

ДОДАТОК А

Графічний матеріал кваліфікаційної роботи

Моделі та методи управління трафіком в корпоративній комп'ютерній мережі

Кваліфікаційна робота

Виконав:
студент гр. СПМ-23-4
Коробейніков М.Б.

Керівник:
зав. каф. ЕОМ, д.т.н.,
проф. Коваленко А.А.

Мета роботи та завдання

2

Метою кваліфікаційної роботи є дослідження існуючих методів та моделей управління трафіком у корпоративних мережах, а також розробка та практична реалізація адаптивного методу управління трафіком у корпоративній мережі з використанням машинного прогнозування навантаження для забезпечення ефективного розподілу мережевих ресурсів, підвищення якості обслуговування та стійкості до перевантажень у динамічних умовах функціонування інформаційної інфраструктури.

Об'єктом дослідження є процеси управління трафіком у корпоративних мережах .

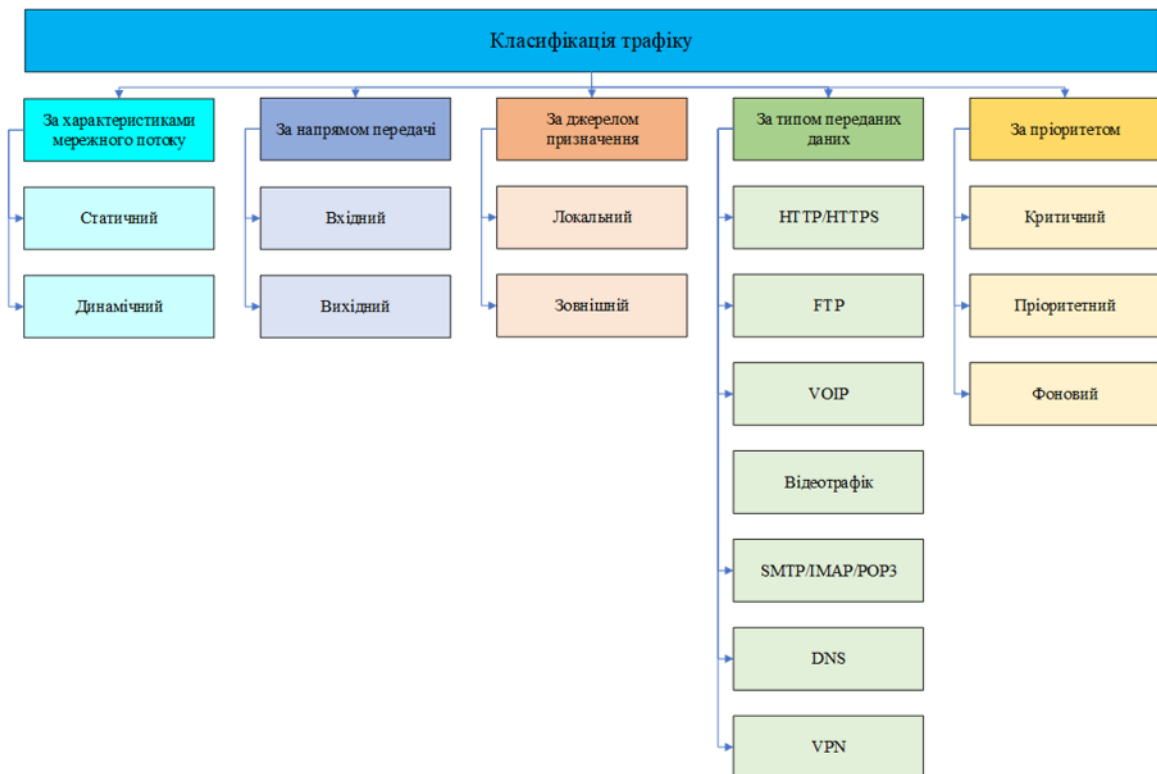
Завдання:

- провести аналіз сучасних моделей та методів управління трафіком у корпоративних мережах, охарактеризувати їх переваги та недоліки;
- здійснити класифікацію існуючих методів управління трафіком за принципами дії, рівнем адаптивності та технологічною сумісністю;
- розробити концептуальну модель адаптивно-пріоритетного методу управління з урахуванням пріоритетності трафіку та прогнозованих навантажень;
- реалізувати побудовану модель у програмному середовищі Google Colab з використанням алгоритмів машинного навчання;
- створити генератор синтетичних типів трафіку для імітаційного моделювання вхідних потоків;
- провести порівняльний аналіз точності прогнозних моделей та оцінити ефективність запропонованого методу за відповідними метриками (MSE, R^2 тощо).

Сучасні методи обробки даних в корпоративній мережі 3



Класифікація мережевого трафіку 4



Порівняльний аналіз ПЗ для управління трафіком

5

Назва ПЗ	Тип моделювання	Платформа	Рівень складності
Cisco Packet Tracer	Візуальна імітація мережі	Windows, Linux	Низький
GNS3	Емуляція мережі з реальним трафіком	Windows, Linux, macOS	Середній
NS-3	Імітаційне моделювання подій	Windows, Linux	Високий
OPNET (Riverbed Modeler)	Імітаційне моделювання	Windows	Високий
OMNeT++	Імітаційне моделювання подій	Windows, Linux	Високий
Назва ПЗ	Переваги	Недоліки	
Cisco Packet Tracer	Легкість у використанні, інтерактивність	Обмеженість у складних сценаріях	
GNS3	Реалістична емуляція, підтримка реального ПЗ	Потребує потужних ресурсів	
NS-3	Висока точність, підтримка скриптів	Складність у використанні	
OPNET (Riverbed Modeler)	Масштабованість, точне моделювання	Комерційне ПЗ, складність налаштування	
OMNeT++	Гнучкість, модульна структура	Складність для початківців	

Методи управління трафіком

6

Метод	Призначення	Переваги	Недоліки
Описова статистика	Оцінка загального стану трафіку	Простота реалізації	Не враховує динаміку
Регресійний аналіз	Прогнозування майбутніх навантажень	Можливість передбачення	Потребує якісних даних
Кореляційний аналіз	Виявлення взаємозв'язків між параметрами	Інформативність	Може давати хибні висновки
Кластеризація	Групування схожих шаблонів трафіку	Автоматизація аналізу	Залежить від обсягу даних
Аналіз сезонності	Виявлення повторюваних змін у трафіку	Використання для планування	Не працює в режимі реального часу

Метод	Призначення	Переваги	Недоліки
Traffic Shaping	Згладжування піків трафіку	Поліпшення QoS	Складність налаштування
Traffic Policing	Контроль перевищення політик	Гнучке обмеження трафіку	Може призводити до втрат
Active Queue Management	Зменшення затримок та втрат	Підвищення стабільності	Вимагає потужного обладнання
Adaptive Routing	Динамічне керування маршрутами	Реакція на зміни	Залежить від топології
Load Balancing	Рівномірний розподіл трафіку	Зниження перевантаження	Може потребувати централізації

Метод	Призначення	Переваги	Недоліки
Нейронна мережа	Автоматичне розпізнавання шаблонів трафіку	Висока точність	Складність реалізації
Нечітка логіка	Гнучке прийняття рішень при нечітких даних	Адаптивність	Потреба у великих обчисленнях
Генетичні алгоритми	Оптимізація мережевих рішень	Ефективність у складних мережах	Труднощі з налаштуванням
Методи класифікації	Виявлення аномалій	Автоматичне навчання	Вимоги до навчальних даних
Машинне навчання	Прогнозування та адаптація	Гнучкість у різних ситуаціях	Можливість помилок при навчанні

Адаптивно-пріоритетний метод управління трафіком з машинним прогнозуванням навантаження

Крок 1. Класифікація трафіку за пріоритетами. Усі вхідні потоки даних поділяються на класи залежно від критичності для бізнес-процесів:

- клас 1 – критичний трафік (голос, відео, транзакції);
- клас 2 – службовий трафік (пошта, бази даних, внутрішні API);
- клас 3 – фоновий або допоміжний трафік (оновлення, резервне копіювання).

Для кожного класу призначається пріоритет: $P1 > P2 > P3$.

Крок 2. Моніторинг поточної інтенсивності трафіку. У режимі реального часу або в контрольованому середовищі проводиться фіксація обсягу трафіку $\lambda(t)$ через регулярні інтервали часу.

Значення можуть бути агреговані в масив: $\Lambda = \{\lambda(t-p), \lambda(t-p+1), \dots, \lambda(t)\}$.

Крок 3. Побудова прогнозу трафіку. На основі масиву Λ будується модель прогнозування:

$$\lambda'(t+1) = f(\Lambda)$$

де f – функція машинного навчання (лінійна регресія, LSTM, SVM або нейронна мережа), що передбачає інтенсивність трафіку на наступний крок.

Крок 4. Аналіз співвідношення пропускної здатності. Система порівнює прогнозовану інтенсивність $\lambda'(t+1)$ з максимально допустимою смугою пропускання мережі μ :

- якщо $\lambda'(t+1) < 0.7\mu$, обслуговуються всі класи без обмежень;
- якщо $0.7\mu \leq \lambda'(t+1)$, застосовується часткове обмеження до класу 3;
- якщо $\lambda'(t+1) \geq \mu$, трафік класу 3 блокується, клас 2 – обмежується, клас 1 – працює без змін.

Крок 5. Застосування адаптивного управління чергами. Реалізується управління чергами в кожному класі:

- черги високого пріоритету обробляються першочергово.
- фоновий трафік затримується або скидається залежно від навантаження.
- застосовуються алгоритми типу Active Queue Management з адаптацією порогів від прогнозу.

Крок 6. Контроль втрат і відновлення. Система реєструє обсяг втрат пакетів, часу очікування, затримок і використовує ці дані для навчання моделі прогнозу в наступних ітераціях.

Алгоритм реалізації розробленого методу в Google Colab

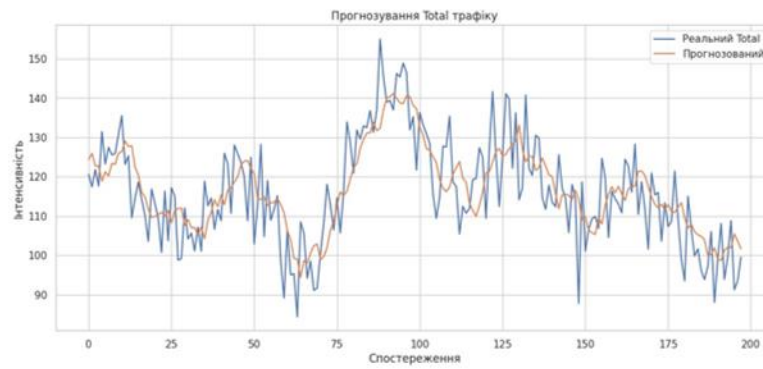
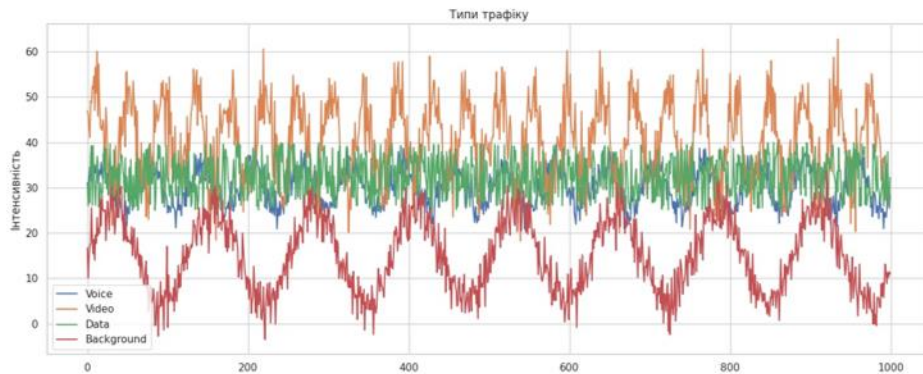
Для реалізації даного алгоритму:

- імпортувати бібліотеки;
- завантажити або згенерувати синтетичні дані трафіку;
- імітувати трафік у вигляді серії $\lambda(t)$ з різними піками;
- розмітити дані за класами трафіку;
- побудувати модель прогнозування;
- візуалізувати результат;
- зробити висновок про ефективність.



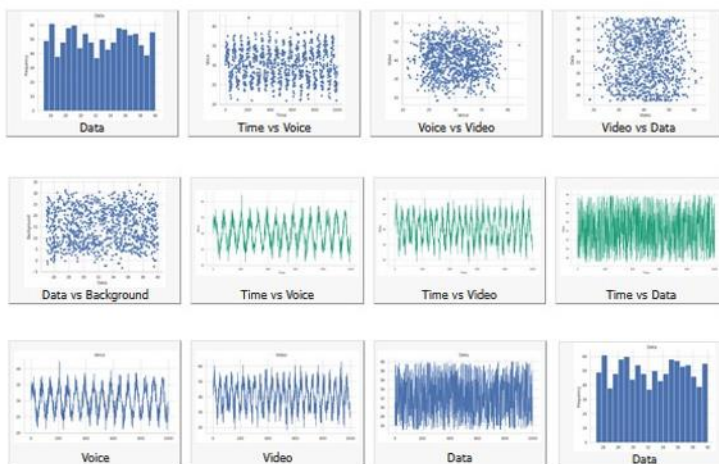
Сгенеровані типи трафіку. Прогнозування

9



10

Генерація різних типів трафіку



```

np.random.seed(42)
n = 1000
time = np.arange(0, n)

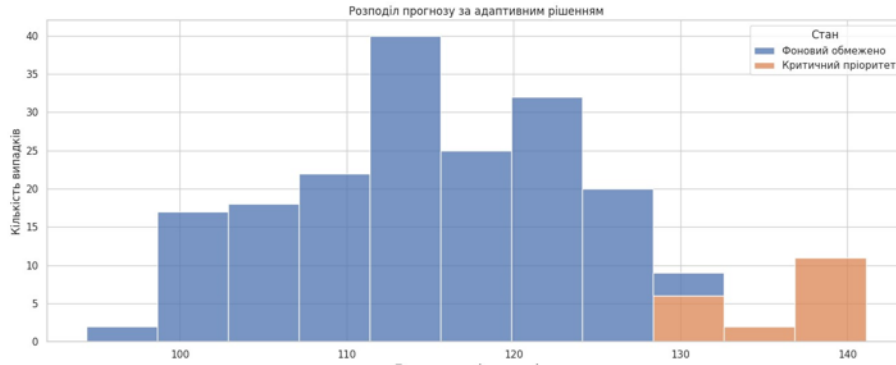
# Типи трафіку: голос (1), відео (2), дані (3), фон (4)
voice = 30 + 5 * np.sin(0.1 * time) + np.random.normal(0, 2, n)
video = 40 + 10 * np.sin(0.15 * time) + np.random.normal(0, 5, n)
data = 25 + 15 * np.random.rand(n)
background = 15 + 10 * np.sin(0.05 * time) + np.random.normal(0, 3, n)

df = pd.DataFrame({
    'time': time,
    'Voice': voice,
    'Video': video,
    'Data': data,
    'Background': background
})

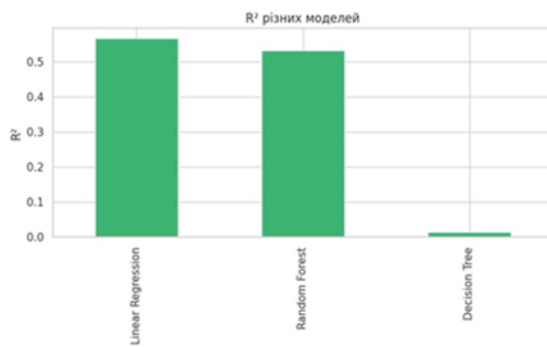
# Сумарний трафік
df['Total'] = df[['Voice', 'Video', 'Data', 'Background']].sum(axis=1)
df.head()
    
```

	Time	Voice	Video	Data	Background	Total
0	0	30.093428	40.906777	31.100507	16.719383	125.816186
1	1	30.222838	46.117550	25.900148	10.142102	112.472528
2	2	32.288724	43.253354	30.232308	14.919445	120.693831
3	3	34.023061	41.114671	20.064971	17.397703	119.701307
4	4	31.478785	40.137541	37.123528	17.538197	135.278051

Розподіл прогнозу за адаптивним рішенням



Порівняння моделей за коефіцієнтом детермінації та точністю



DATA PROCESSING METHODS IN A CORPORATE NETWORK

Abstract. Relevance. In the context of modern digital business transformation, corporate networks have become critically important components of enterprise information infrastructure, integrating numerous systems, devices, and services into a unified information environment. Every day, vast volumes of data are generated, transmitted, stored, and processed within these networks, encompassing all aspects of organizational activity – from financial operations and logistics to customer communications and internal management processes. The successful functioning of a corporate network is impossible without effective data processing methods, which enables the extraction of valuable insights from raw data, supports strategic decision-making, ensures information security, and optimizes resources. Amid intense competition, the rapid growth of information flows, and increasing demands for swift decision-making, data processing methods play a crucial role in enabling the analytical capabilities of enterprises. Data processing methods in corporate environments find applications across diverse business sectors. In the financial domain, they are used for transaction analysis, risk management, market fluctuation forecasting, and fraud detection. In logistics, they support route optimization, warehouse stock control, and real-time supply coordination. In marketing, they facilitate customer behavior analysis, audience segmentation, and personalized communication. In HR departments, they assist in employee performance evaluation and candidate selection. In healthcare, they support patient data analysis, service process optimization, and clinical decision-making. Furthermore, data processing forms the foundation for the development of artificial intelligence systems, digital twins, cybersecurity solutions, and automated enterprise management. Traditional approaches to data processing are gradually being replaced by advanced technologies based on Big Data, real-time stream processing, cloud computing, intelligent data analysis, and machine learning. In this context, issues related to data protection, integrity, availability, and compliance with legal and regulatory requirements for confidentiality gain particular importance. The purpose is to analyze modern methods and technologies for data processing in corporate networks, identify their advantages, disadvantages, and application areas, and classify approaches to data collection, cleaning, storage, and analysis. The subject of research includes methods and technologies for data processing, storage, analysis, and protection. The results. As a result of the conducted research, a comprehensive understanding of modern data processing methods in corporate environments was formed. The architectural models of enterprise information systems were analyzed, key stages of data preparation identified, and approaches to their processing – from classical to intelligent – were characterized. The review revealed a wide range of data processing techniques, from traditional SQL queries and multidimensional analysis to real-time processing and machine learning applications. The choice of a particular approach depends on data volume, data inflow speed, latency sensitivity, and the specific goals of processing.

Keywords: corporate network, data processing, architecture, transaction, cybersecurity, Big Data.

Introduction

The successful operation of modern enterprises largely depends on the efficiency of processing large volumes of information circulating within corporate networks. This involves the collection, systematization, storage, analysis, and transmission of data between various company departments, as well as external counterparts. The complexity of information processing increases with the growing volume of data, the need to ensure prompt access to information, and the tightening requirements for data security and protection. Within a corporate network, data processing encompasses the entire information lifecycle – from its creation or external acquisition to its storage, analysis, and subsequent use in decision-making processes. To fully understand the essence of data processing in this context, it is essential to have a clear understanding of what constitutes a corporate network and the types of data that circulate within it.

A corporate network represents an integrated environment that unites computer systems, servers, storage facilities, network equipment, software, and users into a single information space. Such a network enables uninterrupted communication among all structural units of an organization, regardless of their physical location, allowing for centralized data management, information exchange, and access to resource management, reporting,

and customer support systems.

The data circulating in a corporate environment may vary in format and degree of structure. Structured data includes information arrays with a clearly defined format, such as numerical tables or records in relational databases. Unstructured data, in contrast, lacks a fixed format and may include emails, text documents, audio and video files, images, or social media messages. A separate category is semi-structured data, which contains a certain degree of organization but does not conform to traditional relational database formats.

In this context, data processing entails not only the technical computation of data but also includes stages such as data collection, filtering, aggregation, transformation, analysis, and subsequent presentation in an interpretable format. A key feature of corporate data processing is the necessity to account for complex access hierarchies, user roles, security levels, and adherence to data storage and processing policies in accordance with regulatory requirements.

A defining aspect is that corporate data does not exist in isolation. It is closely integrated into an organization's business processes, forming the context in which it gains significance. This underpins the need for a comprehensive infrastructure capable of handling data processing in alignment with the company's scale, structure, and operational dynamics. Thus, understanding the

entire array of modern enterprise information systems, sensors, web services, or partner company databases.

All these factors create the necessity for a systematic study of processes associated with data handling within corporate networks. Given the growing demands for the speed, reliability, and scalability of information solutions, it is essential to examine not only the general principles of such systems' operation but also the specific methods employed to ensure a complete data cycle – from acquisition to analytical reporting. In this context, the analysis of contemporary approaches to data transformation, selection of processing algorithms, automation tools, and visualization instruments becomes particularly relevant, as they enable not only data storage but also the extraction of practical value for managerial decision-making. The purpose is to analyze modern methods and technologies for data processing in corporate networks, identify their advantages, disadvantages, and application areas, and classify approaches to data collection, cleaning, storage, and analysis.

Main part

The architecture of a corporate data processing system represents a comprehensive, multi-layered structure that ensures interaction between technical, software, and organizational components for the effective management of information flows within an enterprise. Under modern conditions, such an architecture must not only be capable of processing large volumes of data but also of promptly responding to changing business requirements, ensuring scalability, fault tolerance, and uninterrupted access to critical services.

A central element of any corporate architecture is the logic governing the distribution of data and computing resources, which may follow either a centralized or a distributed model. In centralized models, core processing and storage are performed on a single server or within a corporate data center, enabling efficient control over data access, updates, and backup operations. In contrast, distributed systems delegate processing and storage tasks to multiple network nodes, which helps reduce latency, increase flexibility, and bring data closer to end users – particularly in geographically dispersed branches of an enterprise.

The emergence of cloud technologies has led to significant changes in architectural approaches. The virtualization of computing resources, the use of infrastructure as a service and platform as a service [1], have enabled companies to dynamically scale computing power, deploy new services without significant investment in hardware, and integrate with external service providers. In such an environment, data can be stored in cloud repositories, processed through virtual clusters, and analytics can be accessed via web interfaces from anywhere in the world. Despite these advantages, such solutions require special attention to issues of security, access control, data leakage prevention, and compliance with international

standards. An effective architecture for corporate data processing must take into account the specifics of the industry, the scale of the company, the types of data being processed, and the organization's level of digital maturity. Only a flexible, scalable, and secure architecture can enable full automation of analytical processes, reduce infrastructure costs, and significantly improve the quality of managerial decisions based on up-to-date data.

In the context of corporate networks, data processing is impossible without a high-quality and systematic approach to data collection and preparation. This stage determines not only the volume and relevance of available information but also its cleanliness, completeness, and suitability for further analytical processing. In many cases, it is at the collection and preprocessing stages where data suffers the most significant quality losses, which may subsequently lead to erroneous conclusions or ineffective management decisions. Therefore, the question of corporate data collection and preparation methods is of critical importance.

Data in corporate environments can come from a variety of sources, each with its own structure, formatting standards, and update frequency. The main sources include internal enterprise information systems – particularly enterprise resource planning systems [3], accounting and logistics platforms – as well as external web services, API platforms, user behavior monitoring tools, the Internet of Things, and partner databases. Given the high heterogeneity of these sources, the data collection process relies on specialized integration tools that ensure the unification of incoming information and its delivery to a single processing environment.

One of the key methodological approaches to data collection and preparation is the ETL process, which consists of three logically connected stages [4]. The first stage involves extracting data from sources, where it is crucial to ensure data flow continuity, avoid record duplication, and maintain version control. In the transformation stage, data undergoes various operations such as filtering, format standardization, anomaly removal, deduplication, handling of missing values, categorization, or merging of records from multiple tables. The final stage is loading the cleaned and transformed information into target systems, which may include data warehouses, analytical databases, cloud platforms, or local enterprise servers.

Particular attention should be given to the data

Diachenko D., Korobeinikov M., Korobeinikov O., Kovalenko A., Kravchenko P. Data processing methods in a corporate network. Системи управління, навігації та зв'язку, вип.3. Полтава, 2025. С. 81-86.

Висновки

В процесі виконання кваліфікаційної роботи було досліджено, проаналізовано та практично реалізовано сучасні методи й моделі управління трафіком у корпоративному мережевому середовищі. Робота охоплює як теоретичні основи функціонування механізмів контролю потоків даних, так і прикладну частину, в якій було здійснено моделювання, прогнозування та оцінку ефективності запропонованого підходу на основі машинного навчання.

Проведений аналіз засвідчив, що традиційні методи управління трафіком – зокрема, статистичні, динамічні й евристичні – мають як певні переваги в окремих сценаріях, так і обмеження, пов'язані з недостатньою адаптивністю до швидких змін мережевого навантаження. Відтак виникла необхідність у створенні комбінованого підходу, що поєднує точність аналітичних моделей з гнучкістю штучного інтелекту. У відповідь на це завдання було розроблено адаптивно-пріоритетний метод управління трафіком з машинним прогнозуванням навантаження, який дозволяє приймати рішення щодо розподілу мережевих ресурсів на основі передбачення пікових ситуацій.

У роботі реалізовано повнофункціональну модель у середовищі Google Colab, що включає генерацію синтетичних типів трафіку, побудову прогнозу моделі, класифікацію за критичністю та динамічне реагування на зміни в умовах навантаження. Проведене тестування моделі на різних алгоритмах машинного навчання (лінійна регресія, дерева рішень, випадковий ліс) дало змогу оцінити точність і стабільність кожного підходу. Найкращі результати було отримано за допомогою лінійної регресії, яка продемонструвала найменшу середньоквадратичну похибку та високий коефіцієнт детермінації.

Аналіз роботи моделі показав, що в більшості випадків система ефективно ідентифікує сценарій помірного навантаження, застосовуючи політики часткового обмеження фонових процесів. У критичних ситуаціях, коли прогнозований рівень трафіку перевищує допустимі порогові, система коректно перемикається на обслуговування лише вископріоритетного трафіку. Це свідчить про здатність алгоритму до проактивного управління ресурсами мережі з урахуванням прогнозованих ризиків перевантаження.

Загалом, результати дослідження підтверджують ефективність запропонованого методу в умовах динамічного корпоративного середовища. Метод може бути застосований у реальних інфраструктурах для забезпечення стійкості сервісів, покращення якості обслуговування та оптимізації використання смуги пропускання. Крім того, використання хмарного середовища Google Colab забезпечило високу продуктивність і масштабованість реалізації без потреби в локальних ресурсах.

ДОДАТОК Б

Програмний код

Б.1 Підготовка середовища та даних

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error
sns.set(style='whitegrid')
```

Б.2 Генерація різних типів трафіку

```
np.random.seed(42)
n = 1000
time = np.arange(0, n)
# Типи трафіку: голос (1), відео (2), дані (3), фон (4)
voice = 30 + 5 * np.sin(0.1 * time) + np.random.normal(0, 2, n)
video = 40 + 10 * np.sin(0.15 * time) + np.random.normal(0, 5, n)
data = 25 + 15 * np.random.rand(n)
background = 15 + 10 * np.sin(0.05 * time) + np.random.normal(0, 3, n)
df = pd.DataFrame({
    'Time': time,
    'Voice': voice,
    'Video': video,
    'Data': data,
    'Background': background
})
# Сумарний трафік
df['Total'] = df[['Voice', 'Video', 'Data', 'Background']].sum(axis=1)
df.head()
```

Б.3 Візуалізація трафіку за типами

```
plt.figure(figsize=(14, 6))
plt.plot(df['Time'], df['Voice'], label='Voice')
plt.plot(df['Time'], df['Video'], label='Video')
```

```

plt.plot(df['Time'], df['Data'], label='Data')
plt.plot(df['Time'], df['Background'], label='Background')
plt.title('Типи трафіку')
plt.xlabel('Час')
plt.ylabel('Інтенсивність')
plt.legend()
plt.tight_layout()
plt.show()

```

Б.4 Побудова прогнозової моделі на Total трафік. Порівняння прогнозу з реальністю

```

window_size = 10
X, y = [], []
total = df['Total'].values
for i in range(window_size, len(total)):
    X.append(total[i - window_size:i])
    y.append(total[i])
X = np.array(X)
y = np.array(y)
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, shuffle=False)
model = LinearRegression()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)

```

Б.5 Порівняння прогнозу з реальністю

```

plt.figure(figsize=(14, 6))
plt.plot(np.arange(len(y_test)), y_test, label='Реальний Total')
plt.plot(np.arange(len(y_pred)), y_pred, label='Прогнозований')
plt.title('Прогнозування Total трафіку')
plt.xlabel('Спостереження')
plt.ylabel('Інтенсивність')
plt.legend()
plt.show()
print(f'SME: {mean_squared_error(y_test, y_pred):.2f}')

```

Б.6 Реалізація адаптивно-пріоритетного методу

```

threshold = 130
result = []
for real, pred in zip(y_test, y_pred):
    if pred < 0.7 * threshold:
        result.append('Усі класи активні')

```

```

elif pred < threshold:
    result.append('Фоновий обмежено')
else:
    result.append('Критичний пріоритет')
df_result = pd.DataFrame({
    'Прогноз': y_pred,
    'Стан': result
})
df_result['Стан'].value_counts()

```

Б.7 Візуалізація адаптивної реакції

```

plt.figure(figsize=(14, 6))
sns.histplot(data=df_result, x='Прогноз', hue='Стан',
multiple='stack')
plt.title('Розподіл прогнозу за адаптивним рішенням')
plt.xlabel('Прогнозована інтенсивність')
plt.ylabel('Кількість випадків')
plt.tight_layout()
plt.show()

```

Б.8 Порівняння різних моделей прогнозування трафіку

```

from sklearn.ensemble import RandomForestRegressor
from sklearn.tree import DecisionTreeRegressor
from sklearn.metrics import r2_score
# Моделі
models = {
    'Linear Regression': LinearRegression(),
    'Random Forest': RandomForestRegressor(n_estimators=100,
random_state=42),
    'Decision Tree': DecisionTreeRegressor(random_state=42)
}
results = {}
for name, model in models.items():
    model.fit(X_train, y_train)
    pred = model.predict(X_test)
    mse = mean_squared_error(y_test, pred)
    r2 = r2_score(y_test, pred)
    results[name] = {'MSE': mse, 'R2': r2}
results_df = pd.DataFrame(results).T
results_df

```