

УДК 681.324.01

DOI: 10.15587/1729-4061.2022.251637

## Розробка методики навчання штучних нейронних мереж для інтелектуальних систем підтримки прийняття рішень

К. А. Махді, А. В. Шишацький, О. А. Симоненко, Н. М. Протас,  
О. О. Троцько, В. С. Кивлюк, А. А. Шульгін, П. М. Стешенко,  
Е. С. Остапчук, Т. І. Голенковська

*Розроблено методику навчання штучних нейронних мереж для інтелектуальних систем підтримки прийняття рішень. Відмінна особливість запропонованої методики полягає в тому, що вона проводить навчання не тільки синаптичних ваг штучної нейронної мережі, але й виду та параметрів функції належності. В разі неможливості забезпечити задану якість функціонування штучних нейронних мереж за рахунок навчання параметрів штучної нейронної мережі відбувається навчання архітектури штучних нейронних мереж. Вибір архітектури, виду та параметрів функції належності відбувається із врахуванням обчислювальних ресурсів засобу та із врахуванням типу та кількості інформації, що надходить на вхід штучної нейронної мережі. Також відмінною особливістю розробленої методики є те, що для обчислення вхідних даних не потрібні попередні розрахункові дані. Розробка запропонованої методики обумовлена необхідністю проведення навчання штучних нейронних мереж для інтелектуальних систем підтримки прийняття рішень, з метою обробки більшої кількості інформації, при однозначності рішень, що приймаються. За результатами дослідження встановлено, що зазначена методика навчання забезпечує в середньому на 10–18 % більш високу ефективність навчання штучних нейронних мереж та не накопичує похибок в ході навчання. Зазначена методика дозволить проводити навчання штучних нейронних мереж за рахунок навчання параметрів та архітектури, визначити ефективні заходи для підвищення ефективності функціонування штучних нейронних мереж. Використання зазначеної методики дозволить зменшити використання обчислювальних ресурсів систем підтримки прийняття рішень та виробити заходи, що спрямовані на підвищення ефективності навчання штучних нейронних мереж; підвищити оперативність обробки інформації в штучних нейронних мережах.*

*Ключові слова: штучні нейронні мережі, оперативність обробки інформації, системи підтримки прийняття рішень.*

### 1. Вступ

Системи підтримки прийняття рішень (СППР) стали основою вирішення інформаційно-розрахункових завдань в повсякденному житті так і для вирішення досить специфічних (спеціальних завдань). СППР активно застосовуються при обробці великих масивів даних, забезпечення інформаційної підтримки процесу прийняття рішень особами, що приймають рішення. Основу існуючих СППР становлять методи штучного інтелекту [1–11].

Створення інтелектуальних СППР стало подальшим розвитком СППР класичного типу, основним інструментом яких є штучні нейронні мережі (ШНМ), що еволюціонують. Інтелектуальна система підтримки прийняття рішень (Intelligent Decision-Maker Support System, iDMSS) – інтерактивна комп'ютерна система, призначена для підтримки прийняття рішень у різних сферах діяльності стосовно слабкоструктурованих і неструктурованих проблем, яка ґрунтується на використанні моделей і процедур з обробки даних та знань на основі технологій штучного інтелекту.

ШНМ, що еволюціонують, мають як універсальні апроксимуючі властивості. ШНМ, що еволюціонують забезпечують стабільну роботу в умовах нелінійності, апіорної визначеності, стохастичності та хаотичності, різного роду збурювань і завад.

Незважаючи на досить успішне їхнє застосування для розв'язку широкого кола завдань інтелектуального аналізу даних, ці системи мають ряд недоліків, пов'язаних з їхнім використанням. Серед найбільш істотних недоліків можна виділити такі:

- складність вибору архітектури системи. Як правило, модель, заснована на принципах обчислювального інтелекту, має фіксовану архітектуру. У контексті ШНМ це означає, що нейронна мережа має фіксовану кількість нейронів і зв'язків. У зв'язку із цим, адаптація системи до нових даних, що надходять на обробку, що мають відмінну від попередніх даних природу, може виявитися проблематичним;

- утворення “мертвих” нейронів в шарах, що виникають при функціонуванні ШНМ;

- навчання в пакетному режимі та навчання протягом декількох епох вимагає значних часових ресурсів. Такі системи не є пристосованими для роботи в online режимі з досить високим темпом надходження нових даних на обробку;

- багато з існуючих систем обчислювального інтелекту не можуть визначати правила, що еволюціонують, за якими відбувається розвиток системи, а також можуть представляти результати своєї роботи в термінах природньої мови.

Таким чином, актуальним є завдання розробки нових методів (підходів, методик) навчання для ШНМ, які дозволять вирішити зазначені труднощі.

## **2. Аналіз літературних даних та постановка проблеми**

В роботі [3] проведено аналіз властивостей ШНМ, які використовувалися при прогнозуванні концентрації забруднюючих речовин в повітрі. Запропоновано використання екстремальної навчальної машини для ШНМ, що забезпечує високу ефективність узагальнення при надзвичайно високій швидкості навчання. До недоліків підходу слід віднести накопичення похибок ШНМ в ході проведення обчислень, неможливість вибору параметрів та типу функції належності, утворення “мертвих” нейронів в ході навчання.

В роботі [4] представлено моделювання адекватності управління банківським капіталом. В основу зазначеного моделювання покладені трендові моделі прогнозування. Для проведення розрахунків використовується багатозаровий перцептрон. Навчання зазначеного перцептрону обмежено лише навчанням си-

наптичних ваг, причому тільки активованих нейронів. Інших механізмів навчання в зазначеному дослідженні не представлено.

В роботі [5] наведено оперативний підхід з просторового аналізу в морській галузі для кількісної оцінки та відображення супутніх екосистемних послуг. До недоліків зазначеного методу слід віднести неможливість гнучкого налаштування (адаптації) оціночних моделей при додаванні (виключенні) показників і зміні їх параметрів (узгодженості та значущості показників). Також навчання ШНМ обмежено лише класичним навчанням ваг активних нейронів.

В роботі [6] представлена модель машинного навчання для автоматичної ідентифікації запитів та надання інформаційних служб підтримки, що обмінюються між членами інтернет-спільноти. Зазначена модель призначена для обробки великої кількості повідомлень користувачів соціальних мереж. Недоліками зазначеної моделі є відсутність механізмів оцінки адекватності рішень, що приймаються та велика обчислювальна складність. Навчання обмежено лише навчанням синаптичних ваг ШНМ.

В роботі [7] продемонстровано використання ШНМ для виявлення аномалії серцевого ритму та інших захворювань серця. В якості методу навчання ШНМ використовується алгоритм зворотного поширення похибки. Недоліками зазначеного підходу є його обмеженість навчанням лише синаптичних ваг, без навчання виду та параметрів функції належності.

В роботі [8] запропоновано використовувати ШНМ для виявлення сходження лавин. В якості методу навчання ШНМ використовується алгоритм зворотного поширення помилки. Недоліками зазначеного підходу є його обмеженість навчанням лише синаптичних ваг, без навчання виду та параметрів функції належності.

В роботі [9] представлено використання ШНМ для виявлення проблем виявлення аномалій в системах домашньої авторизації. В якості методу навчання ШНМ Кохонена використовується алгоритм “переможець отримує все”. Недоліками зазначеного підходу є накопичення помилки, що обумовлена наявністю неактивованих та мертвих нейронів в процесі навчання, обмеженість навчанням лише синаптичних ваг, а також необхідність зберігання раніше розрахованих даних.

В роботі [10] представлено використання ШНМ для виявлення проблем виявлення аномалій в енцефалограмі людини. В якості методу навчання ШНМ використовується метод тонкого налаштування параметрів ШНМ. Недоліками зазначеного підходу є накопичення помилки в процесі навчання, обмеженість навчанням лише синаптичних ваг без навчання виду та параметрів функції належності.

В роботі [12] представлено використання методів машинного навчання, а саме ШНМ та генетичних алгоритмів. В якості методу навчання ШНМ використовується генетичний алгоритм. Недоліками зазначеного підходу є його обмеженість навчанням лише синаптичних ваг, без навчання виду та параметрів функції належності.

В роботі [13] представлено використання методів машинного навчання, а саме ШНМ та методу диференційного пошуку. В ході дослідження проведено розробку гібридного методу навчання ШНМ, що заснований на використанні

алгоритму зворотного поширення похибки та диференційного пошуку. Недоліками зазначеного підходу є його обмеженість навчанням лише синаптичних ваг, без навчання виду та параметрів функції належності.

В роботі [14] проведено розробку методів навчання ШНМ з використанням комбінованої апроксимації поверхні відгуку, який забезпечує найменші похибки навчання і прогнозування. Недоліком зазначеного методу накопичення помилки в ході навчання та неможливість зміни архітектури ШНМ в ході навчання.

В роботі [15] наведено використання ШНМ для оцінки ефективності роботи агрегату, використовуючи попередній часовий ряд його продуктивності. Для навчання ШНМ використовуються моделі SBM (Stochastic Block Model) та DEA (Data Envelopment Analysis). Недоліками зазначеного підходу є обмеженість в виборі архітектури мережі, навчання тільки синаптичних ваг.

В роботі [16] наведено використання ШНМ для оцінювання геомеханічних властивостей. В якості методу навчання ШНМ використовується алгоритм зворотного поширення помилки. Покращення характеристик алгоритму зворотнього поширення помилки досягається за рахунок збільшення навчальної вибірки. Недоліками зазначеного підходу є його обмеженість навчанням лише синаптичних ваг, без навчання виду та параметрів функції належності.

В роботі [17] наведено використання ШНМ для оцінки інтенсивності дорожнього руху. В якості методу навчання ШНМ використовується алгоритм зворотного поширення помилки. Покращення характеристик алгоритму зворотнього поширення помилки досягається за рахунок використання пропускових з'єднань між кожним шаром, так що кожен шар викладає лише залишкову функцію щодо результатів попереднього шару. Недоліками зазначеного підходу є його обмеженість навчанням лише синаптичних ваг, без навчання виду та параметрів функції належності.

Проведення аналізу наукових праць [1–17] показав, що для навчання штучних нейронних мереж використовуються загальновідомі методи навчання. Зазначені методи орієнтовані, як правило, на навчання синаптичних ваг або функції належності. Використання відомих алгоритмів (методів, методик) навчання штучних нейронних мереж навіть з покращеними характеристиками не задовольняє існуючим та перспективним вимогам, що висувуються до них, а саме:

- збільшення кількості інформації, що спроможні обробити штучні нейронні мережі;
- підвищення достовірності прийняття рішення інтелектуальними системами підтримки прийняття рішень;
- підвищення швидкості адаптації архітектури та параметрів штучних нейронних мереж відповідно до завдань, що виникають;
- недопущення тупикових ситуацій під час навчання штучних нейронних мереж;
- забезпечення прогнозованості процесу навчання штучних нейронних мереж;
- забезпечення однозначності рішень, що приймаються інтелектуальними системами підтримки прийняття рішень;

– забезпечення проведення обчислень великих масивів даних за одну епоху без збереження попередніх розрахунків.

### **3. Мета та завдання дослідження**

Метою дослідження є розробка методики навчання штучних нейронних мереж для інтелектуальних систем підтримки прийняття рішень, яка дозволяє виконувати обробку більшої кількості інформації при однозначності рішень, що приймаються.

Для досягнення мети були поставлені такі завдання:

- провести розробку алгоритма методики навчання штучних нейронних мереж для інтелектуальних систем підтримки прийняття рішень;
- експериментально оцінити ефективність навчання штучних нейронних мереж.

### **4. Матеріали та методи дослідження**

В ході проведеного дослідження використовувалися загальні положення теорії штучного інтелекту – для вирішення задачі навчання штучної нейронної мережі в інтелектуальних системах підтримки прийняття рішень. Тобто теорія штучного інтелекту є основою зазначеного дослідження. В дослідженні використано удосконалений генетичний алгоритм та штучні нейронні мережі, що еволюціонують. Моделювання проводилося з використанням програмного забезпечення MathCad 2014 (США) та ПЕОМ Intel Core i3 (США).

Мережа Кохонена [2, 18–24] відноситься до самоорганізуючих мереж. Це означає, що вони не отримують бажаний вихідний сигнал при надходженні вхідного навчального вектору, а в результаті навчання мережа розділяє вхідні сигнали на класи, таким чином формуючи топологічні карти.

Варто відзначити, що самоорганізуюча карта Т. Кохонена реалізує відображення вхідного простору розмірності  $n$  у вихідний простір розмірності  $m$ .

Самоорганізована карта має дуже просту архітектуру з прямою передачею інформації. Крім нульового (рецепторного) шару, вона містить єдиний шар нейронів, який дуже часто називають шаром Кохонена [25–32].

Розглянемо докладніше архітектуру самоорганізуючої карти. На вхід мережі надходить  $n$ -вимірний вхідний сигнал. Мережа містить єдиний шар з  $m$  нейронів, які утворюють на площині прямокутні решітки.

Нейрони характеризуються своїм місцем розташування в мережі. Кожен нейрон шару Кохонена пов'язаний з кожним входом нульового (вхідного) шару прямими зв'язками, а також з усіма іншими нейронами поперечними зв'язками.

На рис. 1 представлена 1D-карта Кохонена.

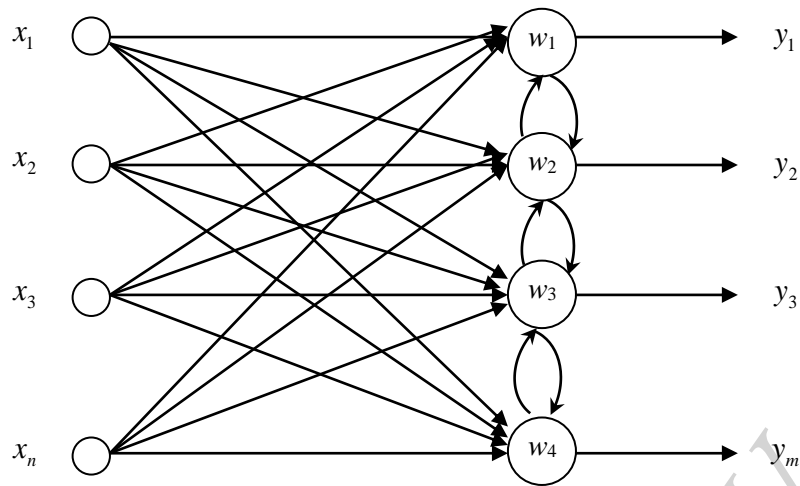


Рис. 1. 1D-карта Кохонена

В процесі навчання сусідні нейрони впливають один на одного сильніше ніж ті, які розташовані далі. Саме латеральні зв'язки в мережі забезпечують збудження одних нейронів і гальмування інших.

Кожен нейрон з шару Кохонена формує зважену суму сигналів  $f(x, w) = \sum_{i=1} w_i x_i$ . При цьому, якщо синапси прискорюють, то  $w_{ij} > 0$ . Якщо ж синапси гальмуючі, то  $w_{ij} < 0$ .

Враховуючі зазначене, класичною процедурою навчання мережі Кохонена є корегування синаптичних ваг, без врахування інших можливостей навчання мережі, таких як вид та параметри функції належності та архітектура мережі.

## 5. Результати дослідження з розробки методики навчання штучних нейронних мереж

### 5.1. Розробка алгоритму методики навчання штучних нейронних мереж

На рис. 2 подано запропонований алгоритм навчання штучної нейронної мережі. Удосконалення зазначеного алгоритму навчання полягає в удосконаленні процедур 2, 3, 8 розробленого раніше методу навчання штучних нейронних мереж [2, 18, 32].

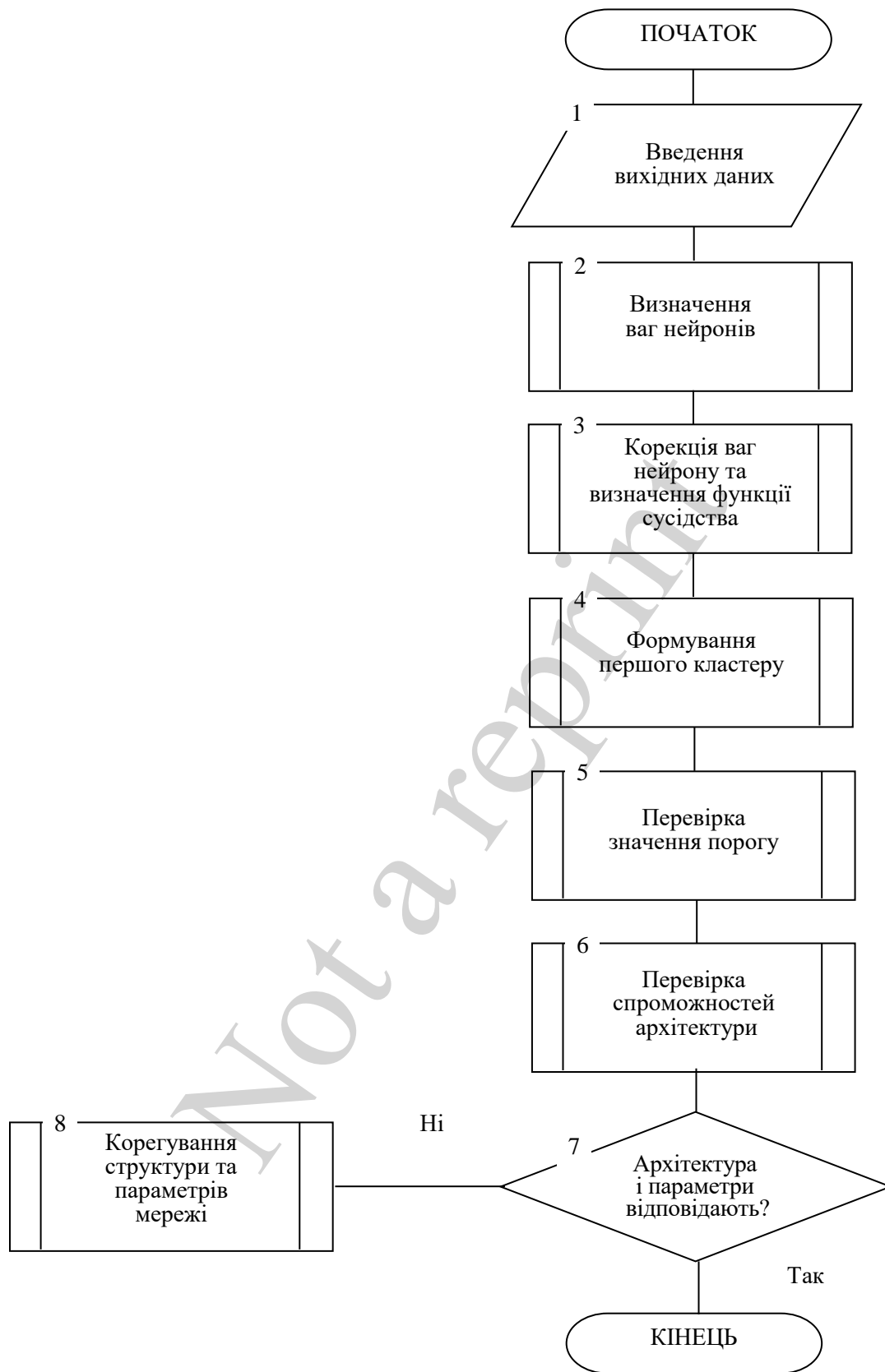


Рис. 2. Алгоритм функціонування та навчання штучної нейронної мережі, що еволюціонує

Коротко наведемо основні етапи реалізації запропонованої методики:

*Крок 1.* Початковим етапом є ініціалізація початкових значень синаптичних ваг.

*Крок 2.* Визначення ваг нейронів.

*Крок 3.* Корекція ваг нейрону та визначення функції сусідства.

*Крок 4.* Формування першого кластеру.

*Крок 5.* Перевірка значення порогу.

*Крок 6.* Перевірка спроможностей архітектури щодо обробки кількості інформації, що надходить на її вхід.

*Крок 7.* Еволюція архітектури системи.

Опишемо детально кроки 2, 3 та 8. Сутність удосконалення полягає в генетико-конкурентному навчанні, що доповнено введенням різних стратегій генетичної оптимізації вагових коефіцієнтів “мертвих” нейронів, розташованих на вихідному шарі мережі. Також, додатково враховується тип невизначеності навчальної вибірки (підхід детально наведений в дослідженні [32]). Запропонована стохастична оптимізація дозволяє скоротити кількість епох навчання мережі Кохонена при досягненні заданого максимального значення векторної помилки квантування та при побудові центроїдів додатково враховуються коефіцієнти невизначеності (повна невизначеність, часткова невизначеність, повна поінформованість) при виборі початкових значень центрів кластерів.

Перед початком роботи алгоритму навчання мережі Кохонена вхідні вектори попередньо нормалізуються [33, 34]:

$$\tilde{x}_i = \frac{x_i}{\sqrt{\sum_i x_i^2}} = \frac{x_i}{\|x\|}, \quad i = 1, 2, \dots, N. \quad (1)$$

Сам алгоритм навчання мережі Кохонена може бути описаний як послідовність кроків:

*Крок 1: Введення вихідних даних.* На даному етапі відбувається ініціалізація початкових значень синаптичних ваг  $w_{ij}=0$ .

Одним з часто застосовуваних способів ініціалізації є присвоювання синаптичним вагам значень, рівних випадково обраним векторам з безлічі спостережень.

*Крок 2: Визначення ваг нейронів.* На даному етапі на вхід системи подається нормалізований вектор сигналів  $\tilde{x}$  та вибирається вектор ваг (нейрон), найближчий до  $\tilde{x}$  тобто вектор, для якого Евклідова відстань до  $x$  буде найменшим:

$$\arg \min_j \|\tilde{x} - w_j\|, \quad j=1, 2, \dots, l. \quad (2)$$

Відбувається послідовність таких дій:

2. 1. Задання параметрів мережі Кохонена (розміру вихідної мережі  $I \times J$ , числа епох навчання  $T \geq 1$ , початкової ширини околиці нейронів  $\sigma_0$ , коефіцієнтів  $\tau, \kappa_0, \eta$ ).



2. 2. Занулення лічильника поточних ітерацій  $t:=0$ , ініціалізація вагових коефіцієнтів  $w_{ij}(1 \leq i \leq I, 1 \leq j \leq J)$  нейронів вихідної решітки випадковим чином, підготовка навчальних даних  $\{x_k\}_{k=1}^M$  вибір стратегії оптимізації  $G'$  вагових коефіцієнтів нейронів вихідної решітки.

Також, на зазначеному етапі відбувається обчислення поточної ширини країв центрів:

$$\sigma(t) = \sigma_0 \cdot \exp\left\{-\tau \cdot \frac{t}{T-1}\right\} \text{ для } T > 1$$

та

$$\sigma(t) = \sigma_0 \text{ для } T=1. \quad (3)$$

*Крок 3: Корекція ваг нейрону та визначення функції сусідства.*

Позначимо  $w_{ij}$  ваговий вектор нейрона, який має координати  $(i, j)$  на вхідній решітці мережі Кохонена ( $i$  – номер строки,  $j$  – номер стовбця). Процес навчання направлений на мінімізацію половинної суми квадратів відстаней між вхідними векторами  $\{x_k\}_{k=1}^M$  навчальної вибірки та векторами нейронів вихідної решітки ( $1 \leq i \leq I, 1 \leq j \leq J$ ).

$$E(w_{11}, \dots, w_{IJ}) = \frac{1}{2} \sum_{k=1}^M \left( D(w_{i_k^* j_k^*}, x_k) \right)^2 \rightarrow \min_{w_{11}, \dots, w_{IJ}}, \quad (4)$$

де  $D(w, x) = \sqrt{(w - x)^T \cdot (w - x)}$  – функція відстаней між парою векторів в евклі-

довому просторі,  $(i_k^*, j_k^*) = \arg \min_{\substack{1 \leq i \leq I \\ 1 \leq j \leq J}} D(w_{ij}, x_k)$  – координати нейрона на вихідному

шарі мережі, ваги якого найбільш близькі до вектору  $x_k$ .

Величина  $\frac{1}{M} \cdot E(w_{11}, \dots, w_{IJ})$ , є похибкою векторного квантування [24]. Використовуючи метод градієнтного спуску, отримуємо наступну формулу для оновлення вагових векторів:  $w_{ij}(1 \leq i \leq I, 1 \leq j \leq J)$ :

$$w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij},$$

$$\Delta w_{ij} = -\kappa \frac{\partial E(w_{11}, \dots, w_{IJ})}{\partial w_{ij}} = \kappa \sum_{k=1}^M \left( \left[ (i, j) = F_{IJ}(x_k) \right] (x_k - w_{ij}) \right), \quad (5)$$

де  $\kappa$  – деяка позитивна константа або функція з областю значень  $(0, 1]$ , що задає швидкість навчання. Зазначимо, що у (5) для оновлення кожного конкретного вагового вектору  $w_{ij}$  використовується тільки частина векторів із навчальної вибірки з найменшою нев'язкою між кожним із них  $w_{ij}$ . Іншими словами, вектор модифікується тоді і лише тоді, коли він є найближчим до навчального вектору  $x_k$  у рамках заданого метричного простору. Причому виправлення вектору  $w_{ij}$  здійснюється на величину, прямо пропорційну різниці між вхідним вектором  $x_k$  і ваговим вектором  $w_{ij}$ . Тим самим між нейронами на вихідній решітці створюється конкуренція за право бути відібраними у найближчі по відношенню до вхідного вектору  $x_k$  кандидати; нейрон, що задовольняє даній вимозі, називається нейроном-переможцем з координатами  $(i_k^*, j_k^*)$ . Зазначимо, що у разі нормування векторів  $w_{ij}$  і  $x_k$  мінімізація  $E(w_{11}, \dots, w_{IJ})$  рівносильна максимізації суми їх скалярних добутків:

$$E(w_{11}, \dots, w_{IJ}) = \frac{1}{2} \cdot \sum_{k=1}^M (w_{i_k^* j_k^*} - x_k)^T \cdot (w_{i_k^* j_k^*} - x_k) = \max_{w_{11}, \dots, w_{IJ}} \sum_{k=1}^M w_{i_k^* j_k^*}^T \cdot x_k - M. \quad (6)$$

Для зменшення конкуренції між нейронами вводиться правило, що дозволяє оновлювати не тільки ваги нейрона-переможця, але й інших нейронів, що лежать у його околі. З цією метою раніше введена характеристична функція  $[(i, j=F_{IJ})(x_k)]$  замінюється експоненційною функцією Гауса

$$\varphi(i, j) = \exp \left\{ -\frac{(i - i_k^*) + (j - j_k^*)^2}{2\sigma^2} \right\},$$

значення якої відображає загасаючу залеж-

ність зміни нейронних ваг зі збільшенням відстані від нейронів до нейрона-переможця на рівні їх координат на вихідній решітці. Чим ближче нейрон розташовується до нейрона-переможця, тим з більшим мультиплікативним коефіцієнтом оновлюються його ваги. Параметр  $\sigma$  називається ефективним шириною околиці [35–37] нейрона-переможця, який може бути інтерпретований як поточне значення радіуса оточення нейрона-переможця. Особливістю алгоритму навчання мережі Кохонена є зменшення значення  $\sigma$  з часом:

$$\sigma(t) = \sigma_0 \cdot \exp \left\{ -\tau \cdot \frac{t}{T-1} \right\}, \quad (t=0, \dots, T-1).$$

Тут параметр  $\sigma_0$  задає початкове значення радіуса околу нейрона-переможця, яке, як правило, встановлюється в

$\sqrt{I^2 + J^2}$ . Параметр  $\tau$  підбирається таким чином, щоб на останній епосі навчання оновленню піддавалися якнайменше число вагових векторів нейронів або зовсім лише один вектор нейрона-переможця. Тим самим було  $\tau = \ln(\sigma_0)$ . Коефіцієнт швидкості навчання обирається таким чином, щоб на початкових епохах алгоритму вагові вектори більшості нейронів оновлювалися з найбільшим темпом. Далі в міру збільшення кількості епох та звуження ширини околиці здійснювалася модифікація дедалі меншої кількості векторів нейронів з більш низькою швидкістю. Використання такого прийому дозволяє будувати кластери, чий

елементи спочатку пристосовуються під загальні характеристики апроксимуючої множини, а потім уточнюють його окремі особливості. Найбільш поширеними представниками з такою характерно спадною залежністю є функції  $\kappa(t)=\kappa_0(t+1)^{-1}$ ,  $\kappa(t)=\kappa_0 \cdot \exp\{-\eta t\}$  [35–37].

Сутність удосконалення процедури 3 полягає у використанні алгоритму конкурентного навчання мережі Кохонена, доповнений запровадженням генетичних операторів.

3. 1. Ініціалізація поточного набору активних нейронів  $V^+ := \emptyset$ .

3. 2. Виконання кроків 3. 2. 1–3. 2. 8 кожного вектору  $x_k$  ( $k=1, \dots, M$ ).

3. 2. 1. Нормування вагових коефіцієнтів нейронів  $w_{ij}$  за допомогою покомпонентного ділення на  $\|w_{ij}\|$  ( $w_{ij} \neq \vec{0}$ ).

3. 2. 2. Нормалізація вектору  $x_k$  за допомогою покомпонентного ділення на  $\|x_k\|$  ( $x_k \neq \vec{0}$ ).

3. 2. 3. Обчислення відстаней між вектором  $x_k$  та кожним ваговим вектором  $w_{ij}$  нейрона:  $d_{ijk} = D(x_k, w_{ij}) = \sqrt{\sum_{l=1}^n (x_{kl} - w_{ijl})^2}$ .

3. 2. 4. Визначення координат нейрона-переможця для вектору  $x_k$ :

$$(i_k^*, j_k^*) = F_{IJ}(x_k) = \arg \min_{\substack{1 \leq i \leq I \\ 1 \leq j \leq J}} d_{ijk}. \quad (7)$$

3. 2. 5. Визначення поточного околу нейрона-переможця  $(i_k^*, j_k^*)$ :

$$V^* = V(i_k^*, j_k^*) = \left\{ (i, j) \mid \left( \begin{array}{l} 1 \leq i \leq I \\ 1 \leq j \leq J \end{array} \right) \wedge (i - i_k^*)^2 + (j - j_k^*)^2 < \sigma^2(t) \right\}. \quad (8)$$

3. 2. 6. Модифікація вагових коефіцієнтів нейронів із координатами  $(i, j) \in V^*$ :  $w_{ij} := w_{ij} + \kappa(t) \cdot \varphi(i, j, t) \cdot (x_k - w_{ij})$ , де функція

$$\varphi(i, j, t) = \exp \left\{ \frac{(i - i_k^*) + (j - j_k^*)^2}{2\sigma^2(t)} \right\}.$$

3. 2. 7. Розширення набору активних нейронів:  $V^+ := V^+ \cup \{(i_k^*, j_k^*)\}$ .

3. 2. 8. Застосування стратегії генетичної оптимізації  $G'$  до ваг  $w_{ij}$  для  $(i, j) \in V^+, V^* \setminus \{(i_k^*, j_k^*)\}$ .

3. 3. Застосування стратегії генетичної оптимізації  $G'$  до ваг  $w_{ij}$  для  $(i, j) \in V^+, V^\dagger = V_{(i_M^*, j_M^*)} \setminus V^+$ .

3. 4. Збільшення лічильника поточних ітерацій:  $t:t+1$ .

3. 5. Перехід до наступного кроку 3. 4 при виконанні умови  $t \geq T$ , інакше перехід до кроку 2.

3. 6. Виключення “мертвих” нейронів з координатами  $(i', j') \in V^\dagger$  на вихідній решітці, де  $V^\dagger = \{(i, j) | \forall k \in \{1, \dots, M\} (i, j) \neq F_{II}(x_k)\}$ .

3. 7. Обчислення порога активації нейронів, що залишилися, з координатами  $(i'', j'') \notin V^\dagger : h_{i'', j''}^- = \min_{1 \leq k \leq M} \{d_{i'' j'' k}^{-1} | (i'', j'') = F_{II}(x_k)\}$ .

*Крок 6.* Перевірка спроможностей архітектури щодо обробки кількості інформації, що надходить на її вхід з оптимізацією архітектури штучної нейронної мережі.

Після модифікації вагових коефіцієнтів при пред'явленні навчального вектору (п. 3. 2. 8) або після виконання кількох епох конкурентного навчання (п. 3. 3) додатково застосовується стохастична оптимізація. Зазначена оптимізація заснована на генетичному алгоритмі стохастична оптимізація вагових коефіцієнтів певних нейронів вихідних решіток карти Кохонена. З цією метою ваги кожного нейрона, що знаходиться на краях поточного нейрона-переможця і жодного разу не активованого, видаються як послідовність генів, що виступають у ролі мінімальної одиниці для вхідного аргументу оператора схрещування ( $O_1$ ). В результаті виконання цього оператора формується пара нових хромосом, у яких переставлені місцями довільно вибрані ділянки ген хромосом. Кожен ген є набором бітів, який можна розглядати як окремих компонент вектору, асоційованого з відповідним нейроном-переможцем або одним з нейронів, що лежать у його околиці. При використанні оператора мутації ( $O_2$ ) здійснюється перестановка пари випадково вибраних бітів усередині одного гена, при використанні оператора інверсії ( $O_3$ ) відбувається інвертування значення випадково вибраного біта. Обидва ці оператори застосовуються тільки до частини мантиси 64-бітного "речового гена". Для імітації процесу еволюції нейронів було розроблено кілька способів генерації породження поколінь. Перший підхід  $A_1$  полягає в застосуванні операторів схрещування, мутації або інверсії довільних нейронів, що знаходяться в поточній близькості від нейрона-переможця. Другий підхід  $A_2$  заснований на геометричних міркуваннях про взаємне розташування нейронів на вихідній решітці щодо нейрона-переможця. Оскільки ваговий вектор кожного нейрона, рівновіддаленого від нейрона-переможця, модифікується з однаковим коефіцієнтом в результаті виконання обраного алгоритму навчання, то оператор схрещування пропонується застосовувати саме до таких нейронів. Третій підхід  $A_3$  передбачає застосування оператора схрещування тільки до найбільш пристосованих особин, у той час як до нейронів, що залишилися, буде використовуватися оператор мутації або інверсії. Тут у ролі функції пристосованості було вибрано зворотне значення величини середнього відхилення при розпізнаванні даним нейроном з ваговим коефіцієнтом  $w_{ij}$  елементів навчальної

$$\{x_k\}_{k=1}^M \text{ вибірки } \{x_k\}_{k=1}^M : \Psi_{ij} = \left( \frac{\sum_{k=1}^M \|x_k - w_{ij}\|}{M} \right)^{-1} \text{ для п. 3. 5 або, } \Psi_{ijk} = \|x_k - w_{ij}\|^{-1}$$

для п. 3. 2. 8. Крім того, було розроблено кілька стратегій для вибору того або іншого способу генерування нейронів: фіксований вибір  $G_1$  визначеного підходу, послідовний чи випадковий перебір усіх підходів  $G_2$  та вибір, заснований на механізмі рулетки з [32]. У стратегії  $G_3$ , згенероване за допомогою обраного підходу потомство нейронів, є більш пристосованим у порівнянні з попередніками. Тоді ймовірність вибору такого підходу в майбутньому збільшується порівняно з рештою підходів, інакше ймовірність його вибору зменшується.

Кожен із підходів  $A_1, A_2, A_3$  застосовується для створення двох дочірніх карт Кохонена, в яких змінам піддаються тільки нейрони з навколишнього середовища нейрона-переможця батьківської карти Кохонена. Ці підходи можуть бути описані як послідовність таких дій, в яких різним є лише крок 3:

*Крок 6. 1.*  $V := \emptyset$ .

*Крок 6. 2* Довільний вибір оператора  $O_{i^*} (i^* \in \{1, 2, 3\}); (3_{A_1})$ .

*Крок 6. 3* Застосування оператора  $O_{i^*}$  до чергового нейрона з координатами  $(i', j') \in V^\dagger \setminus V$  з поточної околиці  $V^*$  нейрона-переможця (якщо  $O_{i^*}$  – оператор схрещування, необхідно додатково вибрати ще один нейрон з координатами  $(i'', j'') (3_{A_2})$ , відмінними від  $(i', j')$  оператора схрещування до пари нейронів з координатами  $(i', j') \in V^\dagger \setminus V$  та

$(i'', j'') \in U = \{(i, j)\}_{(i, j) \in V^* \setminus V \setminus \{(i', j')\} \wedge (i - i_{k(M)}^*)^2 + (j + j_{k(M)}^*)^2 = (i' - i_{k(M)}^*)^2 + (j' - j_{k(M)}^*)^2}$  з околу  $V^*$  нейрона-переможця з координатами  $(i_{k(M)}^*, j_{k(M)}^*)$  (якщо  $U = \emptyset$ , то необхідно застосувати

двічі оператор мутації або інверсії до нейрона з координатами  $(i', j') (3_{A_3})$ . Використання оператора схрещування до нейронів з координатами  $(i', j') = \arg \max_{(i, j) \in V^\dagger \setminus V} \psi_{ij}(k)$  та  $(i'', j'') = \arg \max_{(i, j) \in V^* \setminus V \setminus \{(i', j')\}} \psi_{ij}(k)$  (якщо

$V^\dagger \setminus V \setminus \{(i', j')\} = \emptyset \vee \#(V^\dagger \setminus V) \leq K \cdot \#V^*$ , то необхідно використати двічі оператор мутації або інверсії до нейрона з координатами  $(i', j')$ , де  $0 \leq K \leq 1$  – константа, що визначає відносне число нейронів, до яким повинен застосовуватися лише оператор мутації чи інверсії).

*Крок 6. 4.* додавання двох нових згенерованих за допомогою оператора  $O_{i^*}$  нейронів по одному в кожному з двох дочірніх карт Кохонена із збереженням кожного з цих нейронів у позиції  $(i', j')$  на вихідній решітці.

*Крок 6. 5* Додавання координат  $(i', j')$  до множини  $V : V := V \cup \{(i', j')\}$ .

*Крок 6. 6.* Перехід до кроку 6. 2, якщо  $V \neq V^\dagger$  інакше зупинка.

Кожна зі стратегій  $G_1, G_2, G_3$  маніпулює вибором підходів  $A_1, A_2, A_3$ . Стратегія  $G_1$  є найпримітивнішою з аналізованих і полягає у виборі одного з трьох підходів  $A_1, A_2, A_3$  на кроці 6. 1 алгоритму генетико-конкурентного навчання мережі Кохонена.

Отже, якщо є  $N$  спостережень та  $m$  кластерів з центроїдами  $c_j$ , розрахунки всіх належностей та скореговані координати центроїдів оцінюються згідно від-

ношення:

$$\left\{ \begin{array}{l} u_j(k) = \left( 1 + \left( \frac{\|x(k) - c\|^2}{\mu_j} \right)^{\frac{1}{\beta-1}} \right)^{-1}, \\ c_j = \frac{\sum_{k=1}^N u_j^\beta(k) x(k)}{\sum_{k=1}^N u_j^\beta(k)}, \\ \mu_j(k) = \frac{\sum_{k=1}^N u_j^\beta(k) \|x(k) - c_j\|^2}{\sum_{k=1}^N u_j^\beta(k)}, \\ \psi_{ij}(k) = \left( \frac{\sum_{k=1}^M \|x_k - w_{ij}\|}{M} \right)^{-1}. \end{array} \right. \quad (9)$$

Система рівнянь (9) представляє собою по суті пакетний алгоритм обробки інформації так, що при надходженні спостереження  $x(N+1)$  усі розрахунки повинні бути проведені заново. При достатньо високій частоті надходження даних підхід може виявитися неефективним.

З цією метою потрібно провести розробку рекурентних процедур, що не потребують зберігання відпрацьованих раніше даних. Реалізувати зазначені рекурентні процедури можна на основі двошарової адаптивної нейро-нечіткої мережі з такою архітектурою.

Перший скритий шар мережі утворений звичайними нейронами Кохонена  $N_j^K$ , пов'язані між собою латеральними зв'язками, за якими реалізується процес конкуренції. Вихідний шар мережі, утворений вузлами  $N_j^u$ , призначений для розрахунку рівнів належності кожного спостереження  $x(k)$  кожному  $j$ -му кластеру,  $j=1, 2, 3, \dots, m$ . Для налаштування центрів кластерів використовується рекурентна процедура самонавчання, яка має вигляд [10]:

$$\left\{ \begin{array}{l}
c_j(k+1) = c_j(k) + \frac{u_j^\beta(k)}{k+1} (x(k+1) - c_j(k)), \\
u_j(k+1) = \frac{1}{1 + \left( \frac{\|x(k+1) - c_j(k+1)\|^2}{\mu_j(k)} \right)^{\frac{1}{1-\beta}}}, \\
\mu_j(k+1) = \frac{\sum_{p=1}^{k+1} u_j^\beta(p) \|x(p) - c_j(k+1)\|^2}{\sum_{p=1}^{k+1} u_j^\beta(p)}, \\
\psi_{ij}(k+1) = \left( \frac{\sum_{M=1}^{k+1} \mu_j(k+1) \|x_k - w_{ij}\|}{M} \right)^{-1}.
\end{array} \right. \quad (10)$$

Неважно помітити що перший вираз (10) і є WTM-правилом самонавчання зі звужуючою функцією сусідства  $(k+1)^{-1} u_j^\beta(k)$ .

Дії 4, 5 7 детально описано в попередніх дослідженнях авторів [18, 32], а дії 2, 3, 6 описані відповідно за виразами (1) по (10).

## 5. 2. Експериментальна оцінка ефективності навчання штучних нейронних мереж

Виконано моделювання роботи запропонованої методики в програмному середовищі MathCad 14. Для проведення експерименту використовувався двовимірний штучно згенерований набір даних з різним ступенем невизначеності (для зручності розрахунку по 1500 кожного типу невизначеності. Тобто згенерований набір даних складається з даних:

- що є повністю достовірні (повна поінформованість про стан об'єкту), тобто розвідувальні дані повні та достовірні;
- часткова невизначеність (відсутня інформація по окремому показнику оцінки);
- повна невизначеність (відсутня інформація про об'єкт моніторингу).

Набір даних містив 15 кластерів з різним рівнем перекриття. Вибірка даних містила 4500 спостережень. Дані подавалися на обробку в послідовному режимі.

Для порівняння якості кластеризації використовувалися FCM і система, заснована на еволюційному методі нечіткої кластеризації (Evolving Fuzzy Clustering Method, EFCM) з різними значеннями параметра порогу, EFCM система з методом навчання з роботи [18, 32], K-means++ та K-medoids (Partitioning Around Medoid).

Для проведення наступного експерименту була використана вибірка даних, що описує характеристики об'єкту моніторингу. Кожне спостереження описувалося сімома параметрами:

- кількість особового складу;
- загальна чисельність особового складу;
- кількість організаційно-штатних структур;
- кількість зразків озброєння та військової техніки;
- кількість засобів зв'язку;
- кількість типів зразків озброєння та військової техніки та тип засобів зв'язку.

Перед кластеризацією ознаки спостережень були нормовані на інтервалі [0, 1].

Як критерій оцінки якості кластеризації був використаний індекс Ксі-Бені. У табл. 1 представлені порівняльні результати кластеризації.

Дослідження розробленої методики показало, що зазначена методика навчання забезпечує в середньому на 11–15 % більшу високу ефективність навчання штучних нейронних мереж та не накопичує похибок в процесі навчання (табл. 1).

Зазначені результати видно з результатів в останніх строках табл. 1 як різниця індексу Ксі-Бені.

Таблиця 1  
Порівняльні результати кластеризації

Система	Кількість кластерів	Параметри алгоритму	ХВ (Індекс Ксі-Бені)	Час кластеризації, с
FCM (Fuzzy C-Means)	15	delta=0.1	0.1903	1.4
K-means++	15	delta=0.1	0.1361	0,69
K-medoids	15	delta=0.1	0.1256	0,651
EFCM	15	Dthr=0.24	0.148	0.54
EFCM	15	Dthr=0.19	0.139	0.49
Запропонована роботі [18, 32] система (пакетний режим)	15	delta=0.1	0.12	0.37
Запропонована в роботі [18, 32] система (online режим)	15	delta=0.1	0.1127	0.25
Запропонована система (пакетний режим)	15	delta=0.1	0.1006	0.37
Запропонована система (online режим)	15	delta=0.1	0.10045	0.25

## 6. Обговорення результатів з розробки методики навчання штучних нейронних мереж

Переваги зазначеної методики досягаються виконанням послідовності додаткових процедур, а саме 2, 3, 6, які наведені на рис. 2.



Аналітичні залежності для виконання зазначених процедур наведені в формулах, а саме:

- залежності (2), (3) – крок 2 “Визначення ваг нейронів”;
- залежності (4)–(8) – крок 3 “Корекція ваг нейрону та визначення функції сусідства”;
- залежності (9), (10) – крок 6 “Перевірка спроможностей архітектури”.

Тобто, відбувається корегування не тільки синаптичних ваг, але і архітектури штучних нейронних мереж, що забезпечує меншу кількість епох для навчання.

Основними перевагами запропонованої методики оцінки є такі:

- дозволяє провести обробку більшої кількості інформації за рахунок зменшення удосконаленого алгоритму навчання;
- підвищена достовірність отриманих результатів відсутності накопичення помилки навчання в ході навчання штучних нейронних мереж. Зазначене досягається за рахунок удосконаленої процедури корегування синаптичних ваг та архітектури штучної нейронної мережі;
- широка сфера використання (системи підтримки та прийняття рішень);
- простота математичних розрахунків;
- недопущення явища “перенавчання” за рахунок корегування архітектури мережі;
- відсутність необхідності зберігання результатів попередніх обчислень.

До недоліків запропонованої методики слід віднести:

- необхідність використання високошвидкісних генетичних алгоритмів для корегування синаптичних ваг;
- менша точність оцінювання по окремо взятому параметру оцінювання;
- втрата точності результатів за час переналагодження архітектури штучної нейронної мережі.

Зазначена методика дозволить:

- проводити навчання штучних нейронних мереж;
- визначити ефективні заходи для підвищення оперативності навчання штучних нейронних мереж за рахунок удосконаленої процедури синаптичних ваг та архітектури мереж;
- зменшити використання обчислювальних ресурсів систем підтримки та прийняття рішень;
- виробити заходи, що спрямовані на підвищення ефективності навчання штучних нейронних мереж;
- підвищити оперативність обробки інформації в штучних нейронних мережах.

До обмежень зазначеного дослідження слід віднести:

- необхідність наявності достовірних та повних навчальних та тестових вибірок;
- необхідність врахування часу на збір, обробку та узагальнення інформації;
- для забезпечення заданого ступеню оперативності навчання при програмній реалізації методики необхідно враховувати обчислювальні можливості апаратної частини.

Напрями подальших досліджень слід спрямувати на зменшення обчислювальних витрат при обробці різнотипних даних в системах спеціального призначення.

## **7. Висновки**

1. Розроблено алгоритм навчання методики навчання штучних нейронних мереж для інтелектуальних систем підтримки прийняття рішень.

Підвищення оперативності обробки інформації та зменшення похибки оцінювання досягається за рахунок:

– навчання не тільки синаптичних ваг штучної нейронної мережі, але й виду та параметрів функції належності;

– навчання архітектури штучних нейронних мереж;

– обчислення даних за одну епоху без необхідності зберігання попередніх обчислень. Зазначене зменшує час на обробку інформації за рахунок відсутності необхідності звернення до бази даних;

– відсутності накопичення похибки навчання штучних нейронних мереж в результаті обробки інформації, що надходить на вхід штучних нейронних мереж.

2. Проведений приклад використання запропонованої методики на прикладі кластеризації об'єкту моніторингу. Зазначений приклад показав підвищення ефективності функціонування штучних нейронних мереж на рівні 10–18 % за індексом Ксі-Бені та оперативності обробки інформації за рахунок використання додаткових процедур навчання штучних нейронних мереж.

## **Подяки**

Авторський колектив висловлює подяку за надання допомоги в підготовці статті:

Доктору технічних наук, професору Кувшинову Олексію Вікторовичу – заступнику начальника навчально-наукового інституту Національного університету оборони України імені Івана Черняхівського.

Кандидату технічних наук, доценту Башкирову Олександрю Миколайовичу – провідному науковому співробітнику Центрального науково-дослідного інституту озброєння та військової техніки Збройних Сил України.

## **Література**

1. Kalantaievska, S., Pievtsov, H., Kuvshynov, O., Shyshatskyi, A., Yarosh, S., Gatsenko, S. et. al. (2018). Method of integral estimation of channel state in the multiantenna radio communication systems. *Eastern-European Journal of Enterprise Technologies*, 5 (9 (95)), 60–76. doi: <https://doi.org/10.15587/1729-4061.2018.144085>

2. Kuchuk, N., Mohammed, A. S., Shyshatskyi, A., Nalapko, O. (2019). The method of improving the efficiency of routes selection in networks of connection with the possibility of self-organization. *International Journal of Advanced Trends in Computer Science and Engineering*, 8 (1.2), 1–6. URL: <http://www.warse.org/IJATCSE/static/pdf/file/ijatcse01812sl2019.pdf>

3. Raskin, L., Sira, O. (2016). Method of solving fuzzy problems of mathematical programming. *Eastern-European Journal of Enterprise Technologies*, 5 (4 (83)), 23–28. doi: <https://doi.org/10.15587/1729-4061.2016.81292>
4. Katranzhy, L., Podskrebko, O., Krasko, V. (2018). Modelling the dynamics of the adequacy of bank's regulatory capital. *Baltic Journal of Economic Studies*, 4 (1), 188–194. doi: <https://doi.org/10.30525/2256-0742/2018-4-1-188-194>
5. Manea, E., Di Carlo, D., Depellegrin, D., Agardy, T., Gissi, E. (2019). Multidimensional assessment of supporting ecosystem services for marine spatial planning of the Adriatic Sea. *Ecological Indicators*, 101, 821–837. doi: <https://doi.org/10.1016/j.ecolind.2018.12.017>
6. Çavdar, A. B., Ferhatosmanoğlu, N. (2018). Airline customer lifetime value estimation using data analytics supported by social network information. *Journal of Air Transport Management*, 67, 19–33. doi: <https://doi.org/10.1016/j.jairtraman.2017.10.007>
7. Kachayeva, G. I., Mustafayev, A. G. (2018). The use of neural networks for the automatic analysis of electrocardiograms in diagnosis of cardiovascular diseases. *Herald of Dagestan State Technical University. Technical Sciences*, 45 (2), 114–124. doi: <https://doi.org/10.21822/2073-6185-2018-45-2-114-124>
8. Zhdanov, V. V. (2016). Experimental method to predict avalanches based on neural networks. *Ice and Snow*, 56 (4), 502–510. doi: <https://doi.org/10.15356/2076-6734-2016-4-502-510>
9. Kanev, A., Nasteka, A., Bessonova, C., Nevmerzhitsky, D., Silaev, A., Efremov, A., Nikiforova, K. (2017). Anomaly detection in wireless sensor network of the “smart home” system. 2017 20th Conference of Open Innovations Association (FRUCT). doi: <https://doi.org/10.23919/fruct.2017.8071301>
10. Sreeshakthy, M., Preethi, J. (2015). Classification of human emotion from deap EEG signal using hybrid improved neural networks with Cuckoo Search. *Brain: Broad Research in Artificial Intelligence and Neuroscience*, 6 (3-4), 60–73. URL: <https://brain.edusoft.ro/index.php/brain/article/view/519>
11. Chica, J., Zaputt, S., Encalada, J., Salamea, C., Montalvo, M. (2019). Objective assessment of skin repigmentation using a multilayer perceptron. *Journal of Medical Signals & Sensors*, 9 (2), 88. doi: [https://doi.org/10.4103/jmss.jmss\\_52\\_18](https://doi.org/10.4103/jmss.jmss_52_18)
12. Massel, L. V., Gerget, O. M., Massel, A. G., Mamedov, T. G. (2019). The Use of Machine Learning in Situational Management in Relation to the Tasks of the Power Industry. *EPJ Web of Conferences*, 217, 01010. doi: <https://doi.org/10.1051/epjconf/201921701010>
13. Abaci, K., Yamacli, V. (2019). Hybrid Artificial Neural Network by Using Differential Search Algorithm for Solving Power Flow Problem. *Advances in Electrical and Computer Engineering*, 19 (4), 57–64. doi: <https://doi.org/10.4316/aece.2019.04007>
14. Mishchuk, O. S., Vitynskyi, P. B. (2018). Neural Network with Combined Approximation of the Surface of the Response. *Research Bulletin of the National Technical University of Ukraine “Kyiv Politechnic Institute”*, 2, 18–24. doi: <https://doi.org/10.20535/1810-0546.2018.2.129022>

15. Kazemi, M., Faezirad, M. (2018). Efficiency estimation using nonlinear influences of time lags in DEA Using Artificial Neural Networks. *Industrial Management Journal*, 10 (1), 17–34. doi: <http://doi.org/10.22059/imj.2018.129192.1006898>
16. Parapuram, G., Mokhtari, M., Ben Hmida, J. (2018). An Artificially Intelligent Technique to Generate Synthetic Geomechanical Well Logs for the Bakken Formation. *Energies*, 11 (3), 680. doi: <https://doi.org/10.3390/en11030680>
17. Prokoptsev, N. G., Alekseenko, A. E., Kholodov, Y. A. (2018). Traffic flow speed prediction on transportation graph with convolutional neural networks. *Computer Research and Modeling*, 10 (3), 359–367. doi: <https://doi.org/10.20537/2076-7633-2018-10-3-359-367>
18. Dudnyk, V., Sinenko, Y., Matsyk, M., Demchenko, Y., Zhyvotovskiy, R., Repilo, I. et. al. (2020). Development of a method for training artificial neural networks for intelligent decision support systems. *Eastern-European Journal of Enterprise Technologies*, 3 (2 (105)), 37–47. doi: <https://doi.org/10.15587/1729-4061.2020.203301>
19. Bodyanskiy, Ye., Pliss, I., Vynokurova, O. (2013). Flexible wavelet-neuro-fuzzy neuron in dynamic data mining tasks. *Oil and Gas Power Engineering*, 2 (20), 158–162. URL: [http://nbuv.gov.ua/UJRN/Nge\\_2013\\_2\\_18](http://nbuv.gov.ua/UJRN/Nge_2013_2_18)
20. Haykin, S. (2009). *Neural networks and learning machines*. Pearson, 906. URL: <http://dai.fmph.uniba.sk/courses/NN/haykin.neural-networks.3ed.2009.pdf>
21. Nelles, O. (2001). *Nonlinear System Identification*. Springer, 786. doi: <https://doi.org/10.1007/978-3-662-04323-3>
22. Wang, L.-X., Mendel, J. M. (1992). Fuzzy basis functions, universal approximation, and orthogonal least-squares learning. *IEEE Transactions on Neural Networks*, 3 (5), 807–814. doi: <https://doi.org/10.1109/72.159070>
23. Kohonen, T. (1995). *Self-Organizing Maps*. Springer, 362. doi: <https://doi.org/10.1007/978-3-642-97610-0>
24. Kasabov, N. (2003). *Evolving Connectionist Systems*. Springer, 307. doi: <https://doi.org/10.1007/978-1-4471-3740-5>
25. Sugeno, M., Kang, G. T. (1988). Structure identification of fuzzy model. *Fuzzy Sets and Systems*, 28 (1), 15–33. doi: [https://doi.org/10.1016/0165-0114\(88\)90113-3](https://doi.org/10.1016/0165-0114(88)90113-3)
26. Ljung, L. (1999). *System Identification. Theory for the User*. PTR Prentice Hall, Upper Saddle River, 609. URL: <https://www.twirpx.com/file/277211/>
27. Otto, P., Bodyanskiy, Y., Kolodyazhniy, V. (2003). A new learning algorithm for a forecasting neuro-fuzzy network. *Integrated Computer-Aided Engineering*, 10 (4), 399–409. doi: <https://doi.org/10.3233/ica-2003-10409>
28. Narendra, K. S., Parthasarathy, K. (1990). Identification and control of dynamical systems using neural networks. *IEEE Transactions on Neural Networks*, 1 (1), 4–27. doi: <https://doi.org/10.1109/72.80202>
29. Petruk, S., Zhyvotovskiy, R., Shyshatskyi, A. (2018). Mathematical Model of MIMO. 2018 International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (PIC S&T). doi: <https://doi.org/10.1109/infocommst.2018.8632163>

30. Zhyvotovskiy, R., Shyshatskiy, A., Petruk, S. (2017). Structural-semantic model of communication channel. 2017 4th International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (PIC S&T). doi: <https://doi.org/10.1109/infocommst.2017.8246454>
31. Alieinykov, I., Thamer, K. A., Zhuravskiy, Y., Sova, O., Smirnova, N., Zhyvotovskiy, R. et. al. (2019). Development of a method of fuzzy evaluation of information and analytical support of strategic management. Eastern-European Journal of Enterprise Technologies, 6 (2 (102)), 16–27. doi: <https://doi.org/10.15587/1729-4061.2019.184394>
32. Koshlan, A., Salnikova, O., Chekhovska, M., Zhyvotovskiy, R., Prokopenko, Y., Hurskiy, T. et. al. (2019). Development of an algorithm for complex processing of geospatial data in the special-purpose geoinformation system in conditions of diversity and uncertainty of data. Eastern-European Journal of Enterprise Technologies, 5 (9 (101)), 35–45. doi: <https://doi.org/10.15587/1729-4061.2019.180197>
33. Гороховатський, В., Стяглик, Н., Царевська, В. (2021). Комбінаційний метод прискороного метричного пошуку даних у задачах класифікації зображень. Сучасні інформаційні системи, 5 (3), 5–12. doi: <https://doi.org/10.20998/2522-9052.2021.3.01>
34. Левашенко, В. Г., Ляшенко, А. С., Кучук, Г. А. (2020). Побудова системи підтримки прийняття рішень на основі нечітких даних. Сучасні інформаційні системи, 4 (4), 48–56. doi: <https://doi.org/10.20998/2522-9052.2020.4.07>
35. Meleshko, Y., Drieiev, O., Drieieva, H. (2020). Method of identification bot profiles based on neural networks in recommendation systems. Advanced Information Systems, 4 (2), 24–28. doi: <https://doi.org/10.20998/2522-9052.2020.2.05>
36. Кучук, Н. Г., Мерлак, В. Ю., Скороделов, В. В. (2020). Метод зменшення часу доступу до слабкоструктурованих даних. Сучасні інформаційні системи, 4 (1), 97–102. doi: <https://doi.org/10.20998/2522-9052.2020.1.14>
37. Shyshatskiy, A., Tiurnikov, M., Suhak, S., Bondar, O., Melnyk, A., Bokhno, T., Lyashenko, A. (2020). Method of assessment of the efficiency of the communication of operational troop grouping system. Advanced Information Systems, 4 (1), 107–112. doi: <https://doi.org/10.20998/2522-9052.2020.1.16>