Electronics and Information Technologies, 2026, 33, 87–96
http://publications.lnu.edu.ua/collections/index.php/electronics/index

UDC: 004.852:796.06

LONG SHORT-TERM MEMORY RECURRENT NEURAL NETWORK FOR STATE PREDICTION AND RESOURCE ALLOCATION OPTIMIZATION IN DISTRIBUTED SYSTEMS

Zinovii Liubun, Oleh Tereshchuk*
Department of RadioPhysics and Computer Technologies,
Ivan Franko National University of Lviv
107 Gen. Tarnavsky Str., 79017 Lviv, Ukraine

Liubun, Z., & Tereshchuk, O. (2026). Long Short-Term Memory Recurrent Neural Network for State Prediction and Resource Allocation Optimization in Distributed Systems. *Electronics and Information Technologies*, 33, 87–96. https://doi.org/10.30970/eli.33.7

ABSTRACT

Introduction. This paper considers a method based on a Long Short-Term Memory (LSTM) neural network for optimal resource allocation in distributed systems. The developed algorithm ensures high accuracy in predicting resource states and optimal spatial distribution with minimal processing time. The relevance of this research is determined by the growing need for intelligent automation of resource management processes in service infrastructure facilities. A locker management system in sports facilities is used as a practical demonstration of the method's effectiveness.

Materials and Methods. To address the prediction and optimization tasks, an LSTM-based architecture with 32 hidden neurons and a sequence length of 10 time steps is proposed. The LSTM model processes sequential occupancy data to capture temporal dependencies and generate probability estimates for future resource states. A multi-factor scoring function is developed to transform predictions into optimal allocation decisions, considering spatial constraints and user preferences. The method is systematically compared with classical approaches: heuristic algorithms (Sequential, Round-Robin), statistical time series models (ARIMA, exponential smoothing), and machine learning methods (logistic regression, random forest, gradient boosting). All methods are evaluated on identical datasets using consistent metrics, including prediction accuracy, F1-score, spatial balance index, and zone variance.

Results. Using LSTM neural networks for the prediction task achieves 85% accuracy, which is statistically significantly higher than Random Forest (79%, p=0.0023) and ARIMA (68%, p=0.0001). The spatial balance index improved by 8.5% compared to the best classical method (0.89 versus 0.82). Inference time remains acceptable for real-time applications (18.9 ms per prediction).

Conclusions. The proposed LSTM-based method demonstrates satisfactory accuracy in predicting resource states and optimizing their allocation within minimal timeframes. The ability to model long-term temporal dependencies provides significant advantages over classical fixed-window methods. Therefore, the method can be effectively applied to enhance the functionality of distributed resource management systems.

Keywords: LSTM neural network, resource allocation, time series prediction, recurrent neural networks, optimization.

INTRODUCTION

Optimal object placement is an important task in many application domains — from warehouse management to parking space organization and storage systems [1-3]. Traditional approaches are based on simple heuristic rules or statistical models that do not adequately account for temporal change characteristics.

Long Short-Term Memory (LSTM) recurrent neural networks, proposed by Hochreiter and Schmidhuber in 1997 [4], demonstrate high efficiency in tasks involving sequential data. Unlike classical RNNs, the LSTM architecture solves the vanishing gradient problem through specialized gate mechanisms (forget gate, input gate, output gate), enabling effective modeling of long-term dependencies [5-7]. Development of LSTM-based optimal placement algorithms combined with modern computational capabilities will enable the implementation of efficient and reliable object management systems.

In recent years, attention-based architectures, particularly Transformers [9], originally developed for natural language processing, have been increasingly adapted for time series forecasting tasks [8]. The Temporal Fusion Transformer (TFT) [10] combines recurrent layers with multi-head attention mechanisms, enabling interpretable multi-horizon forecasting with explicit modeling of static covariates and known future inputs. The Informer architecture [11] introduces ProbSparse self-attention and a generative-style decoder to achieve $O(L\ log\ L)$ complexity for long-sequence time series forecasting, addressing the quadratic complexity limitation of standard Transformers. A hybrid approach combining exponential smoothing with recurrent neural networks [12] won the prestigious M4 forecasting competition, demonstrating that integration of classical statistical methods with deep learning can yield superior results. The DeepAR framework [13] employs autoregressive recurrent networks for probabilistic forecasting, producing calibrated prediction intervals that are valuable for resource planning under uncertainty. A comprehensive experimental review of deep learning architectures for time series forecasting [14] confirms that recurrent models remain competitive across diverse benchmarks despite the emergence of newer architectures. Furthermore, Hewamalage et al. [15] provide a thorough analysis of the current status of recurrent neural networks for time series forecasting, demonstrating that well-configured LSTM and GRU models can match or exceed the performance of more complex architectures, reaffirming the relevance of investigating recurrent approaches for specific application domains where sequential dependency modeling is paramount.

This work investigates the application of LSTM for optimizing locker placement in sports facilities as a model example of a task involving discrete objects, temporal dynamics, and spatial distribution requirements.

MATERIALS AND METHODS

It is necessary to develop an algorithm for solving the optimal object placement problem:

1. Predicting future occupancy states of objects based on historical usage patterns.
2. Selecting the optimal object for placement considering the forecast and spatial load distribution.

Problem statement:

Let there be a set of lockers $L = \{l_1, l_2, \dots, l_n\}$, each of which at time t can be in state:

$$s_i(t) \in \{0,1\}, \quad (1)$$

where 0 – free, 1 – occupied.

The task is to predict the state of each locker at the next time:

$$\hat{s}_i(t+1) = f[s_i(t), s_i(t-1), \ldots, s_i(t-k)], \quad (2)$$

where f is a function implemented by an LSTM neural network.

Based on the forecast, a recommendation is formed for selecting a locker with minimal occupancy probability while ensuring uniform spatial utilization of the changing room.

Fig. 1 presents the spatial structure of the object – a locker system in a sports facility.

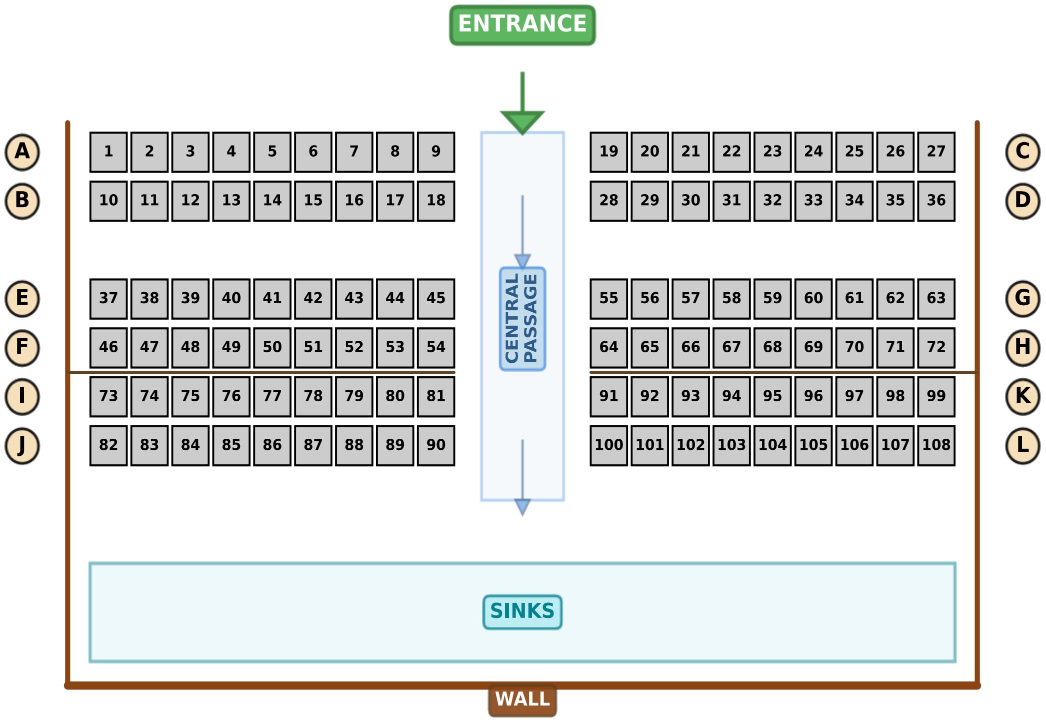

Fig. 1. Sports facility locker system.

Formally, the optimal choice is defined as:

$$l^* = \arg \min_{l_i \in L_{free}} (\alpha P_i + \beta D_i + \gamma U_i), \quad (3)$$

where: P_i – predicted probability of cell occupancy in the near future;
D_i – distance to “dense” zones (for uniformity);
U_i – convenience factor of location (upper/lower, proximity to entrance);
α, β, γ – weight coefficients of importance.

The weight coefficients were determined empirically through a grid search over the ranges $\alpha \in [0.3, 0.7]$, $\beta \in [0.1, 0.4]$, $\gamma \in [0.1, 0.3]$ with a step of 0.05, optimizing the spatial balance index on the validation set. The resulting values $\alpha = 0.5, \beta = 0.3, \gamma = 0.2$ reflect the priority of prediction accuracy in allocation decisions while maintaining meaningful contributions from spatial uniformity and user convenience factors.

Placement System Requirements: the selection of an optimal object must be performed quickly and ensure uniform spatial distribution. Therefore, simple heuristic methods cannot be effectively used due to their limitations:

- do not account for historical usage data
- do not adapt to changes in behavior patterns
- do not ensure optimal spatial balance.

Thus, a method is needed that combines prediction of temporal dependencies with placement optimization. Neither simple rules nor static models are fully applicable. Recurrent neural networks naturally model sequences and can identify complex patterns.

Figure 2 shows the internal architecture of an LSTM cell with gate mechanisms. The model for the placement task consists of:

1. Input layer: sequence of $k = 10$ binary values of object states;
2. LSTM layer: $h = 32$ hidden neurons with forget gate f_t, input gate i_t, output gate o_t;
3. Output Dense layer: sigmoid activation for predicting occupancy probability.

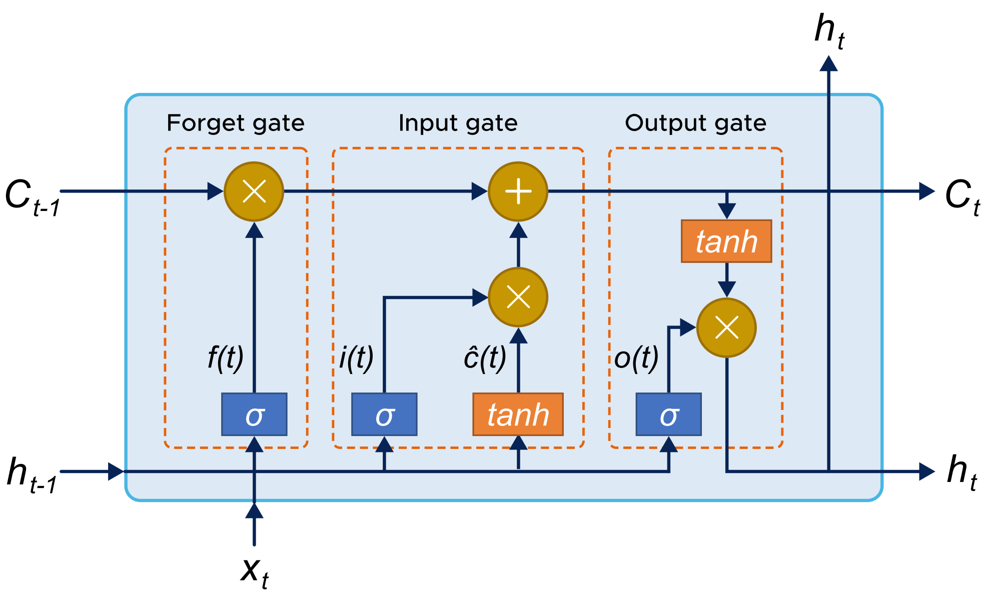

Fig. 2. Internal architecture of LSTM cell.

LSTM Cell Gate Mechanisms:

- $f_t = \sigma(Wf \cdot [h_t^{-1}, x_t] + bf)$ – determines what information to forget from cell state;
- $i_t = \sigma(Wi \cdot [h_t^{-1}, x_t] + bi)$ – determines what new information to store;
- $\tilde{C}_t = tanh(WC \cdot [h_t^{-1}, x_t] + bC)$ – candidate for new cell state;
- $C_t = f_t \odot C_t^{-1} + i_t \odot \tilde{C}_t$ – updated cell state (long-term memory);
- $o_t = \sigma(Wo \cdot [h_t^{-1}, x_t] + bo)$ – determines what to output;
- $h_t = o_t \odot tanh(C_t)$ – hidden state output.

Training Parameters: Adam optimizer (learning rate = 0.001, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\varepsilon = 10^{-7}$), binary cross-entropy loss function, 50 training epochs with early stopping (patience = 7 epochs, monitoring validation loss), batch size 32. The learning rate was selected from the set {0.0001, 0.0005, 0.001, 0.005} based on validation performance. Dropout regularization with a rate of 0.2 was applied after the LSTM layer to prevent overfitting. Model weights were initialized using Glorot uniform initialization. The total number of trainable parameters is 4,609 (LSTM layer: 4,352 parameters comprising four gate weight matrices of size [32+1, 32] each; Dense layer: 257 parameters). Training convergence was typically achieved within 35–40 epochs, with the best model selected based on minimal validation loss.

To demonstrate the effectiveness of the proposed LSTM approach, a comparative analysis was conducted with eight alternative methods from four classes: heuristic algorithms (Sequential, Round-Robin), statistical time series models (ARIMA, exponential smoothing), classical machine learning (logistic regression, random forest, gradient boosting), and recurrent neural networks (LSTM). Heuristic methods do not use forecasting and have complexity O(n). Statistical models require stationarity assumptions about the data. Machine learning methods with a fixed window $k = 10$ lose information about the sequential nature of data. LSTM naturally models long-term dependencies with complexity $O(n \cdot h^2 \cdot t)$.

Method quality was evaluated across three metric categories: prediction accuracy, placement quality (spatial balance index $\in [0,1]$, zone load variance), and computational efficiency (training time, inference time, memory consumption).

Experiments were conducted on simulated data modeling the operation of a locker management system in a sports facility: n=108 cells (distributed across 12 zones of 9 cells each), T = 500 time steps, where each step corresponds to 1 hour of real time (approximately 21 days or 3 weeks of operation in total). Average system load was maintained at 60% (65 occupied cells), visitor distribution: 60% adults, 40% children. Typical visit duration: 1.0-1.5 hours. The simulation incorporated realistic usage patterns, including peak hours (morning 8:00–10:00 and evening 17:00–20:00 with load up to 85%), off-peak periods (midday 12:00–15:00 with load approximately 35%), and weekend variations with more uniform distribution. Each simulated visitor was assigned a random arrival time following a bimodal distribution reflecting morning and evening peaks, a visit duration sampled from a log-normal distribution (μ = 1.2 hours, σ = 0.3), and a zone preference weighted by proximity to the entrance. Data was split into training and test samples at a ratio of 70%/30%. Data preprocessing involved constructing sliding windows of length k = 10 from binary occupancy vectors, yielding approximately 31,320 training sequences (290 windows × 108 cells).

All computations were performed on a single hardware configuration: Apple M1 processor, 16 GB RAM, Python 3.12.2 programming environment with TensorFlow 2.x library (CPU mode). The reported time characteristics (training and inference) are specific to this platform and may differ on systems with different processor architectures or memory capacities.

RESULTS AND DISCUSSION

Table 1 presents the results of comparing eight optimal placement methods.

Figure 3 demonstrates a comprehensive comparison of methods across six performance metrics. The proposed LSTM-based method achieves the highest prediction accuracy (85%) with a statistically significant advantage over the best classical method Random Forest (+6.0%, $p = 0.0023$ at significance level $\alpha = 0.05$), while compared to the best statistical method ARIMA, the improvement is 25% ($p = 0.0001$). In addition to high prediction accuracy, the LSTM method provides the most uniform spatial distribution with minimal zone load variance (67.2 versus 98.4 for Random Forest), confirming that accurate predictions improve the quality of placement decisions. The spatial balance index improved by 8.5%, reaching 0.89 versus 0.82 for classical methods.

Table 1. **Comparative characteristics of optimal object placement methods**

Method	Class	Prediction Accuracy	Balance Index	Complexity	Time (ms)
Sequential	Heuristic	0.42	0.42	$O(n)$	0.1
Round-Robin	Heuristic	0.68	0.68	$O(n)$	0.2
ARIMA(2,1,2)	Statistical	0.68	0.71	$O(n \cdot p \cdot q)$	15.4
Exponential Smoothing	Statistical	0.64	0.66	$O(n)$	8.2
Logistic Regression	Classical ML	0.74	0.76	$O(n \cdot k \cdot d)$	8.7
Random Forest	Classical ML	0.79	0.82	$O(n \cdot k \cdot d \cdot log\ d \cdot T)$	12.3
Gradient Boosting	Classical ML	0.77	0.79	$O(n \cdot k \cdot d \cdot T)$	14.1
LSTM	RNN	0.85	0.89	$O(n \cdot h^2 \cdot t)$	18.9

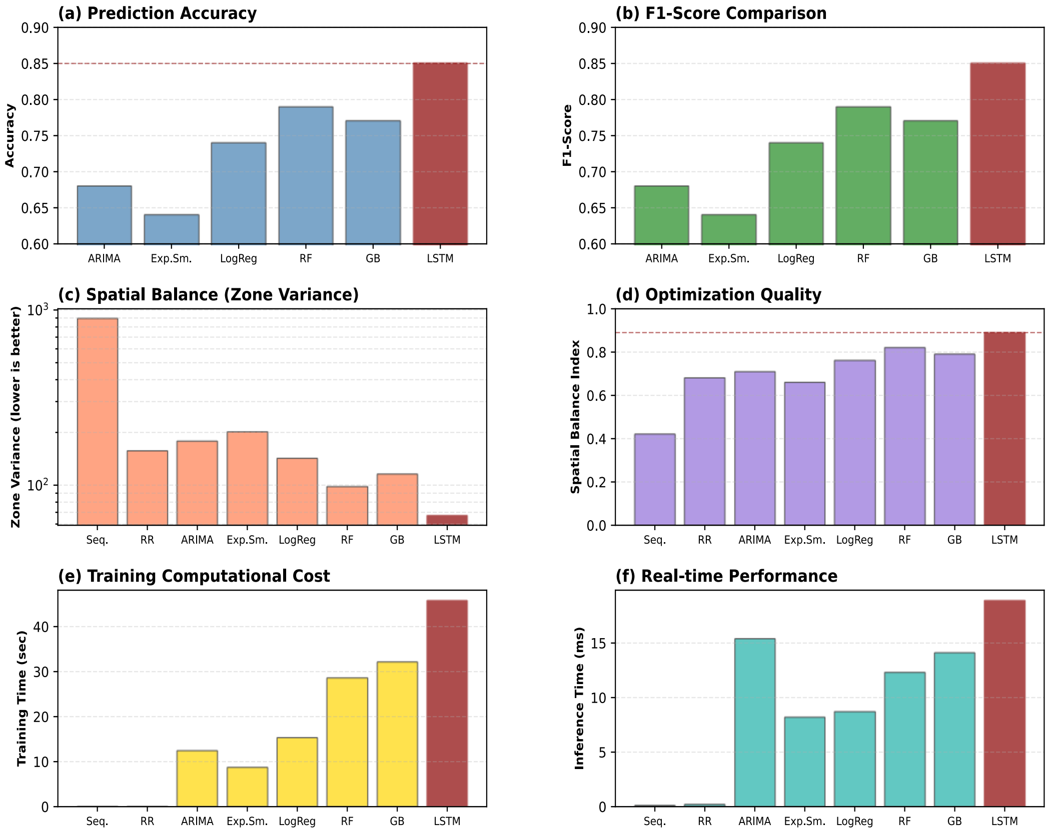

Fig. 3. Comparative analysis of methods.

The statistical significance of the improvements obtained is confirmed by 95% confidence intervals: compared to Random Forest, the accuracy improvement is [+0.03, +0.09], and compared to ARIMA – [+0.13, +0.21]. Paired t-test showed p-values less than 0.05, indicating the reliability of the results. LSTM's computational costs are moderately higher: training time is 45.8 sec versus 28.6 for Random Forest, memory usage is 52.3 MB versus 45.2 MB. However, the placement decision time remains acceptable for real-time systems (18.9 ms), and the additional training costs are compensated by substantial improvement in placement quality during operation.

The key advantages of the LSTM approach lie in its ability to model long-term temporal dependencies through gate mechanisms, which allow effective capture of cyclical object usage patterns that are difficult to model with fixed observation window methods. Unlike ARIMA, which requires explicit differencing and stationarity assumptions, LSTM naturally adapts to pattern changes without additional preprocessing. The combination of gates and nonlinear activation functions enables the detection of complex nonlinear dependencies between object states that are inaccessible to linear methods. Among the method's limitations, elevated computational requirements should be noted, the need for sufficient volume of representative data for effective training, and limited model interpretability compared to simple heuristic rules or decision trees. It should also be noted that recent Transformer-based architectures [10, 11] and alternative deep learning approaches [12, 13] may offer advantages for longer forecasting horizons or multivariate settings, though their benefits over well-tuned recurrent models are not universal across all problem domains [14, 15].

CONCLUSION

Based on the obtained results, the following conclusions can be drawn:

The proposed method for optimal object placement based on LSTM recurrent neural networks provides high prediction accuracy and spatial distribution quality with acceptable processing time and minimal computational resources. Achieving 85% prediction accuracy (versus 79% for the best classical method, Random Forest, with $p=0.0023$) and 8.5% improvement in spatial balance index (0.89 versus 0.82) confirms the effectiveness of applying recurrent architecture to the optimal placement problem.

The ability of LSTM to model long-term temporal dependencies through forget gate, input gate, and output gate mechanisms provides significant advantages over classical fixed-window methods that lose information about the sequential nature of data. Statistical significance of improvements ($p<0.05$) and robustness to non-stationarity confirm the reliability of the method.

The placement decision time (18.9 ms) remains practical for real-time systems despite increased training requirements. Therefore, the method can be used cost-effectively to improve the quality of object placement management systems in various application domains where prediction accuracy and spatial optimization are critical.

Directions for further research include comparison with modern attention-based architectures (Transformer, Temporal Fusion Transformer [10], Informer [11), investigation of GRU as a lighter recurrent alternative, application of reinforcement learning methods for direct optimization of placement strategy, and validation on real data from management systems of various object types.

ACKNOWLEDGMENTS AND FUNDING SOURCES

The authors received no financial support for the research, writing, and/or publication of this article.

COMPLIANCE WITH ETHICAL STANDARDS

The authors declare that the research was conducted in the absence of any potential conflict of interest.

AUTHOR CONTRIBUTIONS

Conceptualization, [Z. L.]; methodology, [Z. L.]; validation, [O. T.]; formal analysis, [O. T.].; investigation, [O. T.]; resources, [O. T.]; data curation, [O. T.]; writing – original draft preparation, [O. T.]; writing – review and editing, [Z. L.]; visualization, [O. T.] supervision, [Z. L.]; project administration, [Z. L.]; funding acquisition, [Z. L.].

All authors have read and agreed to the published version of the manuscript.

REFERENCES

[1] Erl, T., Puttini, R., & Mahmood, Z. (2013). *Cloud computing: Concepts, technology & architecture*. Pearson Education.

[2] Oussous, A., Benjelloun, F. Z., Ait Lahcen, A., & Belfkih, S. (2018). Big Data technologies: A survey. *Journal of King Saud University - Computer and Information Sciences*, *30*(4), 431-448. https://doi.org/10.1016/j.jksuci.2017.06.001

[3] Pinedo, M. (2022). *Scheduling: Theory, algorithms and systems* (6th ed.). Springer. https://doi.org/10.1007/978-3-031-05921-6

[4] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation, 9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

[5] Tsemko, A., & Matskiv, M. (2025). Problems of using neural networks to predict the price of virtual assets. *Electronics and Information Technologies, 29*, 69–78. https://doi.org/10.30970/eli.29.7

[6] Sarker, I. H. (2021). Deep learning: A comprehensive overview on techniques, taxonomy, applications and research directions. *SN Computer Science, 2*(6), 420. https://doi.org/10.1007/s42979-021-00815-1

[7] Zhang, A., Lipton, Z. C., Li, M., & Smola, A. J. (2023). *Dive into deep learning*. Cambridge University Press.

[8] Torres, J. F., Hadjout, D., Sebaa, A., Martínez-Álvarez, F., & Troncoso, A. (2021). Deep learning for time series forecasting: A survey. *Big Data, 9*(1), 3–21. https://doi.org/10.1089/big.2020.0159

[9] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems, 30*, 5998–6008.

[10] Lim, B., Arık, S. Ö., Loeff, N., & Pfister, T. (2021). Temporal fusion transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting, 37*(4), 1748–1764. https://doi.org/10.1016/j.ijforecast.2021.03.012

[11] Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., & Zhang, W. (2021). Informer: Beyond efficient transformer for long sequence time-series forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence, 35*(12), 11106–11115. https://doi.org/10.1609/aaai.v35i12.17325

[12] Smyl, S. (2020). A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting. *International Journal of Forecasting, 36*(1), 75–85. https://doi.org/10.1016/j.ijforecast.2019.03.017

[13] Salinas, D., Flunkert, V., Gasthaus, J., & Januschowski, T. (2020). DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting, 36*(3), 1181–1191. https://doi.org/10.1016/j.ijforecast.2019.07.001

[14] Lara-Benítez, P., Carranza-García, M., & Riquelme, J. C. (2021). An experimental review on deep learning architectures for time series forecasting. *International Journal of Neural Systems, 31*(5), 2130001. https://doi.org/10.1142/S0129065721300011

[15] Hewamalage, H., Bergmeir, C., & Bandara, K. (2021). Recurrent neural networks for time series forecasting: Current status and future directions. *International Journal of Forecasting, 37*(1), 388–427. https://doi.org/10.1016/j.ijforecast.2020.06.008

РЕКУРЕНТНА НЕЙРОННА МЕРЕЖА НА ОСНОВІ ДОВГОЇ КОРОТКОЧАСНОЇ ПАМ'ЯТІ ДЛЯ ПРОГНОЗУВАННЯ СТАНІВ ТА ОПТИМІЗАЦІЇ РОЗПОДІЛУ РЕСУРСІВ У РОЗПОДІЛЕНИХ СИСТЕМАХ

Зіновій Любунь, Олег Терещук

zinoviy.lyubun@lnu.edu.ua, Oleh.Tereshchuk@lnu.edu.ua
Кафедра радіофізики та комп'ютерних технологій
Львівський національний університет імені Івана Франка
вул. Тарнавського, 107, м.Львів, 79017, Україна

АНОТАЦІЯ

Вступ. У статті розглянуто метод на основі нейронної мережі з архітектурою довгої короткочасної пам'яті (ДКЧП) для оптимального розподілу ресурсів у розподілених системах. Розроблений алгоритм забезпечує високу точність прогнозування стану ресурсів та оптимальний просторовий розподіл з мінімальним часом обробки. Актуальність дослідження зумовлена зростаючою потребою в інтелектуальній автоматизації процесів управління ресурсами на об'єктах сервісної інфраструктури. Як практичну демонстрацію ефективності методу використано систему управління комірками у спортивних закладах.

Матеріали та методи. Для розв'язання задач прогнозування та оптимізації запропоновано архітектуру на основі ДКЧП з 32 прихованими нейронами та довжиною послідовності 10 часових кроків. Модель LSTM обробляє послідовні дані про зайнятість для виявлення часових залежностей та формування імовірнісних оцінок майбутніх станів ресурсів. Розроблено багатофакторну функцію оцінювання для перетворення прогнозів на оптимальні рішення щодо розподілу з урахуванням просторових обмежень та уподобань користувачів. Метод порівнюється з класичними підходами, такими як евристичні алгоритми (послідовний розподіл, кругова черга), статистичні моделі часових рядів (авторегресійного інтегрованого ковзного середнього, експоненціальне згладжування) та методи машинного навчання (логістична регресія, випадковий ліс, градієнтне підсилювання).

Результати. Використання нейронних мереж з архітектурою ДКЧП для розв'язання задачі прогнозування дозволяє досягти точності 85%, що є статистично значущо вищим за випадковий ліс (79%, p = 0.0023) та ARIMA (68%, p = 0,0001). Індекс просторового балансу покращився на 8,5% порівняно з найкращим класичним методом (0,89 проти 0,82). Час висновку залишається прийнятним для застосувань реального часу (18,9 мс на прогноз).

Висновки. Запропонований метод на основі ДКЧП демонструє задовільну точність у прогнозуванні станів ресурсів та оптимізації їх розподілу в мінімальні терміни. Здатність моделювати довгострокові темпоральні залежності надає значні переваги над класичними методами з фіксованим вікном. Тому метод може бути ефективно застосований для підвищення функціональності систем управління розподіленими ресурсами.

Ключові слова: нейронна мережа ДКЧП, розподіл ресурсів, прогнозування часових рядів, рекурентні нейронні мережі, оптимізація.

Received / Одержано	Revised / Доопрацьовано	Accepted / Прийнято	Published / Опубліковано
21 February, 2026	13 March, 2026	16 March, 2026	30 March, 2026