

ЗВІТ

Про виконання проекту за програмою «Остпартнершафт ДААД»

з 15.11.22 по 15.01.23

По проекту було призначено керівником проекту проф. Вольфганг Неїдль , менторів від наукового центру ЛЗС ЛУХ та для керівництва студентами ХНУРЕ проф. О. Кузьоміна.

Науковим центром ЛЗС були представлені наступні теми та ментори:

Title: Handwriting noise removal from images

Mentor: again Sajjad Kamali <kamali@l3s.de>

Student: Ripny Vladyslav <vladyslav.ripnyi@nure.ua>

Title: Python- reinforcement learning based flexible manufacturing system with order management for multi-agent reinforcement learning.

Mentor: Rezaul Abedin <abedin@l3s.de>

Student: Kaidan Anastasia <anastasiia.koidan@nure.ua>

Title: Python-based car production environment simulation project

Mentor: Rezaul Abedin <abedin@l3s.de>

Student: Valentyn Prokipets <valentyn.prokipets@nure.ua>

Title: Python-based car production environment simulation project

Mentor: Rezaul Abedin <abedin@l3s.de>

Student: Valentyn Prokipets <valentyn.prokipets@nure.ua>

Title: From Memory-Based Models to Graph-Based Models for Work different types of concept drift

Mentor: Amir Abolfazli <abolfazli@l3s.de>

Student: Kholodov Stanislav (stanislav.kholodov@nure.ua)

Title: Knowing What We Don't Know: Exploring Choice Classification methods

Mentor: Jeles Stefano Rios Vasconcelos <riosvasconcelos@l3s.de>

Student: Denys Berkovskyi (denys.berkovskyi@nure.ua)

Title: joint autoencoder with joint anomaly threshold Detection on production lines

Mentor: Sofiane Laridi <laridi@l3s.de> ,

Student: Anna Shustova (anna.shustrova@nure.ua)

Студент **Валентин Прокопець** виконав майже весь проект і отримав позитивний відгук ментора.

Він виконав наступні етапи роботи:

1. Розширення бази даних.
2. Розробив нові варіанти моделей.
3. Їм було рекомендовано для проекту використання GAN.
4. Створені синтетичні данні.
5. Виконана реплікація файлу CSV за допомогою GAN.
6. Застосовані синтетичні данні до середовища моделювання, що підтвердило високу якість виконаної роботи.

Студент **Владіслав Ріпний** мав таку послідовність завдань :

1. Вивчити проблематику теми проекту.
2. Прочитати роботи та статті на суміжні теми.
3. Вибрати 150 зображень з таблицями та рукописним шумом з вхідного набору кількох тисяч сканованих сторінок.
4. Провести маркування обраних зображень.
5. Реалізувати GNN (Graph Neural Network) модель.
6. Натренувати та протестувати модель.
7. Обробити результати та порівняти їх з існуючими моделями.

Студент виконав дуже ретельно 5 завдань. Зараз ментор тестує його модель (6 етап).

Ментором для **Анастасії Кайдан** були поставлені наступні задачі:

1. Знайомство з предметною областю для проектування та оцінка існуючих методів та програмних засобів для проектування.
2. Розробити програмний код та запустити код на локальному комп'ютері (оцінити результат).
3. Застосувати середовище з rllib для виконання завдання.
4. Налаштувати виконавчі функції програмного коду та знову запустити середовище за допомогою rllib.
5. Поєднайте наш код у rllib і спробувати запустити на ньому алгоритм Q-навчання.

Студентка Анастасії Кайдан дійшла до останнього етапу. Вона гарно виконала свою роботу.

Студент: **Холодов Станіслав** виконав такі кроки які були виконанні у процесі розробки проекту:

Крок 1: Реалізація алгоритму GraphPool (метод на основі графів, який працює з потоками даних із повторюваним дрейфом концепції) за допомогою фреймворку Scikit-Multiflow у Python.

Крок 2: Покращення GraphPool шляхом додавання компонента, який усуває

дисбаланс класів (для цієї частини ми використовували пакет `imbalanced-learn`).

Крок 3: Порівняння результатів із принаймні двома найсучаснішими не збалансованими класифікаторами потоку даних (для оцінки використовували середнє геометричне та показники збалансованої точності).

Крок 4: Використовували попереднє оцінювання (це метод оцінювання, який найчастіше використовують у класифікації потоку даних).

Крок 5: Оформлення звіту з проекту

Звіт містить наступні розділи:

1) Опис завдання

2) Робочий процес

3) Реалізація коду

3.1 Структура проекту

3.2 Код "main.py", дані для роботи та навантаження для навчальної моделі

3.3 Функція для навчання моделі

3.4 Реалізація моделі

3.5 Розбір додаткових даних

3.6 Додаткові утиліти

3.7 Збалансування набору даних

4) Результати роботи

5) Висновок

У цьому проекті була робота над тим, щоб зробити застосувати `GraphPool` і зробити її зручнішою та зробити і сумісною з потоками даних. Додано незбалансоване навчання для створення незбалансованості наборів даних і перевірено, як це працює з різними потоками даних. Можливо відмітити, що дані, які проходять через збалансований потік, працювали чіткіше та виглядали комфортніше. У подальшій роботі над проектом можна покращити роботу `graph pool` використовувати інші можливі бібліотеки.

Наставник поставив Анні Шустовій такі завдання:

1. Ознайомтеся з файлом (прикріпленням до листа) під назвою «OverviewDL», в якому описано матеріал про нейронні мережі, автокодер та варіанти їх використання.

2. Прочитавши цей документ і ознайомившись із цією темою, Анна перейшла до виявлення аномалій і того, як для цього використовуються автокодери.

3. Бібліотека Python (Keras: <https://keras.io>) була розглянута для частини впровадження/застосування.

4. Анна самостійно (за власною ініціативою) прочитала декілька статей про виявлення аномалій та роботу автокодувальника.

5. Створення письмового звіту, близько 10 сторінок, про виявлення аномалії за допомогою автокодера.

6. Формування загального звіту:

6.1 Вступ

6.2 Визначення та використання

6.3 Типи виявлення аномалій

6.4 Різні типи аномалій у виявленні аномалій

6.5 Випадки використання виявлення аномалій

6.6 Висновки

6.7 Література

Студентка має бажання розвивати та повністю завершити цей проект.

Виконавці в основному виконали поставлене завдання, в тому числі перейшли до 5 пункту плану роботи. За відгуками наставників, особливих успіхів досягли Прокопець, Рєпний, Кайдан і Холодов. Борковський втратив зв'язок після завершення двох етапів проекту. Але на останньому етапі всеж таки відправив наставнику свій аналіз статті. Порадившись із наставником, я запропонував залучити до подальшої реалізації проекту ще одного студента – Добре. Тепер наставник оцінює результати доопрацювання проектних звітів. Після хвороби **Анна Шустова** виконувала всі завдання, пов'язані з проектом. Наставник дав їй наступні завдання (які вона успішно виконала):

1. Прочитайте файл (прикріплений до листа) під назвою "OverviewDL", який описує матеріал про нейронні мережі, автокодер та варіанти їх використання.

2. Прочитавши цей документ і ознайомившись з темою, вона взялася за створення алгоритму виявлення аномалій за допомогою автокодера.

3. Бібліотека Python (Keras: <https://keras.io>) розглядалася для впровадження/застосування.

4. Анна Шусова добровільно прочитала декілька статей про виявлення аномалій та роботу автокодера.

5. Підготуйте письмовий звіт приблизно на 10 сторінках про виявлення аномалії за допомогою автокодерів.

6. Створіть загальний звіт.

6.1 Вступ.

6.2 Визначення та використання.

6.3 Типи виявлення аномалій.

6.4 Різні типи аномалій у виявленні аномалій.

6.5 Випадки використання виявлення аномалій.

6.6 Висновки.

6.7 Література.

Цей проект допоміг Анні Шустовій досягти її професійної освіти та цілей сертифікації з проектування інформаційних систем/управління технологіями. А. Шустова впевнена, що її дослідження розвинули навички та особисті знання, необхідні для кар'єри вченого. А.Шустова має бажання розвивати цей проект і доводити його до кінця.

Під час щотижневих обговорень і обговорень зі студентами проміжних результатів дизайну я вказав студентам на відповідні розділи моїх лекцій, які я їм читав, і на практичні питання, які необхідно застосувати для вирішення проблем дизайну.

Враховуючи, що студенти об'єктивно виконали проект в умовах військових дій в Україні з частими обстрілами, відключеннями електроенергії, Інтернету та були мотивовані продовжувати проектування, я вважаю, що напрацювання студентів можна використати для подальшого вдосконалення. Та створити команду виконавців для подальшої роботи над успішними проектами з використанням DL (Deep Learning).

О. Кузьомін

