https://doi.org/10.31891/2307-5732-2025-359-66
УДК 004.4

ГРІНЕНКО ОЛЕНА
Державний університет «Київський авіаційний інститут»
https://orcid.org/0000-0001-9673-6626
email: olena.hrinenko@npp.nau.edu.ua
СУХОВИЙ ОЛЕКСАНДР
Державний університет «Київський авіаційний інститут»
https://orcid.org/0009-0006-2912-5327
email: 1462622@stud.kai.edu.ua

ПОРІВНЯЛЬНИЙ АНАЛІЗ НАВЧАННЯ МОДЕЛЕЙ НА ЖОРСТКИХ МІТКАХ З НАВЧАННЯМ ЧЕРЕЗ ДИСТИЛЯЦІЮ ЗНАНЬ ДЛЯ ЗАДАЧ КЛАСИФІКАЦІЇ ЗОБРАЖЕНЬ В УМОВАХ ОБМЕЖЕНИХ ОБЧИСЛЮВАЛЬНИХ РЕСУРСІВ

У цій роботі представлено порівняльний аналіз двох підходів до навчання моделей штучних нейронних мереж для задач класифікації зображень: традиційного навчання на жорстких мітках і дистиляції знань, яка використовує м'які ймовірнісні виходи попередньо навченої моделі-вчителя для спрямування процесу навчання меншої мережі-учня. Метою дослідження є оцінка відносної ефективності та практичного впливу цих методів у середовищах із обмеженими обчислювальними ресурсами, де зменшення розміру моделі та часу інференсу є критичним без втрати точності прогнозування. У межах експериментальної частини було розроблено та протестовано кілька архітектур мереж. Реалізовано дві схеми передавання знань: дистиляцію від одного вчителя та від ансамблю з десяти незалежно натренованих моделей, що відображає як індивідуальні, так і колективні джерела знань. Усі експерименти проводилися на наборі даних MNIST - стандартному еталоні для розпізнавання рукописних цифр. Основними метриками оцінювання були точність і частота помилок. Кожна конфігурація навчалася та перевірялася десять разів для забезпечення статистичної достовірності результатів і усунення випадкових флуктуацій, пов'язаних з ініціалізацією та оптимізацією моделей. Отримані результати показують, що за протестованих умов дистиляція знань не забезпечила помітного підвищення точності моделей-учнів, а в деяких випадках навіть призвела до незначного зниження результатів порівняно з класичним навчанням на жорстких мітках. Ці висновки свідчать, що хоча дистиляція знань залишається потужною концепцією у великомасштабному глибокому навчанні, її переваги можуть бути обмеженими для простих повнозв'язних архітектур або невеликих наборів даних із низькою варіативністю вхідних даних. Відтак для задач, обмежених апаратними чи енергетичними ресурсами, незалежне навчання на жорстких мітках залишається більш надійною та обчислювально оптимальною стратегією.

Ключові слова: *Машинне навчання, нейронні мережі, дистиляція знань, оптимізація моделей, обмежені обчислювальні ресурси.*

OLENA GRINENKO
SUKHOVYI OLEKSANDR
State university "Kyiv Aviation Institute"

COMPARATIVE ANALYSIS OF TRAINING MODELS ON HARD LABELS VERSUS KNOWLEDGE DISTILLATION FOR IMAGE CLASSIFICATION TASKS UNDER LIMITED COMPUTATIONAL RESOURCES

This paper presents a comparative analysis of two approaches to training artificial neural network models for image classification tasks: traditional training on hard labels and knowledge distillation, which leverages the soft probabilistic outputs of a pretrained teacher model to guide the learning process of a smaller student network. The study aims to evaluate the relative efficiency and practical impact of these methods under constrained computational environments, where reducing model size and inference time is essential without sacrificing predictive performance. Within the experimental framework, several network architectures were developed and tested. Two knowledge transfer schemes were implemented: distillation from a single teacher and from an ensemble of ten independently trained models, representing both individual and collective knowledge sources. All experiments were conducted on the MNIST dataset, a standard benchmark for handwritten digit recognition, using accuracy and error rate as the primary evaluation metrics. Each configuration was trained and validated ten times to ensure statistical reliability and eliminate random fluctuations in model initialization and optimization. The results demonstrate that, under the tested conditions, knowledge distillation did not provide measurable improvement in student model accuracy and, in some cases, led to a moderate decrease compared to classical training on hard labels. These findings indicate that, while knowledge distillation remains a powerful concept in large-scale deep learning, its benefits may be limited for simple fully connected architectures or small datasets with low input variability. Consequently, for tasks constrained by hardware or energy efficiency, direct independent training on hard labels remains a more reliable and computationally optimal strategy.

Keywords: *Machine learning, neural networks, knowledge distillation, model optimization, limited computational resources.*

Стаття надійшла до редакції / Received 28.10.2025
Прийнята до друку / Accepted 15.11.2025

Вступ

У сучасному машинному навчанні однією з проблем залишається навчання моделей нейронних мереж із високою точністю для використання в умовах обмежених обчислювальних. [1] Обмежені обчислювальні ресурси – це така сукупність умов середовища виконання, за яких застосування складних моделей машинного навчання ускладнене або неможливе через дефіцит одного чи кількох критичних апаратних або часових ресурсів. До пристроїв з обмеженими обчислювальними ресурсами можна віднести мобільні телефони, пристрої інтернету речей, мікроконтролери, автономні дрони з живленням від батареї.

Одним із підходів до вирішення цієї проблеми є стиснення (компресія) моделей нейронних мереж. [2] Це дозволяє навчити велику нейромережу, а потім стиснути її без значної втрати точності. Дистиляція знань – одна з технік стиснення нейромереж, яка дозволяє передати знання від одної нейромережі до іншої. Модель що

виступає джерелом знань називають учителем, а модель що навчається – учнем. В ролі учителя може виступати як одна нейромережа, так і група нейромереж, що називається ансамблем. [3]

Дистиляція знань була запропонована у 2015-му році групою інженерів компанії Google DeepMind як метод підвищення точності роботи нейромереж через навчання на м'яких мітках отриманих як усереднене значення роботи ансамблю нейромереж. На відміну від класичного навчання на жорстких мітках це дає більше інформації для якісного навчання (зокрема, через розподіл імовірностей між класами можна передавати інформацію про подібність чи відмінність тих чи інших класів). Надалі ідея з навчанням на м'яких мітках була розширена з навчання на ансамблі нейромереж на навчання на м'яких мітках більших моделей [4].

Метод даного дослідження є аналіз можливостей використання дистиляції знань з ансамблю моделей, а також з однієї моделі в задачах, коли необхідно досягти максимальної точності роботи нейромережі в умовах обмежених обчислювальних ресурсів.

Теоретичні основи дослідження

Суть задачі класифікації зображень полягає у віднесенні вхідного зображення до одного з наперед визначених класів. У ролі базового набору даних для розв'язання цієї задачі був використаний набір даних MNIST (Modified National Institute of Standards and Technology) [5], що є одним із широко застосовуваних стандартних наборів даних у галузі глибокого навчання, комп'ютерного зору та нейронних мереж.

Цей набір даних містить чорно-білі зображення рукописних арабських цифр від 0 до 9, які були зібрані з великої кількості рукописних форм, наданих реальними людьми. Таким чином, зображення охоплюють різні стилі написання цифр, що робить задачу класифікації реалістичною і практично корисною для побудови систем розпізнавання. Кожне зображення представлено у вигляді градацій сірого (тобто 1-канальне зображення) з роздільною здатністю 28×28 пікселів, що дає 784 пікселі (ознаки) для обробки нейронною мережею після перетворення у вектор. Приклади зображень з набору даних MNIST приведено на рисунку 1.

Рис 1. Приклади зображень з набору даних MNIST – рукописні арабські цифри у вигляді зображень 28 на 28 пікселів

Загальна кількість зображень у наборі даних становить 70 000 одиниць, з яких 60 000 прикладів призначені для тренування моделі (тренувальний набір даних), і ще 10 000 зображень входять до складу тестового набору даних, що використовується для оцінки узагальнюючої здатності моделі після завершення навчання.

Набір даних MNIST був обраний через його доступність, популярність у науковій спільноті та легкість обробки, що дозволяє швидко оцінити ефективність різних підходів до навчання моделей, включаючи методи дистиляції знань, стиснення нейронних мереж і порівняння архітектур нейронних мереж. Попри свою простоту, MNIST залишається важливим еталонним набором для тестування нових архітектур нейронних мереж і навчальних стратегій, зокрема тих, що пов'язані з оптимізацією моделі для роботи на пристроях з обмеженими ресурсами.

Кожне зображення з набору даних представляється як двовимірне зображення розміром 28×28 пікселів у відтінках сірого. Щоб подати це зображення на вхід нейронній мережі, необхідно виконати операцію розгортання – перетворити його у вектор довжиною 784 елементи (28 × 28 = 784). Кожен елемент цього вектора містить значення яскравості відповідного пікселя та слугує окремою вхідною ознакою для моделі.

Для класифікації таких зображень застосовується повнозв'язна нейронна мережа. Вхідний шар складається з 784 нейронів – по одному для кожної ознаки. Далі слідує один або кілька прихованих шарів, які складаються з певної кількості нейронів (наприклад, 128, 256 або 800), що виконують лінійні перетворення та передають результати через нелінійну функцію активації. У нашому випадку використовується ReLU (Rectified Linear Unit) [6] – функція, яка повертає нуль для від'ємних значень і лінійно пропускає додатні:

$$ReLU(x) = max(0, x).$$

Завершує архітектуру вихідний шар, який складається з 10 нейронів – по одному для кожного класу від 0 до 9, що відповідають рукописним цифрам. На цьому шарі застосовується softmax-активація, яка перетворює вихідні значення (логіти) на ймовірнісне розподілення між класами, дозволяючи інтерпретувати результат як ймовірності належності зображення до кожної з цифр.

Ця проста архітектура дозволяє моделі ефективно навчатися на задачі класифікації рукописних цифр, забезпечуючи узагальнюваність на тестових прикладах.

Для перевірки якості дистиляції знань від більшої до меншої нейромережі потрібно обрати щонайменше 2 архітектури нейромережі – більшу складнішу, з більшою кількістю нейронів, а також меншу – з малою кількістю нейронів. Для великої нейромережі було обрано архітектуру з п'яти прихованих шарів розмірами 500, 400, 300, 200 та 100 нейронів відповідно, а для малої лише один прихований шар у 100 нейронів.

Для перевірки дистиляції знань від ансамблю нейромереж до одної нейромережі було прийнято рішення використовувати для ансамблю нейромереж ідентичну архітектуру, як в результуючої нейромережі з кількість моделей в ансамблі 10.

Процес навчання як моделі-учителя, так і моделей-учнів у рамках дистиляції знань проводився до повної конвергенції. Під конвергенцією в даному контексті розуміється стабілізація функції втрат, тобто момент, коли її значення припиняє помітно знижуватись протягом подальших епох навчання. Це свідчить про те, що модель вичерпала свій потенціал покращення на основі наявних даних і поточних параметрів оптимізації.

Зазвичай це супроводжується досягненням плато на кривій навчання, яке також корелює з відсутністю покращення точності або інших ключових метрик як на навчальній, так і на валідаційній вибірках.

Як функцію втрати було використано функцію розрідженої категоріальної крос-ентропії:

$$L(y, \hat{y}) = -log(\hat{y}_y)$$

де y – індекс правильного класу (наприклад, 2 для третього класу), $\hat{y}_y$ – ймовірність, яку модель призначила цьому класу. [7]

Для оптимізації використовувався адаптивний оптимізатор Adam (Adaptive Moment Estimation), який є одним із методів стохастичного градієнтного спуску. [8] Adam поєднує в собі переваги двох інших популярних методів – AdaGrad та RMSProp – шляхом адаптації швидкості навчання для кожного параметра моделі на основі оцінок першого моменту (середнє значення градієнтів) і другого моменту (нормалізоване середнє квадратичне значення градієнтів). Такий підхід дозволяє стабільно та ефективно оновлювати ваги навіть у разі рідкісних або шумних градієнтів, що особливо важливо при роботі з великими нейронними мережами чи у процесі дистиляції знань, де моделі-учні мають вловити складну поведінку моделей-учителів.

Таким чином, використання Adam забезпечувало ефективну оптимізацію як у процесі первинного навчання великих моделей-учителів, так і під час дистиляції, де задача ускладняється необхідністю водночас імітувати результат учителя та враховувати справжні мітки. Усі моделі навчались з однаковими гіперпараметрами (окрім архітектури) до досягнення стійкого зниження функції втрат або до зупинки за умови відсутності покращення протягом двох епох.

Дистиляція реалізована через модифікацію функції обрахування втрати:

$$\text{Втрата}_{\text{загальна}} = \alpha \cdot \text{втрата студента} + (1 - \alpha) \cdot \text{втрата учителя}$$

де α – це ваговий коефіцієнт, який визначає баланс між двома складовими втрати – втрати студента та втрати вчителя. [9] При $\alpha = 0$ студент повністю намагається скопіювати вчителя. При $\alpha = 1$ студент вчиться на жорстких мітках. При значеннях $1 < \alpha < 0$ студент вчиться як на даних вчителях так і на жорстких мітках. Для проведення дослідження було обрано коефіцієнт $\alpha = 0.1$. Використання такого значення покликане для того, щоб студент не переймав помилок вчителя і частково тренувався на оригінальних навчальних даних. Імплементація даної функції на мові програмування Python приведено на рисунку 2.

```python
def compute_loss(self, x, y, y_pred, sample_weight=None, allow_empty=False):
    # 1. Отримати soft-розподіли всіх teacher'ів
    teacher_softs = []

    for teacher in self.teachers:
        teacher_logits = teacher(x, training=False)
        teacher_soft = ops.softmax(teacher_logits / self.temperature, axis=1)
        teacher_softs.append(teacher_soft)

    # 2. Усереднити soft targets — mean soft labels
    mean_teacher_soft = ops.stack(teacher_softs)
    mean_teacher_soft = ops.mean(mean_teacher_soft, axis=0)  # середнє по вчителях

    # 3. Обчислити soft student logits
    student_soft = ops.softmax(y_pred / self.temperature, axis=1)

    # 4. Втрата дистиляції
    distill_loss = self.distillation_loss_fn(mean_teacher_soft, student_soft) * (self.temperature ** 2)

    # 5. Втрата студента (класична крос-ентропія)
    student_loss = self.student_loss_fn(y, y_pred)

    # 6. Повертаємо комбіновану втрату
    return self.alpha * student_loss + (1 - self.alpha) * distill_loss
```

Рис 2. Реалізація функції втрати через дистиляцію.

Виклад результатів дослідження

З метою отримання статистично надійних результатів та зменшення впливу випадковостей, пов'язаних з ініціалізацією ваг, стохастичністю навчання та варіативністю у процесі оптимізації, для кожної експериментальної конфігурації моделі було виконано по 10 незалежних повних ітерацій навчання. Зокрема, кожна така ітерація включала повну послідовність етапів, а саме:

1. Навчання окремої моделі-вчителя – навчання нейронної мережі, яка слугує джерелом знань для подальшої дистиляції.

2. Формування ансамблю вчителів – навчання набору моделей нейромереж по 10 моделей, які слугують джерелом знань для подальшої дистиляції

3. Навчання моделей малих нейромережі за допомогою дистиляції знань, які використовували і окрему модель-вчитель, і ансамбль як джерело "м'яких" міток, що мають на меті передати узагальнені знання від складніших моделей до меншої.

4. Незалежне навчання малої нейронної мережі без дистиляції, тобто звичайне навчання за жорсткими мітками із використанням тих самих архітектурних і гіперпараметричних налаштувань, як і в процесі дистиляції, але без використання додаткових знань від вчителя або ансамблю.

Після завершення кожної повної ітерації було проведено оцінювання якості навчених моделей (як з дистиляцією, так і без неї) на тестовому наборі даних. За метрику було обрано частоти помилок, яка обчислюється як відсоток неправильно класифікованих зображень:

$$\text{Частота помилок} = \frac{\text{кількість неправильних передбачень}}{\text{загальна кількість прикладів}}$$

або ж як протилежна метрика до точності

$$\text{Частота помолок} = 1 - \text{Точність}$$

де точність це кількість правильних передбачень поділена на загальну кількість прикладів [10].

Для кожної комбінації моделей та стратегії навчання було отримано 10 окремих значень частоти помилок, після чого було розраховано середнє значення цієї помилки, що забезпечує репрезентативну оцінку якості моделі з урахуванням можливих варіацій між окремими запусками. Отримані середні результати для всіх експериментальних конфігурацій узагальнено у таблиці 1. Графічне представлення результатів зображено на рисунку 3.

Таблиця 1

Середня частота помилки в моделях навчених на жорстких мітках і в дистильованих моделях

Модель	Середня похибка з 10 вимірювань
Велика нейромережа	2.01%
Дистильована велика нейромережа	2.89%
Ансамбль з 10 малих нейромереж	1.81%
Дистильований ансамбль з 10 малих нейромереж	2.68%
Незалежно навчена мала нейромережа	2.26%

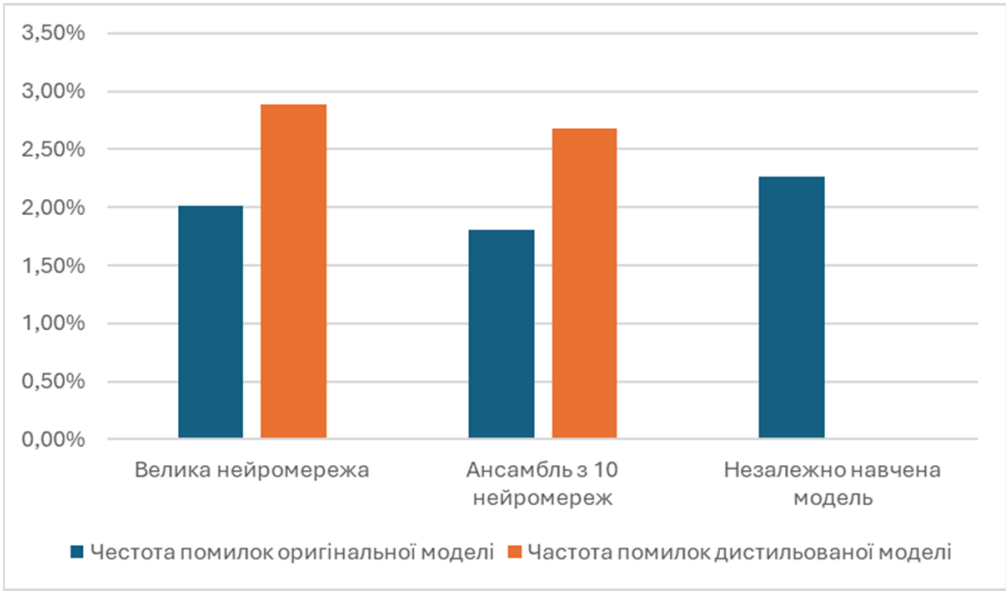

Рис. 3. Частота помилок оригінальних моделей і дистильованих моделей

Як видно з отриманих результатів вимірювань, проведених у рамках серії експериментів, застосування дистиляції знань у всіх протестованих випадках призвело до збільшення частоти помилок класифікації тестових зображень. Це свідчить про те, що в умовах дослідження – зокрема для задачі класифікації зображень з набору MNIST – дистиляція не лише не покращила, а навпаки, негативно вплинула на здатність моделі-учня узагальнювати дані, що може бути зумовлено недостатньою репрезентативністю або складністю обраних моделей-учителів, неадекватною передачею знань або іншими факторами, які потребують подальшого аналізу.

Таким чином, результуюча точність дистильованих моделей виявилася нижчою, ніж у відповідних моделей, які були навчені безпосередньо. Також, точність дистильованих моделей в обох випадках виявилась нижчою ніж точності моделі що була навчена на жорстких мітках незалежно. А, отже, задля досягнення максимальної точності в задачах класифікації зображень в умовах обмежених обчислювальних ресурсів потрібно використовувати підхід з незалежним навчанням моделі на жорстких мітках.

Висновки

Отже, в даному дослідженні проведено аналіз можливостей використання дистиляції знань з ансамблю нейромереж а також з великої нейромережі в задачах коли необхідно досягти максимальної точності роботи нейромережі в умовах обмежених обчислювальних ресурсів. Проведено експериментальну перевірку роботи дистиляції знань з великої нейромережі в малу а також з ансамблю нейромереж.

Література

1. Machine Learning in Real-Time Internet of Things(IoT) Systems: A Survey / J. Bian та ін. *IEEE Internet of Things Journal*. 2022. Т. 9, вип. 8. URL: https://doi.org/10.1109/JIOT.2022.3161050.

2. Lightweight Deep Learning: An Overview / C.-H. Wang та ін. *IEEE Consumer Electronics Magazine*. 2024. Т. 13, вип. 4. С. 51 – 64. ISSN 21622248. URL: https://doi.org/10.1109/MCE.2022.3181759.

3. Compacting Deep Neural Networks for Internet of Things: Methods and Applications / Z. Ke та ін. *IEEE Internet of Things Journal*. 2021. Т. 8. С. 11935–11959. ISSN 23274662. URL: https://10.1109/JIOT.2021.3063497.

4. Knowledge Distillation: A Survey / J. Gou та ін. *International Journal of Computer Vision*. 2021. Т. 129, вип. 6. С. 1789 – 1819. URL: https://doi.org/10.1007/s11263-021-01453-z.

5. Wang P., Fan E., Wang P. Comparative analysis of image classification algorithms based on traditional machine learning and deep learning. *Pattern Recognition Letters*. 2021. Т. 141. С. 61 – 67. ISSN 01678655. URL: https://doi.org/10.1016/j.patrec.2020.07.042.

6. Understanding of Convolutional Neural Network (CNN): A Review / Purwono та ін. *International Journal of Robotics and Control Systems*. 2022. Т. 2, вип. 4. С. 739 – 748. ISSN 27752658. URL: https://doi.org/10.31763/ijrcs.v2i4.888.

7. Hui L., Belkin M. EVALUATION OF NEURAL ARCHITECTURES TRAINED WITH SQUARE LOSS VS CROSS-ENTROPY IN CLASSIFICATION TASKS. *ICLR 2021 - 9th International Conference on Learning Representations*, 3–7 трав. 2021. International Conference on Learning Representations, ICLR, 2021.

8. Tian Y., Zhang Y., Zhang H. Recent Advances in Stochastic Gradient Descent in Deep Learning. *Mathematics*. 2023. Т. 11, вип. 3. ISSN 22277390. URL: https://doi.org/10.3390/math11030682.

9. Yuan M., Lang B., Quan F. Student-friendly Knowledge Distillation. Knowledge-Based Systems. 2024. Т. 296. URL: https://doi.org/10.1016/j.knosys.2024.111915.

10. Vujovic Z. Classification Model Evaluation Metrics. International Journal of Advanced Computer Science and Applications. 2021. Т. 12. С. 599–606. URL: https://doi.org/10.14569/IJACSA.2021.0120670.

References

1. Machine Learning in Real-Time Internet of Things(IoT) Systems: A Survey / J. Bian та ін. *IEEE Internet of Things Journal*. 2022. Т. 9, вип. 8. URL: https://doi.org/10.1109/JIOT.2022.3161050.

2. Lightweight Deep Learning: An Overview / C.-H. Wang та ін. *IEEE Consumer Electronics Magazine*. 2024. Т. 13, вип. 4. С. 51 – 64. ISSN 21622248. URL: https://doi.org/10.1109/MCE.2022.3181759.

3. Compacting Deep Neural Networks for Internet of Things: Methods and Applications / Z. Ke та ін. *IEEE Internet of Things Journal*. 2021. Т. 8. С. 11935–11959. ISSN 23274662. URL: https://10.1109/JIOT.2021.3063497.

4. Knowledge Distillation: A Survey / J. Gou та ін. *International Journal of Computer Vision*. 2021. Т. 129, вип. 6. С. 1789 – 1819. URL: https://doi.org/10.1007/s11263-021-01453-z.

5. Wang P., Fan E., Wang P. Comparative analysis of image classification algorithms based on traditional machine learning and deep learning. *Pattern Recognition Letters*. 2021. Т. 141. С. 61 – 67. ISSN 01678655. URL: https://doi.org/10.1016/j.patrec.2020.07.042.

6. Understanding of Convolutional Neural Network (CNN): A Review / Purwono та ін. *International Journal of Robotics and Control Systems*. 2022. Т. 2, вип. 4. С. 739 – 748. ISSN 27752658. URL: https://doi.org/10.31763/ijrcs.v2i4.888.

7. Hui L., Belkin M. EVALUATION OF NEURAL ARCHITECTURES TRAINED WITH SQUARE LOSS VS CROSS-ENTROPY IN CLASSIFICATION TASKS. *ICLR 2021 - 9th International Conference on Learning Representations*, 3–7 трав. 2021. International Conference on Learning Representations, ICLR, 2021.

8. Tian Y., Zhang Y., Zhang H. Recent Advances in Stochastic Gradient Descent in Deep Learning. *Mathematics*. 2023. Т. 11, вип. 3. ISSN 22277390. URL: https://doi.org/10.3390/math11030682.

9. Yuan M., Lang B., Quan F. Student-friendly Knowledge Distillation. Knowledge-Based Systems. 2024. Т. 296. URL: https://doi.org/10.1016/j.knosys.2024.111915.

10. Vujovic Z. Classification Model Evaluation Metrics. International Journal of Advanced Computer Science and Applications. 2021. Т. 12. С. 599–606. URL: https://doi.org/10.14569/IJACSA.2021.0120670.