



A. I. Пукач, В. М. Теслюк

Національний університет «Львівська Політехніка», м. Львів, Україна

ORCID: <https://orcid.org/0000-0002-5974-9310> – В. М. Теслюк



vasyl.teslyuk@gmail.com

ІНФОРМАЦІЙНА МОДЕЛЬ АНАЛІЗУ ФАКТОРІВ ВПЛИВУ АВТОМАТИЗАЦІЇ ПІДТРИМКИ ПРОГРАМНИХ КОМПЛЕКСІВ З ВИКОРИСТАННЯМ СЕРЕДОВИЩ R ТА PYTHON

Анотація. Основна проблема полягає в необхідності розроблення ефективної та дієвої інформаційної моделі аналізу факторів впливу автоматизації підтримки програмних комплексів. Метою дослідження є пошук оптимальних рішень для розроблення необхідної інформаційної моделі з можливістю її подальшої реалізації, застосування, тестування, аналізу та опрацювання отриманих результатів при вирішенні прикладних практичних задач. Серед основних методів дослідження – методи комп'ютерного моделювання з використанням середовищ R та Python. Основними результатами дослідження є наступні. Розроблена інформаційна модель опрацювання даних та відповідне спеціалізоване програмне забезпечення для моделювання аналізу факторів впливу автоматизації підтримки програмних комплексів з використанням середовищ програмування R та Python. Розроблена модель дає змогу досліджувати процеси впливу різноманітних факторів впливу на результати суб'єктивного сприйняття об'єктів підтримки (підтримуваних програмних комплексів, чи процесів їх підтримки) відповідними суб'єктами взаємодії, що прямо чи опосередковано взаємодіють з цим об'єктом(-ами) підтримки. Розроблена модель реалізує практичний підхід до вирішення науково-прикладної задачі відновлення границь факторів впливу підтримки програмних комплексів з подальшою можливістю аналізу цих факторів впливу, що є однією з комплексу задач науково-прикладної проблеми автоматизації підтримки програмних комплексів. Наведені та представлені відповідні конкретні результати моделювання, отримані в ході реальних експериментальних досліджень задач аналізу факторів впливу підтримки програмних комплексів. Розв'язана прикладна практична задача пошуку члена команди підтримки програмного комплексу з максимальним відхиленням показників впливу факторів відносно середнього значення впливу факторів по команді. Перспективи застосування розробленої моделі та її результатів моделювання не обмежуються виключно підтримкою програмних комплексів, а можуть бути використані також й в інших областях, де важливими є процеси суб'єктивного сприйняття досліджуваних об'єктів чи процесів.

Ключові слова: інформаційна модель, програмний комплекс, підтримка, фактори впливу, комп'ютерне моделювання.

A.I. Pukach, V.M. Teslyuk

Polytechnic National University, Lviv, Ukraine

INFORMATION MODEL FOR AUTOMATION OF SOFTWARE COMPLEXES SUPPORT INFLUENCING FACTORS ANALYSIS WITH USAGE OF THE R-SYSTEM AND PYTHON ENVIRONMENTS

Annotation. The main problem is the need to develop an effective and efficient information model for the automation of software complexes to support influencing factors analysis. The purpose of the research is to find optimal solutions for the development of this information model with the possibility of its further implementation, application, testing, analysis and processing of the obtained results while solving applied practical problems. Among the main research methods are computer modelling methods using the R-system and Python environments. The main results of the research are the following. An information model of data processing, as well as corresponding specialised software, have been developed in this research to model the software complexes support influencing factors analysis automation using R-system and Python programming environments. The developed model provides possibilities for studying the processes of influence of various influencing factors onto the results of subjective perception of supported objects (supported software complexes or processes of their support) by the relevant subjects of interaction, which directly or indirectly interact, with these supported object(s). The developed model implements a practical approach for solving the scientific and applied problem of restoring the boundaries of the influencing factors of software complex support processes with further possibilities of analysing these influencing factors, which is one of the complex tasks of the scientific and applied problem

of software complexes support automation. Relevant specific modelling results obtained during experimental studies of the problem of software complexes support influencing factors analysis have been provided and presented in this research as well. The applied practical problem of finding the software complex support team member with the maximum deviation of influencing factors indicators/levels in comparison with the average value of the influencing factors indications/levels on the team has been solved in this research as an example. The prospects of the developed model as well as its modelling results are not limited exclusively to the software complex support, but can also be used in other areas where the processes of subjective perception of the investigated objects or processes are highly valuable and important.

Keywords: information model, software complex, support, influencing factors, computer modelling.

Вступ.

На сьогодні автоматизація підтримки програмних комплексів є актуальною науково-прикладною проблемою, що включає в себе ряд супутніх науково-прикладних задач, однією з яких є задача аналізу факторів впливу автоматизації підтримки програмних комплексів, що впливають на результати суб'єктивного сприйняття об'єкта підтримки відповідними суб'єктами, що так чи інакше з ним взаємодіють, та впливають на його підтримку.

Взагалі проблеми, питання та задачі автоматизації в області програмних комплексів є актуальними та перспективними, оскільки самі програмні комплекси, технології навколо них, підходи, методики, як і самі розробники та користувачі, продовжують постійно розвиватись в наш час швидких змін та різноманітних ІТ-трендів.

Серед основних напрямків автоматизації в області процесів довкола програмних комплексів можна виділити такі, як: автоматизація тестування, автоматизація DevOps, автоматизація прийняття рішень та багато інших. Розглянемо їх на основі аналізу інформації з відповідних використаних інформаційних джерел.

Автоматизація тестування бере свій початок ще з тих часів, коли технічні можливості такої автоматизації були надзвичайно обмеженими та відверто примітивними, проте тоді цей напрямок видавався його першопрохідцям надзвичайно перспективним і, зрештою, вони мали рацію. Робота [2] далекого 1999 року випуску: описує як структурувати та створювати автоматизований режим тестування, який дасть тривалі переваги у використанні інструментів виконання тестів для автоматизації тестування в середньому та великому масштабі, а також пропонує практичні поради щодо вибору правильного інструменту та впровадження методів автоматизованого тестування в компаніях, та відображає як позитивний, так і негативний досвід автоматизації тестування. Тобто, як бачимо, вже в той час підхід до цього напрямку – автоматизації розробки програмних комплексів, був на доволі високому та серйозному рівні сприйняття. Можливо, саме це і дало свої плоди. Адже сьогодні у 2024-ому році, згідно з інформацією представленою в роботі [5], лише 9% усіх компаній стверджують, що не використовують

автоматизацію в тестуванні, а це автоматично свідчить про часткову (а подекуди, можливо і повну) автоматизацію тестування в решті 91% компаній. А відповідно до інформації, поданої в роботі [9], мова вже йде про автоматизацію тестування на основі технологій та підходів штучного інтелекту, що фактично, є переходом від автоматизованого тестування до наступної фази — повністю автоматичного тестування.

Автоматизація за допомогою DevOps також активно розвивається. Зокрема в роботі [8] розглядаються питання використання штучного інтелекту для автоматизації DevOps процесів в області системної інженерії. Також, в роботі [4] розглядається питання автоматизації реагування (відповіді) на інциденти та загрози як частини автоматизації кібербезпеки. Крім того, в роботі [3] піднімається питання впливу людських факторів на адаптацію DevOps. В роботі [1] розглянуте питання автоматизації хмарних сервісів за допомогою DevOps. Додатково, в роботі [7] піднімається питання оптимізації автоматизації спеціалізованого тестування при використанні DevOps. Тобто, напрямком автоматизації DevOps також постійно досліджується, розвивається та системно вдосконалюється. В свою чергу, автоматизація прийняття рішень використовується в обох з наведених напрямків, а також у багатьох інших додаткових напрямках.

Також доволі ґрунтовною працею в напрямку автоматизації підтримки є робота [6], що піднімає питання розуміння впливу автоматизації на загальну продуктивність системи та когнітивні потреби людини, а також питання негативних аспектів впровадження автоматизації підтримки на противагу отриманим перевагам.

Водночас, в цій роботі автори розглядають усі ці кейси автоматизації (від тестування, до прийняття рішень та інших можливих) з токи зору суб'єктивного сприйняття як кожного з цих кейсів, так і підтримуваних програмних комплексів чи процесів їх підтримки в цілому. Зокрема, основною метою на даному етапі проведеного дослідження є, якраз, аналіз факторів впливу, що впливають на результати суб'єктивного сприйняття підтримки програмних комплексів відповідними суб'єктами взаємодії з ними (від тестувальників, до менеджерів проєктів, та реальних клієнтських користувачів).

Власне, саме з цією метою було розроблене відповідне спеціалізоване програмне забезпечення (ПЗ), представлене в цій роботі, що дає змогу здійснити моделювання аналізу факторів впливу автоматизації підтримки програмних комплексів.

Результати дослідження.

Інформаційна модель опрацювання даних.

Нижче на рисунку представлена розроблена інформаційна модель опрацювання даних для аналізу факторів впливу автоматизації підтримки програмних комплексів з використанням відповідних середовищ програмування R та Python.

Модель розроблена саме для програмного (з використанням вже згаданих R та Python та з допомогою відповідного розробленого спеціалізованого ПЗ, описаного в роботі) моделювання процесів аналізу факторів впливу.

Особливістю розробленої моделі є реалізований симбіоз двох потужних спеціалізованих середовищ програмування: R-системи та Python; а також спеціально розроблений механізм підготовки та опрацювання даних, що забезпечує можливість відповідної ідентифікації та аналізу досліджуваних суб'єктивних факторів впливу.

Розроблена інформаційна модель має трьохкомпонентну структуру, де кожен компонент – це окремий напівавтономний програмний модуль, який може взаємодіяти з іншими модулями напряму або опосередковано шляхом опрацювання спільного потоку даних, що розпочинається із вхідних даних (дані, що містять інформацію про вхідні характеристики об'єкта а також відповідні результати їх суб'єктивної інтерпретації) та

завершується вихідними результатами моделювання (результати, що містять інформацію про величину впливу, представлену в конкретному числовому еквіваленті, кожного із визначених факторів впливу на результуюче представлення досліджуваного об'єкта сприйняття відповідними суб'єктами взаємодії з ним). Перевагою запропонованої трикомпонентної структури розробленої моделі є можливість використання кожного з компонентних модулів як окремо в рамках поточної моделі – для вирішення певних задач, так і поза межами поточної моделі – наприклад, для розроблення нових або інших моделей.

Ще однією надзвичайно важливою складовою розробленої моделі є запропонована структура представлення та опрацювання даних для моделювання, що передбачає чітку організацію вхідних даних у формі дата-сетів двох типів:

- глобального дата-сету – для навчання відповідної інкапсульованої в модель штучної нейронної мережі (ШНМ) типу багатошарового перцептрона (БП) моделі, а також для одалшого моделювання;

- локальних дата-сетів – для можливості виокремлення ізольованого впливу кожного окремо взятого фактора впливу.

Модель забезпечує неперервне опрацювання потоку даних між відповідними компонентами (модулями), аж до отримання фінального результату, а також передбачає виведення всієї необхідної додаткової інформації (зокрема, побудованих зворотніх ланцюжків максимальних wag) разом з результатами.

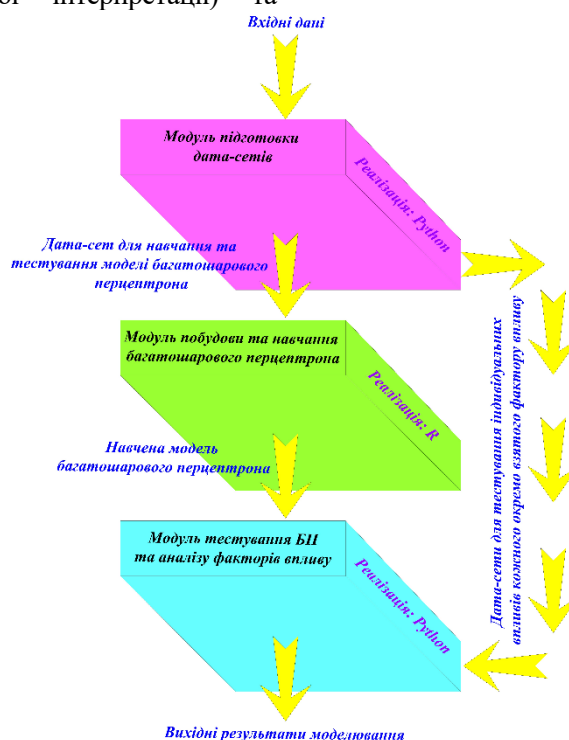


Рисунок 1 – Інформаційна модель аналізу факторів впливу

В якості середовища розробки для моделювання аналізу факторів впливу автоматизації підтримки програмних комплексів було використано комбіноване середовище, що включає:

- R-систему версії R 3.6.3 – доступна на ресурсі [10];

- пакет «neuralnet» для R-системи – повна технічна документація до якого представлена на ресурсі [11];

- Python 3.12 – доступний на ресурсі [15];

- IDE Thonny 4.1.4 – доступний на ресурсі [14];

- також для відлагодження коду розробленого ПЗ та у випадку будь-яких труднощів, що стосувалися саме написання та відлагодження коду, опрацювання виняткових ситуацій, помилок компіляції тощо, були використані ресурси [13] та [12].

Одразу ж варто зазначити, чому саме був здійснений такий вибір середовища розробки. Зокрема, одним з основних аспектів на користь вибору саме такої конфігурації є, насамперед, спеціалізованість обох компонентів обраної конфігурації: як R так і Python спеціалізуються на роботі з опрацюванням даних (data mining), штучним інтелектом, штучними нейронними мережами, в тому числі мережами типу багат шарового перцептрона; володіють широким прикладним інструментарієм в цьому напрямку, відлагодженими та відтестованими рішеннями, значними направами та активною спільнотою фахівців та ентузіастів по всьому світу, відданих своїй справі. Крім того, обидві мови програмування характеризуються надзвичайно лаконічним синтаксисом та розвинутою інтерпретаційною інфраструктурою компілятора, а також наявністю та можливістю простого підключення відповідних спеціалізованих бібліотек для роботи в напрямку опрацювання даних та штучного інтелекту.

Структура розробленого ПЗ комбінована, і включає в себе кілька окремих модулів, що відповідають за конкретні етапи моделювання, які описано нижче.

Модуль підготовки дата-сету для навчання та тестування моделі багат шарового перцептрона, а також відповідних дата-сетів для тестування індивідуальних впливів кожного окремо взятого фактора впливу, відповідає за підготовку вхідних даних в заданому форматі, а також за попередню деперсоналізацію та нормалізацію цих даних, що є обов'язковим у відповідності з загальною практикою роботи з подібними дата-сетами. Цей модуль реалізований в середовищі Python.

Модуль побудови та навчання відповідної моделі багат шарового перцептрона відповідає за побудову та навчання спроектованої моделі багат шарового перцептрона на основі дата-сету,

підготовленого попереднім модулем. Цей модуль реалізований в середовищі R.

Модуль тестування навченої моделі багат шарового перцептрона, а також пошуку та формування зворотніх ланцюжків максимальних ваг для подальшої ідентифікації та аналізу факторів впливу. Цей модуль реалізований в середовищі Python.

Опис процесу дослідження та моделювання. Основна ідея процесу дослідження та відповідного моделювання полягає в реалізації механізму зворотного аналізу факторів впливу для автоматизації підтримки програмних комплексів (власне, розроблене програмне забезпечення, представлене в цій статті, є якраз частковою програмною реалізацією цього механізму) полягає в таких етапах:

- розроблення узагальненої моделі суб'єктивного сприйняття об'єкта підтримки;

- розроблення та інкапсуляція в цю узагальнену модель відповідної штучної нейронної мережі (в загальному випадку – довільного типу, але в цьому конкретному прикладі розглядається саме ШНМ типу БП);

- навчання інкапсульованої моделі БП на загальному дата-сеті;

- підготовка дата-сетів для тестування індивідуальних впливів кожного окремо взятого ізольованого фактора впливу;

- тестування моделі на кожному з індивідуальних дата-сетів та формування відповідних зворотніх ланцюжків максимальних ваг;

- * формування зворотніх ланцюжків максимальних ваг відбувається таким чином: ми проходимо/перебираємо усі шари БП у зворотньому порядку (від останнього прихованого шару, через усі попередні приховані шари, аж до вхідного шару), і додаємо у формований зворотний ланцюжок максимальних ваг ті нейрони (з попереднього шару відносно «поточного нейрона»), внесок яких у активацію «поточного нейрона» був максимальним, після чого такий нейрон автоматично стає новим «поточним нейроном», і ми вже переходимо до опрацювання попереднього шару БП (аж доки не опрацюємо вхідний шар)

- виокремлення серед усіх отриманих зворотніх ланцюжків лише унікальних;

- розрахунок ймовірностей належності нейронів, що входять в ці унікальні зворотні ланцюжки, до відповідних факторів впливу, до яких належить кожен з цих ланцюжків.

Розроблена модель функціонує в повній відповідності до наведеного опису. Відповідно, на виході моделі отримано конкретні числові значення ймовірностей належності кожного з нейронів прихованих шарів (НПШ)

інкапсульованої ШНМ БП до відповідних факторів впливу, що в надалі використано для розрахунку конкретних числових значень величини впливу кожного із факторів впливу на суб'єктивне сприйняття об'єкта підтримки досліджуваним суб'єктом взаємодії.

Обговорення результатів дослідження.

Всі наведені нижче результати стосуються одного і того ж конкретного експериментального

досліджуваного кейса, тож вони безпосередньо пов'язані між собою.

Нижче на рисунку представлений формат даних в дата-сеті (дані вже подані у деперсоналізованій нормалізованій формі), отриманому на виході модуля підготовки дата-сетів розробленого ПЗ.

	A	B	C	D	E	F	G	H	I
1	I1	I2	I3	I4	Oideal	O1	O2	O3	O4
2	0.1	0.8	0.8	0.7	0.5	0	0	1	0
3	0.3	0.2	0.9	0.6	0.3	0	1	0	0
4	0.1	0.2	0.8	0.8	0.3	0	1	0	0
5	0.3	0.3	0.6	0.2	0.3	0	1	0	0
6	0.3	0.9	0.9	0.2	0.5	0	0	1	0
7	0.7	0.1	0.9	0.7	0.5	0	0	1	0
8	0.6	0.6	0.5	0.8	0.5	0	0	1	0
9	0.2	0.1	0.6	0.6	0.3	0	1	0	0
10	0.6	0.7	0.1	0.4	0.3	0	1	0	0
11	0.8	0.1	0.5	0.7	0.5	0	0	1	0
12	0.4	0.5	0.8	0.4	0.5	0	0	1	0
13	0.2	0.3	0.2	0.1	0.1	1	0	0	0
14	0.1	0.7	0.5	0.6	0.3	0	1	0	0
15	0.2	0.2	0.7	0.7	0.3	0	1	0	0
16	0.1	0.6	0.2	0.3	0.1	1	0	0	0
17	0.5	0.3	0.7	0.6	0.5	0	0	1	0
18	0.8	0.2	0.2	0.7	0.3	0	1	0	0
19	0.3	0.8	0.7	0.1	0.3	0	1	0	0
20	0.7	0.9	0.2	0.5	0.5	0	0	1	0
21	0.6	0.4	0.2	0.2	0.3	0	1	0	0
22	0.3	0.7	0.6	0.4	0.3	0	1	0	0
23	0.6	0.9	0.7	0.9	0.7	0	0	0	1
24	0.3	0.1	0.1	0.8	0.3	0	1	0	0
25	0.4	0.3	0.1	0.9	0.3	0	1	0	0

Рисунок 2 – Формат CSV-файла з даними для моделювання.

Особливістю запропонованого формату представлення вхідних даних моделі є, власне, вже згадана їх попередня деперсоналізація та нормалізація, що забезпечує максимальний степінь безпеки для будь-якої потенційно чутливої інформації, а також максимальний

степінь коректності подальшого опрацювання таких вхідних даних.

Нижче на рисунку подано код модуля побудови та навчання відповідної інкапсульованої штучної нейронної мережі типу багатоварового перцептрона.

```
PCObjSup <- read.csv("C:/inputGDataset.csv")
str(PCObjSup)
PCObjSup_train <- PCObjSup[1:1000,]
PCObjSup_test <- PCObjSup[1001:1100,]
library(neuralnet)
my_activation_function <- function(x) {ifelse(x<0, 0.01*x, x)}
informational_model <- neuralnet(O1 + O2 + O3 + O4 ~ I1 + I2 + I3 + I4,
  data = PCObjSup_train,
  hidden = c(5,5,5,5),
  act.fct = my_activation_function,
  linear.output = FALSE,
  threshold=1, stepmax=1000000)
plot(informational_model)
informational_model$weights
```

Рисунок 3 – Код модуля побудови та навчання інкапсульованої штучної нейронної мережі типу багатоварового перцептрона

Представлений модуль побудови та навчання інкапсульованої ШНМ БП розробленої моделі використовує:

- бібліотеку Neuralnet;
- функцію активації LeakyReLU;
- алгоритм зворотного розповсюдження помилки Rprop+;
- параметр `linear.output=false`, що вказує на необхідність застосування функції активації також і для нейронів вихідного шару;

- гіперпараметр `threshold`, що відповідає за величину порогу для часткових похідних функції помилки як критерію зупинки;

- гіперпараметр `stepmax`, що відповідає за максимальну кількість кроків навчання БП.

На рисунку 4 зображено приклад навченої інкапсульованої мережі БП, як результату роботи відповідного модуля.

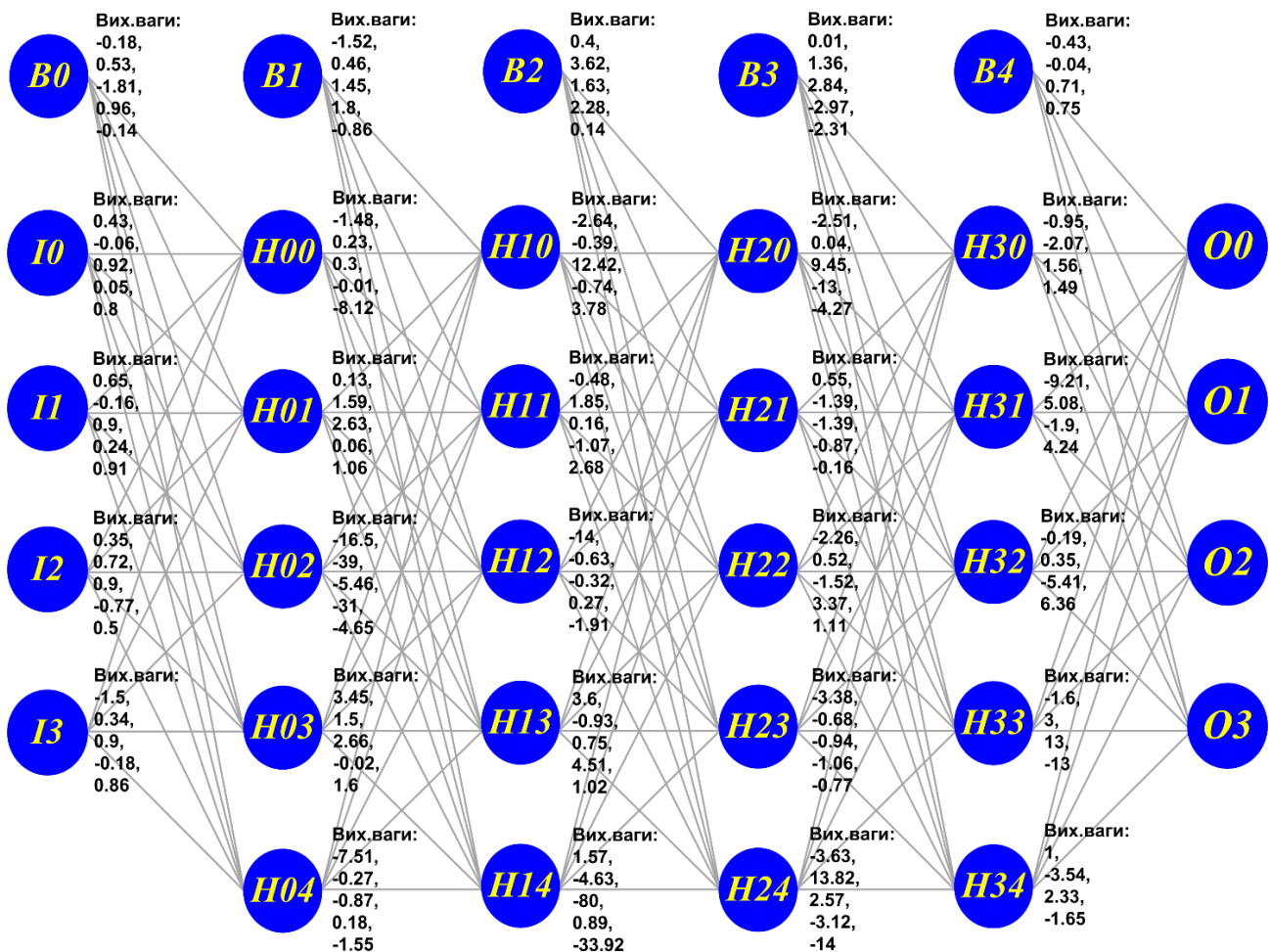


Рисунок 4 – Навчена інкапсульована мережа БП, як результат роботи відповідного модуля

На наступному рисунку зображено приклад роботи модуля тестування БП та аналізу факторів впливу, на якому видно, власне, відповідний процес та результати формування зворотних ланцюжків максимальних ваг.

Модуль відображає детальну інформацію про: значення вхідних параметрів; еталонне значення сумарного десяткового результату еталонні диференційовані значення кожного з

вихідних нейронів; актуальні диференційовані значення кожного з вихідних нейронів; ідентичність актуального та еталонного результатів; а також побудований для кожного тестового кейса зворотний ланцюжок максимальних ваг – від вихідного активного нейрона, через нейрони всіх прихованих шарів, аж до відповідного вхідного нейрона.

```

Shell X
>>> %Run analyze_GDataset.py
#      Input values      Res      Ideal      Actual (0|1)      OK?      O[x]<-HLN[3][x]<-HLN[2][x]<-HLN[1][x]<-HLN[0][x]<-I[x]
1      0.3 0.4 0.4 0.5    0.3      0100      0 1 0 0      Y      O[1]<-HLN[3][0]<-HLN[2][1]<-HLN[1][1]<-HLN[0][1]<-I[2]
2      0.2 0.5 0.9 0.6    0.5      0010      0 0 1 0      Y      O[2]<-HLN[3][2]<-HLN[2][0]<-HLN[1][0]<-HLN[0][3]<-I[1]
3      0.1 0.1 0.2 0.1    0.1      1000      1 0 0 0      Y      O[0]<-HLN[3][1]<-HLN[2][1]<-HLN[1][1]<-HLN[0][3]<-I[1]
4      0.8 0.3 0.3 0.6    0.3      0100      0 1 0 0      Y      O[1]<-HLN[3][0]<-HLN[2][1]<-HLN[1][1]<-HLN[0][1]<-I[2]
5      0.8 0.5 0.3 0.5    0.5      0010      0 0 1 0      Y      O[2]<-HLN[3][2]<-HLN[2][0]<-HLN[1][0]<-HLN[0][3]<-I[1]
6      0.8 0.8 0.2 0.9    0.5      0010      0 0 1 0      Y      O[2]<-HLN[3][2]<-HLN[2][0]<-HLN[1][0]<-HLN[0][3]<-I[1]
7      0.3 0.3 0.8 0.4    0.3      0100      0 1 0 0      Y      O[1]<-HLN[3][0]<-HLN[2][1]<-HLN[1][1]<-HLN[0][1]<-I[2]
8      0.9 0.1 0.4 0.7    0.5      0010      0 0 1 0      Y      O[2]<-HLN[3][2]<-HLN[2][0]<-HLN[1][0]<-HLN[0][3]<-I[1]
9      0.8 0.2 0.9 0.5    0.5      0010      0 0 1 0      Y      O[2]<-HLN[3][2]<-HLN[2][0]<-HLN[1][0]<-HLN[0][3]<-I[1]
10     0.9 0.1 0.2 0.8    0.3      0100      0 1 0 0      Y      O[1]<-HLN[3][0]<-HLN[2][1]<-HLN[1][1]<-HLN[0][1]<-I[2]
11     0.1 0.9 0.3 0.7    0.3      0100      0 1 0 0      Y      O[1]<-HLN[3][0]<-HLN[2][1]<-HLN[1][1]<-HLN[0][1]<-I[2]
12     0.5 0.3 0.3 0.9    0.3      0100      0 1 0 0      Y      O[1]<-HLN[3][0]<-HLN[2][1]<-HLN[1][1]<-HLN[0][1]<-I[2]
13     0.7 0.4 0.2 0.7    0.3      0100      0 1 0 0      Y      O[1]<-HLN[3][0]<-HLN[2][1]<-HLN[1][1]<-HLN[0][1]<-I[2]
14     0.5 0.5 0.7 0.8    0.5      0010      0 0 1 0      Y      O[2]<-HLN[3][2]<-HLN[2][0]<-HLN[1][0]<-HLN[0][3]<-I[1]
15     0.8 0.9 0.8 0.9    0.7      0001      0 0 0 1      Y      O[3]<-HLN[3][3]<-HLN[2][2]<-HLN[1][4]<-HLN[0][1]<-I[2]
16     0.7 0.5 0.3 0.6    0.5      0010      0 0 1 0      Y      O[2]<-HLN[3][2]<-HLN[2][0]<-HLN[1][0]<-HLN[0][3]<-I[1]
17     0.2 0.5 0.5 0.9    0.5      0010      0 0 1 0      Y      O[2]<-HLN[3][2]<-HLN[2][0]<-HLN[1][0]<-HLN[0][3]<-I[1]
18     0.7 0.8 0.4 0.8    0.5      0010      0 0 1 0      Y      O[2]<-HLN[3][2]<-HLN[2][0]<-HLN[1][0]<-HLN[0][3]<-I[1]
19     0.2 0.6 0.7 0.4    0.3      0100      0 1 0 0      Y      O[1]<-HLN[3][0]<-HLN[2][1]<-HLN[1][1]<-HLN[0][1]<-I[2]
20     0.7 0.4 0.7 0.6    0.5      0010      0 0 1 0      Y      O[2]<-HLN[3][2]<-HLN[2][0]<-HLN[1][0]<-HLN[0][3]<-I[1]
21     0.2 0.3 0.7 0.3    0.3      0100      0 1 0 0      Y      O[1]<-HLN[3][0]<-HLN[2][1]<-HLN[1][1]<-HLN[0][1]<-I[2]
22     0.6 0.7 0.8 0.7    0.5      0010      0 0 1 0      Y      O[2]<-HLN[3][2]<-HLN[2][0]<-HLN[1][0]<-HLN[0][3]<-I[1]
23     0.7 0.7 0.8 0.9    0.7      0001      0 0 0 1      Y      O[3]<-HLN[3][3]<-HLN[2][2]<-HLN[1][4]<-HLN[0][1]<-I[2]
24     0.9 0.4 0.9 0.7    0.7      0001      0 0 0 1      Y      O[3]<-HLN[3][3]<-HLN[2][2]<-HLN[1][4]<-HLN[0][1]<-I[2]
25     0.3 0.2 0.9 0.9    0.5      0010      0 0 1 0      Y      O[2]<-HLN[3][2]<-HLN[2][0]<-HLN[1][0]<-HLN[0][3]<-I[1]
26     0.1 0.9 0.7 0.7    0.5      0010      0 0 1 0      Y      O[2]<-HLN[3][2]<-HLN[2][0]<-HLN[1][0]<-HLN[0][3]<-I[1]
27     0.5 0.3 0.6 0.4    0.3      0100      0 1 0 0      Y      O[1]<-HLN[3][0]<-HLN[2][1]<-HLN[1][1]<-HLN[0][1]<-I[2]
28     0.3 0.1 0.3 0.1    0.1      1000      1 0 0 0      Y      O[0]<-HLN[3][1]<-HLN[2][0]<-HLN[1][3]<-HLN[0][2]<-I[0]

```

Рисунок 5 – Приклад роботи модуля тестування БП та аналізу факторів впливу, з побудовою зворотних ланцюжків максимальних ваг

В результаті роботи модуля отримано такі зворотні ланцюжки для кожного з факторів впливу:

- для фактора впливу 1:

$HPH\{3\}\{1\} \leftarrow HPH\{2\}\{0\} \leftarrow HPH\{1\}\{3\} \leftarrow HPH\{0\}\{2\}$
 $HPH\{3\}\{1\} \leftarrow HPH\{2\}\{1\} \leftarrow HPH\{1\}\{1\} \leftarrow HPH\{0\}\{3\}$
 $HPH\{3\}\{1\} \leftarrow HPH\{2\}\{4\} \leftarrow HPH\{1\}\{1\} \leftarrow HPH\{0\}\{1\}$

- для фактора впливу 2:

$HPH\{3\}\{0\} \leftarrow HPH\{2\}\{1\} \leftarrow HPH\{1\}\{1\} \leftarrow HPH\{0\}\{1\}$
 $HPH\{3\}\{1\} \leftarrow HPH\{2\}\{4\} \leftarrow HPH\{1\}\{1\} \leftarrow HPH\{0\}\{3\}$
 $HPH\{3\}\{1\} \leftarrow HPH\{2\}\{4\} \leftarrow HPH\{1\}\{1\} \leftarrow HPH\{0\}\{1\}$
 $HPH\{3\}\{0\} \leftarrow HPH\{2\}\{1\} \leftarrow HPH\{1\}\{1\} \leftarrow HPH\{0\}\{3\}$

- для фактора впливу 3:

$HPH\{3\}\{3\} \leftarrow HPH\{2\}\{0\} \leftarrow HPH\{1\}\{0\} \leftarrow HPH\{0\}\{3\}$
 $HPH\{3\}\{3\} \leftarrow HPH\{2\}\{2\} \leftarrow HPH\{1\}\{4\} \leftarrow HPH\{0\}\{3\}$
 $HPH\{3\}\{2\} \leftarrow HPH\{2\}\{0\} \leftarrow HPH\{1\}\{0\} \leftarrow HPH\{0\}\{3\}$
 $HPH\{3\}\{3\} \leftarrow HPH\{2\}\{2\} \leftarrow HPH\{1\}\{4\} \leftarrow HPH\{0\}\{1\}$
 $HPH\{3\}\{2\} \leftarrow HPH\{2\}\{2\} \leftarrow HPH\{1\}\{4\} \leftarrow HPH\{0\}\{1\}$
 $HPH\{3\}\{2\} \leftarrow HPH\{2\}\{2\} \leftarrow HPH\{1\}\{4\} \leftarrow HPH\{0\}\{3\}$

- для фактора впливу 4:

$HPH\{3\}\{3\} \leftarrow HPH\{2\}\{2\} \leftarrow HPH\{1\}\{4\} \leftarrow HPH\{0\}\{1\}$
 $HPH\{3\}\{3\} \leftarrow HPH\{2\}\{2\} \leftarrow HPH\{1\}\{4\} \leftarrow HPH\{0\}\{3\}$
 $HPH\{3\}\{1\} \leftarrow HPH\{2\}\{4\} \leftarrow HPH\{1\}\{4\} \leftarrow HPH\{0\}\{1\}$

Таким чином, на основі отриманих результатів стає можливим відновлення границь факторів впливу. Відповідно, при подальшому тестуванні: активація того чи іншого нейрона прихованого шару інкапсульованої ШНМ БП свідчатиме про вплив відповідних факторів на процес суб'єктивізації сприйняття досліджуваного об'єкта відповідним суб'єктом взаємодії.

В таблиця 1 подані отримані для висвітленого в роботі тестового прикладу моделювання дані розрахунків ймовірностей належності нейронів прихованих шарів, задіяних у побудованих зворотних ланцюжків, до відповідних факторів впливу.

Таблиця 1

Візуалізація інтерпретованих результатів моделювання в формі таблиці

Нейрон	Ймовірність (%) належності до фактора впливу <№ фактора>			
	<1>	<2>	<3>	<4>
НПШ{3}{1}	27	73	-	0
НПШ{3}{3}	-	-	70	30
НПШ{2}{0}	17	-	83	-
НПШ{2}{1}	1	99	-	-
НПШ{2}{4}	1	98	-	1
НПШ{2}{2}	-	-	67	33
НПШ{1}{1}	1	99	-	-
НПШ{1}{4}	-	-	67	33
НПШ{0}{3}	1	29	69	1
НПШ{0}{1}	1	72	16	11
НПШ{3}{0}	0	1	0	0
НПШ{3}{2}	0	0	0	1
НПШ{1}{0}	0	0	1	0
НПШ{1}{3}	1	0	0	0
НПШ{0}{2}	1	0	0	0

Отримані результати моделювання забезпечують можливість подальшого кількісного розрахунку (в числовому еквіваленті) оцінки ступеня впливу визначених факторів на суб'єктивне сприйняття об'єкта підтримки відповідним досліджуваным суб'єктом взаємодії, та формувати відповідні портрети суб'єктів, наприклад, для пошуку суб'єктів (скажімо, члена команди підтримки, та клієнтського користувача) з максимально наближеним баченням одного і того ж об'єкта підтримки (програмного комплексу, чи частини його функціоналу) для досягнення

максимального ступеня кооперації, порозуміння та взаємодії, а також багато інших задач.

Зокрема, в цій статті в якості прикладу подано застосування отриманих результатів моделювання розробленої моделі для розв'язання прикладної практичної задачі пошуку члена команди підтримки програмного комплексу, на якого досліджувані фактори мають максимальний рівень впливу (порівняно з усіма іншими членами цієї ж команди).

Нижче в таблиці 2 наведені індивідуальні портрети кожного з членів (позначені як «Суб'єкт1» – «Суб'єкт20») досліджуваної команди підтримки програмного комплексу.

Таблиця 2

Індивідуальні портрети членів команди підтримки

Суб'єкти	Фактори			
	1	2	3	4
1	0,1346	0,5387	0,2796	0,0471
2	0,1376	0,1724	0,4597	0,2303
3	0,339	0,2109	0,1399	0,3102
4	0,251	0,1802	0,1014	0,4674
5	0,4319	0,2438	0,1535	0,1708
6	0,1953	0,1647	0,2959	0,3441
7	0,1386	0,3072	0,4493	0,1049
8	0,2517	0,2948	0,2043	0,2492
9	0,1763	0,3762	0,3092	0,1383
10	0,1304	0,1916	0,3759	0,3021
11	0,1906	0,4017	0,3032	0,1045
12	0,2975	0,3846	0,1787	0,1392
13	0,1391	0,5191	0,1528	0,189
14	0,1102	0,3086	0,3801	0,2011
15	0,1074	0,1502	0,3222	0,4202
16	0,1392	0,2736	0,3922	0,195
17	0,2018	0,1203	0,4689	0,209
18	0,3945	0,174	0,103	0,3285
19	0,1231	0,4667	0,2753	0,1349
20	0,1107	0,1835	0,3877	0,3181

Значення кожного із чотирьох факторів впливу кожного члена команди підтримки формується як усереднений результат ймовірності впливу цього фактора на суб'єктивне сприйняття підтримки програмного комплексу на основі розглянутих тестових кейсів моделювання для цього конкретного члена команди.

На основі отриманих даних формується усереднений узагальнений портрет команди підтримки як середнє арифметичне по кожному із стовпців факторів впливу. Після чого здійснюється розрахунок відхилення величини кожного фактора впливу кожного члена команди від відповідного

усередненого значення цього фактора впливу цієї команди. Далі відбувається розрахунок сумарного відхилення індивідуального портрета кожного члена команди як суми відхилень кожного з факторів. І на завершальному етапі обирається максимальне значення сумарного відхилення, що репрезентує члена команди підтримки програмного комплексу з показниками впливу факторів, максимально віддалених від середніх показників по цій команді.

Отримані результати представлені нижче на рисунку 6.

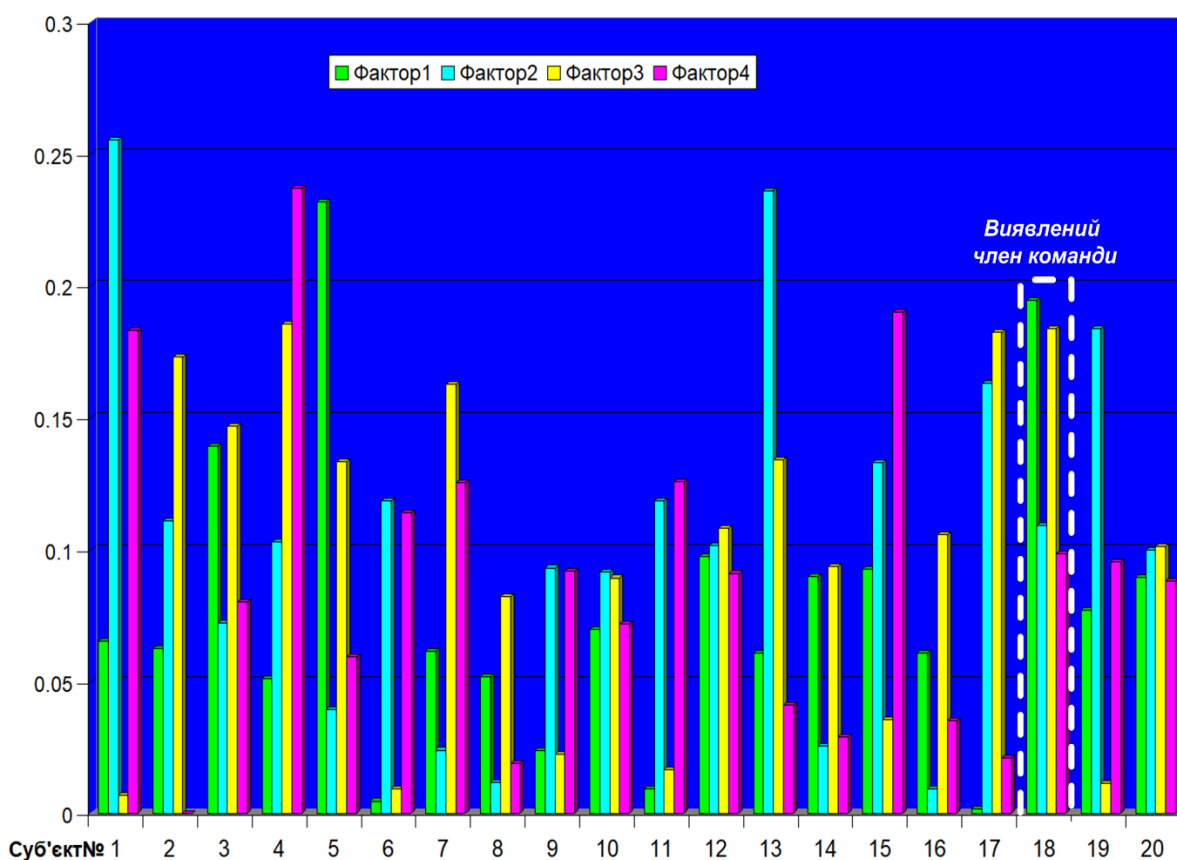


Рисунок 6 – Результати розв'язання практичної задачі пошуку члена команди з максимальним відхиленням показників впливу факторів відносно середнього значення впливу факторів по команді

Таким чином, розроблена інформаційна модель аналізу факторів впливу автоматизації підтримки програмних комплексів з використанням середовищ R та Python дає змогу отримати результати, які можуть бути використані для розв'язання значного числа прикладних практичних задач, серед яких, зокрема, представлена задача пошуку члена команди з максимальним відхиленням показників впливу факторів відносно середнього значення впливу факторів по команді.

Крім того, отримані результати можуть бути також використані для розв'язання таких прикладних практичних задач, як, наприклад:

- формування узагальненого портрета команди підтримки;
- розподіл команди підтримки на підкоманди «однотумців» – членів команди з подібними портретами і, відповідно, наблизеним баченням об'єкта підтримки;
- пошук членів команди з мінімальним відхиленням показників впливу факторів відносно середнього значення впливу факторів по команді;
- пошук взаємозамінних членів команди;
- пошук взаємозамінних членів команди;
- аналіз портретів членів команди з метою виявлення потенційних напрямків їх подальшого розвитку та вдосконалення;

- порівняння узагальнених портретів різних команд підтримки одного й того ж програмного комплексу;

- підбір оптимальних варіантів поєднання різних команд підтримки одного й того ж програмного комплексу в рамках їх спільної взаємодії;

- та багато інших.

Загалом, перспективи застосування розробленої моделі та її результатів моделювання не обмежуються виключно підтримкою програмних комплексів, а можуть бути використані також й в інших областях, зокрема, в тому числі, психології, соціології, маркетингу та багатьох інших, де важливими є процеси суб'єктивного сприйняття досліджуваних об'єктів чи процесів.

Висновки.

В статті розроблена інформаційна модель для аналізу факторів впливу автоматизації підтримки програмних комплексів з використанням середовищ програмування R та Python, а також представлені результати моделювання аналізу факторів впливу автоматизації підтримки програмних комплексів, отримані в результаті функціонування розробленого відповідного спеціалізованого програмного забезпечення з використанням середовищ програмування R та Python. Розроблена модель та ПЗ є частиною програмної реалізації відповідного методу зворотнього аналізу факторів впливу для автоматизації підтримки програмних комплексів.

Розроблена та представлена інформаційна модель для аналізу факторів впливу складається з трьох компонентів (модулів): модуля підготовки дата-сетів (деперсоналізації та нормалізації вхідних даних) – реалізованого на Python; модуля побудови та навчання відповідної інкапсульованої штучної нейронної мережі типу багатошарового перцептрона – реалізованого на R; модуля тестування БП та аналізу факторів впливу – реалізованого на Python. Подано структуру CSV-файла з деперсоналізованими нормалізованими даними, використаними як для навчання інкапсульованого багатошарового перцептрона, так і для тестування навченого БП на загальній вибірці даних (загальному дата-сеті) та аналізу відповідних дата-сетів індивідуальних впливів кожного окремо взятого фактора впливу. Подано опис процесів, на основі яких функціонує розроблена модель, а також здійснюються подальші розрахунки, інтерпретація та візуалізація отриманих результатів моделювання.

Представлені деталізовані результати моделювання на прикладі конкретного реального досліджуваного експериментального кейса. Проведена інтерпретація та візуалізація отриманих результатів моделювання, поданих в зручній формі представлення для кращого

сприйняття. Отримані результати дають наочне представлення моделювання процесу розв'язання науково-прикладної задачі аналізу факторів впливу автоматизації підтримки програмних комплексів, як частини більш комплексної науково-прикладної проблеми автоматизації підтримки програмних комплексів.

Розв'язана прикладна практична задача пошуку члена команди підтримки програмного комплексу з максимальним відхиленням показників впливу факторів відносно середнього значення впливу факторів по команді. Проведений аналіз та обговорення отриманих результатів. Окреслені перспективи подальшого розвитку щодо можливих шляхів використання моделі для розв'язання інших прикладних практичних задач.

Список літератури:

1. Baskaran Jambunathan, Dr. Y. Kalpana, Design of DevOps solution for managing multi cloud distributed environment, International Journal of Engineering & Technology, 2018, 7 (2.33), pp. 637-641, <https://doi.org/10.14419/ijet.v7i2.33.14854>
2. Fewster Mark, Graham Dorothy, (1999), Software Test Automation Effective use of test execution tools, Published by Addison-Wesley, Harlow, Essex, U.K., ISBN: 0-201-33140-3, 574 pages
3. Pérez-Sánchez, J., Rafi, S., Carrillo de Gea, J.M., Nicolás Ros, J., Fernández Alemán, J.L., A Taxonomy on Human Factors that Affect DevOps Adoption. In: Rocha, Á., Adeli, H., Dzemyda, G., Moreira, F., Poniszewska-Marańda, A. (eds) Good Practices and New Perspectives in Information Systems and Technologies. WorldCIST 2024. Lecture Notes in Networks and Systems, vol 987. Springer, Cham. https://doi.org/10.1007/978-3-031-60221-4_31
4. Krishna Neupane, Continuous Automation with DevOps practices for Threat Detection, 2023, 99 pages, <https://doi.org/10.13140/RG.2.2.33472.92169>
5. Montvelisky Joel, Bhamare Lalitkumar, 11th edition of the State of Testing Report, 2024, PractiTest & Tea-time with Testers, <https://www.practitest.com/assets/pdf/stot-2024.pdf>
6. Onnasch Linda, Benefits and Costs of Automation Support: The Role of Function Allocation and Automation Reliability, 2015, 183 pages, <https://d-nb.info/1069899747/34>
7. Sarthak Srivastava, Optimizing Automation and Specialized Testing Techniques in DevOps, International Journal for Research in Applied Science & Engineering Technology (IJRASET), 2023, Volume 11 Issue VII Jul, pp. 1449-1454, <https://doi.org/10.22214/ijraset.2023.54882>
8. Sarthak Srivastava, Utilizing AI systems to automate DevOps processes within the field of Software Engineering, International Journal of Scientific Research and Engineering Development,

2023, Volume 6 Issue 6, Nov-Dec, pp. 906-910, researchgate.net/profile/Sarthak-Srivastava-10/publication/377724783_Utilizing_AI_systems_to_automate_DevOps_processes_within_the_field_of_Software_Engineering/links/65b418bc79007454973bea00/Utilizing-AI-systems-to-automate-DevOps-processes-within-the-field-of-Software-Engineering.pdf

9. Trudova A., Dolezel M., and Buchalcevova A., Artificial intelligence in software test automation: A systematic literature review, in Proceedings of the 15th International Conference on Evaluation of Novel Approaches to Software Engineering, 2020, pp. 181–192, <https://doi.org/10.5220/0009417801810192>

10. The Comprehensive R Archive Network. Available at: <https://cran.r-project.org/> (Accessed: 10 May 2024).

11. Stefan Fritsch, Frauke Guenther, Marvin N. Wright, Marc Suling, Sebastian M. Mueller, (2022), Package ‘neuralnet’, 15 pages, <https://cran.r-project.org/web/packages/neuralnet/neuralnet.pdf>

12. Sign in, RPubs. Available at: <https://rpubs.com/> (Accessed: 10 May 2024).

13. Where developers learn, share, & build careers Stack Overflow. Available at: <https://stackoverflow.com/> (Accessed: 10 May 2024).

14. Thonny, Python IDE for beginners. Available at: <https://thonny.org/> (Accessed: 10 May 2024).

15. Welcome to Python.org. Available at: <https://www.python.org/> (Accessed: 10 May 2024).

References:

1. Baskaran Jambunathan, Dr. Y. Kalpana, (2018), ‘Design of DevOps solution for managing multi cloud distributed environment’, International Journal of Engineering & Technology, 7 (2.33), pp. 637-641, <https://doi.org/10.14419/ijet.v7i2.33.14854>

2. Fewster Mark, Graham Dorothy, (1999), Software Test Automation Effective use of test execution tools, Published by Addison-Wesley, Harlow, Essex, U.K., ISBN: 0-201-33140-3, 574 pages

3. Pérez-Sánchez, J., Rafi, S., Carrillo de Gea, J.M., Nicolás Ros, J., Fernández Alemán, J.L. (2024). A Taxonomy on Human Factors that Affect DevOps Adoption. In: Rocha, Á., Adeli, H., Dzemyda, G., Moreira, F., Poniszewska-Marańda, A. (eds) Good Practices and New Perspectives in Information Systems and Technologies. WorldCIST 2024. Lecture Notes in Networks and Systems, vol 987. Springer, Cham DOI: [10.1007/978-3-031-60221-4_31](https://doi.org/10.1007/978-3-031-60221-4_31)

4. Krishna Neupane, (2023), ‘Continuous Automation with DevOps practices for Threat Detection’, 99 pages,

<https://doi.org/10.13140/RG.2.2.33472.92169>

5. Montvelisky Joel, Bhamare Lalitkumar, (2024), ‘11th edition of the State of Testing Report’, PractiTest & Tea-time with Testers, <https://www.practitest.com/assets/pdf/stot-2024.pdf>

6. Onnasch Linda, (2015), ‘Benefits and Costs of Automation Support: The Role of Function Allocation and Automation Reliability’, 183 pages, <https://d-nb.info/1069899747/34>

7. Sarthak Srivastava, (2023), ‘Optimizing Automation and Specialized Testing Techniques in DevOps’, International Journal for Research in Applied Science & Engineering Technology (IJRASET), Volume 11 Issue VII Jul, pp. 1449-1454, <https://doi.org/10.22214/ijraset.2023.54882>

8. Sarthak Srivastava, (2023), ‘Utilizing AI systems to automate DevOps processes within the field of Software Engineering’, International Journal of Scientific Research and Engineering Development, Volume 6 Issue 6, Nov-Dec, pp. 906-910, researchgate.net/profile/Sarthak-Srivastava-10/publication/377724783_Utilizing_AI_systems_to_automate_DevOps_processes_within_the_field_of_Software_Engineering/links/65b418bc79007454973bea00/Utilizing-AI-systems-to-automate-DevOps-processes-within-the-field-of-Software-Engineering.pdf

9. Trudova A., Dolezel M., and Buchalcevova A., (2020), ‘Artificial intelligence in software test automation: A systematic literature review’, in Proceedings of the 15th International Conference on Evaluation of Novel Approaches to Software Engineering, pp. 181–192, <https://doi.org/10.5220/0009417801810192>

10. The Comprehensive R Archive Network. Available at: <https://cran.r-project.org/> (Accessed: 10 May 2024).

11. Stefan Fritsch, Frauke Guenther, Marvin N. Wright, Marc Suling, Sebastian M. Mueller, (2022), Package ‘neuralnet’, 15 pages, <https://cran.r-project.org/web/packages/neuralnet/neuralnet.pdf>

12. Sign in, RPubs. Available at: <https://rpubs.com/> (Accessed: 10 May 2024).

13. Where developers learn, share, & build careers Stack Overflow. Available at: <https://stackoverflow.com/> (Accessed: 10 May 2024).

14. Thonny, Python IDE for beginners. Available at: <https://thonny.org/> (Accessed: 10 May 2024).

15. Welcome to Python.org. Available at: <https://www.python.org/> (Accessed: 10 May 2024).

© А. І. Пукач, В. М. Теслюк, 2024.

Науково-методична стаття.

Надійшла до редакції 31.05.2024.

Прийнято до публікації 12.06.2024.