УДК 004:891.032.26:616.127–073.7
DOI: 10.15587/1729-4061.2017.110073

Розробка методу навчання ознак та вирішальних правил для прогнозування порушення умов обслуговування в хмарному середовищі

В. В. Москаленко, А. С. Москаленко, С. В. Пімоненко, А. Г. Коробов

Розроблено алгоритм навчання багатошарового екстрактора ознак, що використовує принципи нейронного газу та розрідженого кодування. Запропоновано інформаційно-екстремальний метод двійкового кодування ознакового подання для побудови вирішальних правил. Це дозволяє зменшити вимоги до обсягів навчальних даних і обчислювальних ресурсів та забезпечити високу достовірність прогнозування порушення умов договору про рівень обслуговування в хмарному середовищі

Ключові слова: датацентр, розріджене кодування, нейронний газ, інформаційний критерій, машинне навчання, ройовий алгоритм

1. Вступ

Зростання популярності хмарних сервісів стимулює поширення розподілених центрів обробки даних у глобальному масштабі, що призводить до численних проблем з точки зору планування ресурсів за різними адміністративними доменами. Ефективне планування ресурсами полягає в одночасному забезпеченні мінімізації порушень угод про рівень обслуговування (Service Level Aggrement, SLA), зниження вартості використання хмарних сервісів, підвищення рівня економії енергії, а також підвищення прибутку провайдера послуг. Проте нестаціонарність попиту на хмарні ресурси породжує змінні піки навантаження, що робить неефективними механізми реактивного масштабування сервісів, які ініціалізують процес виділення додаткових ресурсів лише після перевищення певною метрикою критичного значення. Тому активною областю досліджень є проактивні та предиктивні принципи керування ресурсами, які дозволяють завчасно ініціалізувати виділення необхідних ресурсів. Крім того, використання предиктивних механізмів дозволяє забезпечити ефективний перерозподіл ресурсів шляхом визначення невдалих кандидатів (датацентрів чи окремих серверів) на розміщення віртуальних машин. При цьому прогнозування порушення SLA дозволяє зняти невизначеність щодо функціонального стану сервісів на різних рівнях хмарної системи і підвищити ефективність багатокритеріальних оптимізаційних алгоритмів при плануванні розподілу ресурсів [1].

Одним із підходів до прогнозування порушення SLA на різних рівнях хмарної системи є використання ідей та методів машинного навчання, що формують прогностичну модель шляхом аналізу часових рядів зміни ключових показників продуктивності, ключових показників якості та системних повідомлень [2]. Однак використання традиційних однорівневих методів

машинного навчання, для яких характерна експоненційна залежність кількості параметрів моделі від кількості ознак розпізнавання, за умов багатовимірності спостережень призводить до зростання вимог до обчислювальних ресурсів та обсягу навчальних даних. Саме тому найбільш перспективним напрямком синтезу аналітичних інструментів системи керування хмарним середовищем є використання методів навчання ознак. Ці методи призначені для формування інформативного словника незалежних ознак вищого рівня абстрактності з відносно невисокою розмірністю, що значно спрощує синтез вирішальних правил.

Таким чином, розробка методу навчання ознак та вирішальних правил для прогнозування порушення умов SLA є актуальним напрямком дослідження, оскільки спрямований на підвищення ефективності системи керування ресурсами хмарного середовища.

2. Аналіз літературних даних і постановка проблеми

Основною проблемою сервіс-провайдерів хмарного середовища є визначення найкращого компромісу між прибутком і задоволенням користувача. Однак вирішення цієї проблеми ускладнено апріорною невизначеністю щодо функціонального стану сервісу внаслідок нестаціонарнарності попиту та гетерогенністю фізичних та віртуальних компонентів IT-інфраструктури. У працях [3, 4] було запропоновано застосування алгоритмів дерева рішень, випадкового лісу та Наївного Байєса для зняття невизначеності щодо дотримання умов SLA, пов’язаних з перевищенням часу відгуку сервісу, доступності сервісу чи зниженням інформаційної безпеки. Проте у запропонованих підходах контролювалася невелика кількість ознак, що не дозволило отримати високодостовірну прогностичну модель на випереджений проміжок часу, достатній для вживання необхідних заходів. У працях [4, 5] пропонується розглядати тренди використання ресурсів в рамках ковзного вікна заданого розміру для формування ознакового опису прогнозованих функціональних станів. В дослідженні [5] модель прогнозування будується на основі рекурентної нейронної мережі коротко-довгострокової пам’яті (Long Short-Term Memory, LSTM). Застосування такої мережі дозволило зменшити час відгуку сервісів, однак експеримент проводився на віртуальних симуляторах і на даних трасування обмеженого обсягу. При цьому мережа LSTM при розгортанні в часі є досить глибокою і потребує великих обсягів навчальних даних для уникнення ефекту перенавчання (overfitting), що робить модель неефективною протягом тривалого часу функціонування сервісу. У праці [6] розглядається прогнозування порушень SLA внаслідок перевантаження мережевих каналів в IT-інфраструктурі датацентру на основі глибокої моделі, яка так само потребує великого обсягу навчальних даних для уникнення збіжності до локального екстремуму функції втрат.

Розвиток ідеології автономних обчислень в хмарних системах обумовлює дослідження та впровадження технологій предиктивної аналітики на всіх рівнях інфокомунікаційної системи. Дані телеметрії, що накопичуються в системі

керування датацентром, характеризуються високою розмірністю, незбалансованістю обсягу зразків, що представляють функціональні стани сервісів, і відносно невеликим обсягом розмічених даних про порушення SLA, особливо на початку розгортання нових сервісів. У працях [7, 8] для аналізу даних високої розмірності з невеликим обсягом розмічених зразків пропонується використовувати навчання ознак без учителя на повному обсязі даних, а навчання класифікатора функціональних станів здійснювати на розмічених зразках закодованих навченими ознаками. У працях [9, 10] показано високу ефективність нейромережевих алгоритмів навчання ознак на основі стекування автоенкодерів та обмежених машин Больцмана. Однак такий підхід потребує дуже великих обсягів даних та обчислювальних ресурсів, що підвищує накладні витрати на аналіз даних і відтерміновує побудову ефективної моделі для прогнозування стану окремих сервісів. Тому активно досліджуються методи матричної факторизації для аналізу багатовимірних вибірок та стекування в багатошарову структуру на основі нелінійного перетворення та оператора пулінгу [8, 11]. До таких методів відносяться аналіз головних (Principal component analysis, PCA) і незалежних компонент (Independent component analysis, ICA) та факторизація невід'ємних матриць (Non-negative matrix factorization, NMF). При цьому у працях [11, 12] показано, що найефективнішою факторизацією є та, що забезпечує розріджене подання даних. Розріджене кодування дозволяє отримати завадозахищене стисле подання вхідних даних, де кожне спостереження може бути подане як лінійна комбінація невеликої кількості базисних векторів, що спрощує його інтерпретацію та подальший аналіз.

У працях [12, 13] було запропоновано алгоритм розріджено кодуючого нейронного газу, що дозволяє здійснювати інкрементальне навчання без учителя ознакового базису на основі принципів самоорганізації та ортогонального узгодженого переслідування (Orthogonal Matching Pursuit, OMP). При цьому алгоритм розріджено кодуючого нейронного газу є придатним для вибірок обмеженого обсягу. Запропонований алгоритм показав високу ефективність при аналізі зображень та зашумлених сигналів, однак досі не було досліджено його організацію у багатошарову структуру для спрощення аналізу багатовимірних спостережень слабоформалізованого процесу.

Найбільш ефективні методи машинного навчання за розміченими вибірками обмеженого обсягу основані на побудові в рамках геометричного підходу оптимальної роздільної гіперповерхні. У працях [10, 11] розглядається використання методу опорних векторів, що здійснює трансформацію простору для побудови роздільної гіперплощини, однак його застосування потребує обчислювально трудомісткої регуляризації моделі шляхом підбору ядер та коефіцієнта регуляризації. У дослідженнях [14, 15] пропонується метод трансформації простору первинних ознак за допомогою обчислювально ефективних операцій порівняння та "виключаючого АБО" для побудови роздільних "гіперсфер" (гіперпаралелепіпедів) у бінарному просторі вторинних ознак. При цьому бінарне кодування ознак та популяційний алгоритм оптимізації параметрів вирішальних правил за інформаційним критерієм

дозволяє автоматично формувати ефективну модель класифікатора, що обумовлює перспективність застосування підходу для аналізу даних моніторингу хмарних систем.

3. Ціль і задачі дослідження

Мета даної роботи полягає у підвищенні ефективності формування ознакового опису та вирішальних правил для прогнозування порушення умов SLA в хмарному датацентрі.

Для досягнення поставленої мети пропонується розв'язання таких задач:

– розробити метод навчання ієрархічного екстрактора ознак на основі ідей та методів нейронного газу та розрідженого кодування спостережень і порівняти його ефективність з автоенкодером;

– розробити алгоритми машинного навчання системи прогнозування порушення SLA з використанням двійкового кодування ознак та популяційної оптимізації параметрів вирішальних правил за інформаційним критерієм;

– дослідити залежність достовірності прогностичних рішень системи, що приймаються в робочому режимі за тестовими даними, від параметрів екстрактора ознак та вирішальних правил.

4. Алгоритми навчання ознак та вирішальних правил

Збір спостережень для навчання ознак відбувається шляхом сканування архівної історії зміни метрик продуктивності інфокомунікаційного сервісу вікном фіксованого розміру W, в межах якого з заданим кроком Δ в часі зчитуються їх значення. Для навчання вирішальних правил формується вибірка таких вікон з класифікованим станом сервісу у момент часу, що випереджає вікно на Δt кроків. При цьому розглядається два функціональні стани – клас X_1^o – нормальний стан функціонування, та X_2^o – порушення умов SLA.

Важливим кроком аналізу даних є попередня нормалізація з метою усунення лінійної кореляції компонент спостереження і уніфікації первинного ознакового подання. Відбілювання даних за методом ZCA (Zero-phase Component Analysis) є одним з найпоширеніших методів попередньої нормалізації даних. Метод ZCA полягає у виконанні наступних кроків:

1) обчислення середнього вибіркового значення ознак $\mu := mean(X)$;

2) обчислення коваріаційної матриці вибіркових спостережень $\Sigma := cov(X)$;

3) сингулярний розклад коваріаційної матриці $\Sigma \approx VDT^{\mathrm{T}}$;

4) відбілювання кожного спостереження за формулою

$$x_j := VD^{-1/2}V^T(x_j - \mu).$$

У загальному випадку навчання ознакового подання полягає у пошуку за нерозміченими даними набору параметрів, наприклад у вигляді множини базисних векторів C, які далі використовуються алгоритмом кодування для реконструкції розподілу вхідних даних. Для побудови словника базисних векторів C можна використати алгоритми векторного квантування, такі як k-

середніх чи нейронний газ. Нейронний газ, оснований на принципах "м'якої" конкуренції, тому характеризується кращою збіжністю, незалежністю від початкової точки пошуку та оптимальнішим розподілом векторів кодової книги. Формування ознакового подання можна здійснювати одним з методів розрідженої апроксимації, наприклад методом ортогонального узгодженого переслідування (OMP). Однак більш ефективним у сенсі мінімізації норми апроксимаційного залишку є метод оптимізованого ортогонального узгодженого переслідування (Optimized OMP, OOMP) [13]. Реалізація кодування в методі OOMP є ітераційною процедурою і включає такі основні кроки:

1) пошук *l*-го стовбця матриці сформованих базисних векторів C, що ще не був вибраний (не додано в множину *U*), з метою мінімізації норми отриманого залишку на поточному кроці:

$$l_{\min} := \arg\min_{l \notin U} \min_{a} \left\| x_j - C^U a \right\|_2^2;$$

2) оновлення множини вибраних базисних векторів

$$U := U \cup l_{win};$$

3) вирішення оптимізаційної задачі

$$a_j^{OMP} := \arg\min_{a} \left\| x_j - C^U a \right\|_2^2;$$

4) обчислення поточного залишку

$$\varepsilon_j := x_j - C a_j^{OMP};$$

5) перехід до кроку 1 доки не виконано *k*-ітерацій.

Для зменшення обчислювальної складності першого кроку можна використати популяційний алгоритм пошуку, або реалізацію, запропоновану в праці [12], де вводиться тимчасова матриця *R*, яка на початку рівна $R=(r_1,\ldots, r_l,\ldots, r_M)=C$ при $\varepsilon_j^U := x_j^U$, і на кожному кроці уточнюється за формулою

$$r_l := r_l - (r_{l_{win}}^T r_l) r_{l_{win}}, \tag{1}$$

де $r_{l_{win}}$ – стовпчик матриці *R*, що має максимальне перекриття з поточним залишком ε_j^U, індекс якого ще не додано до *U*, але визначено за формулою

$$l_{\text{win}} := \arg\max_{l, l \notin U} (r_l^T \varepsilon_i^U)^2. \tag{2}$$

Так само на кожній ітерації оновлюється значення залишку

$$\varepsilon_i^U := \varepsilon_i^U - (r_{l_{win}}^T \varepsilon_i^U) r_{l_{win}}. \quad (3)$$

Нейронний газ, що використовується для пошуку C, є алгоритмом самоорганізації неструктурованої сітки для виявлення топологічної структури даних. У загальному випадку алгоритм нейронного газу включає такі основні кроки:

1) ініціалізація словника $C=(c_1,\ldots, c_M)$ випадковими значеннями з рівномірного розподілу;

2) вибір t-го вхідного спостереження x з множини X, що має обсяг $t_{\max}$;

3) обчислення коефіцієнтів розміру околу сусідства та швидкості навчання за формулами:

$$\lambda_t := \lambda_0 (\lambda_{\text{final}} / \lambda_0)^{t/t_{\max}}, \quad (4)$$

$$\alpha_t := \alpha_0 (\alpha_{\text{final}} / \alpha_0)^{t/t_{\max}}, \quad (5)$$

де λ_0, λ_{final} – початкове та кінцеве значення коефіцієнту λ_t; α_0, α_{final} – початкове та кінцеве значення коефіцієнту α_t;

4) розрахунок відстані вхідного вектора x до слів кодової книги C та впорядкування їх за зростанням

$$\left\| x - c_{l_0} \right\| \le \ldots \le \left\| x - c_{l_k} \right\| \le \ldots \le \left\| x - c_{l_{M-1}} \right\|;$$

5) виконання M–1 ітерацій оновлення кодових слів за формулою

$$c_{l_k} := c_{l_k} + \alpha_t e^{-k/\lambda_t} (x - c_{l_k}), \; k = \overline{1, M-1};$$

6) перехід до кроку 2, якщо $t<t_{\max}$.

Для адаптації алгоритму векторного квантування до обраної схеми кодування з метою зменшення помилки розрідженої апроксимації пропонується використати модифікацію алгоритму, досліджену в працях [13]. Модифікований алгоритм нейронного газу для розрідженого кодування спостережень складається з таких кроків:

1) ініціалізація словника $C=(c_1,\ldots, c_M)$ випадковими значеннями з рівномірного розподілу;

2) вибір t-го вхідного спостереження x з множини X, яка має обсяг t_{max};

3) нормалізація базисних векторів $c_1,\ldots, c_M$ шляхом їх приведення до одиничного розміру (unit vector);

4) обчислення коефіцієнтів розміру околу сусідства та швидкості навчання за формулами (4) та (5);

5) ініціалізація множини індексів тих стовпців C, які вже були використані протягом t-ітерацій $U = \varnothing$;

6) ініціалізація залишку, що мінімізується, $\varepsilon^U = x$;

7) ініціалізація тимчасової матриці $R=(r_1,\ldots, r_l,\ldots, r_M)=C$, ортонормованої відповідно до C^U;

8) ініціалізація лічильника кроків уточнення залишків $h := 1,\ h = \overline{1, K-1}$;

9) розрахунок відстані (скалярного добутку) вектора r_{l_k} до ε^U та впорядкування їх за зростанням

$$-(r_{l_0}^T \varepsilon^U)^2 \le \ldots \le -(r_{l_k}^T \varepsilon^U)^2 \le \ldots \le -(r_{l_{M-h-1}}^T \varepsilon^U)^2.$$

10) ініціалізація лічильника кроків уточнення кодової книги C,

$$k = 0,\ k = \overline{0, M-h-1};$$

11) оновлення на k-му кроці слів кодової книги з використанням принципів ортогональності до підпростору заданого в C^U і правила Ойа [11]

$$c_{l_k} := c_{l_k} + \Delta_{lk,},$$

$$r_{l_k} := r_{l_k} + \Delta_{lk,},$$

де

$$\Delta_{lk} := \alpha_t \exp(-k / \lambda_t) y(\varepsilon^U - y r_{l_k}),$$

де

$$y := r_{l_k}^T \varepsilon^U;$$

12) нормалізація r_{l_k} шляхом приведення до одиничного розміру (unit vector);

13) якщо $k < 0, M-h-1$ перехід до кроку 11;

14) визначення базису переможця за формулою (2);

15) оновлення матриці R та поточного залишку ε_i^U за формулами (1) та (3);

16) оновлення матриці обраних базисних векторів $U = U \cup l_{win}$;

17) якщо $h<K-1$, то перехід до кроку 11;

18) якщо $t<t_{max}$, то перехід до кроку 2, інакше – закінчення обробки.

Перший шар екстрактора ознак може здійснювати аналіз вхідних сигналів з декількох часових вікон, що перетинаються в часі. Після навчання першого шару екстрактора ознак можна перекодувати всю навчальну вибірку у

розріджене конкатеноване подання і використати його для навчання наступного шару. Перед цим доцільно внести нелінійність в отримане подання та зменшити кількість базисних векторів в новому шарі [11]. Найпростішою нелінійністю є обмеження у вигляді умови невід'ємності ознак, при якому вихід *S*-го шару o_s з розрідженим кодом a_S^{OOMP} можна обчислити за формулою

$$o_S(a_S^{OOMP}) := [\max(0, a_S^{OOMP}), \max(0, a_S^{OOMP})], \quad (6)$$

де 0 – вектор з нульовими компонентами, розмірністю M; max – оператор поелеметного максимуму між двома векторами.

Застосування запропонованої нелінійності (6) збільшує розмірність результуючого коду вдвічі $o_S \in R^{2*M}$, але підвищує інформативність за рахунок можливості роздільного аналізу від'ємних та додатніх відгуків сигналу і зберігає властивість розрідженості. Таким чином, ненульові значення ознакового подання вищого рівня сигналізуватимуть про активацію певної групи низькорівневих ознак. При цьому на класифікатор можна подавати як вихід останнього шару екстрактора ознак, так і виходи нижніх шарів, що дозволятиме здійснювати класифікаційний аналіз з урахуванням специфіки функціонального стану на кожному рівні абстракції.

Алгоритм грубого двійкового кодування вектору ознак для класифікаційного аналізу полягає у порівнянні значення *i*-ї ознаки з відповідним нижнім $A_{B,l,i}$ та верхнім $A_{T,l,i}$ межами несиметричного поля контрольних допусків, які розраховуються за формулами

$$A_{B,l,i} = y_{i,\max}\left[1 - \frac{\delta_{l,i}}{\delta_{\max}}\right],$$

$$A_{T,l,i} = y_{i,\max} \text{ при } l = \overline{1,L},$$

де $\delta_{l,i}$ – параметр *l*-го поля контрольних допусків на значення *i*-ї ознаки.

Формування бінарної навчальної матриці

$$\{x_{k,i}^{(j)} \mid i = \overline{1, L \cdot N};\ j = \overline{1, n_k};\ k = \overline{1, K}\},$$

де *N* – кількість ознак класифікатора, n_m – кількість векторів класу X_m^o та *K* – кількість класів розпізнавання, здійснюється за правилом

$$x_{k,(l-1)*N+i}^{(j)} = \begin{cases} 1, \ if \quad A_{B,l,i} \le y_{k,i}^{(j)} \le A_{U,l,i}; \\ 0, \ \text{else.} \end{cases}$$

Обчислення значень координат двійкового усередненого вектора x_k, відносно якого відбувається побудова в радіальному базисі контейнерів класів, здійснюється за правилом

$$x_{k,i} = \begin{cases} 1, & \text{if} \quad \dfrac{1}{n_k}\sum_{j=1}^{n_k} x_{k,i}^{(j)} > \dfrac{1}{n}\sum_{k=1}^{K}\sum_{j=1}^{n_k} x_{k,i}^{(j)}; \quad \mathrm{i} = \overline{1,\mathrm{N}\cdot\mathrm{L}}, \\ 0, & \text{if} \quad \text{else}; \end{cases}$$

де n – повний обсяг розмічених векторів начальної вибірки.

Як критерій ефективності машинного навчання класифікатора розпізнавати реалізації класу X_k^o розглядається модифікація інформаційної міри Кульбака [14, 15]:

$$J_k = \frac{1-(\alpha_k+\beta_k)}{\log_2(2+\varepsilon)-\log_2\varepsilon}\cdot\log_2\left[\frac{2-(\alpha_k+\beta_k)+\varepsilon}{(\alpha_k+\beta_k)+\varepsilon}\right], \tag{7}$$

де α_k, β_k – оцінки помилок першого та другого роду, які задають робочу область критерію у вигляді нерівностей $\alpha_k \geq 0{,}5$ та $\beta_k \geq 0{,}5$; ε – мале знакододатнє число для уникнення невизначеності при діленні на нуль, рівне, як правило, числу з діапазону $[10^{-4}...10^{-2}]$.

Оптимізація параметрів поля контрольних допусків $\{\delta_{l,i}\}$ полягає в пошуку екстремуму функції КФЕ (7) в гіперпросторі рішень. При цьому як пошуковий алгоритм в даній роботі пропонується використати рій частинок (Particle Swarm Optimization, PSO), який характеризується простотою реалізації та інтерпретабельністю [16]. Оптимізація радіусів контейнерів класів може здійснюватися методом послідовного прямого перебору з заданим кроком, оскільки кількість кроків такого пошуку є відносно малою.

Для підвищення компактності образів та міжкласового зазору в двійковому просторі вторинних ознак алгоритм машинного навчання враховує нечітку компактність образів, що обчислюється для класу X_k^o за формулою

$$L_k = \frac{d(x_k \oplus x_c)}{d_k + d_c}, \tag{8}$$

де d_k, d_c – радіуси контейнера класу X_k^o та найближчого сусіднього до нього класу X_c^o відповідно; $d(x_k \otimes x_c)$ – кодова відстань між центрами контейнерів класів та X_c^o, що розраховується за формулою

$$d(x_k \oplus x_c) = \sum_{i=1}^{N}(x_{k,i} \oplus x_{c,i}).$$

Ефективність кожної частинки популяційного алгоритму, тобто близькість до глобального оптимуму, вимірюється за допомогою наперед визначеної фітнес-функції, роль якої в даному випадку виконує функція критерію ефективності машинного навчання (7). Кожна *j*-та частинка крім її позиції P_j зберігає наступну інформацію: V_j – поточна швидкість частинки, $Pbest_j$ – краща персональна позиція частинки. Краща персональна позиція *j*-ї частинки – це позиція *j*-ї частинки, у якій значення фітнес функції для частинки було максимальним на поточний момент часу. Крім цього, з метою пошуку глобального екстремуму фітнес-функції найкраща частинка шукається в усьому рої, а позиція позначається як *Gbest*.

Проте розглянутий вище ройовий алгоритм пошуку спрямований на підвищення усередненого за алфавітом класів значення критерію ефективності навчання. З метою додаткового підвищення компактності образів слід модифікувати процедуру оновлення значень найкращої персональної $Pbest_j$ позиції агентів пошуку за правилом (9), в якому цільова функція $E(\ldots)$ є усередненим значенням функції критерію (7).

$$\text{якщо } \left| E(P_j) - E(Pbest_j) \right| < \varepsilon$$
$$\text{та } \bar{L}(P_j) > \bar{L}(Pbest_j), \text{ то } Pbest_j := P_j. \qquad (9)$$

Аналогічно потрібно модифікувати процедуру оновлення значень найкращої глобальної $Gbest_j$ позиції агентів пошуку

$$\text{якщо } \left| E(Pbest_j) - E(Gbest) \right| < \varepsilon$$
$$\text{та } \bar{L}(Pbest_j) > \bar{L}(Gbest_j), \text{ то } Gbest := Pbest_j. \qquad (10)$$

У режимі екзамену рішення про належність вектора-реалізації *x* одному з класів алфавіту $\{X_k^o\}$ приймається шляхом обчислення геометричної функції належності

$$\mu_k^*(x) = \max_{\{k\}}\{\mu_k(x)\},$$

в якій $\mu_k(x)$ являє собою функцію належності вектора *x* до контейнера класу X_k^o, яка обчислюється за правилом:

$$\mu_k(x) = 1 - \frac{d(x_k^* \oplus x)}{d_k^*}.$$

Для точнішого врахування розподілу двійкових векторів в гіпер-сферичному контейнері класу X_k^o формула функції належності може бути скорегована і матиме вигляд

$$\mu_k(x) = \begin{cases} 1 - \dfrac{d(x_k^* \oplus x)}{d_k^*}, \ if \ d(x_k^* \oplus x) > d_k^*; \\ \dfrac{n_k(d)}{n_{\max}}, \ \text{else}, \end{cases} \tag{11}$$

де $n_k(d)$ – кількість векторів класу X_k^o, що знаходяться на відстані d від центру x_k; $n_{\max}$ – максимальне значення у масиві $n_k(d)$, тобто $n_{\max} = \max\limits_d \{n_k(d)\}$.

Таким чином, запропоновані алгоритми навчання ознак та вирішальних правил для прогнозування умов порушення SLA є невибагливими до обсягу даних та ресурсів обчислювальної машини, що забезпечує ефективність керування ресурсами на ранніх етапах функціонування сервісу.

5. Результати фізичного моделювання системи прогнозування порушення умов SLA

Перевірка ефективності запропонованих алгоритмів розглядається на прикладі задачі прогнозування перевантаження серверів датацентру, які призводять до порушення SLA в метриках доступності, ємності ресурсу та часу відгуку. Моделювання здійснювалося з використанням фреймворку Clouds [16], де було задано 400 серверів HP ProLiant ML110 G4 (Intel Xeon 3040, 2 ядра×1860 МГц, 4 Гб) та 400 серверів HP ProLiant ML110 G5 (Intel Xeon 3075, 2 ядра ×2660 МГц, 4 Гб). Дані робочого навантаження, зібрані на платформі PlanetLab, було взято з проекту CoMon [16]. Горизонт прогнозу складає 10 хвилин, що достататньо для здійснення міграції віртуальної машини. Архітектура системи прогнозування порушень SLA показана на рис. 1 і містить дворівневий екстрактор ознак. Екстрактор аналізує дані моніторингу заванатаженості процесорного ресурсу віртуальної машини у двох зміщених в часі часових підвікнах з перетином 50 % і кроком зчитування рівним 1 хвилині. Довжина підвікна перевищує в декілька разів горизонт прогнозу і становить 50 хвилин, що обрано на наш розсуд і може бути не оптимальним.

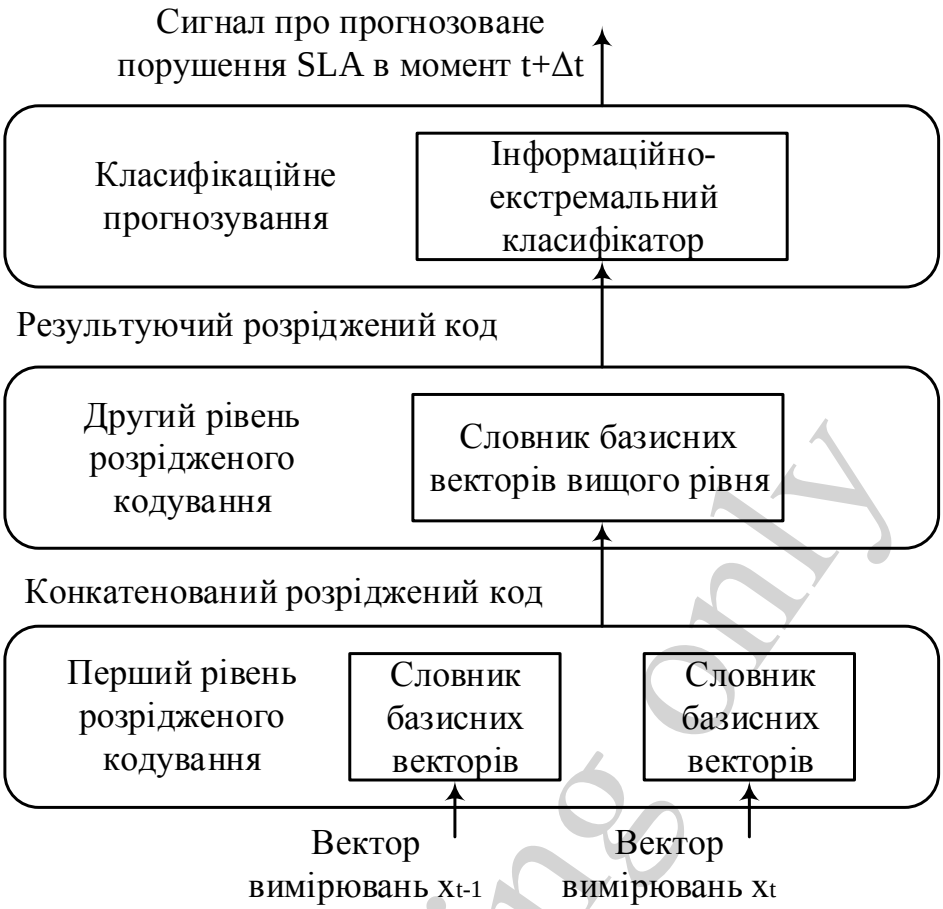

Рис. 1. Структурна схема системи класифікаційного прогнозування порушення умов SLA

Вибірка нерозмічених зразків для навчання дворівневого екстрактора ознак становить 10000 зразків, а обсяг апріорно класифікованої навчальної вибірки кожного з двох класів становить 100 зразків. Тестова вибірка класифікатора має такий же обсяг як і навчальна. У табл. 1 показано результати машинного навчання при різній потужності словника базисних векторів першого та другого рівнів.

Таблиця 1

Результати машинного навчання класифікатора при різних конфігураціях екстрактора ознак

№ з/п	Потужність словника базисних векторів першого рівня	Потужність словника базисних векторів другого рівня	Значення усередненого інформаційного критерію (7)	Точність класифікатора за тестовою вибіркою
1	20	5	0,112	0,8
2	20	10	0,118	0,81
3	20	15	0,118	0,82
4	30	10	0,251	0,91

5	30	15	0,751	0,99
6	30	20	1,000	1,00
7	40	15	1,000	1,00
8	40	20	1,000	1,00
9	40	25	1,000	1,00

Аналіз табл. 1 показує, що найкращою з перевірених конфігурацій екстрактора ознак є 6-та конфігурація, яка забезпечує безпомилкові за тестовою вибіркою вирішальні правила при мінімальній кількості базисних векторів. У табл. 2 показано результати машинного навчання класифікатора з 6-тою конфігурацією екстрактора ознак при різних кількостях контрольних допусків на ознаки розпізнавання.

Таблиця 2

Результати машинного навчання класифікатора при різній кількості контрольних допусків на значення високорівневих ознак

№ з/п	Кількість контрольних допусків на ознаки	Значення усередненого інформаційного критерію	Точність класифікатора за тестовою вибіркою
1	1	0,51	0,98
2	2	0,75	0,99
3	3	1,000	1,00
4	4	1,000	1,00
5	5	1,000	1,00
6	6	1,000	1,00
7	7	1,000	0,99

Аналіз табл. 2 показує, що оптимальна кількість контрольних допусків на значення ознак становить L=3, а подальше нарощування кількості доспусків може призвести до перенавчання, як це видно з таблиці при L=7. При цьому графіки зміни точності отриманих вирішальних правил за навчальною та тестовою вибірками від кількості навчальних векторів екстрактора ознак, показано на рис. 2.

Аналіз рис. 2 показує, що збільшення кількості навчальних векторів екстрактора призводить до підвищення точності за навчальною і тестовою вибірками для класифікатора функціональних станів сервісу. Однак при обсязі навчальної вибірки біля 5000 зразків спостерігається ефект перенавчання системи, а після досягнення 6100 зразків вдається отримати екстрактор, що забезпечує безпомилковість вирішальних правил за тестовою вибіркою.

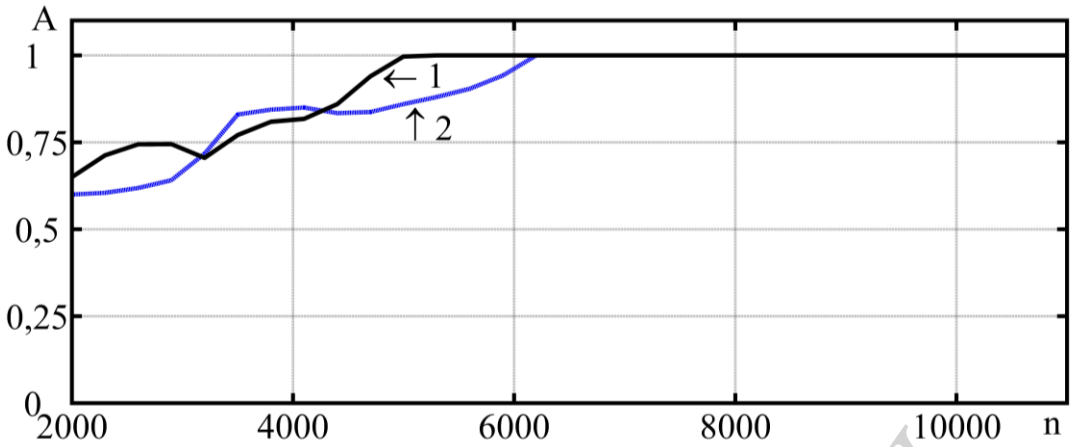

Рис. 2. Графіки залежності ефективності вирішальних правил від кількості навчальних векторів екстрактора ознак: 1 – крива зміни точності за навчальною вибіркою; 2 – крива зміни точності за тестовою вибіркою

Таким чином, розроблений алгоритм навчання ознак та вирішальних правил дозволяє отримати безпомилкові за тестовою вибіркою вирішальні правила з екстрактором, що містить 30 базисних векторів у першому шарі та 20 векторів – у другому. При цьому для навчання екстрактора достатньо 6100 навчальних зразків.

6. Обговорення результатів фізичного моделювання процесу машинного навчання

Використання запропонованого екстрактора та модифікації за правилами (9) та (10) ройового алгоритму оптимізації вирішальних правил, як видно з рис. 2, дозволяє отримати високодостовірні вирішальні правила. При цьому на графіку присутня ділянка перенавчання, шириною в 1100 зразків, в кінці якої точність за тестовою вибіркою досягає граничного максимального значення. Ефект перенавчання має складову як від екстрактора, так і від класифікатора. Щоб оцінити вплив використання правил (9) та (10) на ефект перенавчання, на рис. 3 показано графіки зміни точності отриманих вирішальних правил за навчальною та тестовою вибірками від кількості навчальних векторів екстрактора без використання даних правил.

Як видно з рис. 3, без урахування компактності образів за правилами (9) та (10) для отримання високодостовірних вирішальних правил потрібно використовувати навчальну вибірку набагато більшого обсягу, який в даному випадку становить 8500 зразків.

Для порівняння узагальнюючої здатності запропонованого екстрактора з популярним екстрактором на основі глибокого автоенкодера [9] на рис. 4 показано графіки зміни точності отриманих вирішальних правил за навчальною та тестовою вибірками від кількості навчальних векторів автоенкодера. При цьому автоенкодер має таку конфігурацію : розмірність входу – 75 ознак; кількість вузлів першого прихованого шару – 30; кількість вузлів в прихованому шарі, що відповідає ознаковому поданню – 20.

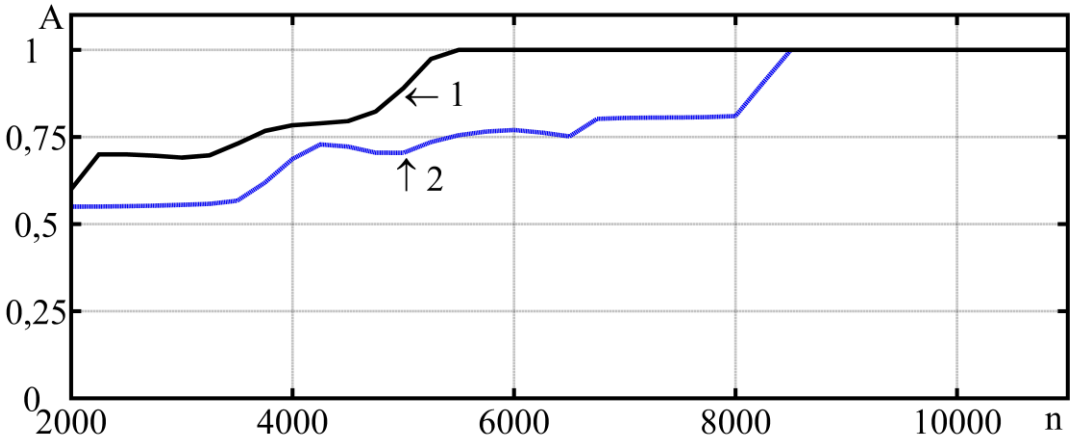

Рис. 3. Графіки залежності ефективності вирішальних правил від кількості навчальних векторів екстрактора ознак без використання правил (9) та (10): 1 – крива зміни точності за навчальною вибіркою; 2 – крива зміни точності за тестовою вибіркою

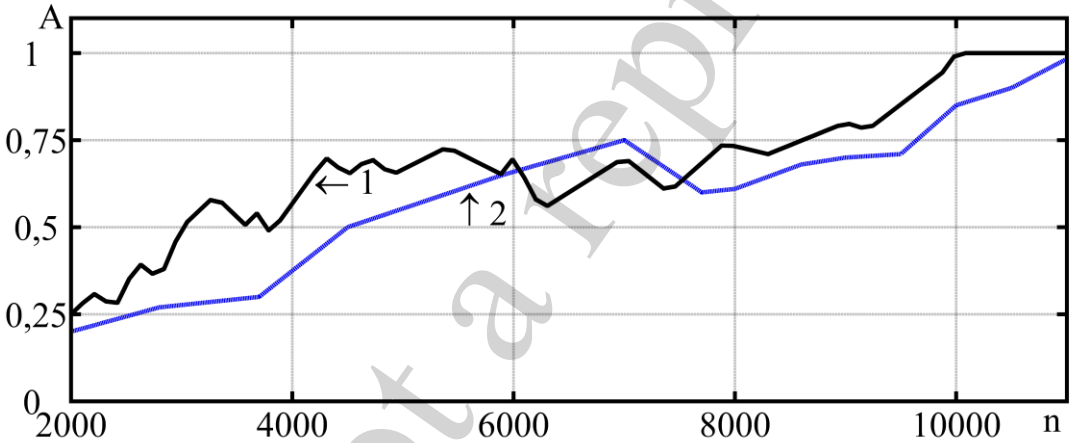

Рис. 4. Графіки залежності ефективності вирішальних правил від кількості навчальних векторів автоенкодера : 1 – крива зміни точності за навчальною вибіркою; 2 – крива зміни точності за тестовою вибіркою

Аналіз рис. 4 показує, що глибокий автоенкодер так само дозволяє отримати безпомилкові за тестовою вибіркою вирішальні правила, але для цього потрібно більше навчальних зразків, обсяг яких перевищує 10 000.

Таким чином, розроблені інформаційне та алгоритмічне забезпечення дозволяють отримати високодостовірні вирішальні правила для прогнозування порушення умов SLA. При цьому реалізовані алгоритми порівняно з автоенкодером потребують меншого обсягу навчальних даних, що дозволяє раніше вводити в дію предиктивні механізми керування відповідними сервісами.

7. Висновки

1. Доведено за результатами фізичного моделювання здатність як запропонованого ієрархічного екстрактора ознак, побудованого на ідеях і методах нейронного газу та розрідженого кодування, так і автоенкодера для отримання безпомилкових за навчальною та тестовою вибірками вирішальних правил. Проте запропонований екстрактор на відміну від аутоенкодера потребує приблизно в 1,6 разів менший обсяг навчальних зразків для досягнення того ж результату, що дозволяє раніше вводити в дію предиктивні механізми керування відповідними хмарними сервісами.

2. Показано, що врахування компактності образів в двійковому просторі вторинних ознак при оптимізації багаторівневої системи контрольних допусків на значення первинних ознак дозволяє значно зменшити негативний ефект перенавчання класифікатора та вимоги до обсягу навчальних зразків.

3. Показано, що запропонована конфігурація екстрактора для задачі прогнозування порушення умов SLA є прийнятною з точки зору точності та складності. При цьому на вході екстрактора використовується два часових вікна, які перетинаються в часі на 50 % і зчитують по 50 ознак. Перший шар кодування екстрактора містить 30 базисних векторів, а другий шар – 20. При цьому міжшаровий пулінг та нелінійність були утворені шляхом конкатенації розріджених кодів кожного з вікон та подовження результуючого коду вдвічі з метою розділення додатніх та від'ємних компонентів коду і перетворення результуючого коду у вектор знакододатніх ознак.

Література

1. Reyhane, A. H. SLA Violation Prediction In Cloud Computing: A Machine Learning Perspective [Electronic resource] / A. H. Reyhane, H. Abdelhakim // arXiv. – 2016. – Available at: https://arxiv.org/pdf/1611.10338.pdf

2. Minarolli, D. Tackling uncertainty in long-term predictions for host overload and underload detection in cloud computing [Text] / D. Minarolli, A. Mazrekaj, B. Freisleben // Journal of Cloud Computing. – 2017. – Vol. 6, Issue 1. doi: 10.1186/s13677-017-0074-3

3. Wajahat, M. Using machine learning for black-box autoscaling [Text] / M. Wajahat, A. Gandhi, A. Karve, A. Kochut // 2016 Seventh International Green and Sustainable Computing Conference (IGSC). – 2016. doi: 10.1109/igcc.2016.7892598

4. Meskini, A. Proactive Learning from SLA Violation in Cloud Service based Application [Text] / A. Meskini, Y. Taher, A. El gammal, B. Finance, Y. Slimani // Proceedings of the 6th International Conference on Cloud Computing and Services Science. – 2016. doi: 10.5220/0005807801860193

5. Ashraf, A. Automatic Cloud Resource Scaling Algorithm based on Long Short-Term Memory Recurrent Neural Network [Text] / A. Ashraf // International Journal of Advanced Computer Science and Applications. – 2016. – Vol. 7, Issue 12. doi: 10.14569/ijacsa.2016.071236

6. Gupta, L. Fault and Performance Management in Multi-Cloud Based NFV using Shallow and Deep Predictive Structures [Text] / L. Gupta, M. Samaka,

R. Jain, A. Erbad, D. Bhamare, H. A. Chan // 7th Workshop on Industrial Internet of Things Communication Networks at The 26th International Conference on Computer Communications and Networks (ICCCN 2017). – Vancluver, 2017.

7. Tarsa, S. J. Workload prediction for adaptive power scaling using deep learning [Text] / S. J. Tarsa, A. P. Kumar, H. T. Kung // 2014 IEEE International Conference on IC Design & Technology. – 2014. doi: 10.1109/icicdt.2014.6838580

8. Flenner, J. A Deep Non-Negative Matrix Factorization Neural Network [Electronic resource] / J. Flenner, B. Hunter // Available at: http://www1.cmc.edu/pages/faculty/BHunter/papers/deepNMF.pdf

9. Li, Y. Learning-based power prediction for data centre operations via deep neural networks [Text] / Y. Li, H. Hu, Y. Wen, J. Zhang // Proceedings of the 5th International Workshop on Energy Efficient Data Centres – E2DC '16. – 2016. doi: 10.1145/2940679.2940685

10. Zhao, Z. Stacked Multilayer Self-Organizing Map for Background Modeling [Text] / Z. Zhao, X. Zhang, Y. Fang // IEEE Transactions on Image Processing. – 2015. – Vol. 24, Issue 9. – P. 2841–2850. doi: 10.1109/tip.2015.2427519

11. Chan, T.-H. PCANet: A Simple Deep Learning Baseline for Image Classification [Electronic resource] / T.-H. Chan, K. Jia, S. Gao, J. Lu et. al. // arXiv. – 2014. – Available at: https://arxiv.org/pdf/1404.3606.pdf

12. Labusch, K. Learning Data Representations with Sparse Coding Neural Gas [Text] / K. Labusch, E. Barth, T. Martinetz // Proceedings of the European Symposium on Artificial Neural Networks. – Bruges, 2008. – P. 233–238.

13. Labusch, K. Sparse Coding Neural Gas: Learning of overcomplete data representations [Text] / K. Labusch, E. Barth, T. Martinetz // Neurocomputing. – 2009. – Vol. 72, Issue 7-9. – P. 1547–1555. doi: 10.1016/j.neucom.2008.11.027

14. Moskalenko, V. Optimizing the parameters of functioning of the system of management of data center it infrastructure [Text] / V. Moskalenko, S. Pimonenko // Eastern-European Journal of Enterprise Technologies. – 2016. – Vol. 5, Issue 2 (83). – P. 21–29. doi: 10.15587/1729-4061.2016.79231

15. Dovbysh, A. S. Information-Extreme Method for Classification of Observations with Categorical Attributes [Text] / A. S. Dovbysh, V. V. Moskalenko, A. S. Rizhova // Cybernetics and Systems Analysis. – 2016. – Vol. 52, Issue 2. – P. 224–231. doi: 10.1007/s10559-016-9818-1

16. Mosa, A. Optimizing virtual machine placement for energy and SLA in clouds using utility functions [Text] / A. Mosa, N. W. Paton // Journal of Cloud Computing. – 2016. – Vol. 5, Issue 1. doi: 10.1186/s13677-016-0067-7