

DOI: <https://doi.org/10.36910/6775-2524-0560-2025-60-27>

УДК 004.02

Радченко Костянтин Олександрович, ст. викладач

<https://orcid.org/0000-0002-1282-6307>

Терейковський Ігор Анатолійович, д.т.н., професор

<https://orcid.org/0000-0003-4621-9668>

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», м. Київ, Україна

МЕТОД ПРОГНОЗУВАННЯ НАВАНТАЖЕННЯ НА ВЕБСЕРВЕР З УРАХУВАННЯМ ВЕЙВЛЕТ-АНАЛІЗУ ВЕБЛОГІВ ТА ВЕБТРАФІКУ

Радченко К. О., Терейковський І. А. Метод прогнозування навантаження на вебсервер з урахуванням вейвлет-аналізу веблогів та вебтрафіку. Стаття присвячена дослідженню та розробці методу прогнозування навантаження на вебсервер із використанням вейвлет-аналізу як інструменту попередньої обробки даних веблогів і вебтрафіку. Актуальність роботи зумовлена необхідністю забезпечення стабільної продуктивності вебсервісів у сучасних умовах зростаючої інтенсивності та нерівномірності навантаження. Запропонований підхід базується на застосуванні дискретного вейвлет-перетворення (DWT), яке забезпечує багаторівневий розклад часових рядів і дозволяє виокремити як глобальні тенденції, так і локальні коливання трафіку. Такий метод підвищує стійкість до шумів та сприяє виявленню прихованих закономірностей у даних. У подальшому оброблені часові ряди використовуються в моделях глибинного навчання, зокрема у рекурентних нейронних мережах типу LSTM або Prophet, що демонструють високу здатність до моделювання нелінійних залежностей і довготривалих кореляцій. Проведено експериментальне порівняння ефективності прогнозування навантаження із застосуванням вейвлет-фільтрації та без неї. Результати підтвердили зниження середньої похибки прогнозу, покращення показників балансування запитів у кластері та підвищення відмовостійкості системи. Отримані результати мають практичну цінність для розроблення інтелектуальних систем управління обчислювальними ресурсами та можуть бути інтегровані у системи моніторингу для підвищення ефективності роботи вебсервісів в умовах динамічного навантаження.

Ключові слова: прогнозування навантаження, вебсервер, вейвлет-аналіз, вебтрафік, часовий ряд, балансування навантаження, кластер, обробка сигналу.

Radchenko K., Tereikovskiy I. Method for predicting web server load based on wavelet analysis of web logs and web traffic. The article is devoted to the study and development of a method for web server load prediction using wavelet analysis as a tool for preprocessing weblog and web traffic data. The relevance of the research lies in the need to ensure stable performance of web services under the conditions of increasing intensity and uneven distribution of workload. The proposed approach is based on the application of the discrete wavelet transform (DWT), which provides a multilevel decomposition of time series and makes it possible to capture both global trends and local fluctuations in traffic. Such decomposition increases noise resistance and facilitates the detection of hidden patterns in the data. The preprocessed time series are further applied in deep learning models, particularly in recurrent neural networks such as LSTM or the Prophet framework, which demonstrate high capability in modeling nonlinear dependencies and long-term correlations. An experimental comparison was carried out to evaluate the effectiveness of load prediction with and without wavelet-based filtering. The results confirmed a reduction in the mean prediction error, improved request balancing within the cluster, and increased overall fault tolerance of the system. The obtained findings have practical value for the design of intelligent resource management systems and can be integrated into monitoring solutions to enhance the efficiency and reliability of web services operating under dynamic load conditions.

Keywords: load forecasting, web server, wavelet analysis, web traffic, time series, load balancing, cluster, signal processing.

Постановка наукової проблеми. Зі зростанням обсягів вебтрафіку, популярності динамічних вебзастосунків та стрімкого розвитку розподілених обчислювальних систем питання ефективного управління ресурсами вебсерверів стає критично важливим. Особливе значення має завчасне прогнозування навантаження, яке дозволяє забезпечити високу доступність сервісів, мінімізувати затримки в обробці запитів та підвищити відмовостійкість усієї інфраструктури.

Однак, вебтрафік має складну часову структуру, схильний до нерегулярності, раптових піків та залежностей, які змінюються впродовж доби, тижня або внаслідок зовнішніх подій. Стандартні підходи до прогнозування, засновані лише на класичних методах статистичного аналізу або машинного навчання, часто виявляються недостатніми для точного урахування таких характеристик. Водночас, веблоги — як джерело історичних даних про активність користувачів — містять багатий набір часових залежностей, які можуть бути використані для вдосконалення моделей прогнозування, але потребують попередньої обробки через їхню високочастотну природу та наявність шуму.

У цьому контексті актуальною є наукова проблема створення методу прогнозування навантаження на вебсервер, що базується на попередньому вейвлет-аналізі веблогів та вебтрафіку. Такий метод має дозволити ефективно вилучати як глобальні, так і локальні закономірності,

підвищити якість часової репрезентації даних та, як наслідок, покращити точність прогнозування навантаження в умовах високої варіативності вхідних даних..

Аналіз досліджень. Проблематика прогнозування навантаження на вебсервери активно досліджується в контексті забезпечення масштабованості, доступності та стабільності вебсервісів. Серед найбільш поширених підходів слід виділити методи на основі часових рядів (ARIMA, SARIMA), статистичного моделювання, регресійного аналізу, а також алгоритми машинного навчання — зокрема, рекурентні нейронні мережі (RNN) та моделі довготривалої короткочасної пам'яті (LSTM).

У роботах [1, 2] підкреслюється ефективність застосування LSTM для моделювання нелінійних залежностей у часі, притаманних запитам до вебсерверів. Проте, ці методи часто мають обмеження в умовах високошумових даних або при наявності багаторівневих трендів і короткочасних аномалій. Для покращення якості вхідних даних та зменшення впливу шумів усе частіше використовуються методи попередньої обробки на основі вейвлет-перетворень [3].

Дискретне вейвлет-перетворення (DWT) дозволяє декомпонувати сигнал на апроксимаційні та детальні компоненти, що дає змогу окремо аналізувати високочастотні сплески активності (деталі) та загальні тенденції (апроксимації). Згідно з дослідженнями [4–6], поєднання вейвлет-аналізу та нейронних мереж дозволяє підвищити точність прогнозу, зменшити середньоквадратичну помилку та підвищити стійкість моделі до аномальних значень.

Окрему увагу приділяється застосуванню вейвлет-фільтрації для аналізу веблогів, які характеризуються нерівномірністю часових міток, великою кількістю запитів та складною структурою запитів користувачів. У дослідженнях [7, 8] зазначено, що попередня обробка веблогів із використанням вейвлетів дозволяє ефективніше виділяти поведінкові шаблони користувачів, які критично важливі для побудови адекватної моделі навантаження.

Таким чином, незважаючи на наявність значного обсягу робіт у сфері прогнозування навантаження, досі залишається актуальним завдання комплексної інтеграції вейвлет-аналізу та моделей глибинного навчання для підвищення точності прогнозування у динамічному середовищі. Існує потреба у методі, який дозволить врахувати багаторівневу структуру вебтрафіку та особливості часових залежностей у веблотах, що й зумовлює доцільність даного дослідження.

Мета роботи. Метою даного дослідження є розробка методу прогнозування навантаження на вебсервер із використанням вейвлет-аналізу для попередньої обробки даних веблогів та вебтрафіку з подальшим застосуванням нейронних мереж типу LSTM, що дозволить підвищити точність прогнозування, забезпечити більш ефективне балансування навантаження у веб-кластерах та підвищити загальну відмовостійкість обчислювальної інфраструктури.

Виклад основного матеріалу й обґрунтування отриманих результатів дослідження. Запропонований метод прогнозування навантаження на вебсервер ґрунтується на використанні визначеної системи критеріїв ефективності, що дозволяють оцінити якість моделі та її здатність передбачати поведінку системи в умовах змінного трафіку.

Багатомасштабна обробка даних реалізується через дискретне вейвлет-перетворення (DWT), що дає змогу аналізувати трафік на різних рівнях: виявляти глобальні тренди (повільні зміни навантаження), періодичні коливання (щоденні або тижневі патерни) та локальні аномалії (наприклад, DoS-атаки чи сплески від рекламних кампаній). Такий підхід дозволяє будувати як довгострокові, так і короткострокові прогнози.

Математично це подається як декомпозиція сигналу на різні частотні компоненти:

$$x(t) = \sum_{j=1}^J \sum_{k=1}^K \psi_{j,k}(t) \cdot c_{j,k},$$

де $x(t)$ – початковий сигнал (дані), J – загальна кількість рівнів декомпозиції, K – число елементів (точок) на кожному рівні, $\psi_{j,k}(t)$ – вейвлет-функція для відповідного масштабу j , $c_{j,k}$ – коефіцієнти вейвлет-перетворення для рівня j .

Виділення деталей здійснюється через фільтрацію високочастотних компонентів:

$$x_{details}(t) = x(t) - \hat{x}(t),$$

де $x(t)$ – початковий сигнал, $\hat{x}(t)$ – його згладжене (наближене) представлення, сформоване за допомогою фільтрації, наприклад, через низькочастотний фільтр або метод усереднення.

Збереження інформації передбачає мінімізацію втрат енергії сигналу при декомпозиції:

$$E = \sum_{j=1}^J \sum_{k=1}^K |c_{j,k}|^2,$$

де E характеризує загальну збережену енергію сигналу, $c_{j,k}$ – відповідні коефіцієнти розкладу для рівня j і точки k .

Це дозволяє зберегти ключову інформацію для точного прогнозування.

Локальне прогнозування виконується на основі обмежених сегментів даних:

$$\hat{y}(t_0) = f(x(t_0), \theta),$$

де $\hat{y}(t_0)$ – прогнозне значення у момент часу t_0 , $x(t_0)$ – відповідний вхідний вектор, $f(\cdot, \theta)$ – модель прогнозування (наприклад, нейронна мережа типу LSTM), θ – параметри цієї моделі.

Інтеграція прогнозів дозволяє об'єднати результати кількох моделей в єдиний прогноз за допомогою вагового середнього:

$$\hat{y}_\Sigma = \sum_{i=1}^m w_i \hat{y}_i,$$

де $\hat{y}_i$ – прогнозне значення від i -ї моделі, w_i – її вага, що може визначатися відповідно до точності або інших критеріїв, m – кількість використаних моделей.

Точність прогнозування оцінюється як частка правильних передбачень у загальній кількості або як відхилення між передбаченим і реальним значенням:

$$Acc = \frac{1}{n} \sum_{i=1}^n I(\hat{y}_i = y_i) * 100\%,$$

де $\hat{y}_i$ – прогноз на i -му кроці, y_i – реальне значення на i -му кроці, $I(\hat{y}_i = y_i)$ – індикаторна функція, яка дорівнює 1, якщо прогноз збігається з фактичним значенням, і 0 – в іншому випадку, n – загальна кількість спостережень.

Об'єднана метрика помилок (Combined Error Metric, CEM) є інтегральним показником, що поєднує в собі декілька класичних метрик оцінювання точності прогнозу. Її формування базується на комбінації середньоквадратичної помилки (RMSE), середньої абсолютної помилки (MAE) та середньої абсолютної відносної помилки (MAPE), що дозволяє всебічно оцінити як абсолютні, так і відносні похибки прогнозування:

$$CEM = w_1 \cdot RMSE + w_2 \cdot MAE + w_3 \cdot MAPE,$$

де w_1, w_2, w_3 – вагові коефіцієнти відповідно до важливості кожного критерію; умова нормування $w_1 + w_2 + w_3 = 1$.

Така агрегована оцінка дозволяє одночасно враховувати різні аспекти похибки. MAE забезпечує загальну оцінку середньої похибки, RMSE – є більш чутливою до екстремальних значень, що особливо критично при аналізі пікового навантаження, а MAPE дозволяє порівнювати ефективність прогнозу на різних масштабах чи в різних моделях.

Обчислювальна складність алгоритму визначається функцією часу виконання залежно від обсягу даних:

$$T(n) = O(f(n)),$$

де $T(n)$ – функція часової складності, $f(n)$ – кількість обчислювальних операцій для обробки n елементів.

Оцінка ресурсної ефективності (RE) може бути подана як співвідношення витрачених обчислювальних ресурсів до обсягу оброблених даних D :

$$RE = \frac{Res}{D},$$

де Res — використані ресурси (пам'ять, процесорний час тощо), а D — кількість оброблених даних.

Стійкість моделі до шуму оцінюється як зміна точності при введенні шумових збурень у вхідні дані:

$$Rob = \frac{Acc_{noise}}{Acc_{clear}} \times 100,$$

де Acc_{noise} – точність моделі на зашумлених даних, Acc_{clear} – точність на чистих даних.

Адаптивність моделі визначається здатністю оперативно змінювати параметри для пристосування до нових умов. Формально це вимірюється кількістю ітерацій до досягнення стабільного прогнозу:

$$Adapt = \frac{I_{conv}}{T_{avail}},$$

де I_{conv} – кількість ітерацій до збіжності, T_{avail} – доступний час для адаптації.

Інтеграційна готовність моделі описується часом, необхідним для впровадження в робочу систему, та кількістю параметрів, які потребують налаштування:

$$Integration\ Time = T_{depl} + R_{config},$$

де T_{depl} – час розгортання, R_{config} – обсяг конфігураційних дій.

Загальна ефективність моделі (Overall Model Quality, OMQ) може бути представлена як зважене середнє значення набору ключових метрик:

$$OMQ = \sum_{i=1}^m w_i M_i,$$

де w_i – вага кожної метрики, M_i – значення відповідної метрики, m – кількість критеріїв оцінювання.

Для оцінювання загальної якості моделі прогнозування на основі вейвлет-декомпозиції, а також ефективності локального прогнозування та інтеграції результатів пропонується наступна комбінована формула:

$$\hat{y}_f = \alpha \sum_{j=1}^J \sum_{k=1}^K |c_{j,k}|^2 + \beta (x(t) - \hat{x}(t)) + \gamma \sum_{i=1}^m w_i \hat{y}_i,$$

де $\hat{y}_f$ – фінальний прогноз, w_i – вага кожної моделі інтеграції прогнозів, $c_{j,k}$ – коефіцієнти вейвлет-декомпозиції, $(x(t) - \hat{x}(t))$ – виділені деталі сигналу, $\hat{y}_i$ – часткові прогнози, α, β, γ – вагові коефіцієнти для кожного з блоків прогнозування.

Отже, багатомасштабний аналіз та виокремлення суттєвих елементів сигналу є ключовими складовими при обробці даних, що безпосередньо впливають на якість прогнозування завдяки збереженню та фокусуванню на значущій інформації. Комбінування результатів прогнозів з різних моделей або методів сприяє підвищенню загальної точності передбачень.

Для забезпечення всебічної оцінки ефективності прогнозної моделі доцільно об'єднати множину критеріїв – як пов'язаних з якістю прогнозу, так і з характеристиками обробки даних – у єдину інтегральну метрику. Такий підхід дозволяє здійснити збалансовану оцінку, яка охоплює не лише точність, але й здатність моделі до багаторівневої обробки, виокремлення ознак, збереження ключової інформації та ефективного об'єднання результатів прогнозування.

Формалізована оцінка прогнозу представлена як:

$$\hat{y}_{final} = w_1 \cdot Acc + w_2 \cdot CEM + w_3 \cdot RE + w_4 \cdot Rob + w_5 \cdot Adapt + \hat{y}_f,$$

де Acc – показник точності прогнозу, CEM – об'єднана метрика помилок (наприклад, MAE, RMSE, MAPE), RE – ресурсна ефективність моделі, $Adapt$ – адаптивність до змін вхідних даних, Rob – стійкість до шумових впливів, w_i – вагові коефіцієнти, що визначають значущість відповідних критеріїв, $\hat{y}_f$ – результат базової моделі.

Таким чином, точність та комбінована помилкова метрика (CEM) дозволяють кількісно оцінити здатність моделі точно відтворювати поведінку вебнавантаження на основі різних типів похибок. Ресурсна ефективність демонструє, наскільки оптимально модель використовує обчислювальні ресурси, адаптивність – її гнучкість до нових умов, а стійкість до шуму – здатність зберігати точність при наявності спотворень у даних.

Послідовність реалізації методу, що враховує комплекс критеріїв ефективності, конкретизується наступними етапами.

Етап 1. Оцінювання стійкості до шуму та зміни вхідних даних

На вхід надходять дані у вигляді множини $D_{in} = \{w_m, \omega, X_{train}, \mu_{sys}\}$, де w_m – базовий (материнський) вейвлет, $\omega = \{\omega_1, \dots, \omega_k\}$ – вектор вагових коефіцієнтів для критеріїв якості вейвлетів, X_{train} – навчальні множини часових рядів, $\mu_{sys} = \{\mu_{CPU}, \mu_{RAM}, \mu_{IO}\}$ – метрики системного навантаження (процесор, оперативна пам'ять, дисковий ввід/вивід).

На цьому етапі виконується тестування моделі на штучно зашумлених вибірках для аналізу чутливості методу до флуктуацій у вхідних даних.

Передбачено попереднє Z-нормування часового ряду за формулою:

$$x_{norm} = \frac{x - \mu}{\sigma},$$

де μ – середнє значення, σ – стандартне відхилення.

Для виявлення та згладжування викидів використовується медіанний фільтр. Аналіз спектра шуму здійснюється на основі розподілу енергії вейвлет-коефіцієнтів:

$$E_j = \sum_t |w_j(t)|^2.$$

Оцінка стабільності прогнозу проводиться за відношенням сигналу до шуму:

$$SNR = \frac{\sum_{i=1}^N f(x_i)^2}{\sum_{i=1}^N (f(x_i) - \hat{f}(x_i))^2},$$

де $f(x_i)$ – істинні значення, $\hat{f}(x_i)$ – прогнозовані значення.

Результатом цього етапу є вихідна множина $D_{out} = \{X, \chi(w_m)\}$, де X – стабілізований часовий ряд, показник чутливості моделі до шуму $\chi(w_m)$ – показник чутливості до шуму, який належить інтервалу $[0,1]$.

Етап 2. Формування вхідної множини ознак

Новий набір вхідних даних представлено як $D_{in} = \{X, \nabla X(t), Tr(X), A_j(t), D_j(t)\}$, де X – сигнал запитів, $A_j(t)$ – апроксимаційні компоненти, $D_j(t)$ – детальні компоненти після вейвлет-декомпозиції, $\nabla X(t)$ – градієнт, $Tr(X)$ – тренд.

Для виявлення локальних трендів застосовується ковзне середнє (SMA, Simple Moving Average):

$$SMA = \frac{1}{n} \sum_{i=t-n+1}^t x_i,$$

де x_i – значення часового ряду у момент часу i , n – розмір вікна згладжування.

Застосовується також ковзна дисперсія для оцінки локальної мінливості:

$$Var = \frac{1}{n-1} \sum_{i=t-n+1}^t (x_i - SMA)^2.$$

Часова залежність враховується через лагові ознаки:

$$Lag_k(x_t) = x_{t-k},$$

що дозволяє моделі враховувати історичні стани системи.

Компоненти вейвлет-декомпозиції, отримані за допомогою алгоритму *à trous*, використовуються як додаткові ознаки, що розкривають поведінку сигналу на різних масштабах часу. Для кожної детальної компоненти D_j вейвлету визначається її енергія E_j , що дозволяє оцінити інформативність частотних діапазонів.

На виході формується матриця ознак $D_{out} = \{X \in \mathbb{R}^{T \times m}\}$, де T – кількість часових відміток, m – кількість ознак (оригінальний ряд, лаги, SMA, вейвлет-компоненти тощо). Кожен рядок матриці – вектор параметрів на момент часу t :

$$X_t = [x_t, Lag_1(x_t), SMA, Var, D_1, E_1, \dots].$$

Етап 3. Вибір моделі та побудова прогнозної системи

На цьому етапі використовується набір вхідних даних $D_{in} = \{X_t, y_t, y_{t+k}, k\}$, де $X_t \in \mathbb{R}^m$ – вектор ознак на момент часу t , $y_t \in \mathbb{R}^m$ – цільова змінна, яка відображає рівень навантаження на вебсервер, y_{t+k} – фактичне значення навантаження через k кроків часу t , $k \in \mathbb{Z}^+$ – горизонт прогнозування.

Метою є побудова моделі, яка апроксимує функціональну залежність:

$$y_t = f(X_t) + \varepsilon_t,$$

де ε_t – випадкова похибка моделювання.

Для прогнозування послідовностей часових даних застосовується архітектура довготривалої короткочасної пам'яті (LSTM), що моделює залежності за допомогою рекурентних структур:

$$h_t = LSTM(x_t, h_{t-1}, c_{t-1}),$$

$$\hat{y}_{t+1} = W_{out} \cdot h_t + b,$$

де x_t – вектор ознак у момент t (наприклад, значення трафіку, його похідні, день тижня), h_t – прихований стан (короткострокова пам'ять), c_t – комірка пам'яті (довгострокова пам'ять), W_{out} і b – вагові коефіцієнти та зсув відповідно.

Цей підхід ефективно захоплює як короткострокові (денні), так і довгострокові (тижневі) залежності у часових рядах.

Доповненням до LSTM є використання вейвлет-нейронної мережі (WNN), що поєднує традиційні нейромережеві структури із вейвлет-базисами як активаційними функціями:

$$\hat{y}_t = \sum_{j=1}^N w_j \psi\left(\frac{x_t - a_j}{b_j}\right),$$

де ψ – материнська вейвлет-функція, a_j – зсув (центр), b_j – масштаб (ширина), w_j – вагові коефіцієнти.

Такий підхід дозволяє вбудовувати спектральне представлення сигналу безпосередньо в мережу, що особливо актуально для обробки імпульсних та нерегулярних даних.

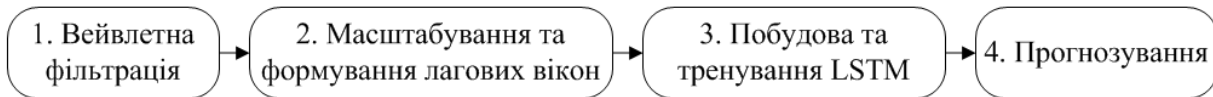


Рис. 1. Стадії реалізації гібридного підходу до прогнозування навантаження

Як показано на рис. 1, запропонована гібридна архітектура передбачає початкову фільтрацію сигналу засобами WNN (вейвлет-декомпозиція та реконструкція), після чого модуль LSTM здійснює часовий прогноз.

Параметри моделі (кількість шарів, кількість нейронів, типи вейвлетів) визначаються шляхом перебору сітки або випадкових стратегій. Загальна форма моделі може бути представлена як:

$$y_t = f(W_{(i,j)}),$$

де $W_{(i,j)}$ – вагові параметри між шарами нейромережі.

Для оптимізації навчання використовується алгоритм Adam (Adaptive Moment Estimation), який комбінує середні значення першого моменту (градієнтів) та другого моменту (їх квадратів). Функцією втрат обрано корінь з середньої квадратичної похибки (RMSE).

Для забезпечення здатності до узагальнення модель проходить процедуру перехресної валідації із поділом даних на тренувальну, валідаційну та тестову частини.

Оцінювання точності моделі здійснюється за допомогою таких метрик як середня абсолютна похибка MAE, середня абсолютна відносна похибка MAPE та коефіцієнт детермінації R^2 , що обчислюється за формулою:

$$R^2 = 1 - \frac{\sum_{t=1}^N (y_t - \hat{y}_t)^2}{\sum_{t=1}^N (y_t - \bar{y})^2}.$$

У результаті цього етапу формується навчена модель, здатна здійснювати ефективне прогнозування навантаження на вебсервер на основі вибраних ознак.

Вихідними результатами етапу є $D_{out} = \{\hat{f}(X_t|\theta), \theta, L(\theta)\}$, де $\hat{f}(X_t|\theta)$ – навчена модель, θ – параметри моделі, що налаштовуються у процесі тренування, $L(\theta) = \{MAE, MAPE, RMSE, R^2\}$ – функціонал оцінювання ефективності моделі.

Етап 4. Прогнозування навантаження вебсервера

На цьому етапі вхідними даними є множина векторизованих ознак $D_{in} = \{Z_t\}$, сформована на попередніх кроках обробки. Для прогнозування значення навантаження на момент часу $t + k$ (де k – горизонт передбачення) використовується набір характеристик, який позначимо як:

$$Z_t = \{x_t, x_{t-1}, x_{t-2}, \dots, x_{t-k}, W_t^{(j)}, \nabla x_t, trend(x_t), \mu_t^{(w)}, \sigma_t^{(w)}, Lag_k(x_t)\}.$$

Цей набір включає: актуальне значення ознаки x_t , попередні значення $x_{t-1}, x_{t-2}, \dots, x_{t-k}$, компоненти вейвлет-декомпозиції $W_t^{(j)}$, похідні характеристики, зокрема градієнти, трендові компоненти, ковзне середнє $\mu_t^{(w)}$, ковзну дисперсію $\sigma_t^{(w)}$ за вікно w , а також лагові змінні $Lag_k(x_t) = x_{t-k}$.

Після формування вхідного вектору відбувається оцінка майбутнього значення навантаження з використанням попередньо навченої моделі:

$$\hat{y}_{t+1} = f(x_t, x_{t-1}, x_{t-2}, \dots, x_{t-k}),$$

де $f(\cdot)$ – модель прогнозування (наприклад, LSTM, WNN), а $\hat{y}_{t+1}$ – очікуване значення навантаження у момент часу $t + 1$.

У випадку гібридної архітектури, що поєднує вейвлет-декомпозицію з прогножною моделлю, майбутнє значення сигналу формується шляхом поелементного прогнозування кожної детальної компоненти $d_j(t)$, після чого виконується реконструкція сигналу:

$$\hat{x}(i) = \sum_{j=1}^{p+1} \hat{W}(i, j),$$

де $\hat{W}(i, j)$ – прогнозоване значення j -тої вейвлет-компоненти на момент часу i , а $p + 1$ охоплює також апроксимаційну складову a_p . Завдяки властивості адитивності вейвлет-розкладу, можливе зважене поєднання прогнозованих компонентів для мінімізації загальної похибки:

$$\hat{x}(i) = \sum_{j=1}^{p+1} \alpha_j \hat{W}(i, j), \quad \sum_{j=1}^{p+1} \alpha_j = 1,$$

де α_j – вагові коефіцієнти, визначені на основі величини похибки для кожної окремої компоненти.

Як результат, вихідні дані $D_{out} = \{\hat{Y}_t, plot(y_t, \hat{y}_t)\}$ містять вектор передбачених значень навантаження $\hat{Y}_t = \{\hat{y}_{t+1}, \hat{y}_{t+2}, \dots, \hat{y}_{t+k}\}$ на заданий часовий горизонт k , а також візуалізацію фактичних та прогнозованих значень у вигляді графіка часових рядів.

Етап 5. Автоматизоване налаштування гіперпараметрів

На цьому етапі як вхідні дані розглядається трійка $D_{in} = \{\theta, X, y\}$, де $\theta = \{\theta_1, \theta_2, \dots, \theta_n\}$ – простір гіперпараметрів моделі, зокрема таких як кількість шарів, кількість нейронів у шарах, розмір вікна, коефіцієнт навчання (learning rate), коефіцієнт відсічення (dropout rate) тощо, X – вже сформована матриця ознак, y – цільові значення (мітки), що відповідають навчальним об'єктам.

Оптимальний підбір гіперпараметрів є важливою умовою побудови високоточної моделі для прогнозування навантаження. Правильно налаштовані параметри дозволяють підвищити здатність моделі до узагальнення, зменшити помилки прогнозування, покращити стабільність системи (рис. 2).

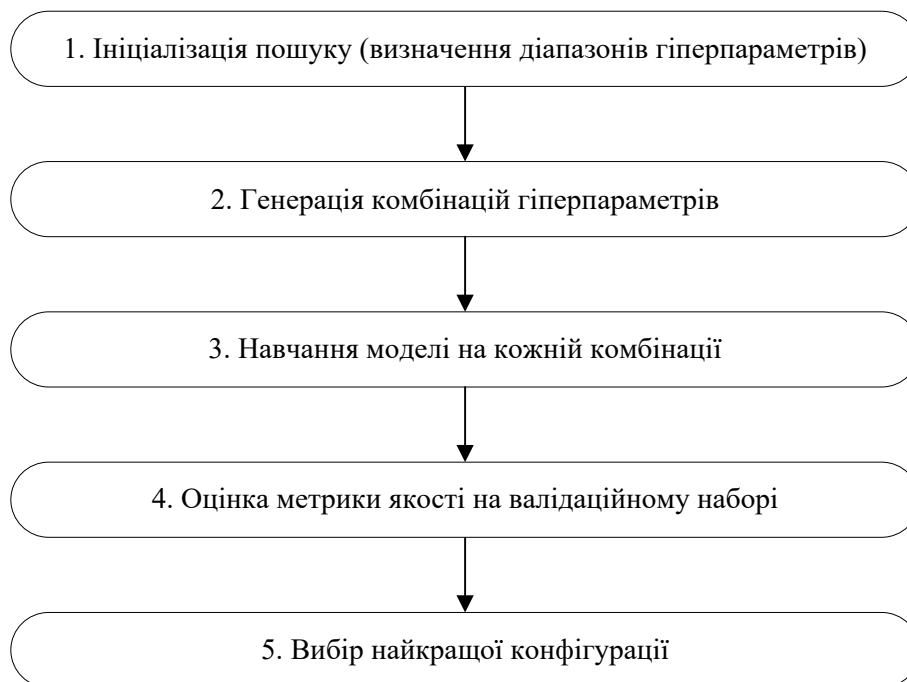


Рис. 2. Стадії автоматизованого налаштування гіперпараметрів

Залежно від характеру вхідних даних та обчислювальних обмежень, для пошуку оптимальних гіперпараметрів використовують такі методи:

- Grid Search – повний перебір усіх можливих комбінацій гіперпараметрів, доцільний за невеликої кількості параметрів;
- Random Search – випадкова генерація комбінацій параметрів, що часто дозволяє знайти вдалі налаштування швидше, ніж Grid Search;

– Bayesian Optimization (на основі бібліотеки Optuna) – адаптивний підхід, який використовує ймовірнісну модель функції втрат для обґрунтованого вибору наступних точок у просторі параметрів. Метод ефективний у випадках із великою вартістю обчислень.

Задача оптимізації формалізується як знаходження таких гіперпараметрів θ^* , що мінімізують функцію втрат L :

$$\theta^* = \arg \min_{\theta} L(y, \hat{y}^{(\theta)}),$$

де θ – вектор гіперпараметрів, $\hat{y}^{(\theta)}$ – прогноз моделі з параметрами θ , y – реальні значення, L – обрана функція втрат (наприклад, RMSE, MAE, R^2).

Для часових рядів також використовуються специфічні метрики, зокрема SMAPE (Symmetric Mean Absolute Percentage Error):

$$SMAPE = \frac{100\%}{N} \sum_{t=1}^N \frac{|y_t - \hat{y}_t|}{(|y_t| + |\hat{y}_t|)/2}$$

і MASE (Mean Absolute Scaled Error):

$$MASE = \frac{\frac{1}{N} \sum_{t=1}^N |y_t - \hat{y}_t|}{\frac{1}{N-1} \sum_{t=2}^N |y_t - y_{t-1}|}.$$

Результатом (вихідними даними) $D_{out} = \{\theta^*, \theta_i \rightarrow M_i, plot(\theta_i, M_i)\}$ є оптимальна конфігурація гіперпараметрів θ^* , яку можна використовувати у подальшому прогнозуванні, таблиця зіставлення кожного набору параметрів $\theta_i = \{\theta_{1i}, \theta_{2i}, \dots, \theta_{ni}\}$ з відповідними метриками якості $M_i = \{MAE_i, MAPE_i, RMSE_i, R^2_i\}$, графічна візуалізація залежностей між параметрами та точністю моделі.

Етап 6. Оцінка якості на різних наборах даних

У якості вхідних даних $D_{in} = \{S_{day}, S_{night}, S_{holiday}\}$ розглядається множина тестових сценаріїв $S = \{S_{day}, S_{night}, S_{holiday}\}$, які моделюють різні умови функціонування вебсервера. Зокрема, сценарій робочого часу S_{day} характеризується високою інтенсивністю запитів та типовим офісним трафіком; нічний режим S_{night} відзначається низьким навантаженням та підвищеною ймовірністю шумів у даних; сценарій святкових днів $S_{holiday}$ відображає нерегулярні запити, сплескову активність або повну їх відсутність, що створює непередбачувані патерни.

На цьому етапі оцінювання ефективності моделі прогнозування вебнавантаження здійснюється шляхом аналізу низки метрик (RMSE, SMAPE, R^2 тощо).

Для візуального дослідження похибок ($y_t - \hat{y}_t$) у межах кожного сценарію будуються гістограми, що дає змогу: визначити наявність систематичного зсуву моделі (схильність до пере- чи недооцінки); виявити “довгі хвости” у розподілі похибок, які відображають рідкісні, але значні відхилення; порівняти характер і дисперсію похибок між різними піднаборами даних.

Додатково модель перевіряється на розширених горизонтах прогнозування ($t + 10, t + 20$), що дозволяє оцінити її стійкість при зростанні відставання та простежити, як змінюються метрики зі збільшенням горизонту, зокрема, чи проявляються систематичні зсуви у передбаченнях.

Таблиця 1. Оцінки метрик для кожного типу даних

Набір	RMSE	SMAPE (%)	R^2
Робочі години	12.5	7.3	0.91
Нічний трафік	4.2	15.7	0.68
Святкові дні	18.6	22.4	0.52

Вихідні результати $D_{out} = \{S_i \rightarrow M_i, Score\}$ включають таблицю оцінок M_i для кожного сценарію S_i (табл. 1), а також узагальнені висновки у вигляді функції $Score = f(Var(R^2_i), max(SMAPE_i))$. Це дозволяє зробити висновки щодо здатності моделі коректно працювати з гетерогенними даними: чи необхідне додаткове навчання, тонке налаштування (fine-tuning) або ж можлива контекстна адаптація (наприклад, із використанням маркерів робочих та вихідних днів).

Етап 7. Візуалізація результатів та формування звіту

У якості вхідних даних $D_{in} = \{\hat{Y}_{WNN}, \hat{Y}_{LSTM}, T, M\}$ розглядаються результати прогнозування, отримані за допомогою моделей WNN та LSTM, часові мітки прогнозу T , а також метрики точності $M = \{MAE, MAPE, SMAPE, RMSE, R^2\}$.

Метою цього етапу є проведення ефективного аналізу результатів прогнозування навантаження на вебсервер. Для цього здійснюється побудова лінійних графіків, що демонструють динаміку зміни трафіку у часі (як фактичних значень, так і прогнозованих). Додатково формується boxplot розподілу похибок для виявлення можливих аномалій та значних коливань у точності прогнозів. Також створюються теплові карти (heatmap), що відображають кореляційні зв'язки між різними змінними, зокрема між прогнозами та реальними даними, що дозволяє виявити приховані закономірності.

Вихідними даними є множина $D_{out} = \{R, V, D\}$, де R – автоматично сформований звіт у форматах PDF/HTML із описом моделей та візуалізаціями, V – набір графічних матеріалів (лінійні графіки, теплові карти, гістограми похибок), D – інтерактивна інформаційна панель (дашборд) для візуального аналізу, реалізована із використанням бібліотек Dash, Streamlit, Plotly. У дашборді передбачена можливість вибору діапазону дат $[t_i, t_j]$, порівняння результатів різних моделей та застосування фільтрації за обраними метриками.

Етап 8. Інтеграція в систему моніторингу

На цьому завершальному етапі відбувається інтеграція побудованої моделі прогнозування у середовище моніторингу вебсервера. Як вхідні дані розглядається множина $D_{in} = \{M^*, A_{API}(t), U(t)\}$, де M^* – обрана ефективна модель прогнозування навантаження, $A_{API}(t) = \{Q(t), V(t), \tau(t), \rho(t)\}$ – набір параметрів, що надходять у момент часу t з API реального вебсервера (кількість запитів $Q(t)$, обсяг трафіку $V(t)$, час відповіді $\tau(t)$, а також використання ресурсів $\rho(t) = \{CPU, RAM, I/O\}$), $U(t)$ – вектор актуальних значень метрик моніторингу.

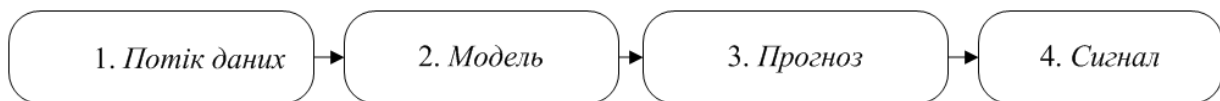


Рис. 3. Схема роботи модуля прогнозування

У даному модулі (рис. 3) реалізується механізм прогнозування через розгортання REST API, що забезпечує взаємодію з іншими компонентами системи (зокрема, з підсистемою моніторингу). REST API приймає дані від вебсервера у форматі HTTP-запиту, здійснює обчислення прогнозу й повертає результат у вигляді JSON-об'єкта.

Після отримання прогнозу система порівнює його зі встановленим порогом. У разі перевищення граничних значень генерується сповіщення для адміністратора або надсилається сигнал у систему балансування навантаження. Порогові значення можуть задаватися вручну або визначатися автоматично на основі попередньо зібраних даних.

Таким чином, усі вхідні дані (кількість запитів, рівень використання процесора та пам'яті тощо) надходять у модуль прогнозування, де здійснюється аналіз та генерується прогноз найближчого навантаження. У випадку перевищення порогів формується попереджувальний сигнал.

Вихідна множина результатів визначається як $D_{out} = \{\hat{Y}_{time}, F, E_{alert}\}$, де $\hat{Y}_{time}$ – прогноз навантаження на найближчий момент часу, F – автоматична фільтрація та передобробка потоку даних, E_{alert} – сигнали й повідомлення про можливі аномалії чи критичні перевантаження.

Запропонований модуль може функціонувати у реальному продакшн-середовищі, безперервно отримуючи дані з вебсервера та надаючи прогнози в режимі реального часу для ефективного моніторингу.

З метою підвищення наочності у таблиці 2 наведено узагальнений та структурований опис ключових етапів реалізації методу прогнозування навантаження на вебсервер.

Процес прогнозування навантаження на вебсервер являє собою багатоступеневу систему, що включає збір логів і метрик, їхню попередню обробку, аналіз, моделювання та інтеграцію прогнозу в робоче середовище. На першому етапі здійснюється збирання даних із вебсерверів (HTTP-логи, показники CPU/RAM, кількість запитів за секунду), які агрегуються у часові ряди з

певною дискретністю (наприклад, 10 секунд або 5 хвилин). Це дозволяє виявляти як короткотермінові, так і довготривалі закономірності у зміні навантаження.

Далі проводиться попередня обробка даних: очищення від шумів, агрегація, а також формування лагових векторів для урахування часових залежностей. Наступним кроком застосовується вейвлет-декомпозиція (DWT, SWT або FWT), яка дає змогу поділити сигнал навантаження на трендову складову та високочастотні коливання (пікові навантаження, сплески).

Прогнозування здійснюється окремо для кожної компоненти: для трендової складової застосовуються статистичні моделі, зокрема Prophet, тоді як для коливань використовуються глибинні архітектури, наприклад LSTM або TDNN. Після цього результати об'єднуються в узагальнений прогноз. Завершальний етап передбачає інтеграцію отриманої моделі у практичну систему: доступ до прогнозу забезпечується через API або інтерактивний дашборд, налаштовуються автоматичні оповіщення про критичні значення навантаження та механізми масштабування ресурсів у хмарному середовищі. Таким чином, методика дозволяє не лише відслідковувати поточний стан, а й адаптивно реагувати на можливі перевантаження.

Таблиця 2. Структура етапи реалізації методу прогнозування навантаження на вебсервер

№	Назва етапу	Вхідні дані D_{in}	Вихідні дані D_{out}
1	Оцінювання стійкості до шуму та зміни вхідних даних	$\{w_m, \omega, X_{train}, \mu_{sys}\}$	$\{X, \chi(w_m)\}$
2	Формування вхідної множини ознак	$\{X, Tr(X), A_j(t), D_j(t)\}$	$\{X\}$
3	Вибір моделі та побудова прогновної системи	$\{X_t, y_t, y_{t+k}, k\}$	$\{\hat{f}(X_t \theta), \theta, L(\theta)\}$
4	Прогнозування навантаження вебсервера	$\{Z_t\}$	$\{\hat{Y}_t, plot(y_t, \hat{y}_t)\}$
5	Автоматизоване налаштування гіперпараметрів	$\{\theta, X, y\}$	$\{\theta^*, \theta_i \rightarrow M_i, plot(\theta_i, M_i)\}$
6	Оцінка якості на різних наборах даних	$\{S_{day}, S_{night}, S_{holiday}\}$	$\{S_i \rightarrow M_i, Score\}$
7	Візуалізація результатів та формування звіту	$\{\hat{Y}_{WNN}, \hat{Y}_{LSTM}, T, M\}$	$\{R, V, D\}$
8	Інтеграція в систему моніторингу	$\{M^*, A_{API}(t), U(t)\}$	$\{\hat{Y}_{time}, F, E_{alert}\}$

У рамках цього дослідження запропоновано гібридний підхід до прогнозування навантаження на вебсервер, який поєднує вейвлет-перетворення з сучасними архітектурами штучних нейронних мереж. Зокрема, розглянута модель WNN+LSTM, де вейвлет-декомпозиція виконує роль попередньої обробки, виокремлюючи ключові тенденції та зменшуючи вплив шумів. Перевага такого підходу полягає у поєднанні здатності Wavelet Neural Network (WNN) виявляти локальні структурні особливості з ефективністю LSTM у моделюванні довготривалих часових залежностей.

Архітектура гібридної моделі включає кілька рівнів: на вхід подаються нормалізовані часові ряди, далі WNN-компоненти аналізують локальні характеристики сигналу, а LSTM-шари відтворюють довгострокові залежності. Остаточний прогноз формується на виході повнозв'язного шару. Для уникнення перенавчання застосовується Dropout з коефіцієнтом 0.2, що підвищує здатність моделі до узагальнення.

Точніше передбачення вебтрафіку у вибраному часовому інтервалі може суттєво покращити оптимізацію програмних рішень, орієнтованих на попереднє кешування та розповсюдження даних. Крім того, воно дозволяє формувати тестові потоки із заданими характеристиками навантаження для перевірки продуктивності систем ще на етапі проектування. Важливим є й те, що попереднє прогнозування дає змогу раціонально використовувати доступні ресурси мережі, запобігаючи перевантаженням та скорочуючи час відгуку системи.

Отримані результати підтверджують, що застосування вейвлет-аналізу у поєднанні з глибинними нейронними мережами дозволяє підвищити точність прогнозування у різних часових

масштабах. Завдяки вейвлет-декомпозиції стає можливим виділення основних структурних компонент сигналу та їх подальший окремий прогноз, що після об'єднання формує більш надійний і точний результат.

Аналіз експериментальних результатів свідчить, що запропонований метод забезпечує суттєве покращення якості прогнозування порівняно з традиційними підходами. Тестування на реальних логах вебнавантаження підтвердило зростання точності прогнозів як у коротко-, так і в довгостроковій перспективі. Особливо важливим є те, що метод успішно передбачає пікові значення навантаження, що є критичним фактором для підтримання безперервної роботи серверної інфраструктури.

У межах дослідження було виконано порівняльні експерименти для оцінки різних методів прогнозування навантаження, використовуючи статистичні дані вебсервера однієї з регіональних аварійно-диспетчерських служб України за період з липня 2018 року до червня 2024 року. Дані містили детальну інформацію про кожен оброблений запит: ім'я хоста, час звернення, запитуваний файл і розмір відповіді у байтах.

Оскільки дані мали розріджену структуру з інтервалом у 5 хвилин, було сформовано первинний набір, нормалізований за середньогодинним діапазоном вебтрафіку, що відображає динаміку оброблених сервером запитів. Попередній аналіз продемонстрував наявність характерних режимів: різке підвищення навантаження у робочі години будніх днів, зниження у вихідні та фоновий рівень активності у нічний час. З цієї причини часовий ряд було поділено на тижневі підсерії з використанням усереднених 5-хвилинних значень для подальших експериментів.

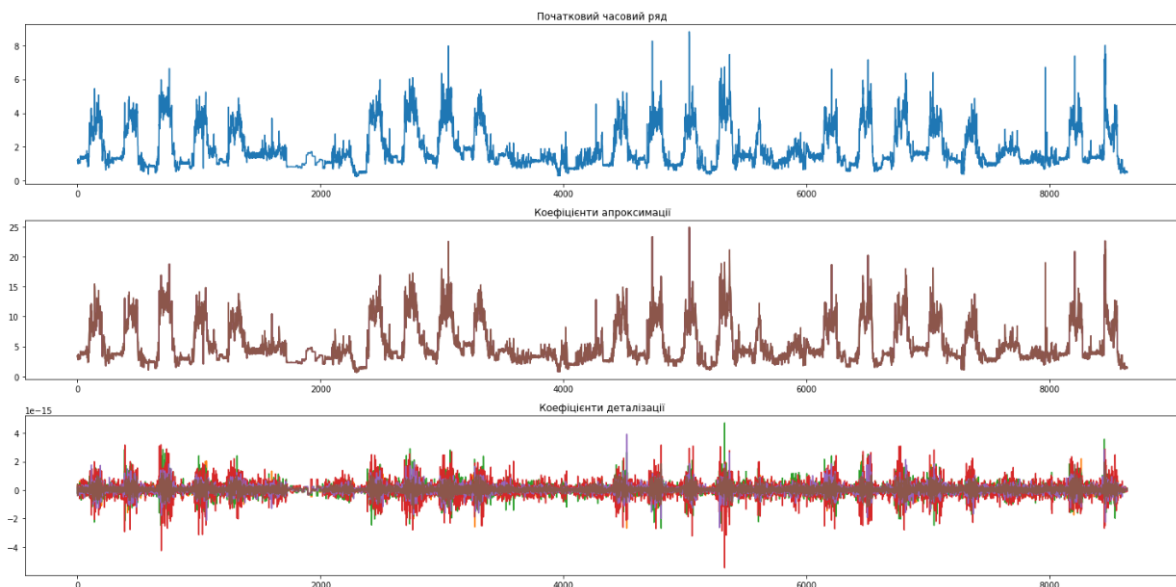


Рис. 4. Розклад рівня навантаженості вебсервера на вейвлет-коефіцієнти

Вибір оптимального базисного вейвлету дозволив провести чисельні експерименти для перевірки точності апроксимації та дослідження можливості використання вейвлет-коефіцієнтів для аналізу локальних особливостей процесу навантаження. Обчислення виконувалися у середовищі Wavelet Toolbox (MATLAB) та з використанням бібліотеки PyWavelets (Python). Отримані результати візуалізовані на рис. 4, де наведено приклади апроксимації рівня вебтрафіку за допомогою вейвлет-спектра різної глибини розкладу.

Аналіз графіків (рис. 4) показав, що досліджуваний процес характеризується наявністю чотирьох періодичних компонентів, кожна з яких відповідає певним часовим інтервалам доби.

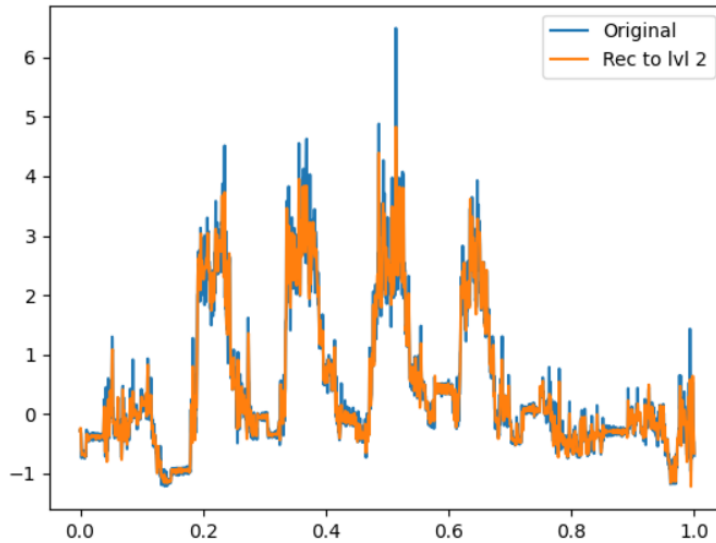


Рис. 5. Спостережуваний і прогнозований рівні навантаження

Крім того, здійснено відновлення сигналу за допомогою зворотного дискретного вейвлет-перетворення на основі отриманих коефіцієнтів. На рис. 5 наведено графіки фактичного та прогнозованого рівнів навантаження.

Отримані результати демонструють високий ступінь подібності між спостережуваними та прогнозними даними. Середньоквадратична похибка апроксимації не перевищила 5 – 8%, тоді як для класичних поліноміальних методів прогнозування, згідно з роботами [6, 12, 13, 17], вона становить близько 10%.

Висновки. У ході виконаного дослідження було розроблено та реалізовано метод прогнозування навантаження на вебсервер, заснований на використанні вейвлет-аналізу. Запропонований метод дав змогу виділити найбільш інформативні характеристики часових рядів навантаження, що забезпечило підвищену точність передбачень. Проведене порівняння з традиційними підходами, зокрема моделлю ARIMA та класичними штучними нейронними мережами, підтвердило перевагу представленого методу у роботі з нестационарними даними та у виявленні багаторівневих змін у процесі формування навантаження.

Практичне застосування методу може стати корисним інструментом для адміністраторів вебсистем, оскільки точне прогнозування дозволяє своєчасно запобігати перевантаженням, підвищувати ефективність серверних ресурсів і гарантувати стабільність функціонування вебсервісів. Високий рівень достовірності прогнозів сприяє також оптимізації використання обчислювальної інфраструктури та зниженню експлуатаційних витрат.

Вибір оптимального типу базисного вейвлету створює основу для проведення числових експериментів, спрямованих на перевірку правильності отриманих результатів шляхом оцінки точності апроксимації статистичних даних. Додатково ці експерименти дозволяють дослідити здатність вейвлет-коефіцієнтів відображати локальні особливості зміни навантаження.

Визначено, що перспективними напрямками подальших досліджень є:

- інтеграція вейвлет-аналізу з методами сучасного машинного навчання (глибинними нейронними мережами, рекурентними моделями, ансамблевими підходами) для підвищення точності прогнозування;
- розробка адаптивних моделей, що автоматично налаштовуються на зміну динаміки навантаження та зберігають ефективність у реальних умовах;
- розширення баз навчальних даних для покращення здатності методу до узагальнення та забезпечення стійкості до різноманітних сценаріїв;
- інтеграція розробленої системи у реальні серверні інфраструктури з подальшим тестуванням у режимі реального часу;
- врахування зовнішніх чинників (сезонних коливань, добових циклів, впливу подій чи рекламних кампаній) для підвищення точності прогнозів.

Отже, подальший розвиток вейвлет-орієнтованих підходів до прогнозування навантаження на вебсервери відкриває значні можливості для вдосконалення управління інформаційними системами, підвищення їхньої стабільності та ефективності використання ресурсів.

Список бібліографічного опису

1. Дронюк І.М. Прогнозування трафіку комп'ютерних мереж для підвищення ефективності використання мережевого обладнання / Дронюк І.М., Федевич О.Ю. // Наук. вісн. НЛТУ України. – 2015. – Вип.25.5. – С.301-307.
2. Платов В. В., Петров В. В. Дослідження самоподібної структури телетрафіку бездротової мережі // Електротехнічні та інформаційні комплекси і системи. – Vol. 3. – 2004. – Р. 38–49.
3. Hu, Z., Terekovskiy, I., Terekovska, L., Tsiutsiura, M., Radchenko, K. (2020). Applying Wavelet Transforms for Web Server Load Forecasting. In: Hu, Z., Petoukhov, S., Dychka, I., He, M. (eds) Advances in Computer Science for Engineering and Education II. ICCSEEA 2019. Advances in Intelligent Systems and Computing, vol 938. Springer, Cham. https://doi.org/10.1007/978-3-030-16621-2_2
4. Радченко К.О. Застосування дискретних вейвлет-перетворень для прогнозування рівня навантаження на вебсервер комп'ютерних мереж загального призначення / К.О. Радченко // Комп'ютерно-інтегровані технології: освіта, наука, виробництво. 2021. № 45. С. 90 – 96. <https://doi.org/10.3691/0/6775-2524-0560-2021-45-13>
5. Радченко К.О. Концептуальна модель забезпечення ефективності прогнозування навантаження на вебсервер / К.О. Радченко // Вчені записки Таврійського національного університету імені В.І. Вернадського. Серія: Технічні науки. 2020. Том 31 (70), № 6. С. 135 – 141. <https://doi.org/10.3283/8/TNU-2663-5941/2020.6-1/23>
6. Радченко К.О. Особливості прогнозування рівня вебтрафіку у комп'ютерних мережах загального призначення / К.О. Радченко // Проблеми інформатизації та управління. 2022. № 3(71). С. 41 – 50. <https://doi.org/10.1837/2/2073-4751.71.17002>
7. Струбицький Р. П. Самоподібна модель завантаженості хмаркових сховищ даних // Вісник Національного університету «Львівська політехніка». – 2015. – №814. – С. 147–156.
8. Федевич О.Ю. Інформаційна технологія аналізу та прогнозування трафіку в комп'ютерних мережах. – Дисертація на здобуття наукового ступеня кандидата технічних наук (доктора філософії) за спеціальністю 05.13.06 «Інформаційні технології» (122 – Комп'ютерні науки). - Національний університет «Львівська Політехніка» МОН України, Львів, 2018. 214 с. <https://lpnu.ua/sites/default/files/2020/dissertation/1651/disfedevychoyu.pdf>

References

1. Droniuk I.M. Forecasting the traffic of computer networks to increase the efficiency of the use of network equipment / Dronyuk I.M., Fedevich O.Yu. // Sciences. release NLTU of Ukraine. – 2015. – Issue 25.5. – P.301-307.
2. Platov V.V., Petrov V.V. Study of the self-similar structure of wireless network teletraffic // Electrotechnical and informational complexes and systems. – Vol. 3. – 2004. – P. 38–49.
3. Hu, Z., Terekovskiy, I., Terekovska, L., Tsiutsiura, M., Radchenko, K. (2020). Applying Wavelet Transforms for Web Server Load Forecasting. In: Hu, Z., Petoukhov, S., Dychka, I., He, M. (eds) Advances in Computer Science for Engineering and Education II. ICCSEEA 2019. Advances in Intelligent Systems and Computing, vol 938. Springer, Cham. https://doi.org/10.1007/978-3-030-16621-2_2
4. Radchenko K.O. Application of discrete wavelet transformations for forecasting the level of load on the web server of general purpose computer networks / K.O. Radchenko // Computer-integrated technologies: education, science, production. 2021. No. 45. С. 90 – 96. <https://doi.org/10.3691/0/6775-2524-0560-2021-45-13>
5. Radchenko K.O. Conceptual model for ensuring the effectiveness of web server load forecasting / K.O. Radchenko // Scientific notes of V.I. Tavri National University. Vernadskyi. Series: Technical sciences. 2020. Volume 31 (70), No. 6. С. 135 – 141. <https://doi.org/10.3283/8/TNU-2663-5941/2020.6-1/23>
6. Radchenko K.O. Peculiarities of predicting the level of web traffic in general purpose computer networks / K.O. Radchenko // Problems of informatization and management. 2022. No. 3(71). С. 41 – 50. <https://doi.org/10.1837/2/2073-4751.71.17002>
7. Strubyskyi R.P. Self-similar model of the load of cloud data storage // Bulletin of the National University "Lviv Polytechnic". – 2015. – No. 814. – pp. 147–156.
8. Fedevich O.Yu. Information technology for traffic analysis and forecasting in computer networks. - Dissertation for obtaining the scientific degree of Candidate of Technical Sciences (Doctor of Philosophy) in the specialty 05.13.06 "Information Technologies" (122 - Computer Sciences). - Lviv Polytechnic National University of the Ministry of Education and Science of Ukraine, Lviv, 2018. 214 p. <https://lpnu.ua/sites/default/files/2020/dissertation/1651/disfedevychoyu.pdf>