UDC 004.8:004.032.26 DOI: 10.31673/2412-4338.2024.019902

Mukhin Vadym
National technical university of Ukraine "Igor Sikorsky Kyiv polytechnic institute"
ORCID 0000-0002-1206-9131

Kulyk Viacheslav
National technical university of Ukraine "Igor Sikorsky Kyiv polytechnic institute"
ORCID: 0000-0003-3833-1529

MODERN MODELS AND METHODS OF RESOURCE MANAGEMENT OF DISTRIBUTED COMPUTER SYSTEMS

Abstract. *The allocation of resources in heterogeneous distributed computer systems is a challenging task, constrained by factors such as task diversity and the decision process for optimal node selection. Traditional scheduling methods face limitations in addressing these complexities. This research proposes an AI-based optimization approach that leverages neural networks and deep learning techniques to efficiently allocate tasks across diverse nodes.*

The core component is a neural network responsible for assigning tasks to nodes based on attributes like computational efficiency, security, fault tolerance, and data transfer latency. Node attributes representing current state are continuously monitored and used to train the neural network, allowing it to learn node capabilities. When a new task arrives, the trained network matches it to the most suitable node by comparing task requirements to learned node attributes.

Extensive experiments compared the performance of feedforward neural networks (FFNN) and convolutional neural networks (CNN) across five datasets of varying sizes (100-2000 rows representing potential nodes). The FFNN demonstrated superior overall accuracy and consistency, achieving 90-98.6% validation accuracy, while the CNN exhibited fluctuating performance.

The proposed AI-based scheduling approach provides an adaptive framework for optimally assigning heterogeneous tasks in distributed environments. Key advantages include adaptability to changing system conditions through continuous training, flexible task-node mapping based on learned capabilities, scalability leveraging deep learning, and optimized resource utilization by fitting tasks to suitable nodes.

However, the experiments revealed no clear superior neural network architecture across all dataset scales. Further research aims to develop a hybrid or adaptive architecture that can dynamically adjust structure and parameters based on input data characteristics, combining strengths of feedforward and convolutional networks for efficient resource allocation tailored to specific datasets.

Keywords: *distributed computer systems, resource allocation, neural networks, deep learning, task scheduling, heterogeneous systems, adaptive optimization.*

Мухін Вадим Євгенович
Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського»
ORCID 0000-0002-1206-9131

Кулик В'ячеслав Олександрович
Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського»
ORCID: 0000-0003-3833-1529

СУЧАСНІ МОДЕЛІ ТА МЕТОДИ УПРАВЛІННЯ РЕСУРСАМИ РОЗПОДІЛЕНИХ КОМП'ЮТЕРНИХ СИСТЕМ

Анотація. *Розподіл ресурсів у гетерогенних розподілених комп'ютерних системах є складним завданням, обмеженим такими факторами, як різноманітність завдань і процес прийняття рішень для вибору оптимального вузла. Традиційні методи планування мають обмеження у вирішенні цих складнощів. Це дослідження пропонує підхід до оптимізації на основі ШІ, який використовує нейронні мережі та методи глибокого навчання для ефективного розподілу завдань між різними вузлами.*

Основним компонентом є нейронна мережа, яка відповідає за призначення завдань вузлам на основі таких атрибутів, як обчислювальна ефективність, безпека, відмовостійкість і затримка передавання даних. Атрибути вузла, що представляють поточний стан, постійно відстежуються та використовуються для навчання нейронної мережі, що дозволяє їй вивчати можливості вузла. Коли надходить нове завдання, навчена мережа зіставляє його з найбільш підходящим вузлом, порівнюючи вимоги до завдання з атрибутами вивченого вузла.

Масштабні експерименти порівнювали продуктивність нейронних мереж прямого зв'язку (FFNN) і згорткових нейронних мереж (CNN) у п'яти наборах даних різного розміру (100–2000 рядків, що представляють потенційні вузли). FFNN продемонстрував високу загальну точність і послідовність, досягнувши 90-98,6% точності перевірки, тоді як CNN показав коливання продуктивності.

Запропонований підхід до планування на основі штучного інтелекту забезпечує адаптивну структуру для оптимального призначення різнорідних завдань у розподілених середовищах. Основні переваги включають адаптивність до мінливих умов системи завдяки безперервному навчанню, гнучке відображення вузлів завдань на основі вивчених можливостей, масштабованість із застосуванням глибокого навчання та оптимізоване використання ресурсів шляхом підгонки завдань до відповідних вузлів.

Однак експерименти не виявили чіткої кращої архітектури нейронної мережі в усіх масштабах набору даних. Подальші дослідження спрямовані на розробку гібридної або адаптивної архітектури, яка може динамічно регулювати структуру та параметри на основі характеристик вхідних даних, поєднуючи переваги прямої та згорткової мереж для ефективного розподілу ресурсів, пристосованих до конкретних наборів даних.

Ключові слова*: розподілені комп'ютерні системи, розподіл ресурсів, нейронні мережі, глибоке навчання, планування завдань, гетерогенні системи, адаптивна оптимізація.*

1. Introduction.

The allocation of resources in distributed computational networks using traditional scheduling methods is significantly constrained by factors such as the heterogeneity of tasks to be distributed and the decision-making process when selecting the most optimal node for a specific task[1-3]. Considering that the tasks are heterogeneous and have dynamic parameter requirements for efficient execution, leveraging neural networks with techniques like reinforcement learning could be the appropriate solution[4, 5]. This approach aims to create software that can be employed in various computational systems, adapting to changing node parameters[1-3, 6].

This research aims to resolve the limitations posed by system heterogeneity in distributed networks, seeking efficient solutions for optimal assignment of tasks to diverse nodes within the system using neural networks and deep learning technology.

2. Resource allocation methods classification & comparison.

2.1. General overview of traditional resource allocation methods.

Below is a general overview of traditional Resource Allocation Methods classification. The schema describes 4 of the traditional categories of methods (Fig. 1) [7]:

- Centralized;
- Decentralized;
- Market-based methods;
- Optimisation-based methods.

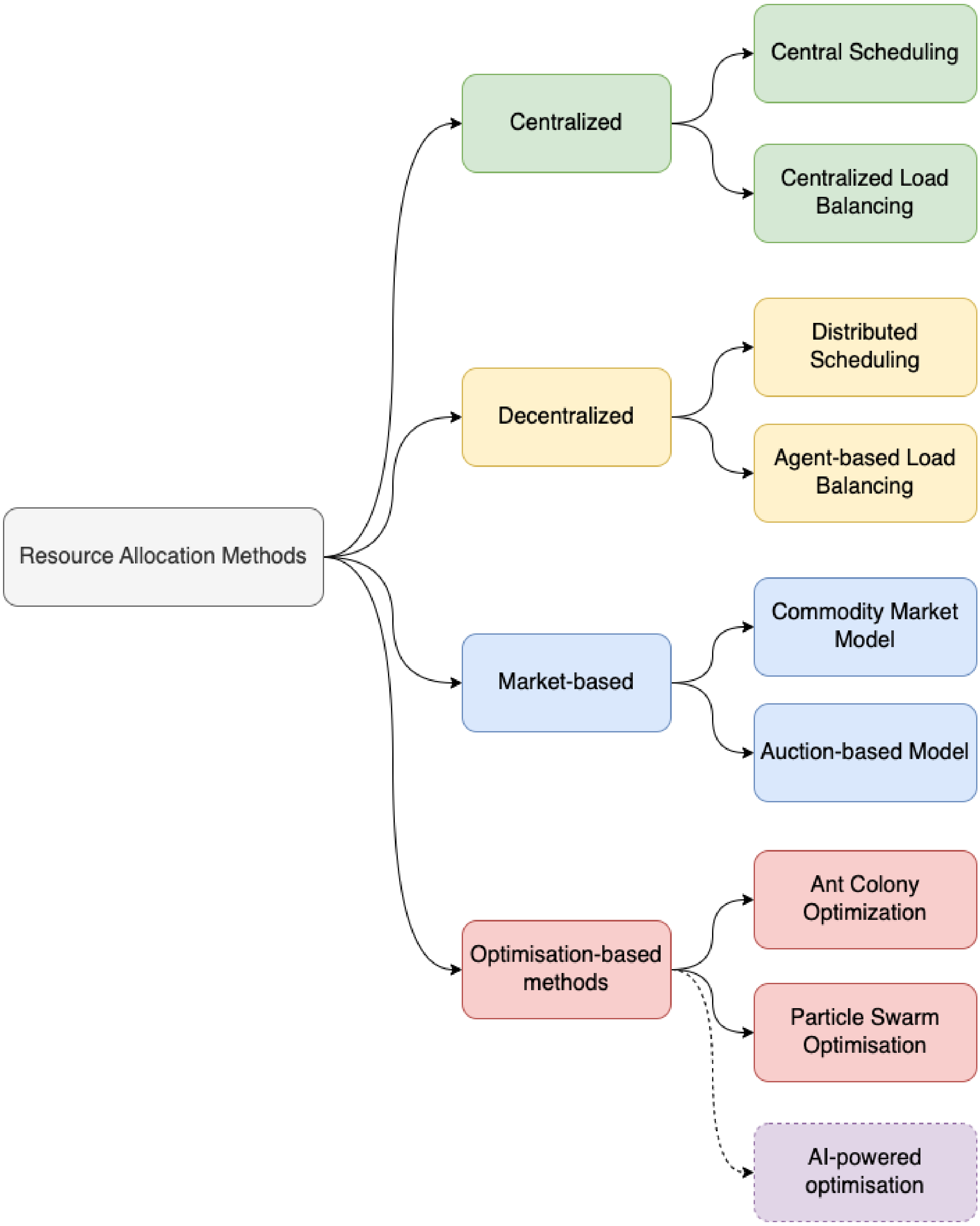

Fig. 1. General overview of resource allocation methods

Table №1 demonstrates the comparison between all the listed methods, including their pros and cons, as well as short form description.

Overview of resource allocation methods (schema):

Table 1

Resource allocation methods comparison:

Method	Description	Advantages	Disadvantages
Central Scheduling	Single centralized scheduler	Simplicity, control	Single point of failure, scalability
Centralized Load Balancing	Single central workload distributor	Implementation simplicity, control	Limits scalability, fault tolerance
Distributed Scheduling	Multiple distributed schedulers	Scalability, fault tolerance	Complexity, communication overhead
Agent-based Load Balancing	Distributed coordinating agents	Adaptability, flexibility	Complex coordination, global optimization
Commodity Market Model	Models system as market economy	Economic efficiency	Complex pricing models
Auction-based Model	Users bid for resources	Demand-based pricing	Susceptible to malicious bidding
ACO	Ant colony optimization agents	Adaptive allocation, scalability	Algorithm complexity
PSO	Cooperative search for optimal allocation	Ease of implementation	Convergence to local optima
AI-powered optimisation	AI-powered task distribution based on nodes attribute values	Adaptability to system changes, control, variability of nodes attributes	Single point of failure

2.2. Description of traditional task distribution algorithms.

Centralized methods involve a single coordinator for allocation. Central scheduling allows simplicity in task sequencing and full control over scheduling policies. However, having a single point of failure reduces fault tolerance[7]. Centralized load balancing simplifies workload distribution implementation but cannot scale efficiently due to the central coordinator bottleneck[8].

Decentralized distributed scheduling provides inherent scalability and redundancy against failures by using multiple schedulers[9]. But extensive communication and coordination between schedulers introduces significant complexity and overhead[10]. Agent-based load balancing is highly flexible and adaptive through autonomous, decentralized agents. However, optimizing global workload distribution through local agent interactions involves extremely complex system modeling and design[11, 12].

Market-based models offer efficient demand-driven resource pricing and allocation[13]. However, accurately modeling user valuations and bidding behaviors as well as mechanisms for price setting and winner selection is non-trivial[14]. Preventing malicious actors who artificially influence prices or auction outcomes is an additional challenge[15].

Swarm intelligence methods like ant colony optimization provide decentralized coordination giving robustness and scalability. But modeling agent interactions requires intricate probabilistic algorithms which are difficult to optimize[16]. Particle swarm optimization simplifies implementation through decentralized cooperative search. But particles may converge prematurely at locally optimal resource configurations and fail to find global optima[17,18].

3. Description of proposed ai-based solution.

The proposed AI-based optimization method employs a neural network for flexible task distribution across heterogeneous system architectures. The neural network is responsible for assigning tasks to nodes based on various attribute parameters of both the nodes and tasks.

Each node provides a set of attributes representing its current state, including metrics like computational efficiency, security level, fault tolerance, and data transfer latency. These node attribute values are collected through continuous monitoring and fed into a dataset used to train the neural network. This allows the neural network to learn the capabilities and current conditions of all nodes in the system

When a new task arrives, the trained neural network can then match the task to the most suitable node by comparing the task requirements to the node attributes it has learned (Fig. 2). Deep learning techniques enable the neural network to continuously update its knowledge through new attribute data from the monitoring system.

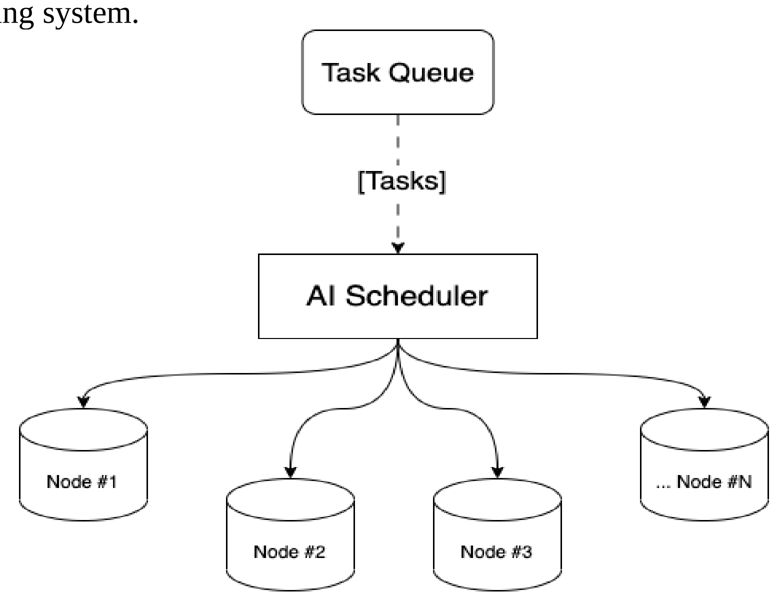

Fig. 2. Schema of AI-based scheduling approach

This approach provides an adaptive framework for optimally assigning heterogeneous tasks across diverse nodes in distributed environments.

Potential Advantages:

- Adaptability to heterogeneous and changing system conditions through continuous neural network training
- Flexible task-node mapping based on learned node capabilities and current states
- Scalable to large systems by leveraging deep learning techniques
- Optimized resource utilization by fitting tasks to best suited nodes

Potential Disadvantages:

- Complex neural network design and training process
- Requires high quality and consistent node monitoring data
- Computationally intensive training for large neural networks

- Challenging to define optimal node attribute parameters and data features

4. Evaluating neural networks architecture while conducting practical experiments.

In order to thoroughly evaluate neural networks for optimized node selection, we conducted extensive comparative experiments with feedforward neural networks (FFNN) and convolutional neural networks (CNN). Five datasets were created containing 100, 500, 1000, 1500, and 2000 rows representing potential nodes, each with attributes for performance, security, baud rate, reliability, and a binary result value. The FFNN and CNN architectures were trained on 80% of each dataset and validated on the remaining 20% to predict node suitability.

A comparison of architectures of developed feedforward and convolutional neural networks is presented in Figure 3 and Figure 4.

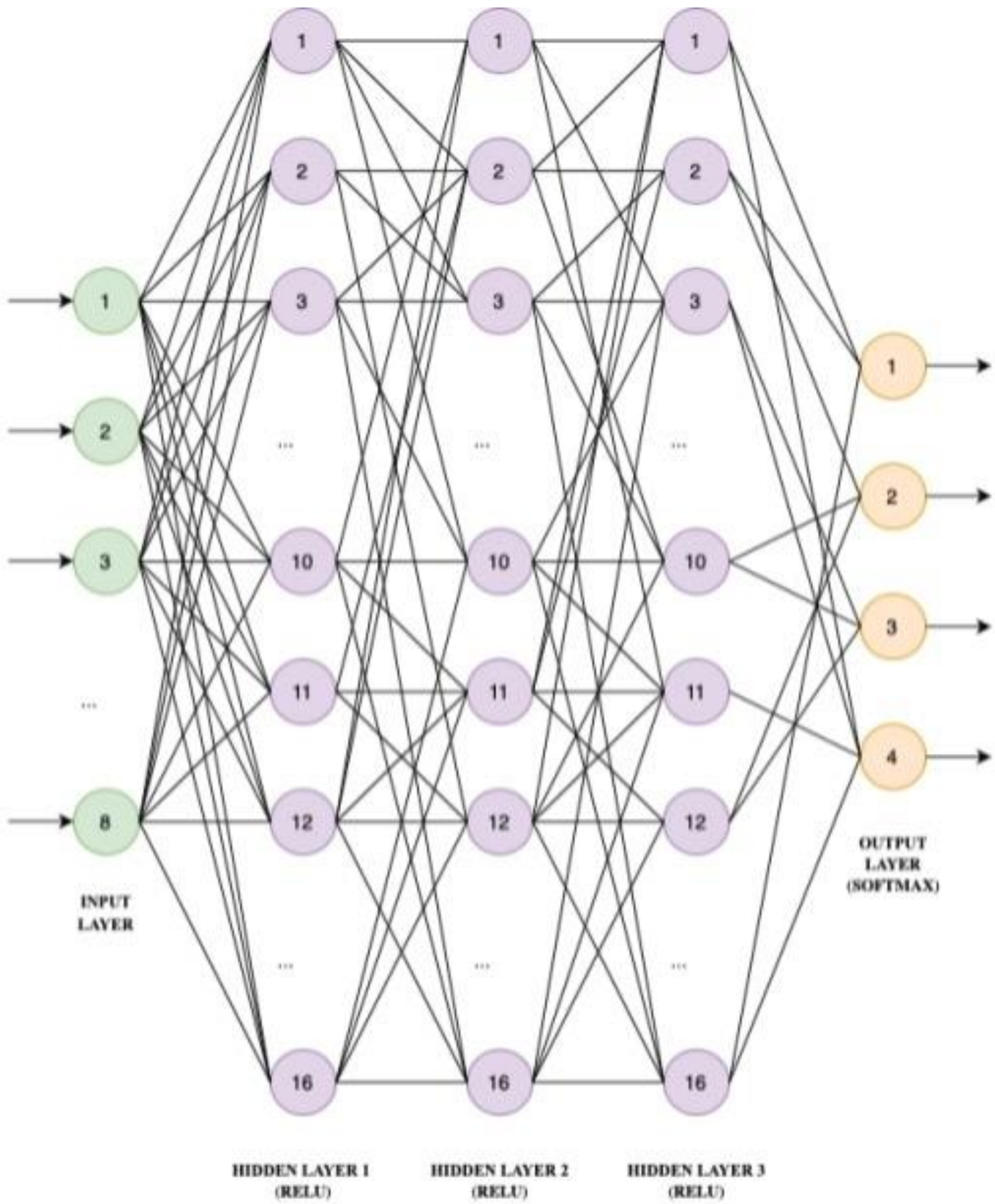

Fig. 3. Model of developed FFNN neural network

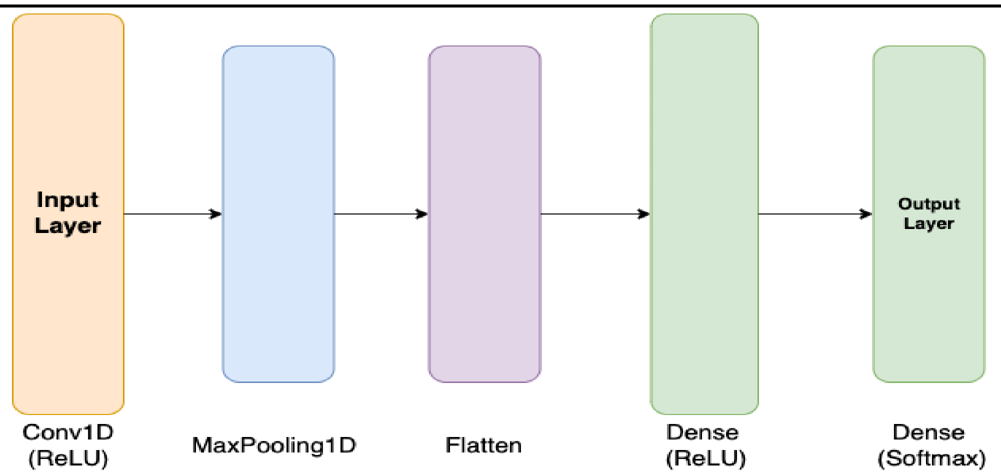

Fig. 4. Model of developed CCNN

Five datasets were created containing 100, 500, 1000, 1500, and 2000 rows representing potential nodes, each with attributes for performance, security, baud rate, reliability, and a binary result value. The FFNN and CNN architectures were trained on 80% of each dataset and validated on the remaining 20% to predict node suitability.

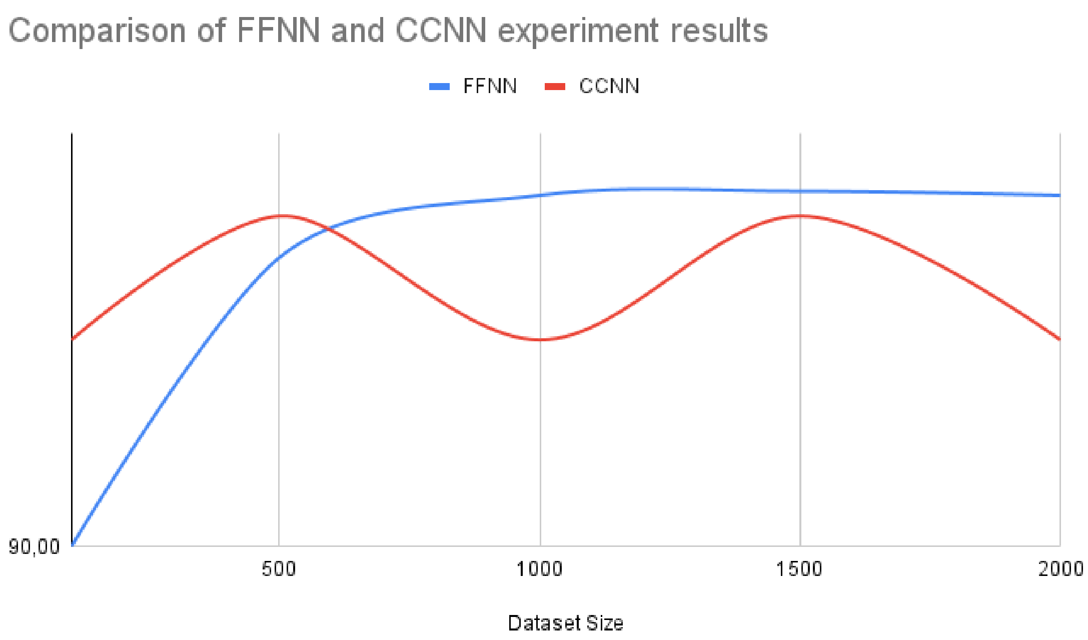

Fig. 5. Diagram of accuracy comparison for FFNN and CCNN architectures

The FFNN demonstrated superior overall accuracy and consistency across all dataset sizes (Fig. 5). Specifically, the FFNN achieved 90%, 97%, 98.5%, 98.6%, and 98.5% validation accuracy on the 100, 500, 1000, 1500, and 2000 row datasets respectively. Meanwhile, the CNN had varying performance of 95%, 98%, 95%, 98%, and 95% accuracy on the same datasets. The FFNN's fully connected layers appear better suited for learning the complex relationships between node attributes and suitability for this classification task.

Based on the experimental results, it was found that for our dataset, it is not possible to definitively determine a more effective neural network architecture. This is because both the feedforward neural network (FFNN) and the convolutional neural network (CNN) architectures exhibited the best and worst performance across different dataset sizes.

Consequently, in the general case, there is no clear-cut choice of a specific neural network architecture to address the given task. There is a need to develop a modified neural network architecture that can provide more effective results for arbitrary dataset dimensions.

The experiments revealed that the performance of the two architectures varied significantly across different dataset sizes. While the FFNN demonstrated superior overall accuracy and consistency, the CNN exhibited fluctuating performance, sometimes outperforming the FFNN and sometimes lagging. This inconsistency highlights the challenge in selecting a single architecture that can perform optimally across diverse dataset scales.

Therefore, further research efforts should focus on designing a hybrid or adaptive neural network architecture that can dynamically adjust its structure and parameters based on the characteristics of the input data. Such an approach could leverage the strengths of both feedforward and convolutional networks, allowing for efficient resource allocation tailored to the specific dataset at hand.

Resources

1. Yang, S.S., 2016. Centralized and decentralized methods of efficient resource allocation in cloud computing (Doctoral dissertation, Arizona State University)

2. Hu, Z., Mukhin, V., Kornaga, Y. *et al.* The Analytical Model for Distributed Computer System Parameters Control Based on Multi-factoring Estimations. *J Netw Syst Manage* 27, 351–365 (2019). https://doi.org/10.1007/s10922-018-9468-x

3. Zhengbing Hu, Vadym Mukhin, Yaroslav Kornaga, Yaroslav Lavrenko, Oksana Herasymenko,"Distributed Computer System Resources Control Mechanism Based on Network-Centric Approach", International Journal of Intelligent Systems and Applications(IJISA), Vol.9, No.7, pp.41-51, 2017. DOI:10.5815/ijisa.2017.07.05

4. Luong, N.C., Hoang, D.T., Gong, S., Niyato, D., Wang, P., Liang, Y.C. and Kim, D.I., 2019. Applications of deep reinforcement learning in communications and networking: A survey. IEEE Communications Surveys & Tutorials, 21(4), pp.3133-3174.

5. Wang, S., Tuor, T., Salonidis, T., Leung, K.K., Makaya, C., He, T. and Chan, K., 2019. Adaptive federated learning in resource constrained edge computing systems. IEEE Journal on Selected Areas in Communications, 37(6), pp.1205-1221.

6. Huang, K., Bi, S. and Zhang, Y.J., 2020. Deep reinforcement learning for online computation offloading in wireless powered mobile-edge computing networks. IEEE Transactions on Mobile Computing, 19(11), pp.2581-2593.

7. Ouyang, W., Xu, R., Wang, X., Zhang, L. and Chang, P., 2019. Adaptive user-managed service placement for mobile edge computing: An online learning approach. IEEE Transactions on Mobile Computing, 20(3), pp.753-766.

8. Liu, J., Mao, Y., Zhang, J. and Letaief, K.B., 2016. Delay-optimal computation task scheduling for mobile-edge computing systems. In IEEE International Symposium on Information Theory (ISIT) (pp. 2760-2764). IEEE.

9. Behmandpoor, P., Patrinos, P. and Moonen, M., 2022. Federated Learning Based Resource Allocation for Wireless Communication Networks. In 2022 30th European Signal Processing Conference (EUSIPCO). IEEE.

10. Zhou, Z., Chen, X., Li, E., Zeng, L., Luo, K. and Zhang, J., 2019. Edge intelligence: Paving the last mile of artificial intelligence with edge computing. Proceedings of the IEEE, 107(8), pp.1738-1762.

11. Wang, X., Han, X., Wang, C.X., Zhang, Q. and Chen, X., 2019. In-edge AI: Intelligentizing mobile edge computing, caching and communication by federated learning. IEEE Network, 33(5), pp.156-165.

12. Dinh, C.T., Tang, J., La, Q.D. and Quek, T.Q., 2017. Offloading in mobile edge computing: Task allocation and computational frequency scaling. IEEE Transactions on Communications, 67(8), pp.5389-5404.

13. Wang, X., Han, Y., Leung, V.C. and Niyato, D., 2020. Convergence of edge computing and deep learning: A comprehensive survey. IEEE Communications Surveys & Tutorials, 22(2), pp.869-904.

14. Mao, Y., You, C., Zhang, J., Huang, K. and Letaief, K.B., 2017. A survey on mobile edge computing: The communication perspective. IEEE Communications Surveys & Tutorials, 19(4), pp.2322-2358.

15. Yu, S., Wang, X. and Langar, R., 2017. Computation offloading for mobile edge computing: A deep learning approach. 2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC) (pp. 1-6). IEEE.

16. Zhang, X., Zhu, H., Han, Y. and Li, Z., 2020. Joint Resource Allocation and Task Offloading in Edge Computing via Deep Reinforcement Learning. In Proc. IEEE Global Communications Conference (GLOBECOM) (pp. 1-6). IEEE.

17. Sellami, B., Hakiri, A., Ben Yahia, S. and Berthou, P., 2020. Deep Reinforcement Learning for Energy-Efficient Task Scheduling in SDN-based IoT Network. In 2020 IEEE 19th International Symposium on Network Computing and Applications (NCA) (pp. 1-4). IEEE.

18. Cui, D. et al., 2020. Cloud Workflow Task and Virtualized Resource Collaborative Adaptive Scheduling Algorithm Based on Distributed Deep Learning. In 2020 IEEE International Conference on Advances in Electrical Engineering and Computer Applications (AEECA) (pp. 137-140). IEEE.