УДК 004.9

Ф. Б. Гуральник

ПІДХІД ДО ПОБУДОВИ ПРОСТОРУ СТАНІВ ПРОГРАМНОЇ СИСТЕМИ НА ОСНОВІ ЛОГІВ ДЛЯ ВИЯВЛЕННЯ АНОМАЛІЙ МЕТОДАМИ КЛАСТЕРИЗАЦІЇ

Вінницький національний технічний університет

Анотація

У роботі запропоновано підхід до аналізу поведінки програмної системи на основі побудови простору її станів за даними логів виконання. На відміну від традиційних методів, що ґрунтуються на аналізі окремих подій або використанні заздалегідь визначених правил, запропонований підхід дозволяє розглядати функціонування системи як цілісний динамічний процес.

Формування простору станів здійснюється шляхом перетворення логів у вектори ознак, що враховують як частотні, так і часові характеристики виконання. Для виявлення аномальних режимів функціонування застосовано методи кластеризації, що дозволяють ідентифікувати типові стани системи та виявляти відхилення від них. Запропонований підхід забезпечує підвищення ефективності автоматизованої діагностики програмних систем.

Ключові слова: логування, простір станів, кластеризація, аномалії, аналіз поведінки, програмні системи.

Abstracts

The paper proposes an approach to analyzing the behavior of a software system based on constructing its state space using execution logs. Unlike traditional methods that rely on the analysis of individual events or predefined rules, the proposed approach enables the system to be considered as a holistic dynamic process.

The state space is formed by transforming logs into feature vectors that capture both frequency-based and temporal characteristics of execution. Clustering methods are applied to detect anomalous operating modes, allowing the identification of typical system states as well as deviations from them. The proposed approach improves the efficiency of automated diagnostics of software systems.

Keywords: logging, state space, clustering, anomalies, behavior analysis, software systems.

Вступ

Сучасні програмні системи характеризуються високим рівнем складності, розподіленістю та динамічністю функціонування, що ускладнює процеси їх моніторингу та діагностики. Існуючі підходи до аналізу логів переважно орієнтовані на обробку окремих подій або використання правил, що обмежує можливості виявлення складних та неявних аномалій. Логи виконання містять значний обсяг інформації про поведінку системи, включаючи часові залежності, послідовності подій та взаємозв’язки між компонентами. Це дозволяє розглядати функціонування програмної системи як процес переходів між станами, які можуть бути формалізовані у вигляді простору ознак.

Об'єктом дослідження процес функціонування програмної системи під час її виконання.

Предметом дослідження методи побудови простору станів програмної системи на основі логів виконання та методи виявлення аномалій у цьому просторі.

Метою роботи Розробка підходу до формалізації поведінки програмної системи у вигляді простору станів та застосування методів кластеризації для виявлення аномальних режимів її функціонування.

Наукова новизна та практичне значення

Наукова новизна отриманих результатів полягає у наступному:

- **вперше** запропоновано підхід до представлення поведінки програмної системи у вигляді простору станів, сформованого на основі логів виконання, що дозволяє враховувати структурні та динамічні характеристики її функціонування;
- **удосконалено** метод виявлення аномалій шляхом застосування кластеризації у просторі ознак, що забезпечує виявлення як явних, так і прихованих відхилень від нормальних режимів роботи;
- **набув подальшого розвитку** підхід до аналізу поведінки програмних систем як послідовності станів та переходів між ними.

Практичне значення отриманих результатів полягає у можливості застосування запропонованого підходу для автоматизованої діагностики програмних систем та підвищення ефективності аналізу їх функціонування.

Матеріали та результати дослідження

У роботі запропоновано формалізацію логів виконання програмної системи у вигляді множини подій:

$$L = \{l_1, l_2, \ldots, l_n\},$$
де кожен запис описується як $l_i = (t_i, e_i, p_i)$,
де t_i — час події, e_i — тип події, p_i — параметри.

У даному підході лог розглядається не як неструктурований текст, а як впорядкована послідовність подій, що формує часовий ряд та відображає процес функціонування програмної системи. Така інтерпретація дозволяє перейти від синтаксичного аналізу логів до семантичного аналізу поведінки системи.

З метою аналізу поведінки системи у часі введено дискретизацію виконання з кроком Δt, у межах якого формується агрегований стан системи S_k. Формально це можна подати як відображення[1]:

$$S\_k = f(L, \Delta t),$$

де функція f виконує агрегацію подій у межах інтервалу часу. Такий підхід дозволяє зменшити розмірність даних та виділити узагальнені характеристики поведінки.

Таким чином, виконання системи подається у вигляді послідовності станів:

$$S = \{S_1, S_2, \ldots, S\_k\},$$

що відображає її поведінку як траєкторію у просторі ознак. З теоретичної точки зору, така модель може розглядатися як дискретна динамічна система, у якій стан змінюється у часі під впливом внутрішніх та зовнішніх факторів.

Кожен стан представлено у вигляді вектора[2]:

$$S\_k = (x_1, x_2, \ldots, x\_m),$$

де ознаки формуються на основі логів та включають кількість подій, середній час відповіді, частоту помилок, інтенсивність навантаження та інші характеристики. Вибір ознак визначає якість представлення системи та впливає на здатність моделі відображати реальну поведінку.

Сформований простір ознак можна інтерпретувати як метричний простір $X \subset \mathbb{R}^m$, у якому вводиться міра відстані між станами, наприклад евклідова метрика. Це дозволяє оцінювати ступінь подібності між різними станами системи.

У сформованому просторі ознак застосовано методи кластеризації (k-means, DBSCAN), які дозволяють виділити групи подібних станів. Теоретично передбачається, що нормальні режими функціонування системи відповідають областям підвищеної щільності у просторі ознак та формують компактні та стійкі кластери. Натомість аномальні стани характеризуються значним відхиленням від центрів кластерів, низькою щільністю або належністю до розріджених областей простору.

З позиції теорії розпізнавання образів, задача виявлення аномалій зводиться до задачі виявлення точок, що не належать жодному з класів нормальної поведінки або мають низьку ймовірність належності до них.

Додатково враховується аналіз переходів між станами, що дозволяє виявляти аномалії як нетипові зміни поведінки системи. У цьому випадку послідовність станів може розглядатися як марковський процес, де ймовірності переходів між станами дозволяють оцінити типовість поведінки. Аномалії проявляються як переходи з низькою ймовірністю або як порушення характерних траєкторій.[4]

Отримані результати показують, що запропонований підхід забезпечує ефективне виявлення аномальних режимів функціонування навіть у випадках відсутності попередньо визначених правил або розмічених даних, що робить його придатним для використання у складних та динамічних програмних системах.

Висновки

У роботі розроблено підхід до побудови простору станів програмної системи на основі логів її виконання, що дозволяє перейти від аналізу окремих подій до аналізу цілісної поведінки системи. Застосування методів кластеризації у сформованому просторі ознак забезпечує можливість виявлення нормальних режимів функціонування та аномальних відхилень без використання жорстко заданих правил. Отримані результати підтверджують ефективність запропонованого підходу для автоматизації процесів діагностики та аналізу складних програмних систем.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Кветний Р. Н., Бойко О. В. Аналіз даних та машинне навчання в інформаційних системах. – Вінниця: ВНТУ, 2021. – 210 с.
2. Кветний Р. Н., Шаховська Н. Б. Інтелектуальні інформаційні системи: методи та технології. – Львів: Видавництво Львівської політехніки, 2020. – 356 с.
3. Савченко В. П., Дорошенко А. Ю. Методи обробки часових рядів у задачах діагностики систем. – Київ, 2021. – 250 с.
4. Xu W., Huang L., Fox A., Patterson D., Jordan M. Detecting Large-Scale System Problems by Mining Console Logs // SOSP. – 2009. 600p.
5. Aggarwal C. C. Outlier Analysis. – Springer, 2017, 350p.
6. He S., Zhu J., He P., Lyu M. Experience Report: System Log Analysis for Anomaly Detection // IEEE Software. – 2016., 430p
7. Nedelkoski S., Cardoso J., Kao O. Anomaly Detection from System Logs Based on Deep Learning: A Survey // Future Internet. – 2020., 850p

Гуральник Фредерік Борисович — студент групи 126-23а, факультет інтелектуальних інформаційних технологій та автоматизації, Вінницький національний технічний університет, м.Вінниця. e-mail: frederikguralnik@gmail.com

Huralnyk Frederik B. — student of Faculty of Intelligent Information Technology and Automation, 126-23a, Vinnytsia National Technical University, Vinnytsia. e-mail: frederikguralnik@gmail.com