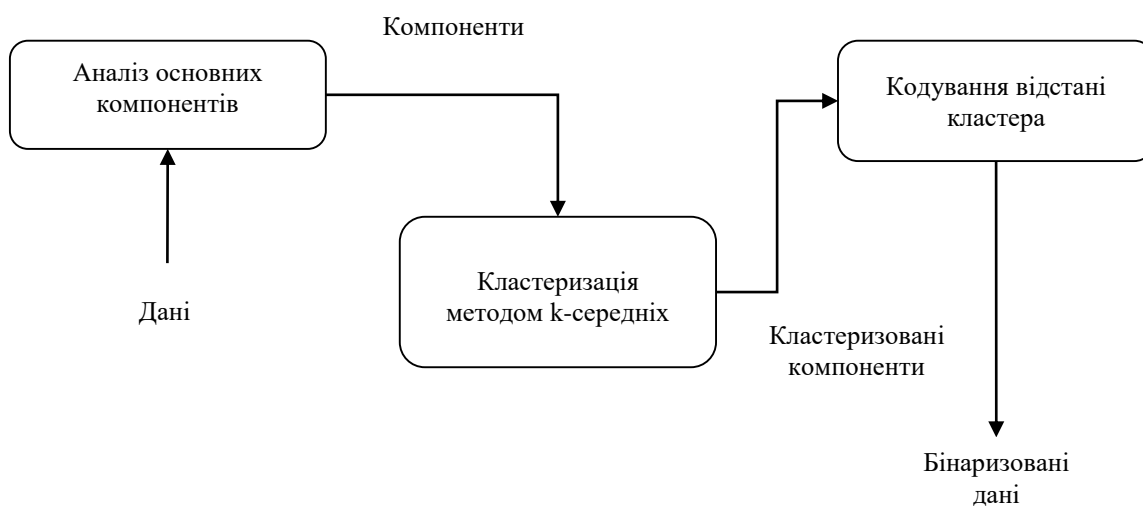


Рис. 4. Діаграма потоку даних методу бінаризації, орієнтованого на значущість, організований у три етапи

Етап 1 (зменшення розмірності). На першому етапі мітки класів видаляються в цьому наборі даних машинного навчання, а потім проводиться аналіз основних компонентів (PCA). PCA генерує ортогональне перетворення набору даних до лінійно некорельованих компонентів, визначених власними значеннями, що представляють їх відсоток дисперсії. Невідповідності у визначенні компонента відкидаються, що призводить до зменшення розмірності.

Етап 2 (класифікація). Після зменшення розмірів за допомогою PCA можливо, що компоненти все ще матимуть власні значення, що вказує на високу кореляцію між ними. У цей момент для кількісної диференціації позицій точок даних застосовується кластеризація методом k-середніх. Метод k-середніх обрано, оскільки це швидкий та масштабований метод, який може поступово адаптуватися до кластерних центроїдів, починаючи з випадкової точки даних. У запропонованому підході використовуємо жорсткий підхід до кластеризації методом k-середніх, який виділяє кожну точку PCA лише одному кластеру. Оскільки відомо, до якого класу належить кожна точка з вихідних міток класу, метод k-середніх дозволяє перевірити їх справжнє значення за допомогою ортогональності.

Етап 3 (двійкове кодування). Визначається відстань між скупченнями, що призводить до методу бінаризації. Використовуємо схему кодування кластерів ліворуч, відстань від справжнього класу, кластеру праворуч, яка може однозначно визначити кластер, максимально зберігаючи ортогональність та зменшення розмірів, отримані від PCA. Визначивши відстані кластера, порогові точки можна оцінити за необробленими наборами даних для двійкового кодування. Для кластерів з перекриттям або вищою точністю порогові значення повторно регулюються або перекодовуються з урахуванням одного або кількох дистанційних кластерів.

0 або 1 присвоюється кожному пункту на основі їх відповідного літералу, виводу пункту, стану, результату підсумовування, очікуваного класу та порогу зворотного зв'язку.

Елемент ймовірності вводиться в налаштування стану, щоб полегшити різноманітність навчання серед пунктів. Елемент ймовірності забезпечується лінійними регістрами зсуву зворотного зв'язку (ЛРЗЗЗ). ЛРЗЗЗ реалізує послідовність випадкових чисел у кожному циклі, яку потім порівнюють із заздалегідь визначеною послідовністю для визначення ймовірності у ланцюзі.

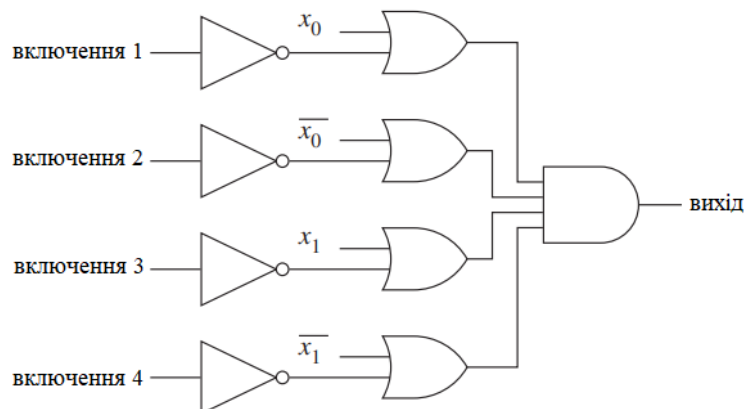
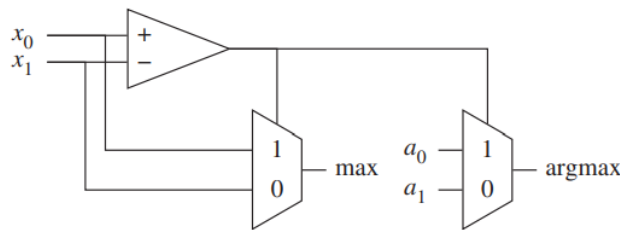


Рис. 5. Логічна реалізація програмування керуючого автомата

Після того як навчання буде завершено за алгоритмом машини Цетліна, логіка посилення стає зайвою, і для висновку потрібен лише стан включення.

На рисунку 5 показана логіка двопозиційного пункту, реалізованого за алгоритмом створення керуючих автоматів на базі машинного навчання. Операція AND реалізована як збалансоване дерево для мінімальної затримки шляху. Включення для кожного літералу використовується для маскуванню літералу, змушуючи відповідний вхід AND підходити до 1, якщо літерал потрібно виключити.

Найпростіша машина «Цетлін» складається з одного класу та використовує її порогову функцію для визначення того, чи є вхід у класі, чи ні. Багатокласна машина Цетліна може бути реалізована шляхом створення декількох класифікаторів Цетліна та вибору класу з найбільшою впевненістю за допомогою блоку argmax . Argmax будується за допомогою дерева компараторів із супутніми мультиплексорами, які проходять через відповідний номер аргументу. На рисунку 6 показана логічна реалізація argmax з двома входами, який використовується для побудови argmax входу з більшим номером. Виходи max та argmax стають входами x_i та a_i для наступного етапу відповідно.

Рис. 6. Логічна реалізація argmax з двома входами

Як правило, пошук на основі програмного забезпечення за допомогою гіперпараметрів має на меті досягнення кращої точності та продуктивності. Апаратний гіперпараметричний пошук дещо відрізняється, оскільки він використовує ці параметри в низькорівневих проектних конфігураціях для економії енергії, досягаючи при цьому прийнятної точності. Таким чином, розроблено комплексний алгоритм створення керуючих автоматів на основі ПЛІС, який можливо використовувати для гнучкого програмування апаратного забезпечення для прискореної перевірки проектування, а також економії енергії та стохастичності за допомогою невеликих та легко доступних наборів даних.

Висновки і перспективи подальших досліджень. У роботі досліджено комплексний алгоритм створення керуючих автоматів на базі машинного навчання. Алгоритм використовує природну здатність ансамблю автоматів Цетліна вчитися з набору навчальних даних. Ґрунтуються автомати Цетліна на принципі енергоощадності. Запропонована архітектура штучного інтелекту є вирішальним кроком на шляху до становлення складних систем штучного інтелекту з можливістю навчання керуючих автоматів, що особливо підходить для додатків, які потребують постійного навчання. Перспективи подальших досліджень ґрунтуються на розробці масштабованої апаратної архітектури, яка дозволить вирішити проблеми машинного навчання за допомогою передових архітектурних розподілів та методів проектування масштабною інтеграції.

Список використаної літератури:

1. *Подчащинський Ю.О.* Проектування та конструювання пристроїв та систем управління : навчальний посібник / *Ю.О. Подчащинський, Ю.О. Шавурський, О.О. Лугових.* – Житомир : ЖДТУ, 2018. – 280 с.
2. *Адамов О.С.* Моделі і методи захисту кіберпростору на основі аналізу великих даних з використанням машинного навчання : дис. ... канд. техн. наук : 05.13.05 / *О.С. Адамов* ; М-во освіти і науки України, Харків. нац. ун-т радіоелектроніки. – Харків, 2019. – 243 с.
3. *Мірошник М.А.* Технології та автоматизація проектування цифрових пристроїв складних комп'ютерних систем на ПЛІС : навч. посіб. / *М.А. Мірошник, Л.А. Клименко, Я.Ю. Корольова.* – Х. : УкрДУЗТ, 2021. – 220 с.
4. *Савченко Є.А.* Аналіз підходів до метанавчання та метамодельовання / *Є.А. Савченко, В.С. Степашко* // Індуктивне моделювання складних систем. – 2017. – Вип. 9. – С. 194–203.
5. *Тарарака В.Д.* Прикладна теорія цифрових автоматів : навчальний посібник / *В.Д. Тарарака.* – Житомир : ЖДТУ, 2019. – 183 с.
6. *Косолапов А.А.* Концептуальне проектування комп'ютерних систем реального часу (моделі, методи і алгоритми) / *А.А. Косолапов, І.В. Жуковицький.* – Дніпро : ДНУЗТ, 2018. – 274 с.
7. Computer Numerical Control Machine Based on Machine Learning / *G.Balavignesh et al.* // International Journal of Pure and Applied Mathematics. – 2017. – Vol. 116. – P. 521–527.
8. *Sharma N.* Comparison of machine learning algorithms for the automatic programming of computer numerical control machine / *N.Sharma, V.Chawla, N.Ram* // International Journal of Data and Network Science. – 2020. – Vol. 4 (1). – P. 1–14. DOI: 10.5267/j.ijdns.2019.9.003.
9. *Chawla V.* Automated CNC Programming by the Restricted Boltzmann Machine Algorithm / *V.Chawla* // Advances in Electromechanical Technologies. – 2021. – P. 691–709. DOI: 10.1007/978-981-15-5463-6_62.
10. *Alves P.R.L.* Detecting chaos and predicting in Dow Jones Index / *P.R.L. Alves, L.G.S. Duarte, L.A.C.P. Da Mota* // Chaos, Solitons & Fractals. – 2018. – Vol. 110. – P. 232–238.
11. *Avrutin V.* Calculation of homoclinic and heteroclinic orbits in 1D maps / *V.Avrutin, B.Schenke, L.Gardini* // Communications in Nonlinear Science and Numerical Simulation. – 2015. – Vol. 22 (1–3). – P. 1201–1214.
12. *Din Q.* Complexity and chaos control in a discrete-time prey-predator model / *Q.Din* // Communications in Nonlinear Science and Numerical Simulation. – 2017. – Vol. 49. – P. 113–134.
13. Generation & control of chaos in a single loop optoelectronic oscillator / *D.Ghosh, A.Mukherjee, N.R. Das, B.N. Biswas* // Optik. – 2018. – Vol. 165. – P. 275–287.
14. *Moe S.* Machine learning in control systems: An overview of the state of the art / *S.Moe, A.M. Rustad, K.G. Hanssen* // International Conference on Innovative Techniques and Applications of Artificial Intelligence. – 2018. – P. 250–265.
15. Nonlinearity and chaos in wireless network traffic / *S.Mukherjee, R.Ray, R.Samanta et al.* // Chaos, Solitons & Fractals. – 2017. – Vol. 96. – P. 23–29.
16. *Scornet E.* Random forests and kernel methods / *E.Scornet* // IEEE Transactions on Information Theory. – 2016. – Vol. 62 (3). – P. 1485–1500.

References:

1. Podchashyns'kyj, Ju.O., Shavurs'kyj, Ju.O. and Lugovyh, O.O. (2018), *Proektuvannja ta konstrujuvannja prystroi'v ta system upravlinnja*, navchal'nyj posibnyk, ZhDTU, Zhytomyr, 280 p.
2. Adamov, O.S. (2019), «Modeli i metody zahystu kiberprostoru na osnovi analizu velykyh danyh z vykorystannjam mashynnogo navchannja», Ph.D. Thesis of dissertation, 05.13.05, M-vo osvity i nauky Ukrainy, Harkiv. nac. un-t radioelektroniky, Harkiv, 243 p.
3. Miroshnyk, M.A., Klymenko, L.A. and Korol'ova, Ja.Ju. (2021), *Tehnologii' ta avtomatyzacija projektuvannja cyfrovych prystroi'v skladnyh komp'juternyh system na PLIS*, navch. posib., UkrDUZT, H., 220 p.
4. Savchenko, Je.A. and Stepashko, V.S. (2017), «Analiz pidhodiv do metanavchannja ta metamodeljuvannja», *Induktyvne modeljuvannja skladnyh system*, Issue 9, pp. 194–203.
5. Tararaka, V.D. (2019), *Prykladna teorija cyfrovych avtomativ*, ZhDTU, Zhytomyr, 183 p.
6. Kosolapov, A.A. and Zhukovyc'kyj, I.V. (2018), *Konceptual'ne proektuvannja komp'juternyh system real'nogo chasu (modeli, metody i algorytmy)*, DNUZT, Dnipro, 274 p.
7. Balavignesh, G. et. al. (2017), «Computer Numerical Control Machine Based on Machine Learning», *International Journal of Pure and Applied Mathematics*, Vol. 116, pp. 521–527.
8. Sharma, N., Chawla, V. and Ram, N. (2020), «Comparison of machine learning algorithms for the automatic programming of computer numerical control machine», *International Journal of Data and Network Science*, Vol. 4 (1), pp. 1–14, doi: 10.5267/j.ijdns.2019.9.003.
9. Chawla, V. (2021), «Automated CNC Programming by the Restricted Boltzmann Machine Algorithm», *Advances in Electromechanical Technologies*, pp. 691–709, doi: 10.1007/978-981-15-5463-6_62.
10. Alves, P.R.L., Duarte, L.G.S. and Da Mota, L.A.C.P. (2018), «Detecting chaos and predicting in Dow Jones Index», *Chaos, Solitons & Fractals*, Vol. 110, pp. 232–238.
11. Avrutin, V., Schenke, B. and Gardini, L. (2015), «Calculation of homoclinic and heteroclinic orbits in 1D maps2», *Communications in Nonlinear Science and Numerical Simulation*, Vol. 22 (1–3), pp. 1201–1214.
12. Din, Q. (2017), «Complexity and chaos control in a discrete-time prey-predator model», *Communications in Nonlinear Science and Numerical Simulation*, Vol. 49, pp. 113–134.
13. Ghosh, D., Mukherjee, A., Das, N.R. and Biswas, B.N. (2018), «Generation & control of chaos in a single loop optoelectronic oscillator», *Optik*, Vol. 165, pp. 275–287.
14. Moe, S., Rustad, A.M. and Hanssen, K.G. (2018), «Machine learning in control systems: An overview of the state of the art», *International Conference on Innovative Techniques and Applications of Artificial Intelligence*, pp. 250–265.
15. Mukherjee, S., Ray, R., Samanta, R., Khondekar, M.H. and Sanyal, G. (2017), «Nonlinearity and chaos in wireless network traffic», *Chaos, Solitons & Fractals*, Vol. 96, pp. 23–29.
16. Scornet, E. (2016), «Random forests and kernel methods», *IEEE Transactions on Information Theory*, Vol. 62 (3), pp. 1485–1500.

Козак Євген Борисович – магістр комп'ютерних наук, розробник програмного забезпечення, інженер-програміст (GAN Inc. C/O Memery Crystal Llp, 165 Fleet Street London EC4A 2DY United Kingdom).

Наукові інтереси:

- інформаційні технології;
- програмне забезпечення систем;
- розробка програмно-інформаційних систем;
- системи управління версіями (CVS, Subversion, TortoiseSVN, Git, Mercurial), C++, Java, IronPython;
- сучасні операційні системи.

E-mail: 1604102018@ukr.net.

Стаття надійшла до редакції 05.08.2021.