

## ДОДАТОК А

## СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА ЗА ТЕМОЮ ДИСЕРТАЦІЇ

<i>Наукові праці, в яких опубліковано основні наукові результати:</i>	
1.V. Ascherkov, "Methods of machine learning in modern metrology," <i>Measuring Equipment and Metrology</i> , vol. 85, no. 1, pp. 57–60, 2024.	
Тип публікації	Публікація в українському виданні, категорія Б
Рік	2024
Ключові слова	Machine learning, metrology, measurement accuracy, instrument calibration, data analysis, forecasting.
DOI	10.23939/istcmtm2024.01
Одноосібне авторство	так
Містить державну таємницю / службову інформацію	ні
Посилання	<a href="https://doi.org/10.23939/istcmtm2024.01">https://doi.org/10.23939/istcmtm2024.01</a>
2.В.О. Ащепков, "Використання моделі ISOLATION FOREST для виявлення аномалій у даних вимірювань," <i>Сучасний стан наукових досліджень та технологій в промисловості</i> , no. 1 (27), pp. 98–113, 2024.	
Тип публікації	Публікація в українському виданні, категорія Б
Рік	2024
Ключові слова	невизначеність, виявлення аномалій, вимірювання, метрологія, обробка даних, алгоритми машинного навчання, статистичні методи.
DOI	10.30837/ITSSI.2023.26
Одноосібне авторство	так
Містить державну таємницю / службову інформацію	ні
Посилання	<a href="https://doi.org/10.30837/ITSSI.2023.26">https://doi.org/10.30837/ITSSI.2023.26</a>
3. В.О. Ащепков, "Дослідження метрологічних характеристик державного первинного еталона одиниці об'ємної та масової витрати рідини при підготовці до участі у міжнародних звіреннях," <i>Український метрологічний журнал</i> , no. 1 (77), pp. 31–37, 2024	
Тип публікації	Публікація в українському виданні, категорія А
Рік	2024
Ключові слова	метрологічні характеристики, міжнародні звірення, стандартна невизначеність, витрата

	рідини, державний еталон.
DOI	10.24027/2306-7039.1.2024.300937
Одноосібне авторство	так
Містить державну таємницю / службову інформацію	ні
Посилання	<a href="https://doi.org/10.24027/2306-7039.1.2024.300937">https://doi.org/10.24027/2306-7039.1.2024.300937</a>
4.V. Ascherkov, "Methods for outlier detection in metrological studies," <i>Measuring Equipment and Metrology</i> , vol. 85, no. 3, pp. 25–29, 2024	
Тип публікації	Публікація в українському виданні, категорія Б
Рік	2024
Ключові слова	<u>metrology</u> , <u>outliers</u> , <u>anomalies</u> , <u>uncertainty</u> , <u>error</u> , <u>Isolation Forest method</u> , <u>robust methods</u> .
DOI	10.23939/istcmtm2024.03.025
Одноосібне авторство	так
Містить державну таємницю / службову інформацію	ні
Посилання	<a href="https://doi.org/10.23939/istcmtm2024.03.025">https://doi.org/10.23939/istcmtm2024.03.025</a>
5.В.О. Ащепков, Д.Ю. Бяллович, В.В. Склярів, "Вплив порогових значень на стандартну невизначеність типу А при вимірюваннях масової витрати рідини," <i>Український метрологічний журнал</i> , no. 3 (30), 2024	
Тип публікації	Публікація в українському виданні, категорія А
Рік	2024
Ключові слова	метрологічні дослідження, вимірювання, викиди, аномалії, невизначеність, похибка, порогові значення, метод ізольованого лісу, робастні методи.
DOI	10.24027/2306-7039.3.2024.312469
Одноосібне авторство	ні
Містить державну таємницю / службову інформацію	ні
Посилання	<a href="https://doi.org/10.24027/2306-7039.3.2024.312469">https://doi.org/10.24027/2306-7039.3.2024.312469</a>
<i>Наукові праці, які засвідчують апробацію матеріалів дисертації:</i>	
1. V. Ascherkov, "Improving the efficiency of processing of measurement results using the machine learning method," in <i>Theses of Reports 1st European Competition of Young Best Metrologists in Ukraine</i> , Ivano-Frankivsk, Ukraine, June 24–28, 2024, pp. 6–8.	
Тип публікації	Тези доповідей конференції
Рік	2024

Ключові слова	machine learning, isolated forest, anomaly detection, processing of measurement results, metrology, statistical method, measurement error, measurement uncertainty.
DOI	-
Одноосібне авторство	так
Містить державну таємницю / службову інформацію	ні
Посилання	<a href="https://www.ifdcsm.com.ua/uk/news/577/vidbuvsia-pershyyi-ievropeiskyi-konkurs-naikrashchykh">https://www.ifdcsm.com.ua/uk/news/577/vidbuvsia-pershyyi-ievropeiskyi-konkurs-naikrashchykh</a>

**ДОДАТОК Б**  
**ДОКУМЕНТИ, ЩО ПІДТВЕРДЖУЮТЬ**  
**ВПРОВАДЖЕННЯ РЕЗУЛЬТАТІВ**



«ЗАТВЕРДЖУЮ»

В.о. генерального директора  
Національного наукового центру  
«Інститут метрології»

Володимир СКЛЯРОВ

*12 листопада* 2024р.

### АКТ

про впровадження результатів дисертаційної роботи на тему:  
«Обробка результатів вимірювань витрати рідини з використанням  
машинного навчання»

здобувача кафедри Інформаційно-вимірювальних технологій  
Харківського національного університету радіоелектроніки  
Ащепкова Валерія Олеговича

Молодшим науковим співробітником ННЦ «Інститут метрології» Ащепковим Валерієм Олеговичем було проведено розробку та аналіз алгоритму на основі моделі машинного навчання «Ізольований ліс» для знаходження викидів у метрологічних дослідженнях вимірювання витрати рідини.

Розроблений алгоритм дозволяє автоматично ідентифікувати та виключати аномальні значення у вимірювальних даних, що особливо важливо для забезпечення високої точності і стабільності під час метрологічних досліджень. Це, у свою чергу, сприяє зниженню похибок і підвищенню надійності результатів, що особливо важливо для забезпечення достовірності під час проведення міжнародних звірень.

Результати дисертаційної роботи впроваджені та використовуються під час проведення міжнародних звірень за напрямком масової та об'ємної витрати рідини на «Державному первинному еталоні одиниці об'ємної і масової витрати рідини та об'єму і маси рідини, що протікає по трубопроводу» (ДЕТУ 03-04-04).

В.о. директора НЦ-3

Дмитро БЯЛЛОВИЧ

Заступник генерального директора  
з науково-метрологічної роботи  
д.ф.-м.н., проф.

Олександр ПРОКОПОВ



«ЗАТВЕРДЖУЮ»

Проректор з наукової роботи  
Харківського національного  
університету радіоелектроніки

Юрій РОМАНЕНКОВ

*12 листопада* 2024р.

### АКТ

про впровадження результатів дисертаційної роботи на тему:  
«Обробка результатів вимірювань витрати рідини з використанням  
машинного навчання»

здобувача кафедри Інформаційно-вимірювальних технологій  
Харківського національного університету радіоелектроніки  
Ащепкова Валерія Олеговича

Ми, що нижче підписалися, д.т.н., с.н.с. Склярів В.В., завідувач кафедри Інформаційно-вимірювальних технологій, д.т.н., проф. Захаров І.П., склали цей акт про те, що результати дисертаційної роботи на здобуття ступеня доктора філософії Ащепкова Валерія Олеговича впроваджено в навчальний процес Харківського національного університету радіоелектроніки на кафедрі Інформаційно-вимірювальних технологій. Розроблений у дисертації підхід до обробки результатів вимірювань витрати рідини під час міжнародних звірень використовується як актуальний метод аналізу вимірювальних даних у дисципліні "Державні еталони України". Він сприяє набуттю студентами практичних навичок роботи з сучасними вимірювальними технологіями, доповнюючи навчальний процес і підвищуючи обізнаність щодо сучасних методів метрологічного контролю та аналізу даних.

Завідувач кафедри Інформаційно-  
вимірювальних технологій

Ігор ЗАХАРОВ

Професор кафедри Інформаційно-  
вимірювальних технологій

Володимир СКЛЯРОВ

«ЗАТВЕРДЖУЮ»

Директор

ПрАТ «Енергооблік»

Ю.н. А.А. Стеценко

13 листопада 2024р.

**АКТ**

**про використання результатів дисертаційної роботи на тему:  
«Обробка результатів вимірювань витрати рідини з використанням  
машинного навчання»**

**здобувача кафедри Інформаційно-вимірювальних технологій  
Харківського національного університету радіоелектроніки  
Ащепкова Валерія Олеговича**

Розглянувши представлені матеріали дисертаційної роботи аспіранта Ащепкова В.О. та його наукові публікації у фахових виданнях, дійшли висновку, що одержані в ході досліджень результати та висновки мають науковий, теоретичний та практичний інтерес. Це стосується, зокрема:

- використання результатів дослідження для аналізу експериментальних даних під час розробки нових моделей витратомірів-лічильників, що дозволить ідентифікувати потенційні джерела похибок та покращити конструкцію пристроїв на етапі розробки;
- забезпечення стабільності роботи проливних установок завдяки виявленню аномалій, що сприятиме швидкому виявленню відхилень від норми, знижуючи ризик виходу обладнання з ладу та зменшуючи кількість позапланових обслуговувань;
- можливості інтеграції алгоритму на основі моделі машинного навчання «Ізольований ліс» у систему контролю якості на виробництві, що дозволить автоматично виявляти аномалії у вимірювальних даних, підвищуючи надійність і точність результатів.

Отримані результати та розроблені методи будуть використані для покращення якості продукції та оптимізації виробничих процесів. Цей акт не є підставою для одержання будь-якої матеріальної вигоди з боку сторін.

Головний метролог

Науменко В.О.

## ДОДАТОК В

### ПРОГРАМНИЙ КОД СТАНДАРТНОЇ МОДЕЛІ ISOLATIONFOREST

**# Імпорт необхідних бібліотек**

```
import pandas as pd
from sklearn.ensemble import IsolationForest
import numpy as np
import matplotlib.pyplot as plt
```

**# Считування даних з файлу Excel**

```
data = pd.read_excel("1.xlsx")
```

**# Визначення параметрів моделі**

```
n_estimators = 1000 # кількість дерев у ансамблі
max_samples = 'auto' # кількість прикладів для навчання кожного дерева
contamination = 0.25 # очікувана частка викидів у даних
num_runs = 10 # кількість запусків моделі
```

**# Цикл для обробки кожного стовбця з файлу "1.xlsx"**

```
for idx, (column_name, column_data) in enumerate(data.items(), start=1):
```

**# Пропускаємо стовбці з порожніми значеннями**

```
if column_data.isnull().all():
```

```
    continue
```

**# Зчитування даних з поточного стовбця**

```
current_data = column_data.dropna()
```

**# Список для зберігання результатів**

```
results = []
```

**# Запускаємо модель num\_runs разів**

```
for _ in range(num_runs):
```

**# Генерація випадкового значення random\_state для кожного запуску**

```
random_state = np.random.randint(1000)
```

**# Ініціалізація моделі Isolation Forest**

```
model = IsolationForest(n_estimators=n_estimators,
```

```

        max_samples=max_samples,
        contamination=contamination,
        random_state=random_state)

# Навчання моделі
model.fit(current_data.values.reshape(-1, 1))

# Передбачення викидів (аномалій)
predictions = model.decision_function(current_data.values.reshape(-1, 1))

# Додавання результатів до списку
results.append(predictions)

# Перетворення результатів в DataFrame
results_df = pd.DataFrame(results, columns=current_data.index)

# Обчислення середнього значення для кожної точки
mean_results = results_df.mean(axis=0)

# Створення DataFrame з даними для зберігання
result_df = pd.DataFrame({
    'Ступінь аномальності': mean_results,
    'Признак аномальності': [1 if x < 0 else -1 for x in mean_results]
})

# Зберігання DataFrame у файл Excel
output_file_name = f"col_{idx}.xlsx"
result_df.to_excel(output_file_name, index=False)

# Передбачення міток класів для кожної точки даних
labels = model.predict(current_data.values.reshape(-1, 1))

# Створення сітки графіків
fig, axs = plt.subplots(1, 2, figsize=(12, 5))

# Графік значень поточного стовпця з відміткою аномалій
for i in range(len(current_data)):
    color = '#E90D0D' if labels[i] == -1 else '#198806' # Якщо точка аномальна,
    то червоний, інакше зелений
    axs[0].scatter(i + 1, current_data.iloc[i], color=color, marker='o')

```

```

axs[0].scatter([], [], color='#E90D0D', label='Аномальні точки', marker='o')
axs[0].scatter([], [], color='#198806', label='Нормальні точки', marker='o')
axs[0].legend()

# Встановлення позначок для осі x без інтервалу
x_ticks = range(1, len(current_data) + 1) # Значення від 1 до кількості вимірювань
вань
axs[0].set_xticks(x_ticks[::1]) # Встановлення позначок для значень з кроком
1
axs[0].set_xticklabels([str(i) for i in x_ticks[::1]]) # Встановлення міток з цифрами з кроком 1
axs[0].set_xlabel('Номер вимірювання')
axs[0].set_ylabel('Значення відносної похибки')
axs[0].set_title(f'Значення відносної похибки в точці витрати {column_name}')

# Графік аномальності точок (усереднений по 10 запускам)
for i in range(len(mean_results)):
    color = '#E90D0D' if labels[i] == -1 else '#198806' # Якщо точка аномальна,
то червоний, інакше зелений
    axs[1].scatter(i + 1, mean_results[i], color=color, marker='o')

# Показуємо маркер для нормальних і аномальних точок на другому графіку
axs[1].scatter([], [], color='#E90D0D', label='Аномальні точки', marker='o')
axs[1].scatter([], [], color='#198806', label='Нормальні точки', marker='o')
axs[1].legend()

# Встановлення позначок для осі x без інтервалу
x_ticks = range(1, len(mean_results) + 1) # Значення від 1 до кількості вимірювань
вань
axs[1].set_xticks(x_ticks[::1]) # Встановлення позначок для значень з кроком
1

```

```
axs[1].set_xticklabels([str(i) for i in x_ticks[:,1]]) # Встановлення міток з циф-  
рами з кроком 1  
axs[1].set_xlabel('Номер вимірювання')  
axs[1].set_ylabel('Коефіцієнт аномальності')  
axs[1].set_title(f'Ступінь аномальності в точці витрати {column_name}')  
plt.tight_layout()  
plt.show()
```

**ДОДАТОК Г**  
**ПРОГРАМНИЙ КОД АЛГОРИТМУ НА ОСНОВІ МОЕЛІ**  
**ІЗОЛЬОВАНОГО ЛІСУ**

**# Імпорт необхідних бібліотек**

```
import pandas as pd
from sklearn.ensemble import IsolationForest
import numpy as np
import matplotlib.pyplot as plt
```

**# Зчитування даних з файлу Excel**

```
data = pd.read_excel("111.xlsx")
```

**# Параметри моделі Isolation Forest**

```
n_estimators = 1000 # кількість дерев в ансамблі
max_samples = 'auto' # кількість прикладів для навчання кожного дерева
contamination_values = np.arange(0.01, 0.5, 0.05) # діапазон значень
contamination
num_runs = 10 # кількість запусків для усереднення
```

**# Обробка кожного стовпця даних**

```
for idx, (column_name, column_data) in enumerate(data.items(), start=1):
    if column_data.isnull().all():
        continue
    current_data = column_data.dropna().values.reshape(-1, 1)
    scores_per_contamination = [] # для збереження результатів для кожного значення contamination
```

**# Запуски моделі з різними значеннями contamination**

```
for contamination in contamination_values:
    scores_per_run = [] # для збереження результатів кожного запуску при поточному значенні contamination
```

**# Запускаємо модель 10 разів для поточного значення contamination**

```
for _ in range(num_runs):
    random_state = np.random.randint(1000)
```

```

model = IsolationForest(
    n_estimators=n_estimators,
    max_samples=max_samples,
    contamination=contamination,
    random_state=random_state
)
model.fit(current_data)
scores = -model.decision_function(current_data) # ступінь аномальності
для поточного запуску
scores_per_run.append(scores) # зберігаємо результат поточного запуску
# Усереднюємо результати 10 запусків для поточного значення
contamination
mean_scores = np.mean(scores_per_run, axis=0)
scores_per_contamination.append(mean_scores) # зберігаємо усереднений
результат для поточної чутливості
# Перетворюємо список результатів у масив для зручності
scores_per_contamination = np.array(scores_per_contamination)
# Усереднення всіх значень за всіма contamination
final_scores = np.mean(scores_per_contamination, axis=0)
# Сортуємо значення ступеню аномальності за зростанням і знаходимо
60% точок
sorted_indices = np.argsort(final_scores)
sorted_scores = final_scores[sorted_indices]
# Розрахунок відстаней у перших 60% точок за зростанням
sixty_percent_index = int(0.6 * len(sorted_scores))
first_60_percent_distances = np.diff(sorted_scores[:sixty_percent_index])
max_first_60_percent_distance = np.max(first_60_percent_distances)
# Пошук порогу у другій половині вибірки
threshold_index = None
for i in range(sixty_percent_index, len(sorted_scores) - 1):

```

```

current_distance = sorted_scores[i + 1] - sorted_scores[i]
# Перевірка на умову відстані та значення другої точки
if current_distance >= 2 * max_first_60_percent_distance and sorted_scores[i +
1] > 0.05:
    threshold_index = i
    break # Зупиняємося, коли знайдено коректний поріг
# Визначення порогового значення
if threshold_index is not None:
    threshold_value = (sorted_scores[threshold_index] +
sorted_scores[threshold_index + 1]) / 2
else:
    threshold_value = None # Якщо викидів немає або не знайдено порогового
значення
# Створення підграфіків
fig, (ax1, ax2, ax3) = plt.subplots(1, 3, figsize=(24, 6))
# Графік для вхідних даних
ax1.scatter(range(1, len(column_data.dropna().values) + 1),
column_data.dropna().values, label='Вхідні дані',
color='green')
ax1.set_xlabel('Номер вимірювання')
ax1.set_ylabel('Відносна похибка вимірювань')
ax1.set_title(f'Точка витрати {column_name}')
ax1.legend()
# Графік у початковому порядку
ax2.scatter(range(1, len(final_scores) + 1), final_scores, label='Ступінь аномаль-
ності', color='blue')
if threshold_value is not None:
    ax2.axhline(y=threshold_value, color='red', linestyle='--', label=f'Поріг (зна-
чення={threshold_value:.2f})')
ax2.set_xlabel('Номер вимірювання')

```

```

ax2.set_ylabel('Ступінь аномальності')
ax2.set_title(f'Ступінь аномальності для {column_name}')
ax2.legend()
# Графік за зростанням
ax3.scatter(range(1, len(sorted_scores) + 1), sorted_scores, label='Ступінь аномальності', color='blue')
if threshold_value is not None:
    ax3.axhline(y=threshold_value, color='red', linestyle='--', label=f'Поріг (значення={threshold_value:.2f})')
ax3.set_xlabel('Номер вимірювання (за зростанням)')
ax3.set_ylabel('Ступінь аномальності')
ax3.set_title(f'Ступінь аномальності для {column_name}')
ax3.legend()
plt.tight_layout()
plt.show()
# Виведення порогового значення або повідомлення про його відсутність
if threshold_value is not None:
    print(f'Порогове значення для {column_name}: {threshold_value:.4f}')
else:
    print(f'Викидів немає для {column_name} або не знайдено коректного порогового значення")

```