Вінницький національний технічний університет

Факультет менеджменту та інформаційної безпеки

Кафедра менеджменту та безпеки інформаційних систем

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

Удосконалення модуля виявлення аномалій ICS/SCADA-систем шляхом прогнозування загроз із використанням LSTM-моделі машинного навчання

Виконав: здобувач 2-го курсу,
групи 2КІТС-24м
спеціальності 125– Кібербезпека
та захист інформації
Освітня програма – Кібербезпека
інформаційних технологій та систем

2КІТС-24м Вязун Дмитро Сергійович

Керівник:
Грицак Анатолій Васильович
«____» _____ 2025 р.

Опонент:
Тарновський Микола Геннадійович
«____» _____ 2025 р.

Допущено до захисту
Голова секції УБ кафедри МБІС

_____ Юрій ЯРЕМЧУК
“_____” _____2025 р.

Вінниця ВНТУ - 2025 рік

Вінницький національний технічний університет
Факультет менеджменту та інформаційної безпеки
Кафедра менеджменту та безпеки інформаційних систем

Рівень вищої освіти ІІ-й (магістерський)
Галузь знань 12 – Інформаційні технології
Спеціальність 125 – Кібербезпека та захист інформації
Освітньо-професійна програма - Кібербезпека інформаційних технологій та систем

ЗАТВЕРДЖУЮ
Голова секції УБ, кафедра МБІС

_____ **Юрій ЯРЕМЧУК**
“_____” _____ 2025 р.

З А В Д А Н Н Я
на магістерську кваліфікаційну роботу студенту
Вязун Дмитро Сергійович
(прізвище, ім`я, по-батькові)

1. Тема роботи Удосконалення модуля виявлення аномалій ICS/SCADA-систем шляхом прогнозування загроз із використанням LSTM-моделі машинного навчання
Керівник роботи Грицак Анатолій Васильович, к.т.н., доцент кафедри
(прізвище, ім`я, по-батькові, науковий ступінь, вчене звання)
затверджені наказом вищого навчального закладу від “24” вересня 2025 року № 313
2. Строк подання студентом роботи за тиждень до захисту
3. Вихідні дані до роботи: електронні джерела, наукові статті, існуюче програмне забезпечення, технічна документація
4. Зміст текстової частини: Вступ. Розділ 1. Теоретичний аналіз підвищення ефективності прогнозування загроз вдосконаленим модулем виявлення аномалій ics/scada-систем
. Розділ 2. Проектування модулю виявлення аномалій ics/scada-систем шляхом прогнозування загроз із використанням машинного навчання. Розділ 3. Програмна реалізація алгоритму. Висновки. Джерела. Додатки.
5. Перелік ілюстративного матеріалу (з точним зазначенням обов`язкових креслень) Поетапна реалізація проекту.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Основна частина			
I	**Грицак А.В., к.т.н., доцент кафедри**		
II	**Грицак А.В., к.т.н., доцент кафедри**		
III	**Грицак А.В., к.т.н., доцент кафедри**		
Економічна частина			
IV	**Ратушняк О.Г., доцент кафедри ЕПВМ, к.т.н.**		

7.Дата видачі завдання 24 вересня 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів магістерської кваліфікаційної роботи	Строк виконання етапів роботи		Примітка
1	**Вибір та узгодження теми**	**24.09.2025**	**30.09.2025**	
2	**Аналіз літературних джерел**	**01.10.2025**	**14.10.2025**	
3	**Побудова моделі та блок-схеми алгоритму**	**15.10.2025**	**10.11.2025**	
4	**Експерименти, тестування**	**11.11.2025**	**23.11.2025**	
5	**Оформлення**	**24.11.2025**	**28.11.2025**	
6	**Підготвока до захисту**	**29.11.2025**	**02.12.2025**	

Студент _____ Вязун Д.С.
(підпис)

Керівник роботи _____ Грицак А.В.
(підпис)

АНОТАЦІЯ

УДК 004.056.5

Вязун Д.С Удосконалення модуля виявлення аномалій ICS/SCADA-систем шляхом прогнозування загроз із використанням LSTM-моделі машинного навчання. Магістерська кваліфікаційна робота зі спеціальності 125 – «Кібербезпека», освітня програма «Кібербезпека інформаційних технологій та систем». Вінниця: ВНТУ, 2025. 93 с.

Бібліогр.: 32 назв; рис.: 12; табл. 6.

У магістерській кваліфікаційній роботі розглянуто проблему підвищення стійкості ICS/SCADA-систем до кіберзагроз шляхом виявлення поведінкових аномалій у технологічних процесах. Проведено аналіз архітектури промислових систем, їхніх вразливостей та сучасних методів захисту. Запропоновано модуль виявлення аномалій, що поєднує рекурентну LSTM-модель і повнозв'язні мережі для обробки різних типів промислових даних. Описано етапи підготовки датасету, побудову моделі та результати її навчання.

Експериментально підтверджено можливість застосування такого підходу для раннього виявлення відхилень у роботі технологічного обладнання. Наведено оцінку економічної доцільності розробки.

Ключові слова: ICS/SCADA, виявлення аномалій, часові ряди, LSTM-модель, кіберзагрози, машинне навчання.

SUMMARY

UDC 004.056.5

Vyazun D.S. Improvement of an ICS/SCADA Anomaly Detection Module through Threat Prediction Using an LSTM Machine Learning Model. Master's Thesis in specialty 125 – Cybersecurity, educational program Cybersecurity of Information Technologies and Systems. Vinnytsia: VNTU, 2025. 93 p.

Bibliography: 32 titles; figures: 12; tables: 6.

The thesis addresses the problem of increasing the resilience of ICS/SCADA systems to cyber threats by detecting behavioral anomalies within technological processes. The work examines the architecture of industrial control systems, typical vulnerabilities, and contemporary protection approaches. A hybrid anomaly detection module is proposed, combining a recurrent LSTM model with fully connected neural networks for processing heterogeneous industrial data. The stages of dataset preparation, model development and training are described.

The experiments demonstrate that the proposed approach can be applied for early detection of deviations in the operation of industrial equipment. An assessment of the economic feasibility of the solution is also provided.

Keywords: ICS/SCADA, anomaly detection, time series, LSTM model, cyber threats, machine learning.

ЗМІСТ

ВСТУП

Актуальність роботи. Упродовж останніх років промислові підприємства все активніше переходять до цифрових технологій, поєднуючи виробниче обладнання з мережевими сервісами та системами автоматизації. Такі зміни значно підвищують ефективність роботи, але це збільшує залежність об'єктів критичної інфраструктури від стану інформаційних систем.

ICS/SCADA-комплекси, на яких тримається керування технологічними процесами, не були створені з урахуванням сучасних кіберзагроз, тому питання їхнього захисту зараз набуває особливої актуальності. Будь-яка помилка або атака в таких системах здатна призвести не лише до збоїв у роботі обладнання, а й до реальних небезпечних наслідків для людей, довкілля та економіки.

У цій ситуації увага дослідників зміщується від класичних методів захисту до підходів, які здатні працювати з поведінковими характеристиками системи та передбачати появу відхилень завчасно. На практиці це означає, що інструменти кіберзахисту повинні не просто фіксувати вже відомі типи атак, а й вміти аналізувати зміни у технологічних процесах, які можуть стати першими ознаками аномалії. Методи машинного навчання виявились найбільш придатними для цього, оскільки дозволяють вивчати структуру промислових даних та виявляти нетипові тенденції, які складно описати правилами.

Метою роботи є розроблення та дослідження модуля виявлення аномалій у ICS/SCADA-системах на основі моделей глибинного навчання. Особливий акцент зроблено на використанні LSTM-архітектури, яка здатна враховувати часову залежність параметрів технологічних процесів. Такий підхід дає можливість прогнозувати нормальну поведінку системи та виявляти відхилення за рівнем помилки реконструкції.

Для досягнення поставленої мети потрібно вирішити наступні задачі:

– провести аналіз особливостей промислових систем керування та сучасних підходів до їх захисту;

– підготувати вхідні дані, які поєднують мережеві журнали, операторські події та технологічні сигнали;

– побудувати та провести навчання моделі нейронної мережі;

– порівняти отримані результатів із іншими методиками.

Об'єктом дослідження є процеси обробки телеметричних та мережевих даних у промислових системах керування.

Предметом дослідження є методи виявлення аномалій у поведінці ICS/SCADA-систем на основі аналізу часових рядів та машинного навчання.

Наукова новизна роботи полягає у використанні гібридного підходу до виявлення загроз, де часові та статистичні характеристики об'єднані в єдину модель.

Практичне значення полягає в можливості адаптувати створений модуль до різних типів промислових процесів і використовувати його як додатковий інструмент моніторингу для операторів.

У роботі застосовуються такі *методи,* як аналіз наукових джерел, методи попередньої обробки даних, методи моделювання поведінки системи, методи оцінювання ефективності.

Результати дослідження можуть бути корисними для фахівців, які займаються впровадженням систем безпеки на промислових об'єктах, а також для подальших наукових розробок у напрямку прогнозування кіберзагроз.

1 ТЕОРЕТИЧНИЙ АНАЛІЗ ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ ПРОГНОЗУВАННЯ ЗАГРОЗ ВДОСКОНАЛЕНИМ МОДУЛЕМ ВИЯВЛЕННЯ АНОМАЛІЙ ICS/SCADA-СИСТЕМ

1.1 Особливості функціонування та архітектури ICS/SCADA-систем

Промислові системи керування (Industrial Control Systems, ICS) займають ключове місце в сучасних виробничих процесах, забезпечуючи автоматизацію, моніторинг та управління технологічними об'єктами. До складу таких систем належать підсистеми різного рівня, серед яких особливе місце займають SCADA-системи (Supervisory Control and Data Acquisition), призначені для диспетчерського контролю та збору даних з віддалених об'єктів у реальному часі [1]. Їх основна мета полягає у підвищенні ефективності виробництва, безпеки технологічних процесів і мінімізації людського фактору.

ICS/SCADA-системи контролюють об’єкти критичної інфраструктури, де ключовими факторами є безперервність процесів, висока доступність та надійне керування фізичними параметрами, що виділяє їх серед інформаційних систем тому що ці системи мають більш жорсткі вимоги до часу реагування і стабільності роботи порівняно з типовими IT-системами [1].

Типова система SCADA складається з кількох ключових компонентів, які працюють у поєднанні для забезпечення безперервного управлінського досвіду [2]. Сюди входять:

1. Дистанційні Термінальні Пристрої (RTUs) – це апаратні пристрої, які розташовані у різних точках промислових процесів для збору даних з сенсорів і актуаторів. Ці пристрої відіграють важливу роль у перетворенні сигналів сенсорів в цифрові дані та їх передачі до центральної системи SCADA для обробки.

2. Програмовані Логічні Контролери (PLCs) використовуються для збору даних в реальному часі та контролю процесів у промислових умовах. PLCs потрібні там, де є вимоги до високошвидкісного аналізу даних та контролю.

3. Інтерфейс Людина-Машина (HMI) служить зв’язком між операторами та системою SCADA та відображає оброблені дані через графічні інтерфейси. Такий підхід дозволяє операторам відстежувати процеси, виявляти наявні відхилення та коректувати, якщо в цьому є необхідність.

4. У центрі SCADA-системи знаходиться централізована система контролю, яка, зазвичай, складається з одного або більше серверів і програмних рішень, розроблених для збору, аналізу та управління даними, отриманими з RTUs і PLCs. Це дозволяє операторам, які приймають рішення, переглядати агреговані дані, аналізувати тенденції та приймати обґрунтовані операційні рішення.

Взаємодія між рівнями здійснюється через промислові мережі, що використовують спеціалізовані комунікаційні протоколи. Традиційні SCADA-системи створювалися в умовах ізольованих виробничих мереж і зазвичай не включали механізми криптографічного захисту або автентифікації користувачів. Наприклад, протокол Modbus TCP передає дані у відкритому вигляді, що дозволяє реалізацію атак типу man-in-the-middle при доступі до мережі [3]. Сучасні протоколи, такі як Secure DNP3 або OPC UA, мають більш розвинуті засоби безпеки, такі, як шифрування, автентифікацію, контроль доступу. Але, зазвичай, у багатьох об'єктах досі використовуються застарілі пристрої й програмне забезпечення без цих можливостей [4,5].

Розвиток SCADA-систем показує поступову інтеграцію з інформаційними мережами та удосконалення комунікаційних технологій (рис.1.1). У першому поколінні (так звані монолітні системи) усі компоненти працювали в межах однієї апаратної інфраструктури без зв'язку з зовнішнім середовищем. У розподілених SCADA (це друге покоління) з'явилася можливість об'єднання кількох підсистем через локальні мережі. Далі вже з’явилися SCADA-системи, які використовують стандартні IP-мережі. Це значно підвищило гнучкість подібних рішень, але водночас створило передумови для нових кіберзагроз [6]. Сучасні системи четвертого покоління інтегровані з технологіями Інтернету речей (IoT) та

хмарними обчисленнями, що забезпечує високу масштабованість і можливість аналітики великих даних, але також збільшує поверхню потенційних кібератак.

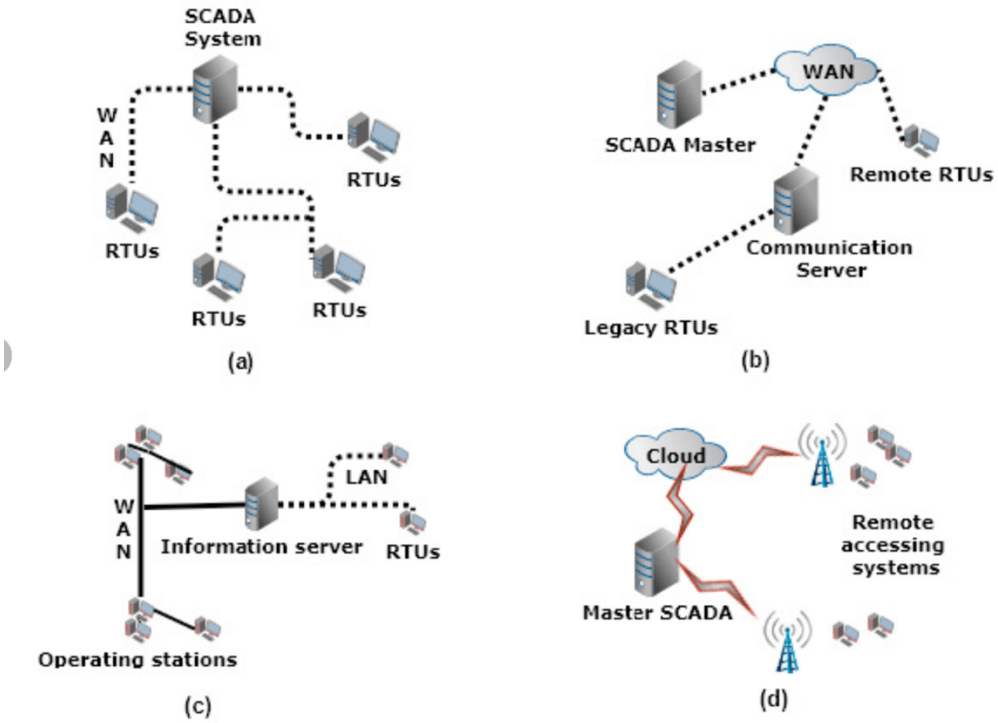


Рисунок 1.1 – Еволюція SCADA-систем: (a) Монолітні SCADA-системи з віддаленими термінальними блоками: перше покоління, (b) Розподілені SCADA-системи: друге покоління, (c) Мережева SCADA-система: третє покоління, (d) SCADA-система на основі IoT-Cloud: четверте покоління [7]

Особливістю сучасних ICS/SCADA-систем є поєднання різних технологічних платформ та поколінь обладнання. У переважної більшості промислових підприємств використовуються гібридні архітектури, у яких нові модулі з сучасними протоколами взаємодіють із застарілими PLC або RTU. Такі рішення створюють певні труднощі для збереження цілісності даних і ускладнюють роботу систем виявлення аномалій. Це пов'язано з тим, що обладнання може бути різних типів, а також із тим, що затримки під час

передавання даних впливають на точність аналізу та прогнозування поведінки процесів.

Важливою властивістю ICS є вимога до високої надійності та доступності. На відміну від звичайних IT-систем, де короткочасне переривання роботи допускається, у промислових системах навіть зупинка на декілька секунд може призвести до аварійних ситуацій або великих фінансових збитків. Тому при розробці модулів виявлення аномалій або прогнозування загроз необхідно враховувати обмеження щодо часу реагування, продуктивності та сумісності із фізичними процесами [8].

Крім того, ICS-системи часто функціонують у середовищах з обмеженими обчислювальними ресурсами. Контролери мають невеликий обсяг пам'яті, обмежену частоту процесора і вони не можуть виконувати складні алгоритми машинного навчання або шифрування. Це змушує дослідників переносити складні обчислювальні завдання на серверний рівень або використовувати гібридні архітектури, де буде здійснюватися первинний моніторинг, а глибокий аналіз відбувається вже централізовано.

Ще однією особливістю є тривалий життєвий цикл таких систем. Типовий план екологічного розвитку IT передбачає заміну серверів та іншого IT-обладнання приблизно кожні 5 років. Натомість, обладнання ICS має набагато довший життєвий цикл, до 30-50 років, а допоміжне програмне забезпечення охоплює кілька поколінь, від Windows 95 до поточних версій [9]. Такі системи важко оновлювати через ризик порушення технологічного процесу, тому оновлення, пов’язані з безпекою, часто відкладаються, а це збільшує ризики компрометації.

Таким чином, архітектура ICS/SCADA-систем характеризується багаторівневою структурою, високими вимогами до надійності та безперервності роботи, а також наявністю обмежень, пов'язаних з обчислювальними ресурсами та сумісністю протоколів. Врахування цих факторів є необхідною умовою для подальшого удосконалення модулів виявлення аномалій і прогнозування загроз.

1.2 Потенційні загрози та уразливості промислових систем керування

Однією з ключових уразливостей ICS/SCADA є використання протоколів, які не передбачають базових засобів безпеки. Наприклад, Modbus TCP передає дані у відкритому вигляді, без шифрування і автентифікації, що дозволяє зловмиснику, який має доступ до мережі, виконувати атаки типу «перехоплення даних» (eavesdropping) або «man-in-the-middle» [10]. У дослідженні [11] говориться, що протокол Modbus TCP/IP як стандарт для багатьох SCADA систем за дизайном не захищений через відсутність шифрування.

Ще одна серйозна загроза – це атаки з використанням підробки даних (false data injection) або «обману» контролюючих компонентів системи. У дослідженні [12] описано сценарії, коли зловмисники надсилали неправдиві вимірювання в систему енергоменеджменту. І оскільки канали передачі часто не захищені, такі підроблені дані можуть обійти механізми виявлення шуму або аномалій, особливо якщо у системі немає достатньої надмірності даних.

Вразливості часто надходять і від слабкого сегментування мереж OT/IT, відсутності адекватного контролю доступу та невідповідної конфігурації пристроїв. Згідно з оглядом [13], недостатня сегрегація дозволяє шкідливому програмному забезпеченню або інсайдерам проникати з мереж IT до OT, і навіть стандартні робочі місця, підключені до корпоративної мережі, можуть стати точками входу.

Крім того, існують загрози пов'язані з мережевими атаками. Наприклад, це може бути відмова в обслуговуванні (Denial of Service, DoS), сканування портів, а також використання вразливих сервісів чи відкритих портів. Такі атаки можуть призвести до втрати доступності чи функціоналу систем, що в ICS часто має дуже високі наслідки через реальний фізичний вплив.

Ще одна важлива група загроз – це загрози від зловмисного програмного забезпечення та атак на рівень програмної логіки. Прикладом є malware, яке спеціально проектують для PLC або RTU, яке може змінювати логіку керування чи

завантажувати шкідливе firmware. Наприклад, у роботі [14] автори наводять приклади того, як шкідливий код може використатися до контролерів у середовищі SCADA і що протоколи без захисту полегшують це.

Також, як і у всіх сферах, де актуальними є кіберзагрози, велике значення має людський фактор. Помилки операторів, неправильні конфігурації, недотримання процедур безпеки, відсутність політик регулярного оновлення – це все те, що дуже розповсюджено. Наприклад, в огляді [13] описано, що багато організацій не оновлюють ПЗ, через побоювання зупинки виробничих процесів та не мають чітких стратегій для оновлень.

Зростаюча інтеграція з зовнішніми мережами, віддалений доступ та підключення пристроїв через IP/бездротові канали також створюють нові вектори атак. У статті [15] говориться, що коли SCADA системи переходять від пропрієтарних чи ізольованих мереж до IP-базованих, збільшується ймовірність експлуатації вразливостей на рівні мережевої передачі, слабких точок доступу та недостатнього захисту протоколів.

1.3 Традиційні методи захисту та їхні обмеження

Якщо говорити про ICS/SCADA системи, то захист, зазвичай, спирався на базові засоби. Це може бути ізоляція мереж, фізичний контроль доступу, використання міжмережевих екранів, правила доступу, періодичне оновлення програмного забезпечення, а також системи виявлення та запобігання вторгнень IDS/IPS, які базуються на сигнатурному аналізі чи заданих правилах [15]. Ці методи багато років забезпечували та й досі забезпечують базовий рівень безпеки в умовах ізольованих чи частково ізольованих виробничих мереж, де кіберзагрози були менш розвинуті та менш поширені.

Однак з часом виявилось, що традиційні підходи мають суттєві обмеження, особливо в контексті сучасних SCADA/ICS, які вже інтегруються із корпоративними мережами, мережею Інтернет або позаміськими точками доступу.

Системи IDS/IPS, які використовують сигнатури, часто не можуть виявити нові атаки або zero-day загрози, бо для них немає попередніх шаблонів атаки. Це означає, що якщо зловмисник застосовує змінені або спеціально адаптовані методи, традиційні системи захисту можуть взагалі не визначити факт атаки або матимуть високу кількість хибно-позитивних (false positives) чи хибно-негативних (false negatives) спрацювань.

Через вимоги дуже високої надійності та безперервності процесів, виробничі системи рідко допускають прості чи часті оновлення або патчі програмного забезпечення, оскільки зміни можуть викликати зупинки обладнання або простої, що може бути надто дорогим або ризиковим. Унаслідок цього багато систем експлуатуються з застарілим ПЗ чи з відкритими вразливостями, які не були усунуті вчасно.

Багато традиційних методів захисту не враховують специфіку OT/ICS щодо часових затримок, ресурсних обмежень та особливостей поведінки пристроїв. Наприклад, використання міжмережевих екранів або шифрування можуть спричинити затримки, які недопустимі в умовах критичних технологічних процесів. Також ресурси PLC або RTU можуть бути надто обмеженими у питанні підтримки складних криптографічних алгоритмів або для повноцінної підтримки багаторівневого аудиту. Потенційно це знижує ефективність більш надійних захисних рішень у таких системах.

Ще одна проблема – це відсутність або неповнота стандартизації. Це й устарілі протоколи, і слабка або взагалі відсутня процедура автентифікації, це використання протоколів, які передають дані без шифрування або контролю доступу. Окремою проблемою стоїть використання стандартних паролей, які досі часто залишають «за замовчуванням». Наприклад, в статті [16] вказується, що під час розробки багатьох протоколів, акцент на безпеку не було зроблено бо автоматизація була більш важливим питанням.

Традиційні методи захисту часто не можуть впоратись із загрозами, які утворюються багатоступневими атаками або атаками на логіку процесів, де

зловмисник комбінує невеликі зміни по багатьох компонентах, щоб залишатися непомітним [12].

З усього цього можна зробити висновок, що традиційні методи захисту мають важливу роль, але вони не здатні в повному обсязі забезпечити захист сучасних ICS/SCADA, особливо в умовах швидких змін, нових типів атак і підвищених вимог до безпеки, доступності та реагування.

1.4 Сучасні підходи до виявлення аномалій

У попередньому підрозділі було зазначено, що сучасні промислові системи керування ставлять більші вимоги до надійності, безпеки та швидкого реагування і традиційні методи виявлення аномалій виглядають вже досить обмеженими і малоефективними, коли це стосується критичної інфраструктури. Сучасні підходи базуються на машинному навчанні (ML), глибокому навчанні (Deep Learning) та гібридних моделях, які здатні знаходити складні просторові і часові залежності, адаптуватись до змін у системі й зменшувати кількість хибних спрацьовувань.

Одним з таких підходів є застосування моделей LSTM-автоенкодерів та Sequence-to-Sequence (послідовність-послідовність) автоенкодерів із механізмами уваги (attention). У роботі [17] запропоновано модель, яка використовує Sequence-to-Sequence автоенкодер на базі LSTM, вбудований (embedding) шар, навчання з підказкою вчителя (teacher forcing) та механізм уваги. Ця модель навчена і протестована на даних SWaT (зменшена водоочисна станція) і показала високі показники виявлення атак, зокрема кращу чутливість (recall) порівняно з іншими існуючими моделями.

Також було розглянуто роботу [18], в якій запропоновано гібридну модель AMCNN-LSTM, що поєднує CNN блоки для виділення особливостей із LSTM для моделювання часових залежностей, а також підхід федеративного навчання, щоб розподілити навантаження та захистити конфіденційність даних.

Досить часто використовуються останнім часом гібридні моделі, які комбінують кілька підходів. Вони поєднують використання автоенкодерів, CNN, RNN, One-Class SVM, Isolation Forest тощо. Такий підхід дозволяє захоплювати як поведінкові, так і мережеві патерни аномалій. Так в статті [19] порівнюється CNN та LSTM підходи та показано, що гібридні рішення, які навчаються автоматично вибирати ознаки, часто перевищують стандартні статистичні чи сигнатурні методи.

Ще одним напрямком у розвитку систем виявлення аномалій є. оцінка невизначеності прогнозів. Сутність такої оцінки полягає в тому, щоб не тільки знаходити підозрілі відхилення, а й визначати те, наскільки модель впевнена в своєму прогнозі. У статті [20] розглядається підхід в якому звичайні моделі (Autoencoder, LSTM, Sequence-to-Sequence) були доповнені методами Monte Carlo Dropout та байєсівськими нейронними мережами. Такий підхід дозволив не тільки ідентифікувати аномалію, але й оцінити, наскільки модель впевнена в своєму прогнозі. Якщо система видає низьку впевненість, може розпочатися додаткова перевірка, а не відразу починається реєстрація аномалії. Це допомагає зменшити кількість хибних спрацювань.

Також описано так званий Lightweight підхід, який орієнтовано на середовище з обмеженими ресурсами або на такі, де немає можливості використовувати «важкі» моделі через затримки чи існуючі обмеження обчислювальної потужності. У статті [21] описано алгоритм ReRe для виявлення аномалій у потокових часових рядах. Він працює в реальному часі, легкий (lightweight) та не потребує ручного налаштування. Його спроєктовано для використання на звичайному обладнанні без потреби специфічних знань.

Однак навіть серед сучасних підходів існують деякі проблеми. Наприклад, багато моделей вимагають великих наборів даних, які будуть позначені як «нормальні» і «аномальні», а такі дані складно зібрати у реальних промислових середовищах без ризику порушення виробничих процесів. Окрім цього, адаптивність моделей до якихось змін в обладнанні чи в налаштуваннях системи

часто обмежена, а це може призвести до зниження точності або до збільшення кількості небажаних спрацювань.

Таким чином, сучасні підходи до виявлення аномалій у ICS/SCADA базуються на методах глибокого навчання та гібридних моделях, використовують механізми уваги, оцінки невизначеності та lightweight архітектури. Вони значно краще, ніж традиційні методи, але потребують обгрунтованого вибору архітектур, настроювання й адаптації під конкретні умови системи.

1.5 Аналіз існуючих рішень

Одним прикладів спеціалізованих рішень є OTNetGuard – сенсор пасивного моніторингу для ICS/SCADA-систем, який здатний захоплювати аналогові, серійні та IP-комунікації й таким чином надавати повний огляд середовища OT-мережі [22]. Перевага цього рішення полягає в тому, що воно охоплює й застарілі комунікації, які часто залишаються поза увагою традиційних систем безпеки. Проте, як і більшість систем подібного типу, OTNetGuard потребує окремого сенсора й налаштування, що може бути непросто в умовах вже діючого виробничого процесу.

В роботі [23] пропонується гібридний підхід із поєднанням 1D-CNN та BiLSTM для аналізу вимірюваних даних (не лише мережевого трафіку) та оптимізації гіперпараметрів за допомогою PSO-алгоритму. Цей підхід демонструє високі результати в тестових наборах даних та підкреслює важливість роботи з самими сигналами контролю (sensors/actuators) замість виключно мережевого аналізу. Обмеженням є те, що тестування проходило на симульованому наборі даних, який може не відображати всіх нюансів реальної інженерної системи.

MADICS - методологія для виявлення аномалій у ICS побудована на напів-керованому машинному навчанні і призначена саме під особливості промислових систем [24]. Методологія включає кроки підготовки даних, відбору ознак, екстракції високорівневих ознак, вибір моделі та валідацію. Завдяки цьому

MADICS досягає високої точності на тестовому майданчику SWaT. Недоліком є необхідність наявності меток нормальної та атакованої поведінки системи, що не завжди можливо в реальних умовах.

Ще один підхід, описаний у [25], аналізує поведінку не лише на рівні мережі, але й фізичного процесу і таким чином виявляє атаки, що маскуються під звичайні операції системи. У тестах система показала високу точність та низький рівень хибних спрацювань. Але для впровадження в реальну систему цей підхід потребує глибокого розуміння технологічного процесу та доступу до фізичних сенсорів.

Хоч ці рішення демонструють значні досягнення, спільною проблемою залишається їхня адаптація до реальних умов тому що багато систем тестуються на симульованих наборах даних, мають складну інтеграцію, або вимагають значних апаратних ресурсів чи доступу до низькорівневого обладнання.

1.6 Переваги використання машинного навчання у кіберзахисті критичної інфраструктури

В даний час розвиток кіберзагроз відбувається значно швидше, ніж з'являються нові засоби захисту. Це особливо актуально для систем критичної інфраструктури, таких як енергетика, транспорт, промисловість чи водопостачання. Традиційні підходи, що ґрунтуються на фіксованих правилах або сигнатурах, як вже було зазначено, не здатні ефективно реагувати на нові види атак. Тому все більшого значення набувають методи машинного навчання, які дозволяють автоматизувати процес виявлення, прогнозування та запобігання кіберінцидентам [26,27].

Однією з головних переваг машинного навчання є можливість швидко аналізувати великі обсяги даних і виявляти приховані закономірності. Алгоритми здатні обробляти журнали подій, мережевий трафік та дані сенсорів у реальному часі, що значно зменшує навантаження на фахівців з безпеки. За даними дослідження [26], використання штучного інтелекту дозволяє автоматизувати

аналіз великих даних та підвищити швидкість виявлення загроз. Подібну позицію висловлюють і практичні фахівці з кіберзахисту, які відзначають, що застосування ML дає змогу скоротити час реагування на інциденти [27].

Ще однією суттєвою перевагою є здатність таких систем виявляти невідомі раніше (zero-day) атаки. На відміну від класичних антивірусів чи систем виявлення вторгнень, які спираються на відомі сигнатури, моделі машинного навчання вивчають нормальну поведінку системи та реагують на будь-які відхилення від неї [28]. Це дає можливість вчасно помічати нові, ще не задокументовані типи атак.

Важливим аспектом є і здатність моделей адаптуватися до змін у середовищі. Під час оновлення програмного забезпечення, зміни конфігурації мережі або введення нового обладнання традиційні системи часто потребують ручного налаштування. Алгоритми машинного навчання, натомість, можуть навчатися на нових даних і самостійно підлаштовувати свої параметри. Це робить захисні системи більш гнучкими та здатними реагувати на нові сценарії загроз [28].

Крім того, застосування машинного навчання допомагає зменшити кількість хибних спрацювань. У багатьох випадках системи безпеки генерують велику кількість попереджень, більшість із яких не мають реального підґрунтя. ML-моделі вчаться розрізняти звичайну активність від потенційно небезпечної, що підвищує точність і зменшує ризик перевантаження операторів [29].

Машинне навчання також сприяє ефективнішій роботі спеціалістів. Завдяки автоматизації рутинних процесів (наприклад, аналізу журналів або фільтрації подій) фахівці можуть зосередитись на складніших завданнях, таких як стратегічне планування чи розробка нових політик безпеки [27]. Це підвищує загальну ефективність управління кіберзахистом на об'єктах критичної інфраструктури.

Сучасні ML-рішення відзначаються масштабованістю – їх можна застосовувати як на рівні окремих пристроїв (edge-середовище), так і у хмарних інфраструктурах. Завдяки цьому стає можливим централізований моніторинг великої кількості об'єктів у режимі реального часу [30].

Окрему увагу нині приділяють прозорості та інтерпретованості результатів, так званим методам Explainable AI. Вони дозволяють пояснити, чому модель зробила певний висновок, що підвищує довіру операторів до автоматизованих систем безпеки [31].

Загалом, застосування машинного навчання сприяє переходу до проактивного підходу в кіберзахисті. Якщо раніше система реагувала на вже здійснені атаки, то тепер можливо прогнозувати загрози ще до того, як вони вплинуть на об'єкт критичної інфраструктури. Це забезпечує більш високий рівень стійкості та надійності таких систем.

1.7 Постановка задачі виявлення аномалій у ICS/SCADA-системах та вибір моделі прогнозування загроз

На відміну від звичайних IT-мереж, ICS/SCADA системи мають стабільний і передбачуваний характер даних, тому навіть невеликі відхилення можуть свідчити про можливу атаку або збій обладнання. Тому основна мета системи захисту – навчити модель розрізняти нормальну поведінку технологічного процесу від потенційно небезпечних змін, що можуть бути результатом кібератак або внутрішніх помилок.

Під час розроблення системи для виявлення аномалій доцільно спиратися на відкриті набори даних, які імітують роботу промислових систем. Найбільш поширеними є SWaT (Secure Water Treatment Dataset) та BATADAL (Battle of the Attack Detection Algorithms). Ці набори містять часові ряди сенсорних показників, станів клапанів, насосів тощо, а також позначення нормальних і атакованих періодів. Їх можна використовувати для навчання та тестування моделей навіть у домашніх умовах, без підключення до реальних виробничих об’єктів.

Оскільки дані з ICS/SCADA-систем мають часову структуру, ефективними виявилися рекурентні нейронні мережі (RNN), зокрема LSTM, які добре враховують залежності у часі. Модель LSTM здатна «запам’ятовувати» попередні

стани процесу й прогнозувати очікувані значення наступних показників. Якщо фактичне значення суттєво відрізняється від прогнозованого, система може інтерпретувати це як аномалію.

Для реалізації в умовах обмежених ресурсів можна використовувати спрощену архітектуру LSTM-автоенкодера, яка складається з вхідного шару, одного шару LSTM для кодування часових залежностей і одного шару для відновлення сигналу. Після навчання модель здатна відтворювати нормальні дані з низькою похибкою, тоді як аномальні відхилення викликатимуть збільшення помилки реконструкції. Такий підхід можна реалізувати за допомогою бібліотек TensorFlow або PyTorch, використовуючи стандартний ПК.

Для підвищення точності передбачення та зниження хибних спрацювань доцільно передбачити попередню нормалізацію даних і використання ковзного вікна, яке дозволяє враховувати контекст кількох попередніх спостережень. Для базового навчання достатньо набору даних обсягом кілька десятків тисяч рядків, що дозволяє експериментувати навіть без графічного процесора.

Таким чином, в роботі ставиться задача розроблення модуля виявлення аномалій у даних SCADA-системи з використанням LSTM-моделі, який:

– навчається на відкритих часових рядах з реальних або симульованих даних;

– прогнозує нормальну поведінку сенсорних сигналів;

– визначає аномалії за перевищенням встановленого порогу помилки реконструкції.

Обґрунтування вибору саме LSTM-моделі полягає в тому, що на основі аналізу сучасних публікацій, було зроблено висновок, що вона добре працює з послідовними промисловими даними, не потребує значних обчислювальних ресурсів і може бути реалізована в навчальних умовах. До того ж, згідно з результатами досліджень, такі моделі демонструють стабільну точність виявлення аномалій на відкритих наборах даних.

1.8 Висновки та постановка задачі

У результаті проведеного теоретичного аналізу було встановлено, що сучасні ICS/SCADA-системи є складними об'єктами кіберзахисту через їхню тривалу експлуатацію, обмеженість ресурсів та використання застарілих комунікаційних протоколів. Традиційні методи захисту, які базуються на сигнатурному виявленні або фільтрації трафіку, виявилися недостатньо ефективними у випадках нових чи модифікованих атак.

Огляд сучасних наукових праць показав, що методи машинного навчання, особливо ті, що враховують часову послідовність даних, мають високу ефективність при аналізі поведінкових аномалій у промислових процесах. Оптимальні результати демонструють моделі, побудовані на базі рекурентних нейронних мереж, зокрема архітектури LSTM. Такі моделі здатні аналізувати динаміку змін сенсорних сигналів і виявляти відхилення, які не завжди можна описати за допомогою традиційних правил.

У ході аналізу існуючих рішень було визначено, що використання LSTM-автоенкодерів дозволяє зменшити кількість хибних спрацювань і забезпечити адаптивність моделі до різних технологічних процесів. Також показано, що для попередніх експериментів можна використовувати відкриті набори даних, такі як SWaT або BATADAL, які імітують роботу реальних SCADA-систем і придатні для досліджень у навчальних умовах.

На основі проведеного аналізу сформульовано задачі дослідження – розробити та дослідити модуль виявлення аномалій для ICS/SCADA-систем, який використовує LSTM-модель машинного навчання для прогнозування поведінки сенсорних сигналів і визначення потенційних загроз. Модель має забезпечувати високу точність при помірних обчислювальних витратах, що дозволить реалізувати її на звичайному комп'ютері.

2 ПРОЕКТУВАННЯ МОДУЛЮ ВИЯВЛЕННЯ АНОМАЛІЙ ICS/SCADA-СИСТЕМ ШЛЯХОМ ПРОГНОЗУВАННЯ ЗАГРОЗ ІЗ ВИКОРИСТАННЯМ МАШИННОГО НАВЧАННЯ

2.1 Формалізація вхідних даних і підхід до їх попередньої обробки

Для побудови модуля виявлення аномалій у промислових системах керування (ICS/SCADA) необхідною передумовою є формалізація та якісна попередня обробка даних. У кваліфікаційній роботі використовується набір даних, наданий компанією 2WAF Security [32], яка спеціалізується на захисті інформаційних систем, в тому числі і об’єктів критичної інфраструктури. Даний набір містить записи журналів подій, мережевого трафіку та системних повідомлень, що моделюють роботу промислових мереж під час типових операцій і можливих кібератак. Такий формат забезпечує достатній рівень наближення до реального середовища без доступу до конфіденційної або технологічної інформації підприємств.

У промислових системах використовується кілька типів даних таких як мережевий трафік (наприклад, пакети Modbus/TCP), журнали подій контролерів (PLC logs), системні повідомлення серверів SCADA, а також записи користувацьких дій операторів. Кожен тип інформації має власну структуру і вимагає специфічної обробки. Числові параметри нормалізуються, текстові потрібно токенізувати, а часові перетворити у послідовності фіксованої довжини.

Основними джерелами даних є мережеві журнали (network logs), записи міжсистемних транзакцій та події системного моніторингу. У структурі кожного запису зазвичай міститься часовий штамп, тип події, IP-адреси джерела і призначення, коди запитів, розмір пакета, статус відповіді та короткий текстовий опис операції. У сукупності ці елементи дозволяють відтворити часову послідовність подій у мережі, визначити характер взаємодії між компонентами та виявити відхилення від нормальної поведінки.

Подальша підготовка даних охоплює декілька основних етапів таких, як очищення, нормалізацію, токенізацію, векторизацію та побудову масивів векторів, які у комплексі забезпечують готовність інформації до машинного аналізу.

Першим кроком є очищення вихідних записів від зайвих або некоректних елементів. Під час збору журналів можливі технічні дублікати, неповні записи, відсутність часових міток чи помилки форматування. Такі випадки негативно впливають на якість навчальної вибірки, тому всі записи з відсутніми ключовими параметрами або дублюванням видаляються. Після попереднього очищення типовий запис журналу може мати вигляд по типу «*PLC1 → PLC2, read_request, 0.42 s, status=OK»*. Для узгодженості даних додатково виконується нормалізація часових міток, що дозволяє порівнювати події різної тривалості. Масштабування здійснюється за формулою (2.1):

$$t' = \frac{t - t_{\min}}{t_{\max} - t_{\min}} \tag{2.1}$$

Таким чином, часові інтервали зводяться до спільного діапазону, що особливо важливо при навчанні моделей типу LSTM, де часові відмінності впливають на послідовність станів нейронів.

Окрім цього проводиться уніфікація форматів часу та кодування текстових полів, щоб забезпечити узгодженість усіх рядків. Також на цьому етапі формується структурована таблиця, у якій кожен запис має стандартизований набір ознак.

Далі потрібно виконати нормалізацію числових значень. У журналах подій різні параметри (кількість байтів, тривалість з'єднання, інтервал між запитами тощо) мають різні діапазони.

Щоб запобігти переважанню одних ознак над іншими, усі числові поля перетворюються до єдиного масштабу. Зазвичай застосовується мін-макс нормалізація, що зводить усі значення до інтервалу [0,1] за формулою (2.2):

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (2.2)$$

У випадках, коли розподіл даних не є рівномірним, доцільно використовувати z-score нормалізацію, яка базується на середньому значенні μ та стандартному відхиленні σ (2.3):

$$x' = \frac{x - \mu}{\sigma} \quad (2.3)$$

Цей підхід краще зберігає структуру вибірки та запобігає домінуванню аномально великих або малих значень.

У системних логах часто зустрічаються текстові повідомлення по типу «Unauthorized write request detected from IP 192.168.1.23» або «Connection timeout - PLC unit unreachable». Щоб створена модель змогла аналізувати такі повідомлення, їх потрібно перетворити у числову форму. Для цього використовується токенізація тобто процес розбиття тексту на окремі смислові одиниці (токени). Наприклад, зазначене вище речення перетвориться у послідовність ["Unauthorized", "write", "request", "detected", "from", "IP", "192.168.1.23"].

Після цього формується словник унікальних токенів, кожен з яких має свій власний індекс. Така форма надає можливість працювати з текстовими даними у форматі числових послідовностей.

Для більшої інформативності можна застосовувати частотні показники TF-IDF, які враховують важливість слова в контексті. Такий підхід дозволяє моделі виділяти рідкі, але важливі сигнали, наприклад, «unauthorized access» або «unexpected PLC response». Для подальшого узагальнення контексту використовується embedding-представлення (Word2Vec, FastText), де близькі за змістом слова отримують схожі вектори. Це забезпечує кращу здатність моделі розпізнавати семантичні закономірності.

Після токенізації кожен елемент потрібно перетворити у вектор числових значень, придатний для обробки нейронними мережами. Існує два основних способи векторизації.

Перший підхід – це one-hot encoding, коли для кожного токена створюється двійковий вектор, у якому лише одна позиція має значення «1», а решта матиме «0».

Другий підхід – це векторизація через embedding-шари, де токени проєктуються у простір меншої розмірності. У цьому просторі схожі за значенням або контекстом слова розташовані ближче одне до одного. Такий метод забезпечує узагальнення та підвищує здатність моделі виявляти закономірності у складних послідовностях подій.

Після векторизації можливе перевантаження моделі великою кількістю параметрів (іноді понад 10 000 векторів). Щоб уникнути надлишкових ознак, використовується метод головних компонент (PCA), який дає змогу скоротити розмірність простору без втрати ключової інформації (2.4):

$$Z = XW \tag{2.4}$$

де X – початкова матриця ознак;

W – матриця головних компонент;

Z – зменшене представлення.

Після перетворення всіх записів у векторну форму дані групуються за часовими вікнами. Наприклад, формується послідовність із 50 або 100 подій, що відповідають певному проміжку часу. Кожен масив векторів відображає поведінку системи у вибраному часовому фрагменті.

Для кожної послідовності додається мітка, що вказує, чи спостерігалася у цей період аномальна активність. Такі структури слугують вхідними даними для моделей прогнозування типу LSTM або Seq2Seq Autoencoder, які вивчають часові залежності між подіями.

Після завершення обробки дані розподіляються на навчальну, валідаційну та тестову частини. Зазвичай співвідношення становить 70/20/10 або 60/30/10 відповідно. Важливо, щоб дані не перетиналися між вибірками за часовими мітками, адже це може призвести до «витоку інформації» та неправильної оцінки ефективності моделі.

2.2 Розробка алгоритму роботи удосконаленого модуля виявлення аномалій

Алгоритм роботи модуля виявлення аномалій побудовано на поетапній обробці вхідних даних із подальшим аналізом їхніх характеристик у часовій послідовності. Основна мета алгоритму – це виявити відхилення у поведінці системи керування, які можуть свідчити про потенційні кіберзагрози або технічні збої.

Робота модуля починається з отримання вхідних даних, що включають інформацію про сеанс взаємодії, параметри мережевої активності та системні події, які пов'язані з поведінкою оператора. Дані подаються у вигляді часових рядів, що дозволяє відстежувати динаміку змін у системі.

На першому етапі виконується попередня обробка даних тобто нормалізація, токенізація, векторизація та формування матриць векторів, які було описано в попередньому підрозділі. Ці кроки забезпечують перетворення різнорідних текстових і числових даних у єдиний формат, придатний для подальшого машинного аналізу. Результатом етапу є стандартизований набір векторних послідовностей, який зберігає часову залежність та структуру вхідних подій.

Далі формується часовий контекст тобто до кожного вектору додаються часові мітки та показники, що відображають динаміку змін у системі. Це дозволяє алгоритму враховувати не тільки поточний стан, а й розвиток подій у певному проміжку часу.

Після цього дані подаються до двох аналітичних гілок. Перша гілка – це Feedforward Neural Network (FNN), яка аналізує статичні параметри, зокрема команди з SCADA-системи, мережеві характеристики та поведінкові ознаки користувача. Ця частина алгоритму виконує оцінку стану системи на основі окремих параметрів без урахування їхньої часової залежності.

Feedforward Neural Network є найпростішою формою штучної нейронної мережі, у якій інформація проходить тільки в одному напрямку – від входу до виходу (рис.2.1). Вона складається з вхідного шару, одного або кількох прихованих шарів і вихідного шару. Кожен нейрон прихованого шару обчислює зважену суму вхідних сигналів і застосовує до неї нелінійну функцію активації. Формально цей процес описується рівнянням (2.5):

$$h_j = \varphi(\sum_{i=1}^{n} w_{ij} x_i + b_j), \tag{2.5}$$

де x_i – вхідні ознаки, w_{ij} – ваги зв'язків, b_j – зсув, а φ – функція активації (зазвичай ReLU).

FNN може виявляти атаки, які не мають часової кореляції, із точністю до 99,9%, тоді як LSTM ефективніше обробляє послідовні дані. Тому поєднання LSTM та FNN в ансамблевому підході повинно підвищувати ефективність системи виявлення вторгнень.

В роботі така архітектура пропонується для аналізу статичних вхідних параметрів, що характеризують окремі операції ICS-системи – наприклад, команди керування, значення ID пакетів, адреси MODBUS/TCP або DNP3 запитів. Ці параметри не потребують урахування часової залежності, тому FNN дає змогу швидко класифікувати їх як нормальні або потенційно підозрілі.

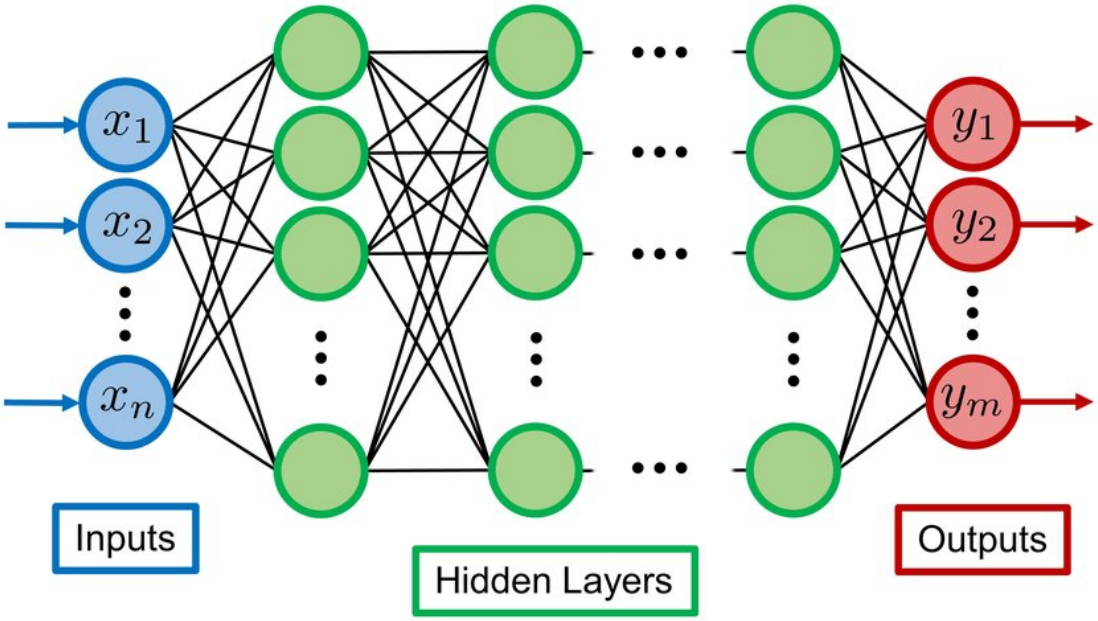


Рисунок 2.1 – Схематичне зображення архітектури FNN

Друга гілка – Recurrent Neural Network з довготривалою пам'яттю (LSTM). Цей тип рекурентних нейронних мереж був запропонований як розв'язання проблеми зникання або вибуху градієнтів під час навчання класичних RNN. Мережа LSTM зберігає внутрішній стан (cell state), який дозволяє передавати інформацію на довгих часових відрізках (рис.2.2).

Основні обчислення LSTM-комірки описуються системою рівнянь (2.6):

$$
\begin{aligned}
f i_t &= \sigma(w_i x_t + u_i h_{t-1} + b_i) \\
f_t &= \sigma\left(w_f x_t + u_f h_{t-1} + b_f\right) \\
o_t &= \sigma(w_o x_t + u_o h_{t-1} + b_o) \\
C_t^{\sim} &= tanh(w_C x_t + u_C h_{t-1} + b_C) \\
C_t &= f_t \odot C_{t-1} + i_t \odot C_t^{\sim} \\
h_t &= o_t \odot \tanh\left(C_t\right)
\end{aligned}
\tag{2.6}
$$

де i_t, f_t, o_t – вхідний, гейт забування та вихідний гейти відповідно;

C_t – стан комірки пам'яті;

h_t – вихідний стан;

x_t – вхідний вектор на кроці часу;

t, w, u, b – вагові матриці та вектори зміщень;

σ – сигмоїдна функція;

tanh – гіперболічний тангенс;

$\odot$ – поелементне множення.

LSTM-мережі є ефективними у навчанні та запам'ятовуванні довгих послідовностей, що робить їх придатними для завдань прогнозування часових рядів і виявлення аномалій.

У проєктованому модулі LSTM-архітектура застосовується для аналізу послідовностей даних. Наприклад, змін стану обладнання або послідовності команд керування, що надходять із SCADA-рівня до контролерів PLC. Це дозволяє виявляти нетипові відхилення, які не можна розпізнати за допомогою статичних параметрів.

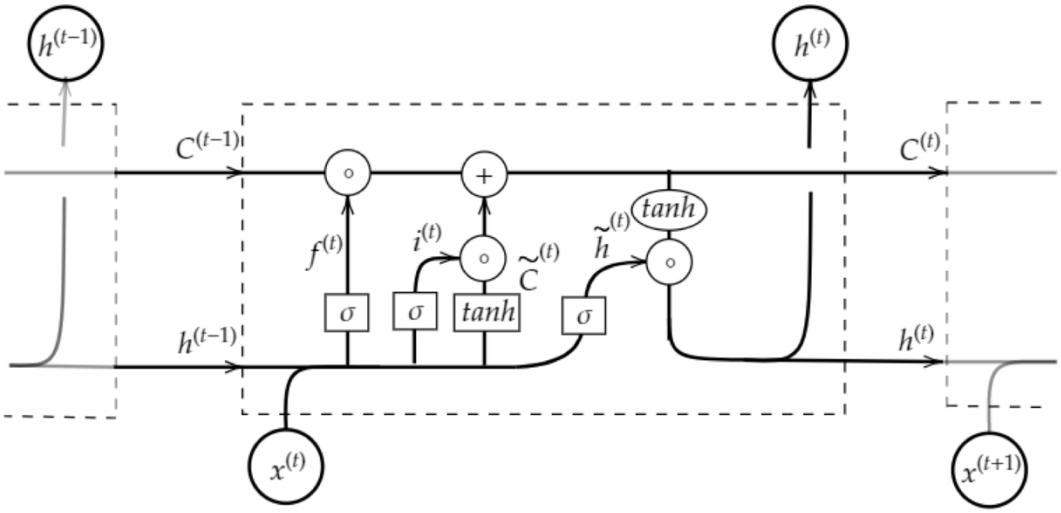


Рисунок 2.2 – Обчислювальний граф LSTM

Для підвищення стабільності навчання використовується метод teacher forcing. Його суть полягає в тому, що на кожному кроці моделі подається не передбачене значення попереднього стану, а реальне з тренувального набору даних. Такий метод прискорює збіжність і зменшує накопичення помилок у довгих послідовностях. У рамках даного дослідження використання teacher forcing дає змогу покращити точність короткострокових прогнозів аномалій та знизити похибку втрат (loss) у перші епохи навчання.

Додатково застосовується механізм уваги (attention), який дозволяє моделі визначати, які саме часові кроки або вхідні ознаки найбільш суттєві для поточного прогнозу. Це особливо важливо для ICS/SCADA-даних, де поведінка системи може змінюватися залежно від типу операційного циклу.

Механізм уваги дозволяє моделі фокусуватися на найбільш важливих часових відрізках даних при прийнятті рішення про наявність аномалії. Замість рівномірного урахування всіх попередніх станів, увага визначає, які саме стани мають найбільший вплив на поточний прогноз.

Принцип уваги базується на обчисленні коефіцієнтів важливості (attention weights) для кожного елемента вхідної послідовності (2.7):

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{dk}})V, \quad (2.7)$$

де Q – вектор запиту (query);

K – ключі (keys);

V – значення (values);

dk – розмірність вектора ключів.

У задачі виявлення аномалій ці ваги дозволяють визначити, які події з історії мають найбільше значення для прогнозу поточного стану системи.

Використання механізму уваги підвищує інтерпретованість і стабільність моделі, дозволяючи відслідковувати, які саме змінні вплинули на рішення

мережі.Такий підхід важливий для критичної інфраструктури, де кожне рішення моделі повинно мати пояснення для оператора.

Результати з обох гілок об'єднуються на рівні шару об'єднання (Concat Result), що формує узагальнений вектор стану системи.

Отримане представлення подається до класифікатора, який визначає тип ситуації і може надати відповідь чи це типова поведінка, якась аномальна подія за якою потрібно спостерігати або критичне відхилення, яке вже є терміновим для реагування.

У разі виявлення аномалії класифікатор генерує сигнал тривоги або рекомендацію для додаткової перевірки. Передбачено також можливість накопичення історії виявлених подій для подальшого аналізу та вдосконалення навчальної моделі.

Таким чином, алгоритм поєднує обробку статичних і динамічних даних, що забезпечує більш точне виявлення загроз у промислових системах керування. Гібридна структура з використанням FNN та LSTM дозволяє одночасно враховувати миттєві характеристики сигналів і їхню часову залежність, підвищуючи надійність роботи системи навіть у нестабільних умовах.

2.3 Розробка архітектури програмного модуля

Архітектура удосконаленого модуля виявлення аномалій була побудована на основі гібридного підходу, який поєднує рекурентні та прямі нейронні мережі. Така структура дозволяє одночасно аналізувати часові послідовності показників сенсорів і статичні характеристики мережевої активності та поведінки оператора. На схемі архітектури представлено основні логічні компоненти та взаємозв'язки між ними (рис.2.3).

Першим блоком йде Вхідний шар (Input layer). На цьому етапі дані надходять у модуль із різних джерел – систем керування технологічними процесами, сенсорних пристроїв, журналів активності користувачів і мережевих моніторів.

Перед обробкою усі дані проходять попередню нормалізацію, токенізацію та векторизацію, що забезпечує єдину числову форму подання. Крім того, для всіх типів даних додається часова мітка, що дозволяє синхронізувати події з різних джерел у межах однієї сесії.

Далі йде блок обробки даних. До цієї групи належать Commands from SCADA, Network Data Metrics і Operator behaviour. Дані цього типу не мають вираженої часової динаміки, тому вони подаються на три паралельні прямі нейронні мережі (FNN). Кожна з них виконує вилучення ключових ознак, виявляє кореляції між командами, мережевою активністю й поведінковими характеристиками оператора. Отримані вектори ознак потім об'єднуються в узагальнене подання.

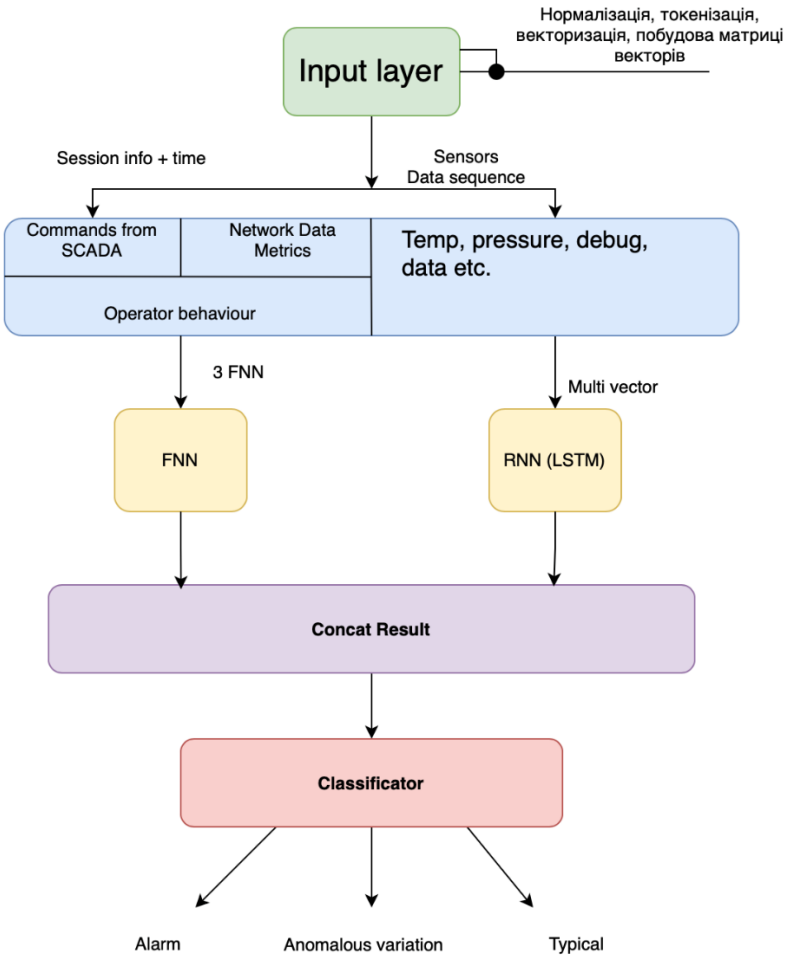


Рисунок 2.3 – Архітектура модуля, що розробляється

Показники технологічного процесу – температура, тиск, сигнали діагностики, індикатори несправностей – подаються у вигляді часових рядів. Для їх обробки використовується рекурентна нейронна мережа типу LSTM, що дозволяє виявляти як короткочасні, так і довготривалі залежності між змінами параметрів системи. На цьому рівні формується багатовимірний вектор (multi vector), який описує поточний стан технологічного середовища з урахуванням попередніх спостережень.

Вихідні вектори з FNN та LSTM проходять процес конкатенації, у результаті чого утворюється узагальнене подання поточної ситуації в системі. Це дозволяє одночасно врахувати як контекст поведінки користувача та мережевих процесів, так і динаміку фізичних параметрів обладнання. Такий підхід забезпечує більшу стійкість моделі до неочікуваних змін і дозволяє зменшити кількість хибнопозитивних спрацьовувань.

Фінальний етап роботи модуля полягає у класифікації поточного стану системи за трьома рівнями:

- Alarm – підтверджена аномалія, що потребує негайного реагування;
- Anomalous variation – відхилення від норми, яке може бути спричинене зовнішніми чи технічними факторами;
- Typical – типовий стан системи без відхилень.

Класифікація здійснюється шляхом аналізу зведеного вектору ознак, який подається на вихідний шар із функцією активації типу Softmax. Це дає змогу оцінити ймовірність належності поточного стану до кожної з категорій.

2.4 Висновки до розділу

У другому розділі було описано підхід до проєктування та побудови модуля виявлення аномалій у промислових системах керування, який орієнтовано на прогнозування кіберзагроз із використанням методів машинного навчання. В основу моделі покладено поєднання двох типів аналізу – статичного та

динамічного, що дозволяє повноцінно оцінити поточний стан мережевих процесів та виявляти потенційні відхилення від нормальної поведінки.

У розділі сформульовано підхід до підготовки вхідних даних, що охоплює етапи очищення, нормалізації, токенізації, векторизації та формування часових рядів. Така багаторівнева попередня обробка буде забезпечувати узгодженість даних і підвищувати стійкість моделі до шуму та нерівномірності розподілу ознак. Було розглянуто особливості застосування TF-IDF для текстових повідомлень системних логів, а також методів зменшення розмірності (PCA) для оптимізації обчислювальних ресурсів під час навчання.

Описано розробку алгоритму роботи удосконаленого модуля виявлення аномалій, який об’єднує дві аналітичні гілки – Feedforward Neural Network (FNN) для обробки статичних параметрів та Long Short-Term Memory (LSTM) для аналізу часових закономірностей. Така комбінація дозволятиме підвищити точність прогнозування аномальних станів за рахунок одночасного врахування поточного стану системи та її динаміки у часі. Крім того, використання механізмів обчислення ймовірності загрози та адаптивного порогу спрацьовування потенційно забезпечує зниження кількості хибнопозитивних результатів.

Представлено архітектуру програмного модуля, розроблену з урахуванням принципів модульності, гнучкості та масштабованості. Архітектура передбачає чітке розмежування між рівнями збору даних, аналітики, обробки та візуалізації, що спрощує інтеграцію модуля. Використання потокової обробки даних дозволяє забезпечити роботу модуля в реальному часі.

Таким чином, у розділі сформовано повну логіку побудови системи виявлення аномалій.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ

3.1 Обгрунтування інструментальних засобів реалізації

Реалізація модуля виявлення аномалій у системах класу ICS/SCADA потребує таких інструментів, які дозволяють одночасно працювати з великими масивами даних, аналізувати часову структуру сигналів, будувати складні моделі машинного навчання та забезпечувати достатню гнучкість для експериментів. Оскільки дані, надані компанією 2WAF Security, містять журнали телеметрії, мережевих подій і операторських дій, їхня структура є часовою. Тобто кожен запис відображає стан тієї чи іншої підсистеми у певний момент часу. Це наклало певні вимоги як до вибору технологій, так і до організації робочого процесу.

Основою для реалізації було обрано мову Python. Це рішення є практичним, адже Python давно став стандартом у сфері аналізу даних, і практично всі сучасні інструменти для роботи з часовими рядами створені саме для нього. Під час роботи над модулем постала необхідність оперативно об'єднувати дані з різних журналів, синхронізувати їх за часовими мітками, вирівнювати пропущені значення та формувати послідовності потрібної довжини. У цьому Python виявився зручним завдяки поєднанню високого рівня абстракції та великої кількості бібліотек, які дозволяють працювати навіть з достатньо «сирими» журналами.

Попередня обробка даних, яка включала фільтрацію, сортування, нормалізацію та формування ковзних вікон, виконувалась за допомогою базових бібліотек. Вони дали змогу легко перетворювати таблиці із записами подій на послідовності, які підходять для навчання рекурентної нейронної мережі. Окремим завданням була обробка різнорідних типів ознак. Частина з них описує фізичні параметри технологічного процесу (тиск, температура тощо), інша частина характеризує мережевий стан або активність оператора. Все це потрібно було об'єднати так, щоб модель отримувала повну картину поведінки системи.

Ключовим елементом проєкту стала LSTM-модель, яка використовувалася для прогнозування та виявлення відхилень у нормальній роботі промислової

мережі. LSTM була обрана не випадково. На відміну від звичайних повнозв'язних моделей вона здатна утримувати інформацію про поведінку системи протягом певного часу, що особливо важливо для ICS/SCADA. У таких системах відхилення рідко виглядають як одинична аномальна точка. Значно частіше це поступові зміни, поява невідповідностей між сенсорами або несинхронні дії між операторами, пристроями і контролерами. Тому модель мала не просто класифікувати окремий стан, а бачити динаміку.

Для реалізації LSTM використовувалися TensorFlow і його високорівнева бібліотека Keras. Це дало змогу будувати архітектуру моделі на основі послідовностей, комбінувати її з додатковими шарами та легко змінювати параметри під конкретні експериментальні налаштування. Під час проєкту довелося працювати з вікнами різної довжини, використовувати кілька підходів до формування навчальних вибірок, а також комбінувати LSTM з додатковою MLP-частиною, що відповідала за обробку агрегованих та контекстних ознак. Таке поєднання виявилось зручним у реалізації саме завдяки гнучкості TensorFlow.

Оскільки навчання рекурентних моделей на великих послідовностях є обчислювально затратним, окремим важливим рішенням стало використання Google Colab. Це середовище надало доступ до графічних прискорювачів, що значно скоротило час навчання моделі. У ході роботи було проведено декілька експериментів: зміна кількості шарів, розмірів прихованих станів, довжини вікна, типів нормалізації та методів розбиття даних на тренувальні та тестові сегменти. Такі експерименти на звичайному ноутбуці потребували б багато днів роботи, тоді як у Colab усе виконувалося значно швидше. Крім того, Colab забезпечив зручне середовище для візуального аналізу кривих навчання та перевірки поведінки моделі на окремих фрагментах вибірки.

Додатково використовувалися інструменти для аналізу якості моделі. У задачі виявлення аномалій важливо не просто визначити, чи правильно класифікується більшість прикладів, а зрозуміти, як модель поводить себе у випадках рідкісних відхилень. Саме тому обов'язковими стали метрики повноти,

точності, F-міра та побудова ROC-кривих. Усі необхідні методи для цього надаються Python, що дозволило швидко оцінювати якість моделі після кожного етапу навчання.

3.2 Формування та опис датасету

Для розроблення та перевірки роботи модуля виявлення аномалій використовувався набір даних, наданий компанією 2WAF Security. У його основу входять фрагменти журналів роботи промислової мережі, подій, що виникали у процесі взаємодії обладнання, а також інформація про реакцію окремих модулів системи. Дані є анонімізованими і не містять технологічно чутливих відомостей, однак зберігають реальну структуру поведінки системи. Їхня форма і спосіб побудови дозволяють відтворити умови, близькі до тих, у яких працюють SCADA-компоненти справжніх виробничих об'єктів.

У загальному вигляді датасет являє собою послідовність записів, кожен з яких відображає стан певної групи параметрів у конкретний момент часу. Дані містять показники фізичних сенсорів технологічних процесів, мережеві характеристики та дії, пов'язані з операторським втручанням або автоматичними командами контролерів. Завдяки такому поєднанню у вибірці присутні як параметри, що описують внутрішню поведінку обладнання, так і події на рівні взаємодії мережевої інфраструктури.

Структура одного запису в датасеті включає такі поля: часову мітку, тиск у контурі, температуру, витрату робочого середовища, стан клапана у конкретну мить, команду, яку виконує PLC, затримку мережевого з'єднання, відсоток втрати пакетів, дію або відсутність дії оператора, тип події, що в цей момент фіксується системою, та ознаку, яка вказує на наявність чи відсутність аномальної поведінки. Це дозволяє відтворювати дуже різноманітні ситуації, оскільки зміни можуть бути помітні як у фізичних параметрах процесу, так і у їхньому поєднанні з мережевими або операторськими характеристиками.

Щоб продемонструвати загальний вигляд цих даних, нижче наведено приклад кількох рядків із датасету у спрощеному текстовому форматі. Їх можна розглядати як фрагмент вихідної вибірки, що використана під час навчання моделі (рис.3.1).

```
2024-05-12 13:04:51  pressure=5.32   temperature=59.9   flow=12.7   valve=0   plc_cmd=HOLD    latency=19ms   loss=0.3%   operator=0   type=normal   anomaly=0
2024-05-12 13:04:52  pressure=5.29   temperature=60.1   flow=12.6   valve=0   plc_cmd=HOLD    latency=20ms   loss=0.3%   operator=0   type=normal   anomaly=0
2024-05-12 13:04:53  pressure=5.27   temperature=60.0   flow=12.4   valve=0   plc_cmd=HOLD    latency=21ms   loss=0.4%   operator=0   type=normal   anomaly=0
2024-05-12 13:04:54  pressure=5.30   temperature=60.2   flow=12.5   valve=0   plc_cmd=HOLD    latency=22ms   loss=0.5%   operator=0   type=normal   anomaly=0
2024-05-12 13:04:55  pressure=5.28   temperature=60.3   flow=12.4   valve=0   plc_cmd=HOLD    latency=23ms   loss=0.5%   operator=0   type=normal   anomaly=0

2024-05-12 13:04:56  pressure=5.25   temperature=60.4   flow=12.2   valve=0   plc_cmd=HOLD    latency=24ms   loss=0.6%   operator=0   type=normal   anomaly=0
2024-05-12 13:04:57  pressure=5.26   temperature=60.5   flow=12.3   valve=0   plc_cmd=HOLD    latency=24ms   loss=0.5%   operator=0   type=normal   anomaly=0
2024-05-12 13:04:58  pressure=5.24   temperature=60.3   flow=12.1   valve=0   plc_cmd=HOLD    latency=25ms   loss=0.7%   operator=0   type=normal   anomaly=0
2024-05-12 13:04:59  pressure=5.19   temperature=60.1   flow=12.0   valve=0   plc_cmd=HOLD    latency=28ms   loss=0.8%   operator=0   type=normal   anomaly=0
2024-05-12 13:05:00  pressure=5.17   temperature=59.8   flow=11.9   valve=0   plc_cmd=HOLD    latency=29ms   loss=0.8%   operator=0   type=normal   anomaly=0

2024-05-12 13:05:01  pressure=5.21   temperature=60.3   flow=12.5   valve=0   plc_cmd=HOLD    latency=21ms   loss=0.4%   operator=0   type=normal   anomaly=0
2024-05-12 13:05:02  pressure=5.18   temperature=60.1   flow=12.4   valve=0   plc_cmd=HOLD    latency=22ms   loss=0.5%   operator=0   type=normal   anomaly=0
2024-05-12 13:05:03  pressure=5.25   temperature=60.2   flow=12.3   valve=0   plc_cmd=HOLD    latency=20ms   loss=0.3%   operator=0   type=normal   anomaly=0

2024-05-12 13:05:04  pressure=4.78   temperature=63.9   flow=10.1   valve=1   plc_cmd=OPEN    latency=48ms   loss=1.1%   operator=0   type=warning  anomaly=1
2024-05-12 13:05:05  pressure=4.51   temperature=67.2   flow= 9.3   valve=1   plc_cmd=OPEN    latency=55ms   loss=1.4%   operator=0   type=warning  anomaly=1
2024-05-12 13:05:06  pressure=4.22   temperature=70.4   flow= 8.2   valve=1   plc_cmd=OPEN    latency=61ms   loss=1.9%   operator=0   type=warning  anomaly=1

2024-05-12 13:05:07  pressure=4.10   temperature=71.8   flow= 7.9   valve=1   plc_cmd=OPEN    latency=65ms   loss=2.1%   operator=0   type=alert    anomaly=1
2024-05-12 13:05:08  pressure=4.05   temperature=73.6   flow= 7.4   valve=1   plc_cmd=OPEN    latency=72ms   loss=2.4%   operator=0   type=alert    anomaly=1
2024-05-12 13:05:09  pressure=3.97   temperature=74.5   flow= 7.1   valve=1   plc_cmd=OPEN    latency=77ms   loss=2.7%   operator=0   type=alert    anomaly=1

2024-05-12 13:05:10  pressure=4.33   temperature=71.9   flow= 8.6   valve=1   plc_cmd=HOLD    latency=61ms   loss=1.8%   operator=1   type=info     anomaly=0
2024-05-12 13:05:11  pressure=4.88   temperature=67.4   flow= 9.5   valve=1   plc_cmd=CLOSE   latency=48ms   loss=1.1%   operator=1   type=info     anomaly=0
2024-05-12 13:05:12  pressure=5.03   temperature=63.1   flow=10.7   valve=1   plc_cmd=CLOSE   latency=32ms   loss=0.9%   operator=1   type=info     anomaly=0

2024-05-12 13:05:13  pressure=5.11   temperature=61.8   flow=11.8   valve=0   plc_cmd=HOLD    latency=27ms   loss=0.7%   operator=0   type=normal   anomaly=0
2024-05-12 13:05:14  pressure=5.16   temperature=61.2   flow=12.1   valve=0   plc_cmd=HOLD    latency=25ms   loss=0.6%   operator=0   type=normal   anomaly=0
2024-05-12 13:05:15  pressure=5.19   temperature=60.9   flow=12.3   valve=0   plc_cmd=HOLD    latency=24ms   loss=0.5%   operator=0   type=normal   anomaly=0
2024-05-12 13:05:16  pressure=5.21   temperature=60.7   flow=12.5   valve=0   plc_cmd=HOLD    latency=23ms   loss=0.5%   operator=0   type=normal   anomaly=0
2024-05-12 13:05:17  pressure=5.23   temperature=60.5   flow=12.6   valve=0   plc_cmd=HOLD    latency=22ms   loss=0.4%   operator=0   type=normal   anomaly=0

2024-05-12 13:05:18  pressure=5.20   temperature=60.4   flow=12.4   valve=0   plc_cmd=HOLD    latency=23ms   loss=0.5%   operator=0   type=normal   anomaly=0
2024-05-12 13:05:19  pressure=5.17   temperature=60.3   flow=12.3   valve=0   plc_cmd=HOLD    latency=24ms   loss=0.5%   operator=0   type=normal   anomaly=0
2024-05-12 13:05:20  pressure=5.14   temperature=60.2   flow=12.1   valve=0   plc_cmd=HOLD    latency=24ms   loss=0.6%   operator=0   type=normal   anomaly=0
2024-05-12 13:05:21  pressure=5.09   temperature=60.0   flow=11.9   valve=0   plc_cmd=HOLD    latency=26ms   loss=0.7%   operator=0   type=normal   anomaly=0
2024-05-12 13:05:22  pressure=5.04   temperature=59.8   flow=11.7   valve=0   plc_cmd=HOLD    latency=27ms   loss=0.8%   operator=0   type=normal   anomaly=0
```

Рисунок 3.1 – Фрагмент використаного для навчання датасету

Цей приклад ілюструє, як фізичні параметри технологічного процесу поєднуються з відповідями системи та мережевим станом. Наприклад, значне зростання температури при одночасному падінні тиску та зміні стану клапана може вказувати на реальну технічну несправність. А збільшення затримки або втрати пакетів може свідчити про проблеми із зв'язком у мережі або підозрілу активність. Такі закономірності і повинні розпізнаватися моделлю як ознаки потенційних аномалій.

Під час підготовки даних до навчання моделі важливу роль відігравав той факт, що всі параметри у датасеті є частиною часових рядів. Метою було не просто оцінити стан системи в окремих моментах, а побачити динаміку змін і виявити характерні послідовності подій, які передують аномалії. Для цього дані були перетворені у формат послідовностей фіксованої довжини. Такий підхід називають

формуванням ковзних вікон. Він полягає в тому, що замість подання моделі окремих значень формується вікно, яке містить значення параметрів за певний попередній відрізок часу. Після цього вікно переміщується на один крок уперед, створюючи нову послідовність. Таким чином, дані перетворюються на множину перекривних відрізків, що дозволяє моделі вивчати не лише самі значення, а й тенденції розвитку процесів.

У даній роботі розмір вікна було підібрано таким чином, щоб модель отримувала достатньо інформації про поведінку системи перед переходом у аномальний стан, але при цьому не перенасичувалась зайвими даними. Кожне таке вікно включало сенсорні, мережеві та операторські показники, синхронізовані між собою за часовою міткою. Саме так формується навчальна вибірка для моделі на основі LSTM, яка працює не з окремими записами, а з послідовностями.

Датасет також містив інформацію про різні типи подій – як звичайні стани, що відображають нормальну роботу обладнання, так і стани, які вважалися аномальними. Ця розмітка дозволяє оцінювати здатність моделі розрізняти нормальні й автентичні проблемні ситуації, характерні для промислових мереж. У сукупності всі ці дані створюють достатню основу для навчання моделі прогнозування загроз, що дозволяє наблизити результати до реальних умов експлуатації на підприємствах.

3.3 Навчання на тестування LSTM-моделі

Після завершення роботи з попередньою обробкою даних наступним етапом стало навчання моделі, яка повинна виявляти відхилення у поведінці промислової системи. Оскільки вихідні дані мали чітко виражену часову структуру, а сама задача передбачала аналіз не окремих значень, а тенденцій і зв'язків між параметрами впродовж певного періоду часу, найбільш логічним було використання рекурентної нейронної мережі типу LSTM. Така архітектура здатна враховувати попередні стани й утримувати інформацію про динаміку параметрів.

Це особливо важливо при роботі з ICS/SCADA-журналами, де аномалія зазвичай не виникає якось раптово, а формується як поведінкова послідовність.

Перед початком навчання вибірку було поділено на дві частини. Перша частина – це дані для тренування моделі, а друга частина – це окрема частина для тестування. Для збереження коректності часових залежностей розбиття виконувалося не випадковим чином, а за принципом хронологічного поділу. Це означає, що модель навчалася на більш ранніх записах і перевірялася на тих, які йшли за ними в реальному порядку. Такий спосіб є більш адекватним для систем керування тому що дозволяє перевірити, чи зможе модель виявити зміну поведінки у майбутньому відносно того, що вона бачила в минулому.

Після цього всі параметри було нормалізовано. Це стандартна процедура, проте в задачах із промисловими журналами вона відіграє окрему роль. Різні типи показників (фізичні, мережеві та операторські) мають різні діапазони, і модель може почати надавати надмірну вагу тим параметрам, які мають більші абсолютні значення. Тому всі ознаки було приведено до спільного масштабу. Це дозволило уникнути ситуацій, коли, наприклад, температура мала б більшу вагу лише через те, що її значення чисельно вищі за величини втрати пакетів або затримки мережі.

Сформовані ковзні вікна стали основою для навчальних послідовностей. Довжина вікна визначалася таким чином, щоб у моделі була достатня кількість інформації для розуміння контексту, але водночас вона не перевантажувалася зайвими даними. У промислових системах зміна стану сенсорів або команд часто має інерційний характер. З цієї причини типовим є аналіз періоду в кілька десятків секунд. Саме такий підхід було реалізовано. Тобто кожне вікно містило певну кількість послідовних записів (у роботі використовувалася довжина 50), які повністю зберігали хронологію процесу. Далі ці вікна використовувалися як вхідні дані для LSTM-мережі.

Архітектура моделі складалася з рекурентного блоку LSTM, який обробляв часову частину, та додаткового багатошарового персептрону, що працював із агрегованими або додатковими ознаками. Така комбінація виникла природно

внаслідок особливостей датасету. Частина показників не мала часової природи, але могла підсилювати прогноз. Наприклад, тип події, що класифікується системою, або інформація про реакцію оператора. LSTM відповідала за динаміку, а MLP – за контекст. У поєднанні це дозволило покращити стабільність та зменшити кількість хибних спрацювань у випадках, коли зміни параметрів не супроводжувалися жодними справжніми ознаками загрози.

Навчання проводилося у середовищі Google Colab із використанням графічного прискорювача (рис.3.2). Це рішення було практичним насамперед через обчислювальну складність LSTM-моделей. Робота з довгими послідовностями, великою кількістю ознак та перекривними вікнами значно збільшує навантаження. Під час перших експериментів було відмічено, що навчання на центральному процесорі займає значний час і не дозволяє оперативно проводити серії експериментів. Використання GPU, навпаки, дало можливість швидко тестувати різні параметри мережі, змінювати розмір прихованого стану, кількість шарів і швидко оцінювати отримані метрики.

Рисунок 3.2 – Фрагмент навчання у Google Colab

Одним із технічних аспектів, на які довелося звернути увагу, була проблема затухання градієнтів. Це природне явище для рекурентних мереж, особливо при

роботі з довгими послідовностями. У ході налаштування моделі довелося підібрати таку довжину вікна, при якій модель утримує достатньо інформації про попередні стани, але не переходить у режим, коли градієнт починає зникати або, навпаки, «вибухати». Крім того, у процесі роботи використовувався прийом скидання прихованого стану на початку кожної послідовності. Це дозволило уникнути накопичення інформації між різними вікнами та забезпечило стабільніші результати при навчанні.

У процесі тестування модель аналізувала нові послідовності, яких не було в тренувальній вибірці. Це дало змогу оцінити її здатність узагальнювати поведінку та виявляти аномалії на основі тих закономірностей, які були засвоєні на навчальних даних. Особливо важливо було перевірити модель на випадках, де аномалія розвивалася поступово. Одним із таких фрагментів був відрізок, де температура зростала протягом кількох секунд, тоді як тиск одночасно спадав, клапан переходив у відкритий стан, а мережеві параметри демонстрували зростання затримки та втрат. Модель успішно відтворювала логіку виявлення таких ситуацій, оскільки вони містили чітку часову структуру.

Результати тестування показали, що модель досить добре розрізняє нормальні стани та відхилення, які характерні для промислових систем. Важливо, що вона не реагувала надмірно на поодинокі коливання параметрів. Такі ситуації часто можна зустріти на реальних технологічних процесах і вони не є ознакою небезпеки. Модель більш впевнено реагувала на саме сукупність змін, що й визначає її придатність до роботи у реальних умовах. Поступове формування відхилення, характерне для промислових аварій або мережевих атак, відтворювалося моделью з достатнім рівнем достовірності.

У підсумку навчання LSTM-моделі дозволило створити інструмент, здатний не просто класифікувати поодинокі записи, а розпізнавати форми поведінки промислової системи. Саме це може бути сигналом про небезпечні ситуації. Використання рекурентної архітектури дало змогу врахувати динаміку, а додаткова MLP-структура дозволила інтегрувати контекстні ознаки. У поєднанні з коректною

попередньою обробкою та достатнім обсягом послідовностей це створило основу для подальшого аналізу точності та порівняння моделі з іншими методами.

3.4 Порівняльний аналіз ефективності з існуючими методами

Якість роботи запропонованого методу перевірялась в зіставленні з іншими підходами, які на практиці досить часто використовують для виявлення відхилень у даних промислових систем. Такий підхід дає можливість побачити, наскільки запропонована модель поводиться краще або гірше у ситуаціях, які можуть траплятися під час роботи обладнання.

Дані для порівняння були підготовлені у вигляді окремих часових фрагментів однакової тривалості. Кожен фрагмент містив показники кількох сенсорів, які відображають основні стани технологічного процесу. Окрім самих сигналів, для кожного фрагмента були сформовані числові характеристики, що узагальнюють поведінку параметрів за цей проміжок часу: середні значення, розмах змін, характер коливань і взаємозалежності між каналами. Завдяки такому поданню даних вдалося протестувати як моделі, що працюють із послідовностями, так і алгоритми, які орієнтуються лише на зведені показники.

На рис. 3.3 показано хід оцінки ймовірності аномалії, яку повертає запропонована модель, у зіставленні з фактичними мітками.

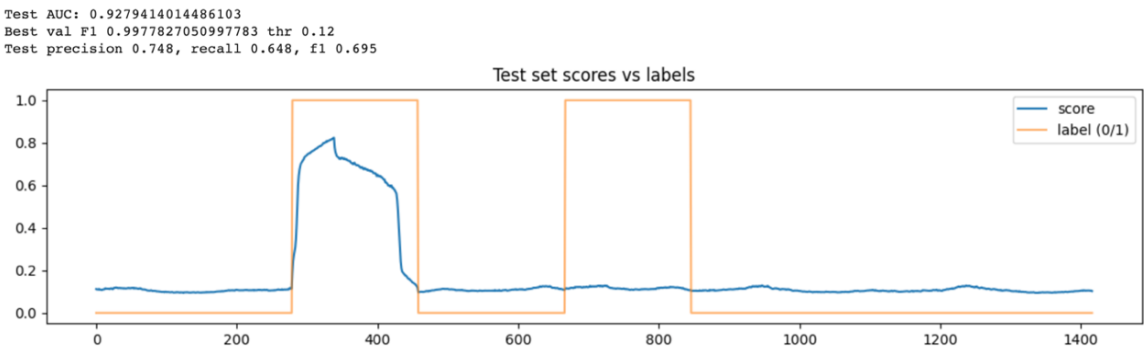


Рисунок 3.3 – Реакція запропонованої моделі на тестових даних

На отриманому графіку можна побачити, що модель дає плавне підвищення оцінки на всій проблемній ділянці, а не тільки у точці різкого відхилення від норми. Це говорить про те, що модель реагує не на окремі піки, а на характер змін у часі. Тобто там, де сенсорні дані поводяться якось нестандартно протягом тривалого часу, оцінка буде поступово підвищуватися. І на весь період аномалії буде триматися на високому рівні.

Для зрівняння із створеним підходом було обрано кілька широко використовуваних методів. Першим був простий варіант із пороговими правилами, коли рішення щодо аномалії приймається залежно від того, чи перевищив певний параметр свій допустимий діапазон. Такі правила працюють у найпростіших випадках, однак часто не реагують на ситуації, коли параметри поводяться нетипово, але ще не виходять за межі.

До наступної групи віднесено моделі, які здатні працювати з ознаками без урахування часових зв’язків усередині вікна. Це Isolation Forest і One-Class SVM. Ці алгоритми намагаються знайти відхилення у розподілі даних, не аналізуючи порядок зміни значень. Для промислових сигналів такий підхід інколи дає непоганий результат, але у випадках, коли аномалія формується поступово, він виявляється недостатнім.

На наступному графіку зіставлено оцінки трьох різних алгоритмів: LSTM, Isolation Forest та One-Class SVM (рис.3.4). Класичні моделі генерують серії піків, які не завжди збігаються з фактичними мітками. На деяких ділянках їх реакція або занадто різка, або навпаки – надто запізніла. Натомість LSTM показує більш рівну поведінку. Вона чітко покриває проміжки, де аномалія триває в часі, навіть якщо її окремі значення не виходять за межі порогів.

Запропонована модель поєднує два рівні подання інформації. Сам часовий фрагмент і набір узагальнених ознак. Завдяки цьому вона враховує не лише загальні характеристики вікна, але й те, як саме зміна сенсорних показників розгортається в часі. Саме така властивість виявилась ключовою під час аналізу

порівняльних графіків, оскільки поведінка параметрів у промислових системах рідко змінюється миттєво.

Рисунок 3.4 – Порівняльна реакція моделей на одному фрагменті

Також було створено теплову карту одного фрагменту (рис.3.5).

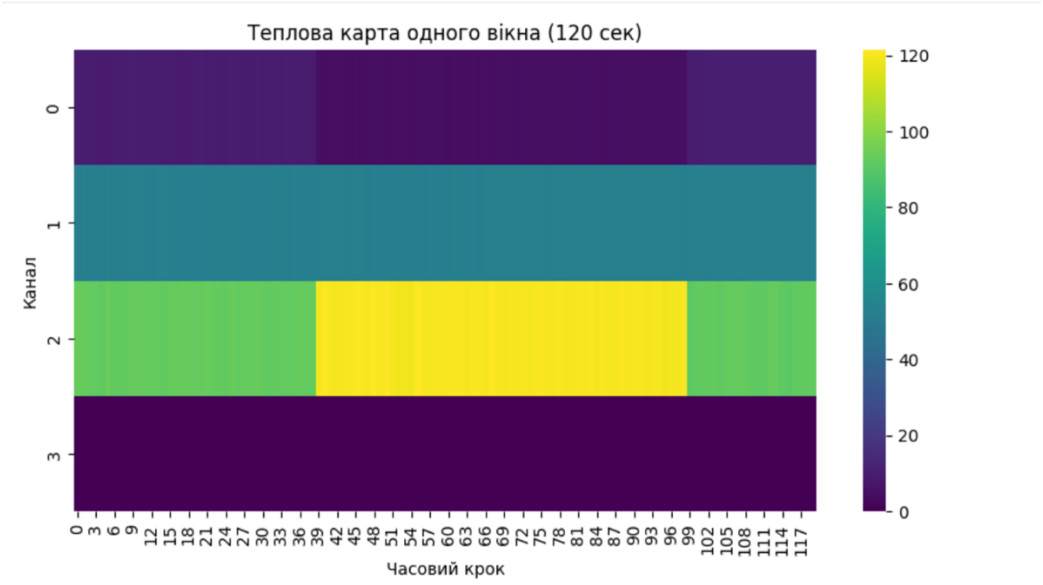


Рисунок 3.5 – Теплова карта одного фрагмента

Теплова карта демонструє, як змінювалися сенсорні значення в межах одного вікна тривалістю 120 секунд. Канал витрати (третя смуга) різко змінює свою інтенсивність на ділянці, що відповідає аномалії, у той час як інші канали змінюються повільніше. Саме такі відмінності у формі кривої LSTM вміє зачепити, оскільки аналізує не тільки числові характеристики, а й сам темп змін у часі.

3.5 Аналіз результатів та оцінка точності виявлення аномалій

Отримані результати дали змогу оцінити не лише формальні метрики запропонованої LSTM-моделі, але й особливості її поведінки на різних типах аномальних ділянок. Порівняння з іншими підходами показало, що рекурентна архітектура краще відтворює характер зміни параметрів у промислових процесах, що суттєво вплинуло на кінцеву якість.

Розглянемо ROC-криву (рис.3.6). Це графічне представлення діагностичної здатності бінарного класифікатора. Для створення ROC-кривої в машинному навчанні частота істинних спрацьовувань (TPR) порівнюється з частотою хибних спрацьовувань (FPR).

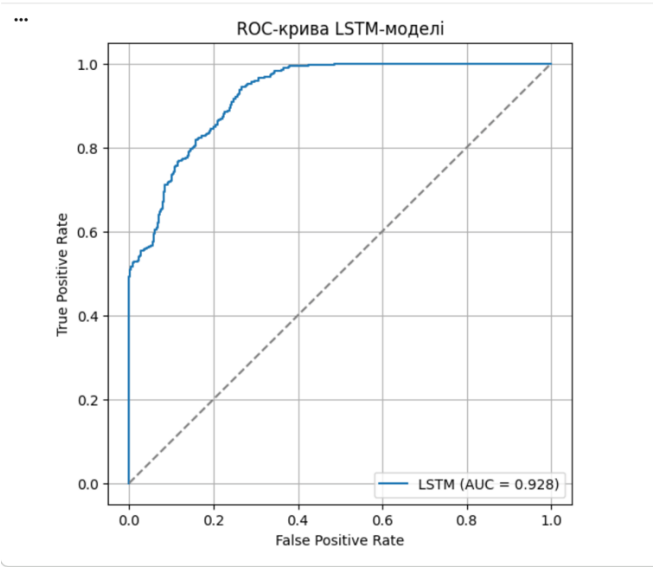


Рисунок 3.6 – ROC-крива LSTM-моделі

На ROC-кривій видно майже вертикальний підйом на початковому відрізку. Це свідчить про здатність моделі ефективно відокремлювати аномальні стани вже при малих хибних спрацюваннях. Площа під кривою становить близько 0.93. Для задачі з послідовною природою сигналів це відповідає доволі високому рівню впевненості моделі у своїх розрізненнях класів.

Далі була побудована PR-крива, яка демонструє, як поведінка моделі змінюється при різних рівнях порогу (рис.3.7).

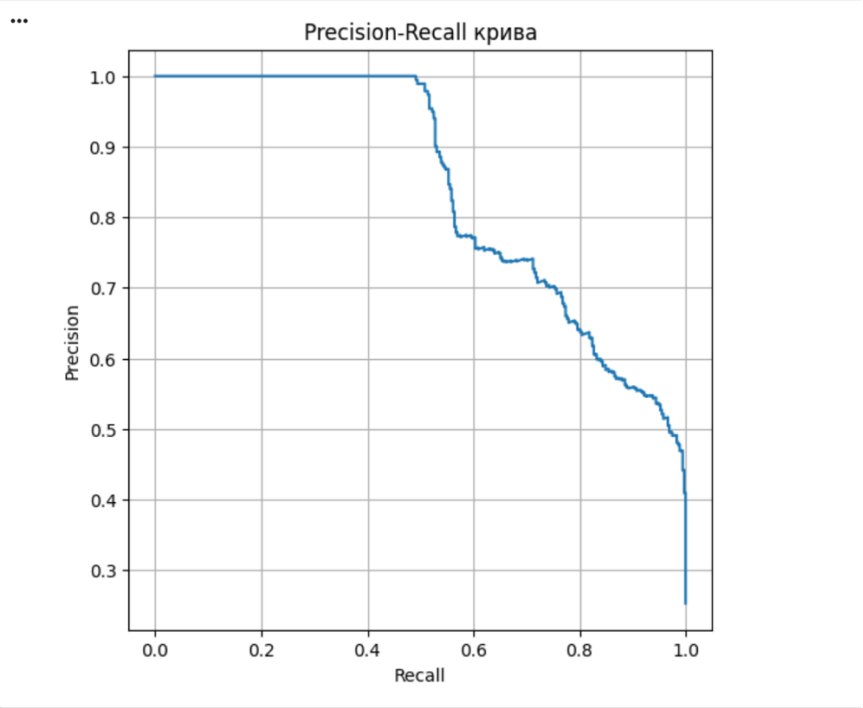


Рисунок 3.7 – Precision-Recall крива

На початку графіка видно довгий відрізок із точністю, близькою до 1.0. Це говорить про те, що при суворих порогах модель майже не помиляється. У міру зростання повноти точність поступово зменшується, але залишається достатньо високою. Це є досить важливим у ситуаціях, коли потрібно зловити якомога більше аномальних фрагментів.

Аналіз поведінки моделі на часових фрагментах показав, що LSTM відтворює структуру аномалії не у вигляді окремих сплесків, а через плавне підвищення вихідного сигналу протягом усього проблемного відрізка. Як було вже сказано в 3.4, така форма реакції добре узгоджується з природою процесів у реальних промислових середовищах, де аномалії рідко проявляються як миттєві точки відхилення.

Порівняння з іншими підходами допомогло побачити різницю у способах реагування на нетипову поведінку. Isolation Forest схильний давати множинні короткі піки, які не завжди збігаються з реальними межами аномалії. One-Class SVM, навпаки, часто демонстрував запізнення та нечутливість до слабких змін, що відобразилося й у дуже низьких метриках у таблиці (рис.3.8). LSTM-модель працює інакше. Її вихід не розбивається на окремі піки, а формує «площу» над аномальною ділянкою, що дозволяє точніше виділити часові межі події.

	Model	AUC	Precision	Recall	F1
0	LSTM + MLP	0.927941	0.748387	0.648045	0.694611
1	Isolation Forest	0.723039	0.521739	0.502793	0.512091
2	One-Class SVM	0.171079	0.083591	0.150838	0.107570
3	Threshold baseline	0.800052	1.000000	0.396648	0.568000

Рисунок 3.8 – Порівняння ефективності моделей

Найвищий рівень F1-міри отримано саме для запропонованої LSTM+MLP архітектури. Водночас простіший пороговий підхід показав прийнятну, але відчутно нижчу повноту. Це свідчить, що окремі нестандартні ділянки, які не виходять за межі фіксованих порогів, такі правила не відловлюють. Рекурентна мережа реагує не на максимальні відхилення, а на форму та швидкість зміни сигналів. Тому вона бачить і ті фрагменти, де параметри лише наближаються до проблемного значення.

Загалом аналіз показав, що запропонований підхід найкраще працює у ситуаціях, де аномалії мають тривалу або фазову структуру. У випадках, де ознаки проявляються протягом десятків секунд, LSTM використовує цю інформацію максимально ефективно, чого не можуть забезпечити моделі, які не охоплюють часовий контекст. У поєднанні з агрегованими ознаками це дало можливість підвищити точність виявлення та зменшити кількість хибних спрацювань у порівнянні з традиційними методами.

3.6 Висновки до розділу 3

У третьому розділі описано результати, які були отримані під час побудови й перевірки модуля виявлення аномалій. Після обробки даних і серії запусків стало зрозуміло, як саме модель поводиться на різних фрагментах і що для неї виявляється найпомітнішим.

Було проведено порівняння з іншими підходами. Для максимально зрозумілої інтерпретації всі методи були накладені на один і той самий часовий відрізок. Деякі алгоритми реагують дуже різко, інші, навпаки, пропускають ділянки, які візуально виглядають підозріло. На фоні таких прикладів поведінка LSTM виявилась більш рівною. Це не означає, що вона ідеальна, але у багатьох випадках її реакція збігається з тим, що реально відбувається в сигналах.

Таблиця з метриками підтвердила те, що було видно на графіках. Результати запропонованої моделі загалом вищі за результати інших підходів. Особливо це помітно там, де зміни в даних розтягнуті в часі й не проявляються одним різким стрибком. Простим методам у таких ситуаціях складно вловити початок або кінець аномальної ділянки. А рекурентна частина моделі читає послідовність і сприймає її не як набір окремих значень, а як цілісну траєкторію.

Запропонований підхід виявився оптимальним для подібних задач. Він краще і стабільніше працює там, де потрібно врахувати зміну сигналів у часі, і де проста перевірка на поріг не дає бажаного результату.

4 ЕКОНОМІЧНА ЧАСТИНА

4.1 Проведення комерційного аудиту розробки

Метою проведення економічного аналізу є визначення комерційної доцільності впровадження створеного модуля виявлення аномалій у ICS/SCADA-системах. Розробка базується на використанні методів глибокого навчання (LSTM), що забезпечує підвищену точність та швидкість реагування на аномальні процеси.

Для проведення комерційного аудиту залучають не менше 3-х незалежних експертів. Експертами, що підтвердили потенціал розробки, виступають:

– Грицак А.В., доцент кафедри МБІС,ВНТУ;

– Салієва О.В., д.ф., доцент кафедри МБІС,ВНТУ;

– Єпіфанова І.Ю.,д.е.н.,проф.,проректор з наукової роботи, ВНТУ.

Комерційний потенціал визначається такими факторами:

– зниження витрат підприємства внаслідок запобігання простоям;

– зменшення збитків від кіберінцидентів;

– відсутність потреби у дорогому обладнанні;

– гнучка інтеграція в існуючу інфраструктуру.

Оцінювання комерційного потенціалу буде здійснене за критеріями, що наведені в таблиці 4.1.

Таблиця 4.1 – Рекомендовані критерії оцінювання науково-технічного рівня і комерційного потенціалу розробки та бальна оцінка

Критерії оцінювання та бали (за 5-ти бальною шкалою)					
№	0	1	2	3	4
1	2	3	4	5	6
Технічна здійсненність концепції:					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено роботоздатність продукту в реальних умовах
Ринкові переваги (недоліки):					

Продовження таблиці 5.1

2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів
4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
Ринкові перспективи					
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає
Практична здійсненність					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витрачати значні кошти та час на навчання наявних	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні.	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування

Продовження таблиці 5.1

10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві
11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3- х років
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

Результати оцінювання комерційного потенціалу експертами розробки зведено в таблицю 4.2.

Таблиця 4.2 – Результати оцінювання комерційного потенціалу розробки

Критерії	Прізвище, ініціали, посада експерта		
	Грицак А.В.	Салієва О.В.	Єпіфанова І.Ю.
	Бали, виставлені експертами:		
1	4	4	4
Ринкові переваги (недоліки):			
2	3	2	3
3	3	3	3

Продовження таблиці 4.2

4	4	4	4
5	3	4	3
Ринкові перспективи			
6	3	3	3
7	3	4	3
Практична здійсненність			
8	4	3	4
9	3	3	3
10	3	3	3
11	4	4	4
12	4	4	4
Сума балів	СБ1=41	СБ2=38	СБ3=41
Середньоарифметична сума балів СБ	40		

За даними таблиці 4.2 можна зробити висновок, щодо рівня комерційного потенціалу розробки. Зважимо на результат й порівняємо його з рівнями комерційного потенціалу розробки, що представлено в таблиці 4.3.

Таблиця 4.3 – Науково-технічні рівні та комерційні потенціали розробки

Середньоарифметична сума балів СБ, розрахована на основі висновків експертів	Рівень комерційного потенціалу розробки
0 – 10	Низький
11 – 20	Нижче середнього
21 – 30	Середній
31 – 40	Вище середнього
41 – 48	Високий

Рівень комерційного потенціалу розробки, становить 40 балів, що відповідає рівню «вище середнього».

Досягнутий рівень комерційного потенціалу пояснюється тим, що розроблений модуль істотно розширює функціональні можливості існуючих

рішень завдяки поєднанню аналізу часових рядів та механізмів точкового виявлення аномалій, що забезпечує вищу точність і стабільність роботи в умовах промислових мереж. Переваги розробки досягнуті за рахунок підвищеної чутливості до відхилень, можливості адаптації до різних технологічних процесів та зменшення ймовірності хибних спрацювань, що безпосередньо підвищує її конкурентоспроможність на ринку систем моніторингу ICS/SCADA.

4.2 Розрахунок витрат на здійснення розробки

Проведемо прогнозування витрат на виконання науково-дослідної, дослідно-конструкторської та конструкторсько-технологічної роботи для розробки програмного забезпечення, яке складається з таких етапів:

– перший етап – розрахунок витрат, які безпосередньо стосуються виконавців даного розділу роботи;

– другий етап – розрахунок загальних витрат на виконання даної роботи;

– третій етап – прогнозування загальних витрат на виконання та впровадження результатів даної роботи.

Результати розрахунків зведемо до таблиці 4.4.

Основна заробітна розробника-дослідника $З_о$ (4.1):

$$З_о = \frac{М}{Т_р} * t \text{ (грн)} \quad (4.1)$$

де М – місячний посадовий оклад;

$Т_р$ – число робочих днів в місяці (приблизно Тр = (22) дні);

t – число робочих днів роботи розробника-дослідника – 42.

Так як в даному випадку розробляється програмний продукт, то розробник виступає одночасно і основним робітником, і тестувальником розроблюваного програмного продукту.

Таблиця 4.4 – Основна заробітна плата розробників

Найменування посади	Місячний посадовий оклад, грн.	Оплата за робочий день, грн.	Число днів роботи	Витрати на заробітну плату, грн.
Керівник проекту	15000	682	5 днів	3410
Програміст	20000	910	35	31850
Всього				35260

Додаткова заробітна плата $З_д$ розробника розраховується як 10% від основної заробітної плати:

$$З_д = 0{,}10 * 35260 = 3526 \text{ (грн)}.$$

Нарахування на заробітну плату $Н_{зп}$ розробника становить (4.2):

$$Н_{зп} = (З_о + З_д) * \frac{\beta}{100} \text{ (грн)} \quad (4.2)$$

де $З_о$ – основна заробітна плата розробника;

$З_д$ – додаткова заробітна плата розробника;

β – ставка єдиного внеску на загальнообов'язкове державне соціальне страхування – 22%.

$$Н_{зп} = (35260 + 3526) * 0{,}22 = 8533 \text{ (грн)}.$$

Амортизація обладнання, комп'ютерів та приміщень, які використовувались під час виконання даного етапу роботи. Дані відрахування розраховують по кожному виду обладнання, приміщенням тощо.

У спрощеному вигляді амортизаційні відрахування розраховуємо за формулою: (4.3)

$$А = (Ц * Т)/(12 * Т_в) \text{ (грн)} \quad (4.3)$$

де Ц – загальна балансова вартість обладнання, приміщення тощо, грн;

Т – фактична тривалість використання, міс;

$Т_в$ – термін використання обладнання, приміщень тощо, роки.

Розробка програмного забезпечення проводилася протягом 2 місяців. Зроблені розрахунки зведено до таблиці 4.5.

Таблиця 4.5 – Амортизаційні відрахування

Найменування	Балансова вартість, грн	Термін використання, роки	Фактична трив. використання, міс.	Величина амортизаційних відрахувань, грн
Ноутбук	40000	2	2	3333
Монітор	7000	2	2	583
Всього				3916

Інформацію про матеріали, що використовуються при розробці даного продукту внесено до таблиці 4.6.

Таблиця 4.6 – Матеріали, що використовуються при виготовленні даного продукту

Найменування матеріалу	Ціна за одиницю, грн.	Витрачено, шт.	Вартість витраченого матеріалу, грн
Папір (пачка)	180	1	180
Ручка	15	2	30
Флеш-накопичувач USB 64 GB	300	1	300
Всього	510		

Під час розробки програмного продукту використовувались лише безкоштовні програмні засоби.

Витрати на енергію визначаються на основі витрат на одиницю продукції та тарифів на енергію за допомогою формули (4.4):

$$В_е = В * П * Ф * К_п \text{ (грн)}, \quad (4.4)$$

де В – вартість 1кВт електроенергії (рахується як 12,5 грн);

П – установлена потужність обладнання, кВт;

Ф – фактична кількість годин роботи при створенні програмного продукту, годин;

$К_п$ – коефіцієнт використання потужності. Отже, витрати на енергію становлять:

$$В_е = 12,5*0,05*180*0,7 = 78,75 \text{ (грн)}.$$

Також потрібно врахувати витрати на доступ до мережі Інтернет, що використовувався під час виконання роботи.

Витрати за доступ до Інтернет можна розрухувати за формулою (4.5):

$$В_{ді} = Ц_{ді} * Т \text{ (грн)}, \quad (4.5)$$

де $Ц_{ді}$ – це ціна доступу за місяць;

Т – кількість місяців використання доступу до мережі.

$$В_{ді} = 650*2 = 1300 \text{ (грн)}.$$

Інші витрати охоплюють витрати на управління організацією, оплату службових відряджень, витрати на утримання, ремонт та експлуатацію основних засобів, витрати на опалення, освітлення, водопостачання, охорону праці тощо.

Інші витрати Ів можна прийняти як 150% від суми основної заробітної плати розробника:

$$В_{ін} = 1{,}5 * 35260 = 52890 \text{ (грн)}.$$

Сума всіх попередніх статей витрат дає витрати на виконання даної частини роботи – В обчислюється за формулою (4.6):

$$В = З_о + З_д + Н_{зп} + А + В_{мат} + В_е + В_{ін} \text{ (грн)} \quad (4.6)$$

$$В = 35260 + 3526 + 8533 + 3916 + 510 + 78{,}75 + 1300 + 52890 = 106014 \text{ (грн)}$$

Далі проведемо розрахунок загальних витрат на виконання даної роботи. Загальна вартість всієї наукової роботи визначається за формулою (4.7):

$$ЗВ_{заг} = \frac{В}{\eta} \text{(грн)} \quad (4.7)$$

де η коефіцієнт, який характеризує етап (стадію) виконання науково-дослідної роботи. Так, якщо науково-технічна розробка знаходиться на стадії: науково-дослідних робіт, то η=0,1; технічного проектування, то η =0,2; розробки конструкторської документації, то η =0,3; розробки техно- логій, то η =0,4; розробки дослідного зразка, то η =0,5; розробки промисло- вого зразка, то η =0,7; впровадження, то η =0,9.

Підставивши дані у формулу, отримуємо:

$$ЗВ_{заг} = \frac{106014}{0{,}5} = 212028 \text{ (грн)}.$$

4.3 Розрахунок економічної ефективності науково-технічної розробки від її комерціалізації потенційним інвестором

В ринкових умовах узагальнювальним позитивним результатом, що його може отримати потенційний інвестор від можливого впровадження результатів тієї чи іншої науково-технічної розробки, є збільшення у потенційного інвестора величини чистого прибутку. Саме зростання чистого прибутку забезпечить потенційному інвестору надходження додаткових коштів, дозволить покращити фінансові результати його діяльності, підвищить конкурентоспроможність та може позитивно вплинути на ухвалення рішення щодо комерціалізації цієї розробки.

Для того, щоб розрахувати можливе зростання чистого прибутку у потенційного інвестора від можливого впровадження науково-технічної розробки необхідно:

– вказати, з якого часу можуть бути впроваджені результати науково-технічної розробки;

– зазначити, протягом скількох років після впровадження цієї науково-технічної розробки очікуються основні позитивні результати для потенційного інвестора (наприклад, протягом 3-х років після її впровадження);

– кількісно оцінити величину існуючого та майбутнього попиту на цю або аналогічні чи подібні науково-технічні розробки та назвати основних суб’єктів (зацікавлених осіб) цього попиту;

– визначити ціну реалізації на ринку науково-технічних розробок з аналогічними чи подібними функціями.

При розрахунку економічної ефективності потрібно обов’язково враховувати зміну вартості грошей у часі, оскільки від вкладення інвестицій до отримання прибутку минає чимало часу. При оцінюванні ефективності інноваційних проектів передбачається розрахунок таких важливих показників:

– абсолютного економічного ефекту (чистого дисконтованого доходу);

– внутрішньої економічної дохідності (внутрішньої норми дохідності);

– терміну окупності (дисконтованого терміну окупності).

Аналізуючи напрямки проведення науково-технічних розробок, розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором можна об’єднати, враховуючи визначені ситуації з відповідними умовами.

Оцінити збільшення чистого прибутку підприємства ΔПі для кожного із років, протягом яких очікується отримання позитивних результатів від впровадження розробки можливо використовуючи формулу (4.8):

$$\Delta\text{Пі} = \sum_i^n (\Delta\text{П}_0 * N + \text{Ц}_0 * \Delta N) * \lambda * \rho * (1 - \frac{\nu}{100}) \text{ (грн)}, \qquad (4.8)$$

де $\Delta\text{П}_0$– покращення основного оціночного показника від впровадження результатів розробки у даному році. Зазвичай таким показником може бути ціна одиниці нової розробки;

N – основний кількісний показник, який визначає діяльність підприємства у даному році до впровадження результатів наукової розробки;

ΔN – покращення основного кількісного показника діяльності підприємства від впровадження результатів розробки;

Ц_0 – основний оціночний показник, який визначає діяльність підприємства у даному році після впровадження результатів наукової розробки;

n – кількість років, протягом яких очікується отримання позитивних результатів від впровадження розробки;

λ – коефіцієнт, який враховує сплату податку на додану вартість.

ρ – коефіцієнт, який враховує рентабельність продукту. Рекомендується приймати ρ = 0,25;

ν – ставка податку на прибуток (18%).

У результаті впровадження розробленого модуля виявлення аномалій для ICS/SCADA-систем покращується якість роботи кінцевого програмного продукту.

Підвищується точність аналізу телеметрії, зменшується кількість хибних спрацювань, а сам процес інтеграції стає простішим для інженерів. Завдяки цим змінам компанія-розробник може збільшити вартість ліцензії на програмний продукт у середньому на 60000 грн порівняно з базовою ціною 100000 грн, яка застосовувалася до модернізації.

Окрім цього, очікується, що кількість впроваджень поступово зростатиме: протягом першого року – до 12 впроваджень, другого – до 20 впроваджень, а третього – до 32 впроваджень.

Орієнтовно реалізація продукції до впровадження результатів наукової розробки складала 1 ліцензію на рік.

Спрогнозуємо збільшення чистого прибутку підприємства від впровадження результатів наукової розробки у кожному році відносно базового.

Збільшення чистого прибутку підприємства $\Delta П_i$ протягом першого року складе:

$$\Delta П1 = (60000 * 12 + 100000 * 11) * 0{,}88 * 0{,}25 * (1 - \frac{18}{100}) = 328328 \text{ (грн)}$$

Збільшення чистого прибутку підприємства $\Delta П_i$ протягом другого року (відносно базового року, тобто року до впровадження результатів наукової розробки) складе:

$$\Delta П2 = (60000 * 20 + 100000 * 19) * 0{,}88 * 0{,}25 * (1 - \frac{18}{100}) = 559240 \text{ (грн)}$$

Збільшення чистого прибутку підприємства Пі протягом третього року (відносно базового року, тобто року до впровадження результатів наукової розробки) складе:

$$\Delta\text{ПЗ} = (60000 * 32 + 100000 * 31) * 0{,}88 * 0{,}25 * \left(1 - \frac{18}{100}\right) = 905608 \text{ (грн)}$$

Приведена вартість всіх чистих прибутків ПП розраховується за формулою (4.9):

$$\text{ПП} = \sum_i^m \frac{\Delta\text{Пі}}{(1+\tau)^t} \text{ (грн)}, \quad (4.9)$$

де Пі – збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої НДДКР, грн;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні;

t – період часу (в роках) від моменту отримання чистого прибутку до точки «0».

Ставка дисконтування τ дорівнює 0,1. Отримаємо:

$$\text{ПП} = \frac{328328}{(1+0{,}1)^1} + \frac{559240}{(1+0{,}1)^2} + \frac{905608}{(1+0{,}1)^3} = 1441057(\text{грн})$$

4.3.1 Розрахунок абсолютного економічного ефекту для потенційного інвестора

Основними показниками, які визначають доцільність фінансування наукової розробки інвестором, є абсолютна і відносна ефективність вкладених інвестицій та термін їх окупності. Розрахунок ефективності вкладених інвестицій передбачає проведення таких робіт:

Далі розраховується теперішня вартість інвестицій PV, що вкладаються в наукову розробку. Такою вартістю можемо вважати прогнозовану величину

загальних витрат $З_в$ на виконання та впровадження результатів НДДКР, розраховану за формулою, тобто будемо вважати, що

$$PV = Зв * kinv \quad (4.10)$$

де кінв – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію, це можуть бути витрати на підготовку приміщень, розробку технологій, навчання персоналу, маркетингові заходи тощо; зазвичай кінв=2...5, але може бути і більшим

$$PV = 212028 * 2 = 424056 \text{ грн.}$$

Розрахуємо абсолютну ефективність вкладених інвестицій $Е_{абс}$. Для цього використовується формула (4.11):

$$Е_{абс.} = (ПП - PV)(грн), \quad (4.11)$$

$$Е_{абс.} = (1441057 - 424056) = 1017001$$

Якщо $Е_{абс} > 0$, то результат від проведення наукових досліджень та їх впровадження принесе прибуток, але це ще не свідчить про те, що інвестор буде зацікавлений у фінансуванні даного проекту (роботи).

Оскільки $Е_{абс} > 0$, то вкладання коштів на виконання та впровадження результатів НДДКР є доцільним.

4.3.2 Розрахунок внутрішньої дохідності інвестицій

Розраховуємо відносну (щорічну) ефективність вкладених в наукову розробку інвестицій $Е_в$. Для цього використовуємо формулу (4.12):

$$E_в = \sqrt[Тж]{1 + \frac{E_{абс}}{PV}} - 1\,, \qquad (4.12)$$

де $E_{абс}$ – абсолютна ефективність вкладених інвестицій, грн;

PV –теперішня вартість інвестицій, грн;

$Т_ж$ – життєвий цикл наукової розробки, роки.

Далі, розрахована величина $E_в$ порівнюється з мінімальною (бар'єрною) ставкою дисконтування $\tau_{мін}$, яка визначає ту мінімальну дохідність, нижче за яку інвестиції вкладатися не будуть.

У загальному вигляді мінімальна (бар'єрна) ставка дисконтування $\tau_{мін}$ визначається за формулою (4.13):

$$\tau_{мін} = d + f, \qquad (4.13)$$

де d – середньозважена ставка за депозитними операціями в комерційних банках, d = 0,09;

f – показник, що характеризує ризикованість вкладень, зазвичай, величина f = 0,05.

Якщо величина $E_в > \tau_{мін}$, то інвестор може бути зацікавлений у фінансуванні даної наукової розробки. В іншому випадку фінансування наукової розробки здійснюватися не буде.

Спочатку спрогнозуємо величину $\tau_{мін}$. Припустимо, що за даних умов $\tau_{мін}$ = 0,09 + 0,05 = 0,14. Тоді відносна (щорічна) ефективність вкладних інвестицій в проведення наукових досліджень та впровадження їх результатів складе:

$$E_в = \sqrt[3]{1 + \frac{1017001}{424056}} - 1 = 0{,}503$$

Оскільки $E_в > \tau_{мін}$, то у інвестора буде зацікавленість вкладати гроші в дану розробку, оскільки значно більші прибутки він отримає від того, що інвестує кошти в розробку, а не розмістить гроші на депозиті у комерційному банку.

4.3.3 Розрахунок періоду окупності інвестицій, вкладених потенційним інвестором

Розраховуємо термін окупності вкладених у реалізацію наукового проекту інвестицій. Термін окупності вкладених у реалізацію наукового проекту інвестицій $Т_{ок}$ можна розрахувати за формулою (4.14):

$$Т_{ок} = \frac{1}{E_в} \text{ (років)} \quad (4.14)$$

Для нашої розробки термін окупності вкладених у реалізацію проекту інвестицій $Т_{ок}$ складе:

$$Т_{ок} = \frac{1}{0{,}503} = 1{,}99 \text{ року}$$

Відносна ефективність інвестицій є більшою за мінімальну ставку дисконтування. Це свідчить про економічну доцільність фінансування наукової розробки відповідно до методичних рекомендацій.

Висновки до розділу 4

В четвертому розділі було проведено оцінювання комерційного потенціалу розробки модуля для виявлення аномалій у промислових системах керування (ICS/SCADA).

До аналізу було залучено трьох незалежних експертів. За результатами аналізу визначено, що рівень комерційного потенціалу розробки становить вище середнього.

Проведений аналіз показав, що створений модуль надає ширші можливості, ніж більшість існуючих рішень. Основна перевага полягає у більш точному виявленні відхилень у технологічних процесах та зниженні кількості хибнопозитивних спрацювань, що є критично важливим для промислових підприємств.

Аналіз комерційного потенціалу розробки показав, що програмний продукт за своїми характеристиками випереджає аналогічні програмні продукти і є перспективною розробкою. Він має кращі функціональні показники, а тому є конкурентоспроможним товаром. Сукупність технічних характеристик і доступнішої вартості робить модуль конкурентоспроможним продуктом і створює підґрунтя для його подальшого поширення на ринку.

ВИСНОВКИ

Метою дослідження було перевірити, наскільки моделі машинного навчання, а саме LSTM, можуть допомогти у виявленні аномалій у поведінці ICS/SCADA-систем. Під час аналізу існуючих наукових джерел стало зрозуміло, що для промислових мереж характерні умови, які ускладнюють раннє виявлення загроз. Причиною тому є різнорідність обладнання, суміш нових і застарілих протоколів, обмеження в оновленнях та велика залежність від стабільності технологічного процесу. Саме через це традиційні засоби захисту не завжди дають гідні результати. Тому актуальною є задача пошуку методів, здатних враховувати динаміку системи, а не лише фіксувати окремі події.

Для перевірки можливостей обраного підходу був підготовлений набір даних із технологічних сигналів та системних журналів, у якому поєднуються як нормальні стани, так і відхилення. Дані виявилися досить неоднорідними. Частина записів мала пропуски, відмінні масштаби, різні формати подій. Це вимагало окремої уваги до етапу попередньої обробки та формування часових послідовностей, оскільки саме якість підготовки значною мірою впливала на результат.

LSTM-модель показала себе придатною для таких задач. На відміну від статичних підходів, вона змогла уловити часову структуру сигналів і реагувати на відхилення, які не завжди видно на рівні окремих точок даних. Під час навчання стало помітно, що модель краще працює тоді, коли вхідні послідовності містять контекст, а не лише фрагмент події. У порівнянні з класичними методами аналізу логів, LSTM-підхід дає змогу виявляти потенційні загрози раніше, ще до моменту, коли аномалія призводить до збою у роботі обладнання.

Комбінування LSTM із повнозв’язними мережами для обробки статичних параметрів дало гарний результат. Така гібридна структура дозволила моделі врахувати як динаміку технологічних процесів, так і характеристики мережевої активності. Це підвищило стабільність результатів на різних частинах набору

даних. Також важливо, що сама модель не вимагала значних обчислювальних ресурсів. Навчання довелося проводити на GPU, але інференс може виконуватися й на менш потужних системах, що є суттєвим аргументом для промислових середовищ.

Під час роботи проявилися й певні обмеження. Модель досить чутлива до якості даних та способу формування вікон послідовностей. У реальних ICS/SCADA-мережах дані можуть бути неповними або нерівномірними, що ускладнює завдання. Також відсутність великих обсягів маркованих промислових даних обмежує можливість порівняти модель із більш складними підходами чи застосувати їх до ширшого класу сценаріїв.

Отримані результати дають підстави говорити, що використання LSTM для виявлення аномалій у промислових системах є перспективним напрямом. Модель продемонструвала стійку роботу на різнорідному наборі даних і змогла виявляти нетипові стани, які важко зафіксувати традиційними методами. Економічний аналіз також показав, що створення подібного рішення має практичну значимість. Бо навіть часткове зменшення кількості інцидентів або більш швидке виявлення відхилень здатні знизити витрати підприємства.

Запропонований підхід має потенціал для інтеграції у системи моніторингу промислових об’єктів. Подальший розвиток може бути пов’язаний із розширенням наборів даних, дослідженням трансформерних архітектур та адаптацією моделі до конкретних типів обладнання. Це дозволить зробити систему більш точною та надійною, а також наблизити її до реального впровадження на об’єктах критичної інфраструктури.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Yadav, Geeta, Paul, Kolin. Architecture and Security of SCADA Systems: A Review. 2020. DOI:10.48550/arXiv.2001.02925.

2. SCADA Systems Security. [Електронний ресурс]. - Режим доступу:https://www.vpnunlimited.com/ua/help/cybersecurity/scada.

3. Tidrea, A., Korodi, A., Silea, I. Cryptographic Considerations for Automation and SCADA Systems Using Trusted Platform Modules. Sensors. 2019. Vol. 19. No. 4191. DOI: 10.3390/s19194191.

4. Marksteiner, S. Reasoning on Adopting OPC UA for an IoT-Enhanced Smart Energy System from a Security Perspective. 2018 IEEE 20th Conference on Business Informatics (CBI), Vienna, Austria, 2018. P. 140-143. DOI:10.1109/CBI.2018.10060.

5. Erba, A., Müller, A., Tippenhauer, N. O. Security Analysis of Vendor Implementations of the OPC UA Protocol for Industrial Control Systems. In: Proceedings of the 4th Workshop on CPS & IoT Security and Privacy. 2022.

6. Altaleb, H., Rajnai, Z. A Comprehensive Analysis and Solutions for Enhancing SCADA Systems Security in Critical Infrastructures. 2024.

7. Tariq, N., Asim, M., Khan, F. Securing SCADA-Based Critical Infrastructures: Challenges and Open Issues.Procedia Computer Science. 2019. Vol. 155. P. 612-617. DOI: 10.1016/j.procs.2019.08.086.

8. ICS/SCADA Security. [Електронний ресурс]. - Режим доступу: https://www.micromindercs.com/blog/ics-scada-security.

9. SCADA Security Guide. [Електронний ресурс]. - Режим доступу:https://publications.gc.ca/collections/collection_2016/rddc-drdc/D68-3-C007-2015-eng.pdf.

10. Biggest Threats to ICS/SCADA Systems. [Електронний ресурс]. - Режим доступу:https://www.infosecinstitute.com/resources/scada-ics-security/biggest-threats-to-ics-scada-systems.

11. Stranahan, J., Soni, T., Carpenter, J., Heydari, V. SCADA Testbed Implementation, Attacks, and Security Solutions.In: Arai, K. (Ed.) Advances in Information and Communication. FICC 2021. Advances in Intelligent Systems and Computing, vol. 1363. Springer, Cham. DOI: 10.1007/978-3-030-73100-7_53.

12. Teixeira, A., Dán, G., Sandberg, H., Johansson, K. H. A Cyber Security Study of a SCADA Energy Management System: Stealthy Deception Attacks on the State Estimator. IFAC Proceedings Volumes. 2011. Vol. 44(1). P. 11271-11277.

13. Biggest Threats to ICS/SCADA Systems. [Електронний ресурс]. - Режим доступу:https://www.infosecinstitute.com/resources/scada-ics-security/biggest-threats-to-ics-scada-systems.

14. Alanazi, M., Mahmood, A., Chowdhury, M. J. M. SCADA Vulnerabilities and Attacks: A Review of the State-of-the-Art and Open Issues. Computers & Security. 2023. Vol. 125. 103028.

15. Kim, H. Security and Vulnerability of SCADA Systems over IP-Based Wireless Sensor Networks. International Journal of Distributed Sensor Networks. 2012. Vol. 8(11). 268478.

16. Bonney, G., Höfken, H., Paffen, B., Schuba, M. ICS/SCADA Security Analysis of a Beckhoff CX5020 PLC. In: 2015 International Conference on Information Systems Security and Privacy (ICISSP). IEEE, 2015. P. 1-6.

17. Zare, F., Mahmoudi-Nasr, P., Yousefpour, R. A Real-Time Network-Based Anomaly Detection in Industrial Control Systems. International Journal of Critical Infrastructure Protection. 2024. Vol. 45. 100676.

18. Liu, Y., Garg, S., Nie, J., Zhang, Y., Xiong, Z., Kang, J., Hossain, M. S. Deep Anomaly Detection for Time-Series Data in Industrial IoT: A Communication-Efficient On-Device Federated Learning Approach. IEEE Internet of Things Journal. 2020. Vol. 8(8). P. 6348-6358.

19. Kabore, R., Kouassi, A., N'goran, R., Asseu, O., Kermarrec, Y., Lenca, P. Review of Anomaly Detection Systems in Industrial Control Systems Using Deep Feature Learning Approach. Engineering. 2021. Vol. 13(1). P. 30-44.

20. Birihanu, E., Soullami, A., Lendák, I. Enhancing Industrial Control Systems Security: Real-Time Anomaly Detection with Uncertainty Estimation. In: International Conference on Discovery Science. Cham: Springer Nature Switzerland, 2024.

21. Lee, M. C., Lin, J. C., Gan, E. G. ReRe: A Lightweight Real-Time Ready-to-Go Anomaly Detection Approach for Time Series. In: 2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC). IEEE, 2020. P. 322-327.

22. OTNetGuard. [Електронний ресурс]. - Режим доступу: https://cynalytica.com/otnetguard/.

23. Zhao, X., Zhang, L., Cao, Y., Jin, K., Hou, Y. Anomaly Detection Approach in Industrial Control Systems Based on Measurement Data. Information. 2022. Vol. 13(10). 450. DOI: 10.3390/info13100450.

24. Perales Gómez, Á. L., Fernández Maimó, L., Huertas Celdrán, A., García Clemente, F. J. MADICS: A Methodology for Anomaly Detection in Industrial Control Systems. Symmetry. 2020. Vol. 12(10). 1583. DOI:10.3390/sym12101583.

25. Ike, M., Phan, K., Sadoski, K., Valme, R., Lee, W. Scaphy: Detecting Modern ICS Attacks by Correlating Behaviors in SCADA and Physical. In: 2023 IEEE Symposium on Security and Privacy (SP). IEEE, 2023. P. 20-37.

26. Пантюшенко О. О., Гринько І. О. Штучний інтелект у сфері кібербезпеки: інновації, виклики та перспективи розвитку. - КПІ ім. Ігоря Сікорського, 2023. [Електронний ресурс]. - Режим доступу:https://ela.kpi.ua/server/api/core/bitstreams/66385947-f42b-4cde-982e-0fbadd9e8647/content.

27. Застосування ШІ у кібербезпеці: роль та переваги. Wezom, 2024. [Електронний ресурс]. - Режим доступу:https://wezom.com.ua/ua/blog/zastosuvannya-shi-u-kiberbezpetsi-rol-ta-perevagi.

28. Mohamed, N. Artificial Intelligence and Machine Learning in Cybersecurity: A Deep Dive into State-of-the-Art Techniques and Future Paradigms. Knowl. Inf. Syst. 2025. Vol. 67. P. 6969-7055. DOI: 10.1007/s10115-025-02429-y.

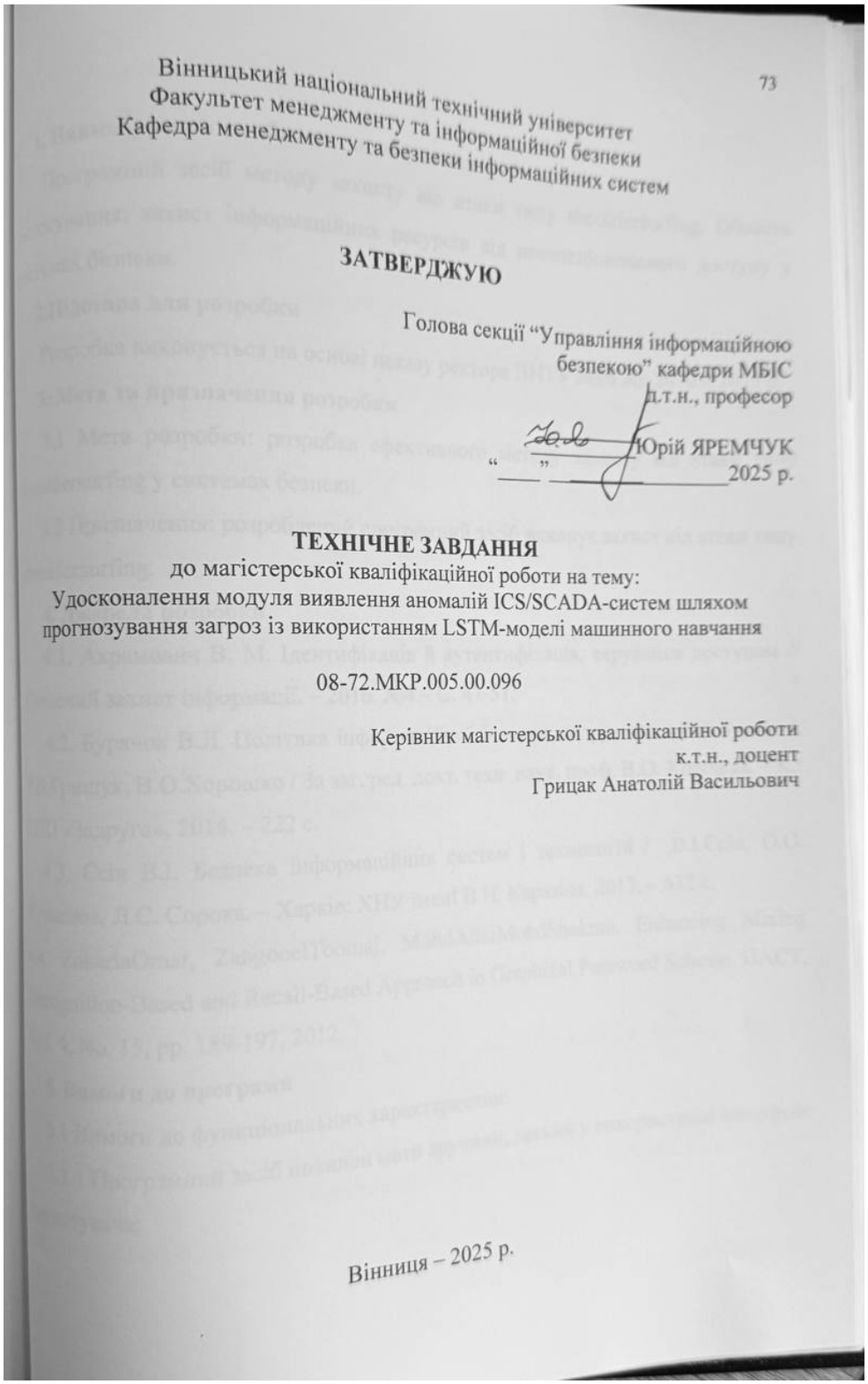
29. Kumar, A., Gutierrez, J. A. Impact of Machine Learning on Intrusion Detection Systems for the Protection of Critical Infrastructure. Information. 2025. Vol. 16(7). 515. DOI: 10.3390/info16070515.

30. Salem, A. H., Azzam, S. M., Emam, O. E. Advancing Cybersecurity: A Comprehensive Review of AI-Driven Detection Techniques. Journal of Big Data. 2024. Vol. 11. 105. DOI: 10.1186/s40537-024-00957-y.

31. Ajayi, A., Akerele, J., Odio, P., Collins, A., Babatunde, G., Mustapha, D. Using AI and Machine Learning to Predict and Mitigate Cybersecurity Risks in Critical Infrastructure. 2025. P. 204-224.

32. Офіційний ресурс компанії 2 WAF Security. [Електронний ресурс]. - Режим доступу: https://2waf.com

ДОДАТОК А Технічне завдання



Вінницький національний технічний університет
Факультет менеджменту та інформаційної безпеки
Кафедра менеджменту та безпеки інформаційних систем

73

ЗАТВЕРДЖУЮ

Голова секції "Управління інформаційною безпекою" кафедри МБІС
д.т.н., професор
Юрій ЯРЕМЧУК
"__" _____ 2025 р.

ТЕХНІЧНЕ ЗАВДАННЯ
до магістерської кваліфікаційної роботи на тему:
Удосконалення модуля виявлення аномалій ICS/SCADA-систем шляхом прогнозування загроз із використанням LSTM-моделі машинного навчання

08-72.МКР.005.00.096

Керівник магістерської кваліфікаційної роботи
к.т.н., доцент
Грицак Анатолій Васильович

Вінниця – 2025 р.

1. Найменування та область застосування

Програмний засіб методу захисту від атаки типу shouldersurfing. Область застосування: захист інформаційних ресурсів від несанкціонованого доступу у системах безпеки.

2.Підстава для розробки

Розробка виконується на основі наказу ректора ВНТУ №96 від 20. 03. 2025 р.

3. Мета та призначення розробки

3.1 Мета розробки: розробка ефективного методу захисту від атаки типу shouldersurfing у системах безпеки.

3.2 Призначення: розроблений програмний засіб виконує захист від атаки типу shouldersurfing.

4. Джерела розробки

4.1. Ахрамович В. М. Ідентифікація й аутентифікація, керування доступом // Сучасний захист інформації. – 2016. №4.– С. 47-51.

4.2. Бурячок В.Л. Політика інформаційної безпеки: підручник. / В.Л.Бурячок, Р.В.Грищук, В.О.Хорошко / За заг. ред. докт. техн. наук, проф. В.О. Хорошка. – К.: ПВП «Задруга», 2014. – 222 с.

4.3. Єсін В.І. Безпека інформаційних систем і технологій / В.І.Єсін, О.О. Кузнецов, Л.С. Сорока. – Харків: ХНУ імені В.Н. Каразіна, 2013. – 632 с.

4.4. ZakariaOmar, ZangooeiToomaj, MohdAfiziMohdShukran. Enhancing Mixing Recognition-Based and Recall-Based Approach in Graphical Password Scheme. IJACT, Vol. 4, No. 15, pp. 189-197, 2012.

5. Вимоги до програми

5.1 Вимоги до функціональних характеристик:

5.1.1 Програмний засіб повинен мати зручний, легкий у використанні інтерфейс користувача;

5.1.2 Реалізація методу не повинна вимагати спеціальних ліцензійних програмних додатків;

5.1.3 Програмний засіб повинен виконувати процес автентифікації користувачів у системі.

5.2 Вимоги до надійності:

5.2.1 Програмний засіб повинен працювати без помилок, у випадку виникнення критичних ситуацій необхідно передбачити виведення відповідних повідомлень;

5.2.2 Бази даних повинні бути налаштовані на автоматичне створення резервних копій;

5.2.3 Програмний засіб повинен виконувати свої функції.

5.3 Вимоги до складу і параметрів технічних засобів:

– процесор – Pentium 1500 МГц і подібні до них;

– оперативна пам’ять – не менше 512 Mb;

– середовище функціонування – операційна система сімейство Windows;

– вимоги до техніки безпеки при роботі з програмою повинні відповідати існуючим вимогам та стандартам з техніки безпеки при користуванні комп’ютерною технікою.

6. Вимоги до програмної документації

6.1 Обов’язкова поетапна інструкція для майбутніх користувачів, наведена у пункті 3.4

7. Вимоги до технічного захисту інформації

7.1 Необхідно забезпечити захист розроблюваного програмного засобу від несанкціонованого використання.

7.2 Неможливість отримання доступу незареєстрованих користувачів до інформаційних ресурсів.

8. Техніко-економічні показники

8.1 Цінність результатів використання даного проекту повинна перевищувати витрати на його реалізацію.

8.2 Має бути реалізований таким чином, щоб підходити для використання широкого загалу.

8.2 Має бути реалізований таким чином, щоб підходити для використання 76
широкого загалу.

9. Стадії та етапи розробки

№ з/п	Назва етапів магістерської кваліфікаційної роботи	Початок	Закінчення
1	Визначення напрямку магістерської роботи, формулювання теми	24.09.25р.	30.09.25р.
2	Аналіз предметної області обраної теми	01.10.25р.	02.10.25р.
3	Апробація отриманих результатів	04.10.25р.	10.10.25р.
4	Розробка алгоритму роботи	15.10.25р.	20.10.25р.
5	Написання магістерської роботи на основі розробленої теми	20.10.25р.	27.10.25р.
6	Розробка економічної частини	28.10.25р.	02.11.25р.
7	Передзахист магістерської кваліфікаційної роботи	03.11.25р.	03.11.25р.
8	Виправлення, уточнення, корегування магістерської кваліфікаційної роботи	17.11.25р.	29.11.25р.
9	Захист магістерської кваліфікаційної роботи	10.12.25р.	10.12.25р.

10. Порядок контролю та прийому

10.1 До приймання магістерської кваліфікаційної роботи надається:

- ПЗ до магістерської кваліфікаційної роботи;
- програмний додаток;
- презентація;
- відзив керівника роботи;
- відзив опонента

Технічне завдання до виконання прийняв _____ Вязун Д.С.

ДОДАТОК Б Лістинг коду

```
# Cell A
!pip install --upgrade pip setuptools wheel
!pip install torch torchvision torchaudio --index-url https://download.pytorch.org/whl/cu118
!pip install scikit-learn matplotlib tqdm

import os, time
import numpy as np
import torch, random
import torch.nn as nn
from torch.utils.data import Dataset, DataLoader
from sklearn.metrics import roc_auc_score, precision_recall_fscore_support
import matplotlib.pyplot as plt
from tqdm.notebook import tqdm
device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')
print("Device:", device)

def gen_synthetic_series(length=3600, F=4, seed=42):
    # length — секундами; F — число сенсоров (например pressure,temp,flow,cmd)
    rnd = np.random.RandomState(seed)
    t = np.arange(length)
    data = np.zeros((length, F), dtype=float)
    # channel 0: pressure — base sine + slow drift
    data[:,0] = 10 + 0.5*np.sin(2*np.pi*t/300) + 0.005*(t/100) + 0.2*rnd.randn(length)
    # channel 1: temperature — correlated with channel0 + noise
    data[:,1] = 50 + 0.3*np.sin(2*np.pi*t/300 + 0.2) + 0.01*(t/200) + 0.15*rnd.randn(length) +
0.2*data[:,0]
    # channel 2: flow — roughly inverse relation with pressure + noise
    data[:,2] = 100 - 0.8*data[:,0] + 0.5*rnd.randn(length)
    # channel 3: command flag binary (0/1), random operator actions
    data[:,3] = (rnd.rand(length) < 0.01).astype(float)
    return data
```

```
# insert anomalies: returns labels vector (0 normal,1 anomaly)
def insert_anomalies(data, n_anom=8, width=30):
    length = data.shape[0]
    labels = np.zeros(length, dtype=int)
    rnd = np.random.RandomState(123)
    for _ in range(n_anom):
        pos = rnd.randint(200, length-200)
        kind = rnd.choice(['spike','substitute','phys_inconsistency'])
        if kind=='spike':
            ch = rnd.randint(0,data.shape[1]-1)
            data[pos:pos+width,ch] += rnd.normal(5,2)  # large spike
        elif kind=='substitute':
            # replace sensor 0 with constant false value
            data[pos:pos+width,0] = data[pos:pos+width,0] + rnd.normal(8,3)
        else:
            # make physical inconsistency: set flow to high while pressure drops
            data[pos:pos+width,2] += rnd.normal(30,5)
            data[pos:pos+width,0] -= rnd.normal(3,1)
        labels[pos:pos+width] = 1
    return data, labels

# Create series
series = gen_synthetic_series(length=7200, F=4)
series, labels = insert_anomalies(series, n_anom=12, width=60)
print("Series shape:", series.shape)
print("Anomaly ratio:", labels.mean())
# parameters
SEQ_LEN = 120  # window length in seconds
STEP = 1

# online aggregator functions
def agg_features(window):
    # window: (seq_len, F)
    # produce aggregated features vector (mean,std,slope,correlations...)
```

```
    feats = []
    feats.extend(window.mean(axis=0).tolist())     # mean per sensor
    feats.extend(window.std(axis=0).tolist())      # std per sensor
    # slope (difference mean)
    diffs = np.diff(window, axis=0)
    feats.extend(diffs.mean(axis=0).tolist())
    # simple pairwise correlations (upper triangle)
    F = window.shape[1]
    flat_corrs = []
    for i in range(F):
        for j in range(i+1, F):
            c = np.corrcoef(window[:,i], window[:,j])[0,1]
            flat_corrs.append(0.0 if np.isnan(c) else c)
    feats.extend(flat_corrs)
    # percent missing (here none) and entropy approx (discrete bins)
    # entropy per sensor
    for i in range(F):
        hist,_ = np.histogram(window[:,i], bins=10)
        p = hist / (hist.sum()+1e-9)
        ent = -np.sum([pi*np.log1p(pi) for pi in p])
        feats.append(ent)
    return np.array(feats, dtype=np.float32)

# Build dataset of windows
X_seq = []
X_agg = []
Y = []
idxs = []
for start in range(0, series.shape[0]-SEQ_LEN+1, STEP):
    w = series[start:start+SEQ_LEN]
    agg = agg_features(w)
    label = labels[start:start+SEQ_LEN].max()  # if any point in window anomalous -> label window
anomalous
    X_seq.append(w.astype(np.float32))
```

```
    X_agg.append(agg)
    Y.append(label)
    idxs.append(start)
X_seq = np.stack(X_seq)   # shape (N_windows, SEQ_LEN, F)
X_agg = np.stack(X_agg)   # shape (N_windows, feat_dim)
Y = np.array(Y).astype(np.int64)
print("Windows:", X_seq.shape, X_agg.shape, Y.mean())
class SCADADataset(Dataset):
    def __init__(self, seqs, aggs, labels):
        self.seqs = torch.from_numpy(seqs)  # float32
        self.aggs = torch.from_numpy(aggs)
        self.labels = torch.from_numpy(labels).float()
    def __len__(self): return len(self.labels)
    def __getitem__(self, i):
        return self.seqs[i], self.aggs[i], self.labels[i]

# split train/val (time-based)
N = len(Y)
train_cut = int(0.6 * N)
val_cut = int(0.8 * N)
ds_train = SCADADataset(X_seq[:train_cut], X_agg[:train_cut], Y[:train_cut])
ds_val   = SCADADataset(X_seq[train_cut:val_cut], X_agg[train_cut:val_cut], Y[train_cut:val_cut])
ds_test  = SCADADataset(X_seq[val_cut:], X_agg[val_cut:], Y[val_cut:])

loader_train = DataLoader(ds_train, batch_size=32, shuffle=True)
loader_val   = DataLoader(ds_val, batch_size=64, shuffle=False)
loader_test  = DataLoader(ds_test, batch_size=64, shuffle=False)
class LSTM_MLP_Fusion(nn.Module):
    def __init__(self, input_dim, agg_dim, hidden_size=128, lstm_layers=1, mlp_hidden=64):
        super().__init__()
        self.lstm = nn.LSTM(input_dim, hidden_size, num_layers=lstm_layers, batch_first=True)
        # optional: layernorm on last hidden
        self.fc_seq = nn.Sequential(
            nn.Linear(hidden_size, hidden_size//2),
```

```
            nn.ReLU(),
            nn.Dropout(0.1)
        )
        self.mlp = nn.Sequential(
            nn.Linear(agg_dim, mlp_hidden),
            nn.ReLU(),
            nn.Dropout(0.1),
            nn.Linear(mlp_hidden, mlp_hidden//2),
            nn.ReLU()
        )
        cat_dim = (hidden_size//2) + (mlp_hidden//2)
        self.classifier = nn.Sequential(
            nn.Linear(cat_dim, 64),
            nn.ReLU(),
            nn.Dropout(0.1),
            nn.Linear(64, 1),
            nn.Sigmoid()
        )
    def forward(self, x_seq, x_agg):
        # x_seq: (B, T, F)
        out, (h_n, c_n) = self.lstm(x_seq)   # h_n: (num_layers, B, hidden_size)
        h = h_n[-1]                          # last layer hidden (B, hidden_size)
        h_seq = self.fc_seq(h)
        h_feat = self.mlp(x_agg)
        h_cat = torch.cat([h_seq, h_feat], dim=1)
        out = self.classifier(h_cat).squeeze(-1)
        return out

# create model
device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')
agg_dim = X_agg.shape[1]
model = LSTM_MLP_Fusion(input_dim=series.shape[1], agg_dim=agg_dim).to(device)
optimizer = torch.optim.Adam(model.parameters(), lr=1e-3)
criterion = nn.BCELoss()
```

```
clip = 1.0

def train_epoch(loader):
    model.train()
    running_loss = 0
    for x_seq, x_agg, y in loader:
        x_seq = x_seq.to(device)
        x_agg = x_agg.to(device)
        y = y.to(device)
        optimizer.zero_grad()
        preds = model(x_seq, x_agg)
        loss = criterion(preds, y)
        loss.backward()
        torch.nn.utils.clip_grad_norm_(model.parameters(), clip)
        optimizer.step()
        running_loss += loss.item() * x_seq.size(0)
    return running_loss / len(loader.dataset)

def evaluate(loader):
    model.eval()
    ys = []
    ps = []
    with torch.no_grad():
        for x_seq, x_agg, y in loader:
            x_seq = x_seq.to(device)
            x_agg = x_agg.to(device)
            y = y.to(device)
            preds = model(x_seq, x_agg)
            ys.append(y.cpu().numpy())
            ps.append(preds.cpu().numpy())
    ys = np.concatenate(ys)
    ps = np.concatenate(ps)
    return ys, ps
```

```
# training
best_auc = 0.0
for epoch in range(1, 21):
    t0 = time.time()
    train_loss = train_epoch(loader_train)
    ys_val, ps_val = evaluate(loader_val)
    try:
        auc = roc_auc_score(ys_val, ps_val)
    except:
        auc = 0.0
    print(f"Epoch {epoch:02d} | train_loss {train_loss:.4f} | val_auc {auc:.4f} | time {time.time()-
t0:.1f}s")
    if auc > best_auc:
        best_auc = auc
        torch.save(model.state_dict(), "best_model.pth")
model.load_state_dict(torch.load("best_model.pth"))
ys_test, ps_test = evaluate(loader_test)
auc_test = roc_auc_score(ys_test, ps_test)
print("Test AUC:", auc_test)

# choose threshold by F1 on val set
from sklearn.metrics import f1_score
ys_val, ps_val = evaluate(loader_val)
best_f1 = 0
best_thr = 0.5
for thr in np.linspace(0.01, 0.99, 99):
    f1 = f1_score(ys_val, ps_val>thr)
    if f1 > best_f1:
        best_f1 = f1; best_thr = thr
print("Best val F1", best_f1, "thr", best_thr)

preds_bin = (ps_test > best_thr).astype(int)
prec, rec, f1, _ = precision_recall_fscore_support(ys_test, preds_bin, average='binary',
zero_division=0)
```

```
print(f"Test precision {prec:.3f}, recall {rec:.3f}, f1 {f1:.3f}")

# plot sample scores
plt.figure(figsize=(14,3))
plt.plot(ps_test, label='score')
plt.plot(ys_test*1.0, label='label (0/1)', alpha=0.6)
plt.legend(); plt.title("Test set scores vs labels"); plt.show()
from sklearn.metrics import roc_curve, auc

fpr, tpr, _ = roc_curve(ys_test, ps_test)
roc_auc = auc(fpr, tpr)

plt.figure(figsize=(6,6))
plt.plot(fpr, tpr, label=f"LSTM (AUC = {roc_auc:.3f})")
plt.plot([0,1],[0,1],'--',color='grey')
plt.xlabel("False Positive Rate")
plt.ylabel("True Positive Rate")
plt.title("ROC-крива LSTM-моделі")
plt.legend()
plt.grid(True)
plt.savefig("roc_lstm.png", dpi=300)
plt.show()
from sklearn.metrics import precision_recall_curve

prec_curve, rec_curve, _ = precision_recall_curve(ys_test, ps_test)

plt.figure(figsize=(6,6))
plt.plot(rec_curve, prec_curve)
plt.xlabel("Recall")
plt.ylabel("Precision")
plt.title("Precision-Recall крива")
plt.grid(True)
plt.savefig("pr_lstm.png", dpi=300)
plt.show()
```

```
from sklearn.ensemble import IsolationForest
from sklearn.svm import OneClassSVM
import numpy as np

# тестовий набір для агрегованих ознак
X_test_agg = X_agg[val_cut:]

# Isolation Forest
iso = IsolationForest(contamination=0.01, random_state=42)
iso.fit(X_agg[:train_cut])
iso_scores = -iso.decision_function(X_test_agg)

# One-Class SVM
svm = OneClassSVM(kernel='rbf', nu=0.01)
svm.fit(X_agg[:train_cut])
svm_scores = -svm.decision_function(X_test_agg)

# нормалізація (min-max)
iso_norm = (iso_scores - iso_scores.min()) / (np.ptp(iso_scores) + 1e-9)
svm_norm = (svm_scores - svm_scores.min()) / (np.ptp(svm_scores) + 1e-9)

# графік
plt.figure(figsize=(12,4))
plt.plot(ps_test, label='LSTM score')
plt.plot(iso_norm, label='IsolationForest norm')
plt.plot(svm_norm, label='OCSVM norm')
plt.plot(ys_test, label='True label', alpha=0.5)
plt.legend()
plt.title("Порівняння реакції моделей на тестовому фрагменті")
plt.grid(True)
plt.savefig("compare_models.png", dpi=300)
plt.show()
import seaborn as sns
```

```
sample_idx = 200   # будь-яке вікно
seq = X_seq[sample_idx]

plt.figure(figsize=(10,5))
sns.heatmap(seq.T, cmap="viridis")
plt.title("Теплова карта одного вікна (120 сек)")
plt.xlabel("Часовий крок")
plt.ylabel("Канал")
plt.savefig("window_heatmap.png", dpi=300)
plt.show()
from sklearn.metrics import roc_auc_score, precision_recall_fscore_support

# ===== LSTM =====
auc_lstm = roc_auc_score(ys_test, ps_test)
pred_lstm = (ps_test > best_thr).astype(int)
prec_lstm, rec_lstm, f1_lstm, _ = precision_recall_fscore_support(
    ys_test, pred_lstm, average='binary', zero_division=0)

# ===== Isolation Forest =====
# нормалізовані скоринги (вже зробили вище)
iso_auc = roc_auc_score(ys_test, iso_norm)
iso_pred = (iso_norm > 0.5).astype(int)
prec_iso, rec_iso, f1_iso, _ = precision_recall_fscore_support(
    ys_test, iso_pred, average='binary', zero_division=0)

# ===== One-Class SVM =====
svm_auc = roc_auc_score(ys_test, svm_norm)
svm_pred = (svm_norm > 0.5).astype(int)
prec_svm, rec_svm, f1_svm, _ = precision_recall_fscore_support(
    ys_test, svm_pred, average='binary', zero_division=0)

# ===== Threshold baseline (simple rule) =====
# приклад: anomaly if pressure < median-std or > median+std
pressure = X_test_agg[:,0]
```

```
thr_lo = np.median(pressure) - np.std(pressure)
thr_hi = np.median(pressure) + np.std(pressure)
baseline_pred = ((pressure < thr_lo) | (pressure > thr_hi)).astype(int)
prec_b, rec_b, f1_b, _ = precision_recall_fscore_support(
    ys_test, baseline_pred, average='binary', zero_division=0)
auc_b = roc_auc_score(ys_test, pressure)

# ===== Таблиця як pandas DataFrame =====
import pandas as pd

table = pd.DataFrame({
    "Model": ["LSTM + MLP", "Isolation Forest", "One-Class SVM", "Threshold baseline"],
    "AUC": [auc_lstm, iso_auc, svm_auc, auc_b],
    "Precision": [prec_lstm, prec_iso, prec_svm, prec_b],
    "Recall": [rec_lstm, rec_iso, rec_svm, rec_b],
    "F1": [f1_lstm, f1_iso, f1_svm, f1_b],
})

print(table)

table.to_csv("comparison_table.csv", index=False)
```

ДОДАТОК Г Ілюстраційний матеріал

Удосконалення модуля виявлення аномалій ICS/SCADA-систем шляхом прогнозування загроз із використанням LSTM-моделі машинного навчання

Виконав: ст.гр.2КІТС-24м Вязун Д.С.

Керівник: Грицак А.В.

2

2222

Мета та задачі роботи

Метою роботи є розроблення та дослідження модуля виявлення аномалій у ICS/SCADA-системах на основі моделей глибинного навчання.

Для досягнення поставленої мети в роботі **поставлені і розв'язані наступні задачі**:

1. Провести аналіз особливостей промислових систем керування та сучасних підходів до їх захисту.
2. Підготувати вхідні дані, які поєднують мережеві журнали, операторські події та технологічні сигнали.
3. Побудувати та провести навчання моделі нейронної мережі.
4. Порівняти отримані результатів із іншими методиками.

3

Проблема та актуальність

1. ICS/SCADA працюють на змішаному обладнанні, де є слабкі протоколи, довгі життєві цикли та неможливість часто оновлювати ПЗ
2. У таких системах аномалія може швидко призвести до фізичних наслідків.
3. Більшість традиційних засобів виявляють тільки відомі атаки.
4. Потрібен підхід, який бачить поведінкові відхилення, а не сигнатури.

4

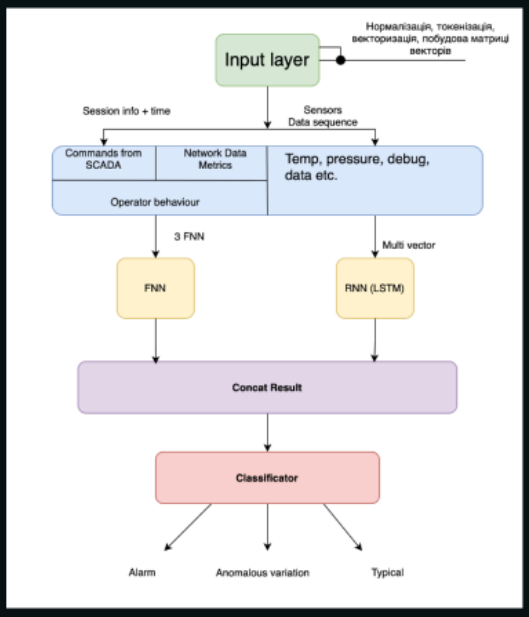
Обмеження існуючих рішень

1. Системи моніторингу часто працюють лише з мережевим трафіком, ігноруючи фізичні сигнали.
2. Частина моделей надто важка для використання на промислових об'єктах.
3. Середня точність сильно падає при появі нових типів аномалій.

В роботі запропоновано поєднання статичних та часових ознак, чого в базових моделях немає

5

Основна ідея розробки



Гібридний модуль:
FNN + LSTM, що аналізує одночасно команди, мережеву активність і динаміку технологічних параметрів.

Дозволяє бачити як «миттєві» відхилення, так і зміни в часі.

6

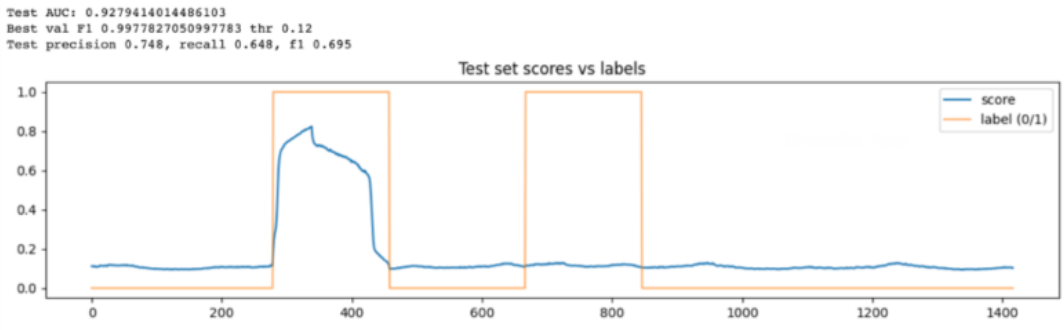
Фрагмент використаного датасету

```
2024-05-12 13:04:51  pressure=5.32  temperature=59.9  flow=12.7  valve=0  plc_cmd=HOLD   latency=19ms  loss=0.3%  operator=0  type=normal   anomaly=0
2024-05-12 13:04:52  pressure=5.29  temperature=60.1  flow=12.6  valve=0  plc_cmd=HOLD   latency=20ms  loss=0.3%  operator=0  type=normal   anomaly=0
2024-05-12 13:04:53  pressure=5.27  temperature=60.0  flow=12.4  valve=0  plc_cmd=HOLD   latency=21ms  loss=0.4%  operator=0  type=normal   anomaly=0
2024-05-12 13:04:54  pressure=5.30  temperature=60.2  flow=12.5  valve=0  plc_cmd=HOLD   latency=22ms  loss=0.5%  operator=0  type=normal   anomaly=0
2024-05-12 13:04:55  pressure=5.28  temperature=60.3  flow=12.4  valve=0  plc_cmd=HOLD   latency=23ms  loss=0.5%  operator=0  type=normal   anomaly=0

2024-05-12 13:04:56  pressure=5.25  temperature=60.4  flow=12.2  valve=0  plc_cmd=HOLD   latency=24ms  loss=0.6%  operator=0  type=normal   anomaly=0
2024-05-12 13:04:57  pressure=5.26  temperature=60.5  flow=12.3  valve=0  plc_cmd=HOLD   latency=24ms  loss=0.5%  operator=0  type=normal   anomaly=0
2024-05-12 13:04:58  pressure=5.24  temperature=60.3  flow=12.1  valve=0  plc_cmd=HOLD   latency=25ms  loss=0.7%  operator=0  type=normal   anomaly=0
2024-05-12 13:04:59  pressure=5.19  temperature=60.1  flow=12.0  valve=0  plc_cmd=HOLD   latency=28ms  loss=0.8%  operator=0  type=normal   anomaly=0
2024-05-12 13:05:00  pressure=5.17  temperature=59.8  flow=11.9  valve=0  plc_cmd=HOLD   latency=29ms  loss=0.8%  operator=0  type=normal   anomaly=0

2024-05-12 13:05:01  pressure=5.21  temperature=60.3  flow=12.5  valve=0  plc_cmd=HOLD   latency=21ms  loss=0.4%  operator=0  type=normal   anomaly=0
2024-05-12 13:05:02  pressure=5.18  temperature=60.1  flow=12.4  valve=0  plc_cmd=HOLD   latency=22ms  loss=0.5%  operator=0  type=normal   anomaly=0
2024-05-12 13:05:03  pressure=5.25  temperature=60.2  flow=12.3  valve=0  plc_cmd=HOLD   latency=20ms  loss=0.3%  operator=0  type=normal   anomaly=0

2024-05-12 13:05:04  pressure=4.78  temperature=63.9  flow=10.1  valve=1  plc_cmd=OPEN   latency=48ms  loss=1.1%  operator=0  type=warning  anomaly=1
2024-05-12 13:05:05  pressure=4.51  temperature=67.2  flow= 9.3  valve=1  plc_cmd=OPEN   latency=55ms  loss=1.4%  operator=0  type=warning  anomaly=1
2024-05-12 13:05:06  pressure=4.22  temperature=70.4  flow= 8.2  valve=1  plc_cmd=OPEN   latency=61ms  loss=1.9%  operator=0  type=warning  anomaly=1

2024-05-12 13:05:07  pressure=4.10  temperature=71.8  flow= 7.9  valve=1  plc_cmd=OPEN   latency=65ms  loss=2.1%  operator=0  type=alert    anomaly=1
2024-05-12 13:05:08  pressure=4.05  temperature=73.6  flow= 7.4  valve=1  plc_cmd=OPEN   latency=72ms  loss=2.4%  operator=0  type=alert    anomaly=1
2024-05-12 13:05:09  pressure=3.97  temperature=74.5  flow= 7.1  valve=1  plc_cmd=OPEN   latency=77ms  loss=2.7%  operator=0  type=alert    anomaly=1

2024-05-12 13:05:10  pressure=4.33  temperature=71.9  flow= 8.6  valve=1  plc_cmd=HOLD   latency=61ms  loss=1.8%  operator=1  type=info     anomaly=0
2024-05-12 13:05:11  pressure=4.88  temperature=67.4  flow= 9.5  valve=1  plc_cmd=CLOSE  latency=48ms  loss=1.1%  operator=1  type=info     anomaly=0
2024-05-12 13:05:12  pressure=5.03  temperature=63.1  flow=10.7  valve=1  plc_cmd=CLOSE  latency=32ms  loss=0.9%  operator=1  type=info     anomaly=0

2024-05-12 13:05:13  pressure=5.11  temperature=61.8  flow=11.8  valve=0  plc_cmd=HOLD   latency=27ms  loss=0.7%  operator=0  type=normal   anomaly=0
2024-05-12 13:05:14  pressure=5.16  temperature=61.2  flow=12.1  valve=0  plc_cmd=HOLD   latency=25ms  loss=0.6%  operator=0  type=normal   anomaly=0
2024-05-12 13:05:15  pressure=5.19  temperature=60.9  flow=12.3  valve=0  plc_cmd=HOLD   latency=24ms  loss=0.5%  operator=0  type=normal   anomaly=0
2024-05-12 13:05:16  pressure=5.21  temperature=60.7  flow=12.5  valve=0  plc_cmd=HOLD   latency=23ms  loss=0.5%  operator=0  type=normal   anomaly=0
2024-05-12 13:05:17  pressure=5.23  temperature=60.5  flow=12.6  valve=0  plc_cmd=HOLD   latency=22ms  loss=0.4%  operator=0  type=normal   anomaly=0

2024-05-12 13:05:18  pressure=5.20  temperature=60.4  flow=12.4  valve=0  plc_cmd=HOLD   latency=23ms  loss=0.5%  operator=0  type=normal   anomaly=0
2024-05-12 13:05:19  pressure=5.17  temperature=60.3  flow=12.3  valve=0  plc_cmd=HOLD   latency=24ms  loss=0.5%  operator=0  type=normal   anomaly=0
2024-05-12 13:05:20  pressure=5.14  temperature=60.2  flow=12.1  valve=0  plc_cmd=HOLD   latency=24ms  loss=0.6%  operator=0  type=normal   anomaly=0
2024-05-12 13:05:21  pressure=5.09  temperature=60.0  flow=11.9  valve=0  plc_cmd=HOLD   latency=26ms  loss=0.7%  operator=0  type=normal   anomaly=0
2024-05-12 13:05:22  pressure=5.04  temperature=59.8  flow=11.7  valve=0  plc_cmd=HOLD   latency=27ms  loss=0.8%  operator=0  type=normal   anomaly=0
```

7

8

Реакція запропонованої моделі на тестових даних

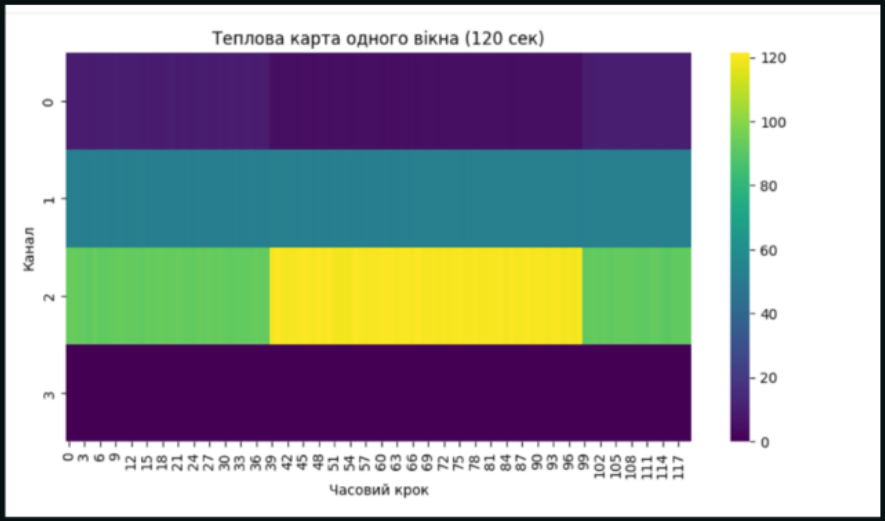


9

Порівняльна реакція моделей на одному фрагменті

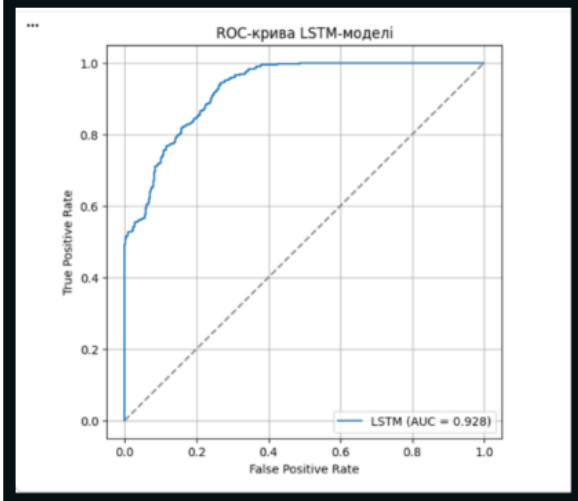
10

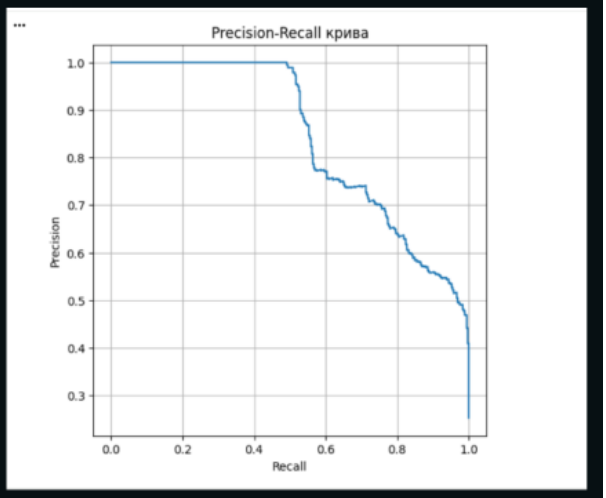
Теплова карта одного фрагмента



11

ROC-крива LSTM-моделі та Precision-Recall крива





12

Порівняння ефективності моделей

	Model	AUC	Precision	Recall	F1
0	LSTM + MLP	0.927941	0.748387	0.648045	0.694611
1	Isolation Forest	0.723039	0.521739	0.502793	0.512091
2	One-Class SVM	0.171079	0.083591	0.150838	0.107570
3	Threshold baseline	0.800052	1.000000	0.396648	0.568000

13

Висновки

1

Розроблено та досліджено гібридний модуль виявлення аномалій для ICS/SCADA, який поєднує аналіз статичних параметрів і часових змін, завдяки чому вдалося підвищити точність розпізнавання відхилень від нормальної поведінки технологічних процесів.

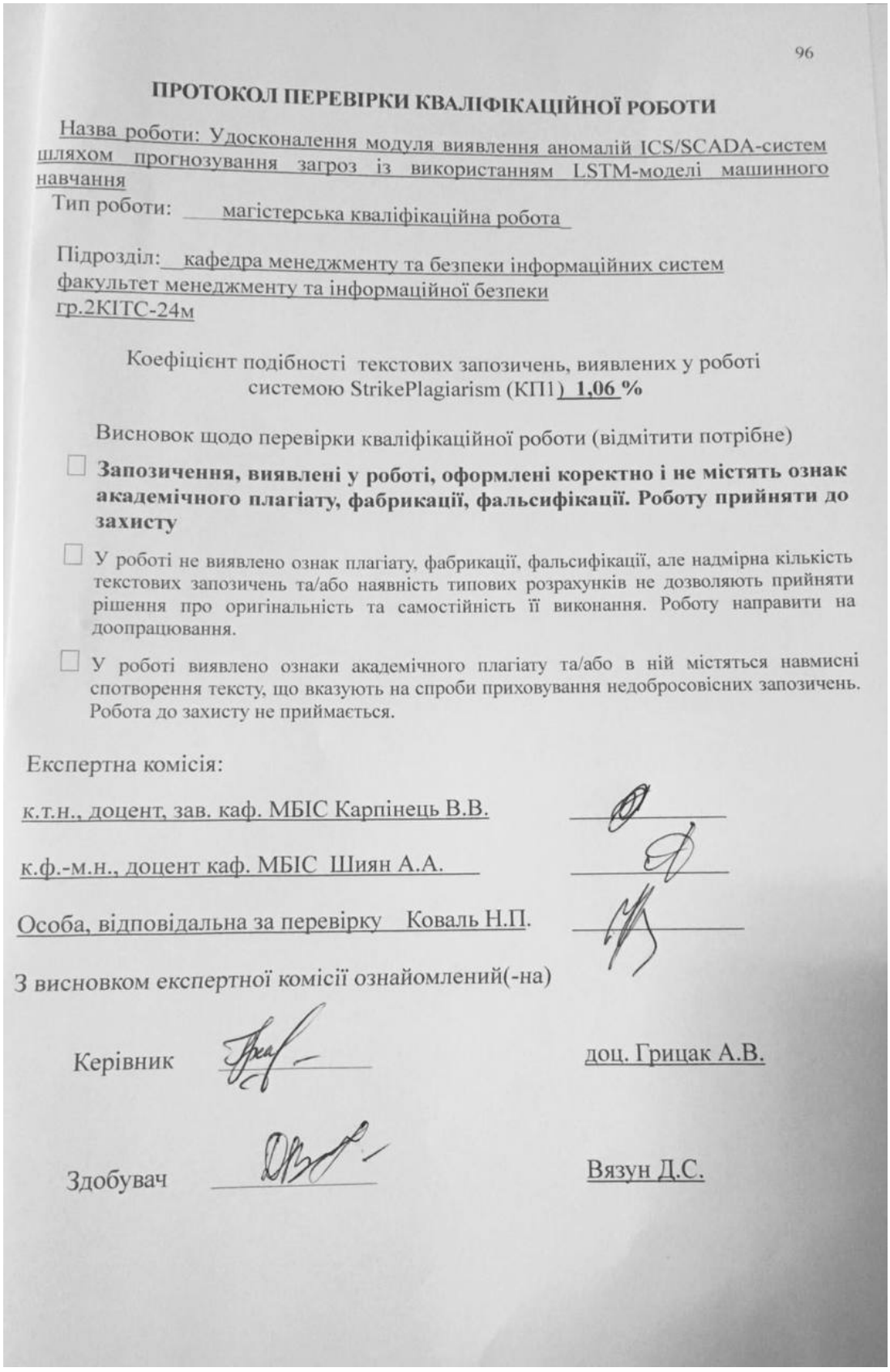
2

Запропонована архітектура з послідовною обробкою даних та урахуванням контексту роботи обладнання дала змогу зменшити похибку під час прогнозування та знизити кількість хибних спрацювань у порівнянні з традиційними підходами.

3

Проведене дослідження підтвердило практичну придатність запропонованого методу: модуль може бути інтегрований у системи моніторингу промислових об'єктів, адаптований під різні типи технологічних процесів та використаний як основа для подальшого удосконалення засобів кіберзахисту.

ДОДАТОК Д Протокол перевірки на антиплагіат



96

ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ

Назва роботи: Удосконалення модуля виявлення аномалій ICS/SCADA-систем шляхом прогнозування загроз із використанням LSTM-моделі машинного навчання

Тип роботи: магістерська кваліфікаційна робота

Підрозділ: кафедра менеджменту та безпеки інформаційних систем
факультет менеджменту та інформаційної безпеки
гр.2КІТС-24м

Коефіцієнт подібності текстових запозичень, виявлених у роботі системою StrikePlagiarism (КП1) **1,06 %**

Висновок щодо перевірки кваліфікаційної роботи (відмітити потрібне)

☐ **Запозичення, виявлені у роботі, оформлені коректно і не містять ознак академічного плагіату, фабрикації, фальсифікації. Роботу прийняти до захисту**

☐ У роботі не виявлено ознак плагіату, фабрикації, фальсифікації, але надмірна кількість текстових запозичень та/або наявність типових розрахунків не дозволяють прийняти рішення про оригінальність та самостійність її виконання. Роботу направити на доопрацювання.

☐ У роботі виявлено ознаки академічного плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень. Робота до захисту не приймається.

Експертна комісія:

к.т.н., доцент, зав. каф. МБІС Карпінець В.В. _____

к.ф.-м.н., доцент каф. МБІС Шиян А.А. _____

Особа, відповідальна за перевірку Коваль Н.П. _____

З висновком експертної комісії ознайомлений(-на)

Керівник _____ доц. Грицак А.В.

Здобувач _____ Вязун Д.С.