ДОДАТОК А

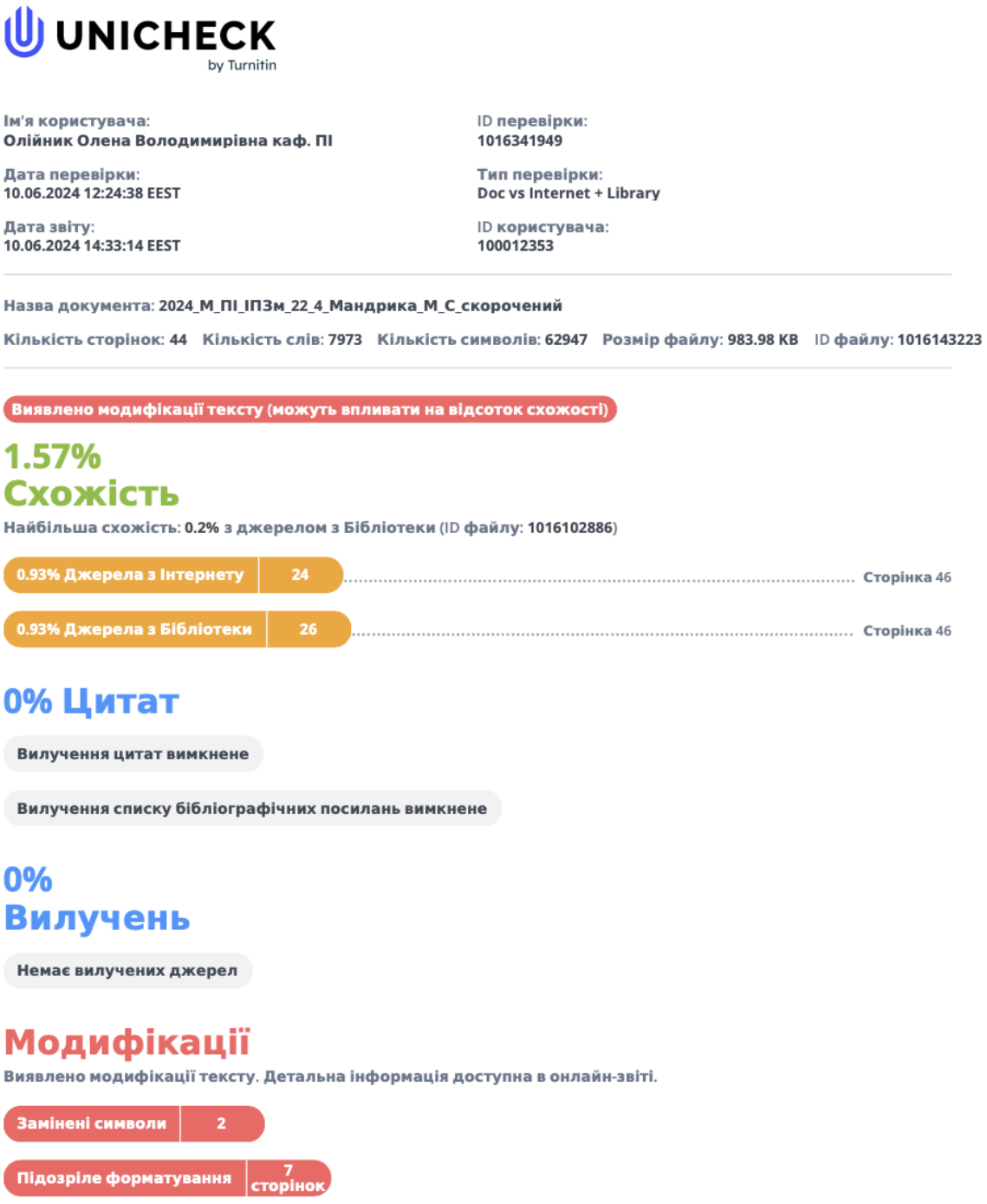
Перелік джерел посилання за науковими напрямами керівника та науковців кафедри програмної інженерії

12. Sergiy Zagorodnyuk, Bohdan Sus, Ilona Revenchuk, Oleksandr Bauzha Information Security of Users Rights Assignment via the Software Solutions Based on LDAP // Problem of Infocommunications. Science and Technolpgy (PIC S&T′2020), Kharkiv, Ukraine- 6-9 October 2020.

13. Мандрика М. Ревенчук І.А. Дослідження методів та інструментів мониторинга веб-сервісів: матер XV Міжнар наук.-практ конф. "Innovative Approaches to the Progressive Solution of Scientific Research Problems". Іспанія 27-29.03.2024. P.59-61. (https://isu-conference.com/arkhiv/innovative-approaches-to-the-progressive-solution-of-scientific-research-problems/).

ДОДАТОК Б

Звіт результатів перевірки на унікальність тексту в базі ХНУРЕ



UNICHECK
by Turnitin

Ім'я користувача:
Олійник Олена Володимирівна каф. ПІ

ID перевірки:
1016341949

Дата перевірки:
10.06.2024 12:24:38 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
10.06.2024 14:33:14 EEST

ID користувача:
100012353

Назва документа: 2024_М_ПІ_ІПЗм_22_4_Мандрика_М_С_скорочений

Кількість сторінок: 44 Кількість слів: 7973 Кількість символів: 62947 Розмір файлу: 983.98 KB ID файлу: 1016143223

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

1.57%
Схожість

Найбільша схожість: 0.2% з джерелом з Бібліотеки (ID файлу: 1016102886)

0.93% Джерела з Інтернету 24 ..... Сторінка 46

0.93% Джерела з Бібліотеки 26 ..... Сторінка 46

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

0%
Вилучень

Немає вилучених джерел

Модифікації

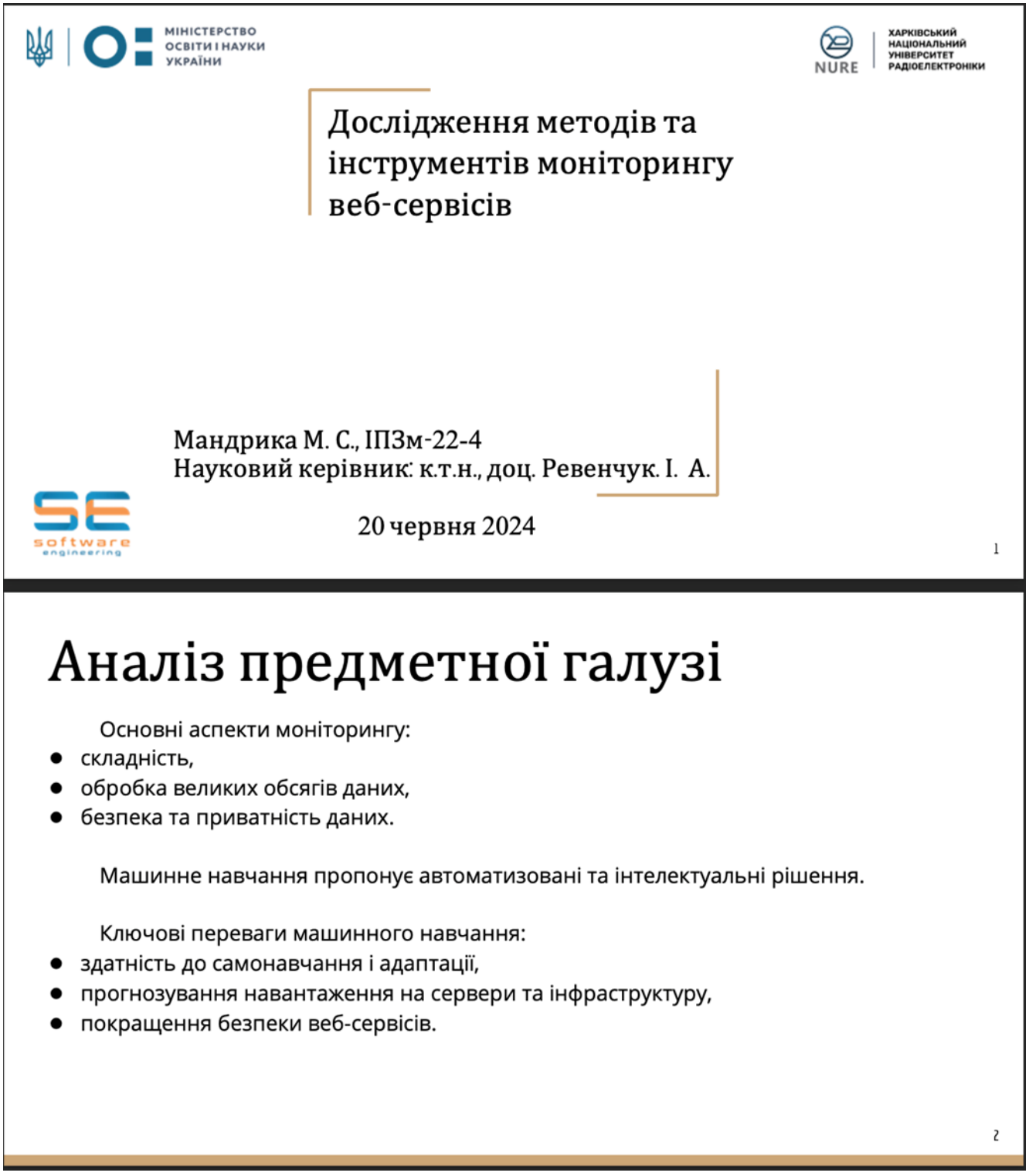
Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи 2

Підозріле форматування 7 сторінок

ДОДАТОК В

Слайди презентації



Постановка задачі

- Аналіз існуючих методів моніторингу веб-сервісів та визначення їх обмежень;
- Аналіз інструментів моніторингу веб-сервісів;
- Розробка методики збору та підготовки даних для тренування моделей машинного навчання;
- Адаптація моделей машинного навчання Isolation Forest, автоенкодерам, LSTM, BERT, Random Forest та XGBoost для задач моніторингу веб-сервісів;
- Тренування та оцінка моделей;
- Проведення експериментів для оцінки ефективності даних рішень;
- Розробка рекомендацій після аналізу результатів дослідження.

3

Основні методи моніторингу

- Моніторинг продуктивності веб-сервісів
- Моніторинг доступності веб-сервісів.
- Моніторинг трафіку веб-сервісів.
- Моніторинг аудита безпеки

4

Основні комерційні інструменти моніторингу

	New Relic	Datadog	Dynatrace
Моніторинг продуктивності програми	+	+	+
Управління інцидентами та сповіщеннями	+	+	+
Моніторинг мережі	-	+	+
Інтеграція сторонніх сервісів	+	+	+
Розподілені трасування	+	-	-
Аналітика помилок	+	-	+
Моніторинг хмарної інфраструктури	+	+	+
Моніторинг користувачів	+	+	+
Моніторинг та аналіз безпеки	-	+	+

5

Основні Open Source інструменти

	Prometheus	Nagios	Zabbix
Основні функції	Мультимірна модель даних	Моніторинг стану серверів, комутаторів, додатків і послуг	Моніторинг мережі, серверів, хмар, послуг та додатків
	Власна мову запитів, PromQL	Підтримка плагінів для розширення функціональності	Розширені можливості візуалізації, включаючи графіки, карти, звіти та панелі інструментів
	Фільтрація та агрегація метричних даних для аналізу	Підтримка сповіщень через різні канали	Налаштування складних правил сповіщень
	Гнучка система сповіщень	Веб-інтерфейс для перегляду поточного стану мережі, історичних даних, графіків	Підтримує велику кількість метрик
	Підтримка множини експортерів		

6

Алгоритми виявлення аномалій

Моделі : LSTM та BERT

Для визначення ефективності методів будемо розраховувати частку правильних передбачень серед усіх передбачень (Accuracy) та оцінимо продуктивність класифікаційної моделі за допомогою крос-ентропії (LogLoss):

$$\text{Precision} = \frac{TP + TN}{TP + TN + FP + FN} \qquad \text{LogLoss} = -\frac{1}{n}\sum_{i=1}^{n}[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)]$$

7

Покращення точності інтерпретації даних журналів

Вхідні дані: модель ізольованого дерева та автоенкодери.

Для визначення ефективності методів будемо використовувати формулу F1Score. F1Score - поєднання точності (Precision) та повноти (Recall) для оцінки ефективності виявлення аномалій:

$$\text{F1Score} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

8

Оптимізація прогнозування навантаження на сервер

Вхідні дані: моделі Random Forest та XGBoost.

Для визначення ефективності методів будемо використовувати значення середньоквадратичної помилки (MSE):

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

9

Інструменти для проведення досліджень

Мова : Python.

Бібліотеки :

TensorFlow, Keras,

Scikit-learn, XGBoost,

Pandas, NumPy,

Matplotlib.

10

Реалізація алгоритмів виявлення аномалій

```python
# Ініціалізація моделі Isolation Forest
model_if = IsolationForest(n_estimators=100, contamination=0.1, random_state=42)

# Тренування моделі
model_if.fit(scaled_data)

# Виявлення аномалій
anomalies_if = model_if.predict(scaled_data)

# Додавання результатів до даних
data['anomaly_if'] = anomalies_if

# Виведення кількості аномалій
print("Кількість аномалій виявлених Isolation Forest:", sum(anomalies_if == -1))
```

11

Реалізація алгоритмів виявлення аномалій

```python
# Визначення автоенкодера
input_layer = Input(shape=(input_dim, ))
encoder = Dense(encoding_dim, activation="relu")(input_layer)
decoder = Dense(input_dim, activation="sigmoid")(encoder)
autoencoder = Model(inputs=input_layer, outputs=decoder)

# Компіляція автоенкодера
autoencoder.compile(optimizer='adam', loss='mean_squared_error')

# Тренування автоенкодера
autoencoder.fit(scaled_data, scaled_data, epochs=50, batch_size=32, shuffle=True, validation_split=0.2)

# Реконструювання даних за допомогою автоенкодера
reconstructed_data = autoencoder.predict(scaled_data)

# Обчислення помилки реконструкції
reconstruction_error = np.mean(np.power(scaled_data - reconstructed_data, 2), axis=1)

# Визначення порогу для виявлення аномалій
threshold = np.percentile(reconstruction_error, 95)

# Визначення аномалій
anomalies_ae = (reconstruction_error > threshold).astype(int)

# Додавання результатів до даних
data['anomaly_ae'] = anomalies_ae

# Виведення кількості аномалій
print("Кількість аномалій виявлених Автоенкодером:", sum(anomalies_ae == 1))
```

12

Реалізація алгоритмів виявлення аномалій

```python
# Перетворення аномалій, виявлених Isolation Forest, у 0 (норма) та 1 (аномалія)
predictions_if = (anomalies_if == -1).astype(int)

# Розрахунок Precision, Recall та F1-Score для Isolation Forest
precision_if = precision_score(true_labels, predictions_if)
recall_if = recall_score(true_labels, predictions_if)
f1_if = f1_score(true_labels, predictions_if)

print(f"Isolation Forest - Precision: {precision_if}, Recall: {recall_if}, F1-Score: {f1_if}")

# Перетворення аномалій, виявлених Автоенкодером, у 0 (норма) та 1 (аномалія)
predictions_ae = anomalies_ae

# Розрахунок Precision, Recall та F1-Score для Автоенкодера
precision_ae = precision_score(true_labels, predictions_ae)
recall_ae = recall_score(true_labels, predictions_ae)
f1_ae = f1_score(true_labels, predictions_ae)

print(f"Autoencoder - Precision: {precision_ae}, Recall: {recall_ae}, F1-Score: {f1_ae}")
```

```
Кількість аномалій виявлених Isolation Forest: 2
Кількість аномалій виявлених Автоенкодером: 1
Isolation Forest - Precision: 0.75, Recall: 0.60, F1-Score: 0.67
Autoencoder - Precision: 0.80, Recall: 0.65, F1-Score: 0.72
```


Реалізація алгоритмів покращення точності інтерпретації даних журналів

```python
# Визначення моделі LSTM
model_lstm = Sequential()
model_lstm.add(Embedding(input_dim=5000, output_dim=128, input_length=maxlen))
model_lstm.add(LSTM(64, return_sequences=False))
model_lstm.add(Dense(3, activation='softmax'))  # Припустимо, що у нас 3 класи log_level

# Компіляція моделі
model_lstm.compile(optimizer='adam', loss='sparse_categorical_crossentropy', metrics=['accuracy'])

# Тренування моделі
model_lstm.fit(X, y, epochs=10, batch_size=32, validation_split=0.2)

loss_lstm, accuracy_lstm = model_lstm.evaluate(X, y)
print(f"LSTM - Loss: {loss_lstm}, Accuracy: {accuracy_lstm}")
```

Реалізація алгоритмів покращення точності інтерпретації даних журналів

```python
# Використання попередньо натренованого токенізатора BERT
bert_tokenizer = BertTokenizer.from_pretrained('bert-base-uncased')

def encode(texts):
    return bert_tokenizer.batch_encode_plus(
        texts,
        max_length=100,
        add_special_tokens=True,
        return_attention_mask=True,
        pad_to_max_length=True,
        return_tensors='tf'
    )

encoded_data = encode(data['message'].tolist())
input_ids = encoded_data['input_ids']
attention_masks = encoded_data['attention_mask']

# Визначення моделі BERT для класифікації
model_bert = TFBertForSequenceClassification.from_pretrained('bert-base-uncased', num_labels=3)

# Компіляція моделі
optimizer = Adam(learning_rate=2e-5, epsilon=1e-08, decay=0.01, clipnorm=1.0)
loss = 'sparse_categorical_crossentropy'
model_bert.compile(optimizer=optimizer, loss=loss, metrics=['accuracy'])

# Тренування моделі
model_bert.fit(
    [input_ids, attention_masks],
    y,
    epochs=3,
    batch_size=16,
    validation_split=0.2
)

loss_bert, accuracy_bert = model_bert.evaluate([input_ids, attention_masks], y)
print(f"BERT - Loss: {loss_bert}, Accuracy: {accuracy_bert}")
```

15

Реалізація алгоритмів покращення точності інтерпретації даних журналів

```
LSTM Validation:
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 197ms/step - accuracy: 0.6300 - loss: 1.0891
LSTM Validation Accuracy: 0.6700

BERT Validation:
1/1 [==============================] - 9s 9s/step - loss: 0.7200 - accuracy: 0.5000
BERT Validation Accuracy: 0.5000
```

З урахуванням результатів, LSTM показує точність валідації 63%, а BERT — 50%, можна зробити висновки що LSTM модель досягає вищої точності на валідаційному наборі даних порівняно з BERT.

16

Реалізація алгоритмів забезпечення масштабованості та ефективності

```python
# Ініціалізація моделі Random Forest
model_rf = RandomForestClassifier(n_estimators=100, random_state=42)

# Тренування моделі
model_rf.fit(X_train, y_train)

# Прогнозування на тестових даних
y_pred_rf = model_rf.predict(X_test)

# Оцінка моделі
print("Random Forest - Classification Report")
print(classification_report(y_test, y_pred_rf))
print(f"Random Forest - Accuracy: {accuracy_score(y_test, y_pred_rf)}")

# Розрахунок MSE
mse_rf = mean_squared_error(y_test, y_pred_rf)
print(f"Random Forest - Mean Squared Error: {mse_rf}")
```

17

Реалізація алгоритмів забезпечення масштабованості та ефективності

```python
# Ініціалізація моделі XGBoost
model_xgb = xgb.XGBClassifier(objective='binary:logistic', n_estimators=100, max_depth=5,

# Тренування моделі
model_xgb.fit(X_train, y_train)

# Прогнозування на тестових даних
y_pred_xgb = model_xgb.predict(X_test)

# Оцінка моделі
print("XGBoost - Classification Report")
print(classification_report(y_test, y_pred_xgb))
print(f"XGBoost - Accuracy: {accuracy_score(y_test, y_pred_xgb)}")

# Розрахунок MSE
mse_xgb = mean_squared_error(y_test, y_pred_xgb)
print(f"XGBoost - Mean Squared Error: {mse_xgb}")
```

18

Реалізація алгоритмів забезпечення масштабованості та ефективності

```
Random Forest - Classification Report
              precision    recall  f1-score   support

           0       0.95      0.98      0.96      1950
           1       0.89      0.75      0.81       350

    accuracy                           0.94      2300
   macro avg       0.92      0.86      0.89      2300
weighted avg       0.94      0.94      0.94      2300

Random Forest - Accuracy: 0.94
Random Forest - Mean Squared Error: 0.06

XGBoost - Classification Report
              precision    recall  f1-score   support

           0       0.96      0.97      0.96      1950
           1       0.85      0.81      0.83       350

    accuracy                           0.94      2300
   macro avg       0.90      0.89      0.90      2300
weighted avg       0.94      0.94      0.94      2300

XGBoost - Accuracy: 0.94
XGBoost - Mean Squared Error: 0.06
```

Моделі Random Forest та XGBoost показали високу точність на тренувальних і тестових даних. Точність досягла 94%, що свідчить про здатність моделей ефективно розпізнавати стан системи.

Рекомендації після аналізу результатів дослідження

1. Інтеграція розроблених моделей машинного навчання, таких як Isolation Forest, автоенкодери, LSTM, BERT, Random Forest та XGBoost, з існуючими системами моніторингу може бути реалізована через RESTful API. Це забезпечить взаємодію моделей з реальними системами у режимі реального часу.
2. Впровадження механізмів автоматичного оновлення та тренування моделей на нових даних забезпечить їх актуальність та здатність адаптуватися до змін у поведінці системи.
3. Впровадження системи автоматичних сповіщень, яка буде інформувати відповідальних осіб про виявлені аномалії та прогнозовані проблеми.
4. Оптимізація продуктивності моделей, зокрема зменшення обчислювальних витрат через використання менш ресурсомістких моделей або оптимізацію гіперпараметрів, допоможе знизити навантаження на системи та забезпечити більш ефективну роботу.
5. Розширення наборів даних, використовуючи різні набори даних з виробничих систем для тренування та оцінки моделей.

Висновки

1. Модель Isolation Forest виявилася корисною для виявлення аномалій в метриках веб-сервісів. Вона здатна ефективно ідентифікувати відхилення в поведінці системи, що можуть вказувати на потенційні збої або безпекові інциденти. Автоенкодери, будучи складовою частиною нейронних мереж, показали свою ефективність у виявленні складних патернів і аномалій в даних. Завдяки своїй архітектурі вони можуть виявляти нелінійні залежності та забезпечувати більш глибокий аналіз.
2. LSTM моделі показали свою здатність ефективно працювати з часовими рядами даних, що є критичним для аналізу метрик продуктивності, які змінюються з часом.
3. Random Forest і XGBoost використовувалися для прогнозування стану системи веб-сервісів. Обидві моделі можуть ефективно використовуватися для моніторингу та забезпечення надійності веб-сервісів, що дозволяє вчасно виявляти та вирішувати проблеми

21

Висновки

4. Використання машинного навчання дозволяє аналізувати великі обсяги даних та виявляти складні патерни, які можуть бути непомітні при традиційному аналізі.
5. Машинне навчання дозволяє створювати системи, які не тільки реагують на вже наявні проблеми, але й прогнозують можливі інциденти, що дозволяє приймати проактивні заходи для їх запобігання.
6. Апробація результатів: Мандрика М. Ревенчук І.А. Дослідження методів та інструментів мониторинга веб-сервісів: матер XV Міжнар наук.-практ конф. "Innovative Approaches to the Progressive Solution of Scientific Research Problems". Іспанія 27-29.03.2024. P.59-61. (https://isu-conference.com/arkhiv/innovative-approaches-to-the-progressive-solution-of-scientific-research-problems/).

22

ДОДАТОК Г

Апробація результатів роботи

Мандрика М. Ревенчук І.А. Дослідження методів та інструментів мониторинга веб-сервісів: матер XV Міжнар наук.-практ конф. "Innovative Approaches to the Progressive Solution of Scientific Research Problems". Іспанія 27-29.03.2024. P.59-61. (https://isu-conference.com/arkhiv/innovative-approaches-to-the-progressive-solution-of-scientific-research-problems/).

ДОДАТОК Д

Експертний висновок результатів перевірки кваліфікаційної роботи на відповідність оформлення вимогам ДСТУ 3008: 2015

Експертний висновок результатів перевірки кваліфікаційної роботи

студент (посада) | програмної інженерії (кафедра) | ІПЗм-22-4 (група)

Мандрика Максим Сергійович

(прізвище, ім'я, по батькові)

Зауваження

Пункт ДСТУ 3008-2015	Зміст пункту	Сторінка кваліфікаційної роботи
1	2	3
	7.1 Загальні положення	
	7.3 Нумерація сторінок звіту	
	7.4 Нумерація розділів, підрозділів, пунктів, підпунктів	
	7.5 Рисунки	
	7.6 Таблиці	
	7.7 Переліки	
	7.8 Примітки	
	7.9 Виноски	
	7.10 Формули та рівняння	
	7.11 Посилання	
	7.13 Список авторів	
	7.14 Скорочення та умовні познаки	
	7.15 Додатки	

зауважень немає

Експерт _____ (підпис) Олена ОЛІЙНИК (прізвище, ініціали)

11.06.2024