

УДК (UDC) 004.94

Тележенко Денис *аспірант*Олександрович *Харківський національний університет імені В.Н. Каразіна,
майдан Свободи, 4, Харків-22, Україна, 61022*e-mail: denisque75@gmail.com<https://orcid.org/0000000283778517>

Прогнозування та аналітика у віртуальних розподілених системах: Використання моделей машинного навчання та аналітичних інструментів

Дана наукова стаття присвячена розробці концептуальної моделі синтезу архітектури віртуальних розподілених систем (ВРС). У статті розглядаються ключові аспекти віртуальних розподілених систем, включаючи апаратне забезпечення, гіпервізори, віртуальні машини та модулі управління. Висвітлюються методологічні принципи синтезу архітектури, починаючи від аналізу системних вимог, проектування архітектури, реалізації та тестування, закінчуючи оцінкою та оптимізацією продуктивності ВРС. У статті підкреслюється важливість кожного етапу в цьому процесі, наголошується на необхідності глибокого розуміння системних вимог і вибору відповідних технологій. Особливу увагу в статті приділено ролі гіпервізорів і віртуальних машин у ВРС, їх підключенню до апаратного забезпечення та можливостям управління ресурсами. Ця стаття буде корисною для дослідників і практиків у сфері віртуалізації та обчислювальних систем, які розробляють або оптимізують віртуальні розподілені системи.

Актуальність. Стаття присвячена важливими питаннями, пов'язаними з прогнозуванням та оптимізацією віртуальних розподілених систем, що є ключовим елементом сучасних технологічних інфраструктур. З розвитком обчислювальних технологій та штучного інтелекту, зростає потреба у ефективному управлінні ресурсами та підтримці високої пропускної спроможності в обчислювальних системах.

Мета. Метою цієї наукової роботи є дослідження та аналіз застосування алгоритмів машинного навчання, зокрема LSTM (Long Short-Term Memory) та механізму уваги, для прогнозування та оптимізації віртуальних розподілених систем. Стаття прагне детально розглянути, як ці технології можуть покращити управління ресурсами, забезпечити більш високу ефективність та пропускну спроможність систем, а також проаналізувати, як вони допомагають виявляти потенційні проблеми та оптимізувати розподіл ресурсів, спираючись на точні прогнози та аналіз історичних даних.

Методи дослідження. У даній роботі використовуються різноманітні методи дослідження, які включають теоретичний аналіз та практичне застосування алгоритмів машинного навчання, особливо архітектури LSTM (Long Short-Term Memory) і механізму уваги, для оцінки їх ефективності у прогнозуванні поведінки віртуальних розподілених систем. Методологія включає збір та аналіз великих обсягів даних з віртуальних систем, проведення експериментальних тестувань для оцінки пропускної спроможності та ефективності ресурсів, а також застосування статистичних інструментів для оцінки точності прогнозів.

Результати. Результати дослідження показують, що застосування алгоритмів LSTM (Long Short-Term Memory) та механізму уваги значно покращує здатність віртуальних розподілених систем до ефективного прогнозування та управління ресурсами. Експерименти та аналіз даних виявили, що ці техніки дозволяють точніше аналізувати та передбачати шаблони використання ресурсів, що веде до оптимізації розподілу навантаження та підвищення загальної пропускної спроможності систем. Також було зазначено, що інтеграція механізму уваги з LSTM дозволяє більш детально розуміти довготривалі залежності в даних, що сприяє більш точному прогнозуванню та реагуванню на зміни у використанні ресурсів.

Висновки. Використання алгоритмів LSTM та механізму уваги в значній мірі покращує прогнозування та управління ресурсами віртуальних розподілених систем, відкриваючи нові можливості для їх оптимізації та підвищення ефективності.

Ключові слова: віртуальні розподілені системи (ВРС), синтез архітектури, LSTM, прогнозування навантаження, Input Gate, Forget Gate, Output Gate, механізм уваги.

Як цитувати: Тележенко Д. О. Прогнозування та аналітика у віртуальних розподілених системах: Використання моделей машинного навчання та аналітичних інструментів. *Вісник Харківського національного університету імені В.Н. Каразіна, серія Математичне моделювання. Інформаційні технології. Автоматизовані системи управління*. 2023. вип. 60. С.46-51. <https://doi.org/10.26565/2304-6201-2023-60-05>

How to quote: Telezhenko D., "Forecasting and analytics in virtual distributed systems: Using machine learning models and analytical tools", *Bulletin of V.N. Karazin Kharkiv National University, series Mathematical modelling. Information technology. Automated control systems*, vol. 60, pp.46-51, 2023. <https://doi.org/10.26565/2304-6201-2023-60-05> [In Ukrainian]

1. Вступ

У сучасному світі, де обсяги даних та їх складність зростають з кожним днем, віртуальні розподілені системи стають невід'ємною частиною багатьох технологічних процесів. Ці системи відіграють ключову роль у різноманітних галузях, від обчислювального хмарного сервісу до штучного інтелекту та великих даних. Однак, разом з розвитком технологій, зростає і потреба у ефективних інструментах для прогнозування та аналізу поведінки цих систем. Надійність, ефективність та безпека віртуальних розподілених систем залежить від здатності точно прогнозувати їх поведінку у відповідь на змінні умови та запити. [1]

У цій статті ми розглядаємо застосування алгоритму Long Short-Term Memory (LSTM) у контексті віртуальних розподілених систем, що є значущим внеском у сферу комп'ютерних наук і технологій. Наукова новизна полягає в адаптації та інтеграції LSTM для прогнозування та оптимізації поведінки віртуальних розподілених систем, що є важливим аспектом у сучасному світі, де обсяги даних зростають надзвичайно швидко. Віртуальні розподілені системи стають основою численних технологічних процесів, від хмарних обчислень до аналізу великих даних, і їх надійність, ефективність та безпека безпосередньо залежать від точності та ефективності прогнозування їх поведінки.

Сучасні методи машинного навчання та аналітичні інструменти відкривають нові горизонти у цій області. Вони дозволяють не лише збирати та аналізувати величезні обсяги даних, а й робити точні прогнози щодо поведінки системи. Цей підхід допомагає виявити потенційні проблеми до їх виникнення, оптимізувати ресурси та підвищити ефективність роботи системи. [2]

В цій роботі ми зосереджуємося на детальному аналізі використання LSTM в різноманітних сценаріях, що включають прогнозування навантаження на сервери та оптимізацію розподілу ресурсів у віртуальних розподілених системах. Ми демонструємо, як LSTM може використовуватися для точного прогнозування поведінки системи, що сприяє підвищенню ефективності та надійності. Окрім цього, ми досліджуємо інтеграцію механізму уваги (attention mechanism) з LSTM, що дозволяє моделі більш точно фокусуватися на релевантних даних, враховуючи специфіку розподілених систем.

Завдяки цьому аналізу можна буде краще оцінити, яким чином сучасні технології машинного навчання та аналітики можуть вдосконалити процеси управління та оптимізації віртуальними розподіленими системами.

2. Застосування алгоритму LSTM в віртуальних розподілених системах.

Оптимізація ресурсів в обчислювальних віртуальних розподілених системах за допомогою машинного навчання є ключовою для забезпечення максимальної ефективності та пропускну здатності у великомасштабних розподілених обчисленнях. Така система може аналізувати використання ресурсів додатками, щоб визначити найкращий додатковий ресурс для кожного. Для даної задачі можна використовувати алгоритм LSTM (Long Short-Term Memory). [3] Це архітектура рекурентної нейронної мережі (RNN), яка широко використовується в глибокому навчанні. Він чудово вловлює довготривалі залежності, що робить його ідеальним для завдань прогнозування послідовності. [4] Прогнозування навантаження на сервери є важливим аспектом управління ресурсами в хмарних обчисленнях та віртуальних розподілених системах.

Основним поняттям алгоритму LSTM є клітинний стан (cell state), що допомагає моделі LSTM в навчанні та запам'ятовуванні інформації протягом довгих послідовностей. Клітинний стан в LSTM схожий на конвеєр, який працює через всю послідовність вхідних даних та переносить інформацію у часі. [5] Він дозволяє LSTM вибірково додавати чи видаляти інформацію з клітинного стану, використовуючи спеціалізовані структури, що називаються гейтами. Три основні компоненти, які складають клітину LSTM, це:

- **Гейт вводу (Input Gate):** Цей гейт визначає, яку інформацію з поточного вводу слід додати до клітинного стану. Він бере поточний вхід та попередній прихований стан як вхідні дані, обробляє їх через сигмоїдну функцію активації, щоб отримати значення між 0 та 1 (0 означає "відкинути", а 1 означає "зберегти"), а потім через тангенс гіперболічний активації, щоб отримати нові кандидатські значення для клітинного стану. Ці значення потім множаться на вихід гейту вводу, щоб визначити, що додати до клітинного стану.

- **Гейт забуття (Forget Gate):** Гейт забуття визначає, яку інформацію з попереднього клітинного стану слід забути чи зберегти. Він бере попередній прихований стан та поточний вхід, оброблює їх аналогічно гейту вводу та виробляє значення між 0 та 1, щоб визначити, які частини попереднього клітинного стану слід забути (0) або зберегти (1).
- **Гейт виведення (Output Gate):** Гейт виведення визначає, яку інформацію з поточного клітинного стану слід використовувати для створення виведення клітини LSTM. Він бере поточний вхід та попередній прихований стан, оброблює їх і комбінує з клітинним станом для створення поточного прихованого стану. Прихований стан потім передається наступному кроку часового ряду, і його також можна використовувати як виведення LSTM, якщо це потрібно. [6]

Модель LSTM може бути описана математично за допомогою набору рівнянь, що керують її роботою:

$$it = \sigma(W_i \cdot [h_{t-1}] + b_i) \quad (2.1)$$

Гейт вводу (Input Gate). Визначає, яка нова інформація зберігається у стані клітини. Де it визначає, яка нова інформація повинна бути додана до стану клітини

$$ft = \sigma(W_f \cdot [h_{t-1}] + b_f) \quad (2.2)$$

Гейт забуття (Forget Gate). Цей гейт вирішує, яка інформація буде відкинута зі стану комірки. Де ft використовується для визначення, яка інформація з попереднього стану клітини C_{t-1} повинна бути "забута" або відкинута.

$$\begin{aligned} \hat{C}_t &= \tanh(W_c \cdot [h_{t-1}] + b_c) \\ C_t &= f_t \cdot C_{t-1} + i_t \cdot \hat{C}_t \end{aligned} \quad (2.3)$$

Представляє нові кандидатські значення, які можуть бути додані до стану. Де $\hat{C}_t$ кандидат на оновлення стану клітини, що створює список кандидатів на оновлення, що можуть бути додані до стану клітини. C_t – оновлення стану клітини, що поєднує інформацію з гейту забуття та вхідного гейту для оновлення стану клітини.

$$\begin{aligned} o_t &= \sigma(W_o \cdot [h_{t-1}] + b_o) \\ h_t &= o_t \cdot \tanh(C_t) \end{aligned} \quad (2.4)$$

Гейт виведення (Output Gate). Цей гейт контролює вивід стану клітини. Де, o_t визначає, яка частина стану клітини повинна бути виведена на наступний шар або використана як вихід. h_t – це фактичний вихід одиниці LSTM, що базується на оновленому стані клітини та вихідному гейті.

У цих рівняннях σ - це сигмоїдна активаційна функція, яка перетворює вхідні значення у діапазон від 0 до 1, а $\tanh$ - гіперболічний тангенс, який нормалізує значення у діапазон від -1 до 1. Ваги W та зміщення b визначають силу зв'язків між різними одиницями у мережі та їх вплив на гейти та стан клітини. А $[h_{t-1}]$ означає вихід LSTM-одиниці з попереднього кроку часу. Це ключовий елемент у рекурентних нейронних мережах, який дозволяє мережі "пам'ятати" інформацію з минулих кроків.

3. Прогнозування навантаження на сервери у віртуальних розподілених системах.

Прогнозування навантаження на сервери у віртуальних розподілених системах є критичним аспектом для підтримки та розподілу завдань. LSTM (Long Short-Term Memory), у поєднанні з механізмом уваги (attention mechanism), становить передовий підхід у цій галузі. [7] Механізмом уваги це техніка, яка дозволяє моделі зосереджуватися на певних частинах вхідної послідовності при генерації певної частини вихідної послідовності. Цей механізм особливо корисний у завданнях, пов'язаних із послідовними даними, такими як обробка природної мови або аналіз часових рядів. Механізм уваги дозволяє моделі призначати різні ваги різним частинам вхідних даних, вказуючи, скільки "уваги" слід приділити кожній частині під час виконання обчислення.

Цей процес допомагає моделі вловлювати залежності та важливі особливості у даних, особливо в сценаріях, де вхідна послідовність є довгою, а різні частини послідовності по-різному впливають на результат. Наприклад, у перекладі мов механізм уваги допомагає моделі зосередитися на відповідних словах у джерельному реченні при перекладі певного слова на цільову мову. Механізм уваги у машинному навчанні можна описати за допомогою формул, які ілюструють, як модель визначає, на яку частину вхідних даних зосередити увагу.

Розрахунок оцінок уваги:

$$e_{t,i} = f(\text{decoder\_state}_t, \text{encoder\_state}_i) \quad (3.1)$$

де, $e_{t,i}$ це оцінка уваги для i -го елемента вхідної послідовності на часовому кроці t . Функція f може бути, наприклад, добутком векторів або мережею перцептронів.

$$\alpha_{t,i} = \frac{\exp(e_{t,i})}{\sum_{j=1}^{T_x} \exp(e_{t,j})} \quad (3.2)$$

Нормалізовані ваги уваги $\alpha_{t,i}$ визначають, скільки "уваги" слід приділити кожному елементу вхідних даних.

$$\text{context}_t = \sum_{j=1}^{T_x} \alpha_{t,i} \times \text{encoder\_output}_i \quad (3.3)$$

Контекстний вектор context_t утворюється як зважена сума вихідних даних кодувальника, де ваги визначаються на основі ваг уваги.

Ці кроки уможливають моделі динамічно вибирати, на яку частину вхідної послідовності зосередити увагу при генерації кожного елемента вихідної послідовності, покращуючи здатність моделі обробляти довгі та складні послідовності даних.

4. Архітектура прогнозування навантаження на сервери у віртуальних розподілених системах.

На рисунку 4.1 зображена архітектура прогнозування навантаження на сервери у віртуальних розподілених системах. Розглянемо призначення різних компонентів архітектури на сервери у віртуальних розподілених системах:

- Символ 1 (рис. 4.1), містить вхідний набір даних, який містить інформацію про навантаження на сервери віртуальних розподілених систем (час, дані заліза, навантаження, температуру і т.д.).
- Символ 2 (рис. 4.1), процес нормалізує або масштабує дані таким чином, щоб всі значення знаходились у певному діапазоні, зазвичай від 0 до 1. Це полегшує тренування моделі, оскільки масштабовані дані зазвичай сприяють кращій збіжності.
- Символ 3 (рис. 4.1), SMOTE (Synthetic Minority Over-sampling Technique): цей метод використовується для боротьби з дисбалансом класів у наборі даних. Він створює синтетичні приклади міноритарного класу для вирівнювання кількості прикладів між класами, що допомагає уникнути упередженості моделі до домінуючого класу.
- Символ 4 (рис. 4.1), перетворення, де дані перетворюються (reshaping) для подачі у LSTM шари. LSTM мережі потребують вхідних даних у формі тривимірного масиву (зазвичай [приклад, часові кроки, ознаки]), тому дані мають бути переформатовані відповідно.
- Символ 5 (рис. 4.1), набір для навчання (Train Dataset) – це частина даних, яка використовується для тренування моделі. Вона проходить через LSTM шари для вивчення залежностей в часі та ідентифікації шаблонів.
- Символ 6 (рис. 4.1), позначає LSTM модель.
- Символ 7 (рис. 4.1), позначає ворота забування (forget gate) в LSTM, які вирішують, яку інформацію з попереднього стану слід відкинути.
- Символ 8 (рис. 4.1), позначає ворота модуляції вхідних (input modularity gate) даних в LSTM, які регулюють внесок нової інформації в стан комірки
- Символ 9 (рис. 4.1), позначає ворота вводу (input gate) в LSTM, які контролюють, яка нова інформація додається до стану комірки.

- Символ 10 (рис. 4.1), позначає стан комірки в LSTM, який зберігає інформацію про весь час тренування.

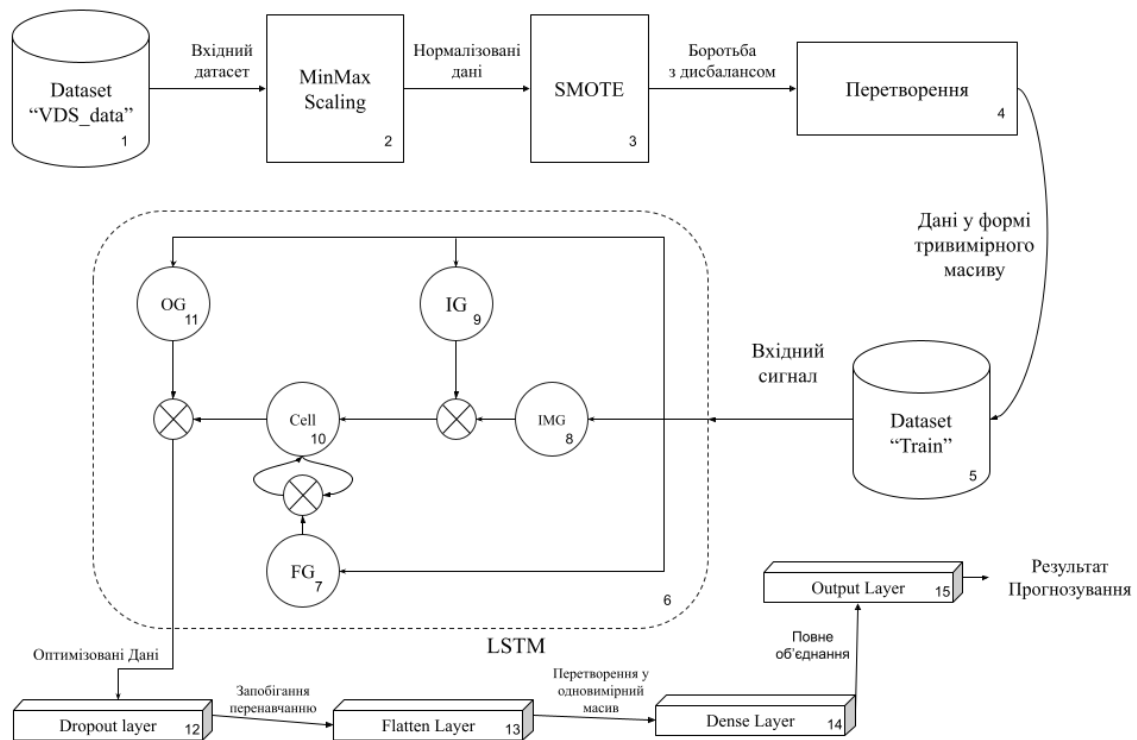


Рис. 4.1 Архітектура прогнозування навантаження на сервери у віртуальних розподілених системах

- Символ 11 (рис. 4.1), позначає ворота виводу (output gate) в LSTM, які визначають, яка інформація передається до виходу з стану комірки.
- Символ 12 (рис. 4.1), позначає шар (Dropout Layer), який запобігає перенавчанню шляхом випадкового відключення деяких нейронів під час тренування.
- Символ 13 (рис. 4.1), позначає шар (Flatten Layer), який перетворює вихідні дані з LSTM шарів у одновимірний масив для подальшої обробки.
- Символ 14 (рис. 4.1), позначає шар нейронний (Dense Layer) для подальшої обробки інформації перед передачею до вихідного шару.
- Символ 15 (рис. 4.1), позначає шар для виведення кінцевого прогнозу або результату класифікації.

Висновок.

У статті розглядається використання алгоритму LSTM (Long Short-Term Memory) для оптимізації віртуальних розподілених систем, зокрема для прогнозування навантаження на сервери. Основні нововведення та аспекти статті включають:

- Застосування LSTM в обчислювальних віртуальних розподілених системах: Підкреслюється важливість LSTM для аналізу використання ресурсів додатками і визначення оптимального розподілу ресурсів для кожного додатку.
- Ключові компоненти LSTM: Стаття детально описує клітинний стан (cell state) і його роль у навчанні LSTM, а також три головні гейти: гейт вводу (Input Gate), гейт забуття (Forget Gate) і гейт виведення (Output Gate).
- Математичний опис моделі LSTM: В статті подано набір рівнянь, які використовуються для опису роботи LSTM, включаючи процеси в гейтах та оновлення клітинного стану.
- Механізм уваги у поєднанні з LSTM: Обговорюється, як механізм уваги може покращити здатність LSTM моделей до обробки довгих та складних послідовностей даних.

Описуються процеси розрахунку оцінок уваги, нормалізації оцінок та формування контекстного вектора.

- Підсумовується, як ці технології можуть підвищити точність прогнозів у контексті віртуальних розподілених систем, сприяючи більш ефективному управлінню та розподілу ресурсів.

СПИСОК ЛІТЕРАТУРИ

1. Sarker IH, Furhad MH, Nowrozy R. Ai-driven cybersecurity: an overview, security intelligence modeling and research directions. SN Comput Sci. 2021;2:1–18.
2. Zhang C, Lu Y. Study on artificial intelligence: the state of the art and future prospects. J Industrial Inform Integr. 2021;23: 100224.
3. Aibin M (2020) LSTM for Cloud Data Centers Resource Allocation in Software-Defined Optical Networks. In: 2020 11th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON). IEEE, New York, p 0162–0167
4. IBM. (n.d.). Recurrent Neural Networks (RNN) - Overview. Retrieved from <https://www.ibm.com/topics/recurrent-neural-networks>
5. J. Kumar, R. Goomer, A.K. Singh, Long short term memory recurrent neural network (lstm-rnn) based workload forecasting model for cloud datacenters. Procedia Comput Sci 125, 676–682 (2018)
6. Ashawa, M., Douglas, O., Osamor, J. et al. Improving cloud efficiency through optimized resource allocation technique for load balancing using LSTM machine learning algorithm. J Cloud Comp 11, 87 (2022). <https://doi.org/10.1186/s13677-022-00362-x>
7. Zhu, Y., Zhang, W., Chen, Y. et al. A novel approach to workload prediction using attention-based LSTM encoder-decoder network in cloud environment. J Wireless Com Network 2019, 274 (2019). <https://doi.org/10.1186/s13638-019-1605-z>

Telezhenko Denys

PhD student

*V.N. Karazin Kharkiv National University, Svobody Square, 4,
Kharkiv-22, Ukraine, 61022*

Forecasting and analytics in virtual distributed systems: Using machine learning models and analytical tools

This scientific article is devoted to the development of a conceptual model for the synthesis of the architecture of virtual distributed systems (VDS). The article examines key aspects of virtual distributed systems, including hardware, hypervisors, virtual machines, and management modules. The methodological principles of architecture synthesis are highlighted, starting from the analysis of system requirements, architecture design, implementation and testing, ending with the evaluation and optimization of VRS performance. The article emphasizes the importance of each stage in this process, emphasizes the need for a deep understanding of system requirements and the selection of appropriate technologies. The article pays special attention to the role of hypervisors and virtual machines in VRS, their connection to hardware and resource management capabilities. This article will be useful for virtualization and computing researchers and practitioners who are designing or optimizing virtual distributed systems. The article is devoted to important issues related to forecasting and optimization of virtual distributed systems, which is a key element of modern technological infrastructures. With the development of computing technologies and artificial intelligence, the need for effective resource management and support of high throughput in computing systems is increasing. The purpose of this scientific work is to research and analyze the application of machine learning algorithms, in particular LSTM (Long Short-Term Memory) and the attention mechanism, for forecasting and optimization of virtual distributed systems. The paper seeks to examine in detail how these technologies can improve resource management, provide higher efficiency and throughput of systems, and analyze how they help identify potential problems and optimize resource allocation based on accurate forecasts and historical data analysis. This paper uses various research methods, which include

Keywords: *virtual distributed systems (VDS), architecture synthesis, LSTM, load forecasting, Input Gate, Forget Gate, Output Gate, attention mechanism.*