

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**Сумський державний університет**  
**Факультет електроніки та інформаційних технологій**  
**Кафедра інформаційних технологій**

«До захисту допущено»

В.о. завідувача кафедри

\_\_\_\_\_ Світлана ВАЩЕНКО

\_\_\_\_\_ 2024 р.

**КВАЛІФІКАЦІЙНА РОБОТА**

**на здобуття освітнього ступеня магістр**

зі спеціальності 122 «Комп'ютерні науки»,

освітньо-наукової програми «Інформаційні технології проектування»

на тему: Інформаційна технологія інтелектуального аналізу хімічного складу сталі для підтримки прийняття рішень оператора спектральної лабораторії ливарного виробництва

Здобувача (ки) групи ІТ.м-21н Бельдієва Андрія Сергійовича  
(шифр групи) (прізвище, ім'я, по батькові)

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

\_\_\_\_\_ (підпис)

\_\_\_\_\_ (Ім'я та ПРІЗВИЩЕ здобувача)

Керівник \_\_\_\_\_ д.т.н., проф, Євгеній ЛАВРОВ \_\_\_\_\_  
(посада, науковий ступінь, вчене звання, Ім'я та ПРІЗВИЩЕ) (підпис)

Суми – 2024

Сумський державний університет

Факультет електроніки та інформаційних технологій

Кафедра інформаційних технологій

Спеціальність 122 «Комп'ютерні науки»

Освітньо-наукова програма «Інформаційні технології проектування»

**ЗАТВЕРДЖУЮ**

**В.о. завідувача кафедри ІТ**

Світлана ВАЩЕНКО

«\_\_\_» \_\_\_\_\_ 2024 р.

## **ЗАВДАННЯ**

**на кваліфікаційну роботу магістра студентіві**

Бельдієву Андрію Сергійовичу

(прізвище, ім'я, по батькові)

**1 Тема кваліфікаційної роботи** Інформаційна технологія інтелектуального аналізу хімічного складу сталі для підтримки прийняття рішень оператора спектральної лабораторії ливарного виробництва

затверджена наказом по університету від «01» лютого 2024 р. № 0096-VI

**2 Термін здачі студентом кваліфікаційної роботи** «10» травня 2024 р.

**3 Вхідні дані до кваліфікаційної роботи** Інформація про діяльність працівників спектральної лабораторії ДП «Завод ОБ та ВТ), літературні джерела з вирішення проблем класифікації даних в різних предметних областях, математичні моделі класифікаторів.

**4 Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити)** Аналіз предметної області, Постановка задачі та методи дослідження, Моделювання та проектування інформаційної технології, Розробка інформаційної технології.

**5 Перелік графічного матеріалу (з точним зазначенням обов'язкових слайдів презентації)** актуальність проекту, постановка задачі, огляд аналогів, вимоги до інформаційної технології, загальна схема інформаційної технології, моделювання інформаційної технології, компоненти програми, діаграма варіантів використання, результати дослідження, робота модулів, засоби розробки та тестування, висновки

**6. Консультанти випускної роботи із зазначенням розділів, що їх стосуються:**

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

Дата видачі завдання \_\_\_\_\_.

Керівник \_\_\_\_\_

(підпис)

Завдання прийняв до виконання \_\_\_\_\_

(підпис)

**КАЛЕНДАРНИЙ ПЛАН**

№ п/п	Назва етапів кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1	Аналіз предметної області, визначення мети та задач дослідження	До 20.02.2024 р.	
2	Проектування інформаційної технології	До 27.03.2024 р.	
3	Розробка методів дослідження	До 12.04.2024 р.	
4	Розробка інформаційної технології	До 24.05.2024 р.	
5	Оформлення пояснювальної записки	До 10.05.2024 р.	
6	Створення презентації	До 20.05.2024 р.	
7	Захист дипломного проекту	22.05.2024 р.	

Магістрант \_\_\_\_\_

Андрій БЕЛЬДІСВ

Керівник роботи \_\_\_\_\_

д.т.н., проф. Євгеній ЛАВРОВ

## АНОТАЦІЯ

Тема роботи «Інформаційна технологія інтелектуального аналізу хімічного складу сталі для підтримки прийняття рішень оператора спектральної лабораторії ливарного виробництва».

Метою роботи є розробка інформаційної технології інтелектуального аналізу хімічного складу сталі для підтримки прийняття рішень оператора спектральної лабораторії ливарного виробництва.

Під час роботи було досліджено предметну область, актуальність та її проблеми, визначено мету та задачі дослідження, проведено аналіз існуючих моделей класифікаторів та підходів до вирішення задач класифікації у схожих предметних областях. Виявлена необхідність у створенні інформаційної технології. Виконано аналіз і проектування інформаційної технології, вибір засобів реалізації, розробка.

Результатом роботи є інтегрована система інтелектуального аналізу хімічного складу сталі для підтримки прийняття рішень оператора спектральної лабораторії ливарного виробництва, що працює на базі СППР «Flare».

Результати доповідались на конференції ІМА 2024. Опубліковано тези доповіді (Додаток Б).

Пояснювальна записка містить 59 сторінок, повний обсяг 124 сторінки, 29 рисунків, 70 літературних джерел, 2 таблиці, 4 додатки.

Ключові слова: Інформаційна технологія, інтелектуальний аналіз, нейронна мережа, автоматизація, моделювання, дані, Java, оператор, лабораторія, підтримка діяльності, складна система, ливарне виробництво, спектральний аналіз.

## ЗМІСТ

ВСТУП	6
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ «КЛАСИФІКАЦІЯ ЗРАЗКІВ В СПЕКТРАЛЬНІЙ ЛАБОРАТОРІЇ ЛИВАРНОГО ВИРОБНИЦТВА»	8
1.1 Аналіз проблем діяльності оператора спектрального аналізу ДП «Завод ОБ та ВТ»	8
1.2 Аналіз підходів до класифікації	10
1.3 Висновки на предмет подальших досліджень	18
2 ПОСТАНОВКА ЗАДАЧІ ТА МЕТОДИ ДОСЛІДЖЕННЯ	20
2.1 Постановка задачі дослідження	20
2.2 Опис функціональних вимог до інформаційної технології	21
3 МОДЕЛЮВАННЯ ТА ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ	22
3.1 Моделювання інформаційної технології	22
3.2 Проектування інформаційної технології	26
4 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ	35
4.1 Розробка інформаційної технології	35
4.2 Аналіз результатів	40
ВИСНОВКИ	46
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	47
ДОДАТОК А. Повний список робіт проекту	56
ДОДАТОК Б. Копія публікації	64
ДОДАТОК В. Копія акту впровадження	65
ДОДАТОК Г. Лістинг програми	66

## ВСТУП

**Актуальність.** Якість ливарної продукції на металургійному підприємстві сильно залежить від того, наскільки хімічний склад співпадає з визначеними відхиленнями марки сталі, що затверджені державними стандартами. Для забезпечення якості продукції потрібна точна класифікація виробленої марки сталі. Нажаль, існуючі алгоритми трудозатратні, бо потребують налаштування з боку оператора лабораторії. Тому постає питання розробки інтелектуальної технології, розробленої на базі нейромережових схем. Замовником роботи є центральна заводська лабораторія дочірнього підприємства «Завод обважених бурильних та ведучих труб» міста Суми (далі ДП «Завод ОБ та ВТ»).

**Об'єкт дослідження.** Підтримка прийняття рішень оператора спектральної лабораторії ливарного виробництва.

**Предмет дослідження.** Інтелектуальний аналіз хімічного складу сталі.

**Суперечність.** Потрібно розробити інформаційну технологію, яка дозволить підвищити якість ливарної продукції, інтелектуально визначаючи марку сталі з прийнятною точністю, але наразі відсутні готові системи, які могли б вирішити дану задачу.

**Мета дослідження.** Розробити інформаційну технологію інтелектуального аналізу хімічного складу сталі для підтримки прийняття рішень оператора спектральної лабораторії ливарного виробництва.

**Задачі.** Для досягнення мети проекту необхідно виконати наступні задачі:

- 1) Дослідити діяльність оператора спектральної лабораторії ДП «Завод ОБ та ВТ»;
- 2) Поставити задачу класифікації зразків ливарної продукції;
- 3) Розробити метод класифікації злитків на технології машинного навчання;
- 4) Розробити вимоги і структуру системи до інформаційної технології.
- 5) Розробити програмну реалізацію інформаційної технології.

**Гіпотеза дослідження.** Якщо використовувати нейромережеві моделі аналізу складу зразків, які будуть натреновані на накопичених в сховищі даних лабораторії зразках, то задачу визначення марки сталі зразка, що надходить в лабораторію, можна вирішити з достатньою для практики точністю.

**Новизна результатів.** На відміну від існуючих наукових підходів, що використовують алгоритмічний підхід до аналізу марок сталі хімічного складу зразків, запропонована модель основана на моделях інтелектуального аналізу накопичених в сховищах даних лабораторії про визначення марок сталі зразків.

**Апробація.** Матеріали дослідження доповідались на міжнародній конференції ІМА 2024 (Суми, Астана)

**Публікації.** За матеріалами дослідження опубліковано тези доповіді (копія публікації наведена в додатку Б).

**Впровадження.** Результати впровадження у виробничий процес спектральної лабораторії ДП «ОБ та ВТ» (м. Суми), про що є відповідний акт (Додаток В).

## **1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ «КЛАСИФІКАЦІЯ ЗРАЗКІВ В СПЕКТРАЛЬНІЙ ЛАБОРАТОРІЇ ЛИВАРНОГО ВИРОБНИЦТВА»**

### **1.1 Аналіз проблем діяльності оператора спектрального аналізу ДП «Завод ОБ та ВТ»**

Перед тим, як отримати ливарну продукцію, на виробництві відбувається багато етапів: від доставки сировини до формування труб із заготовок. Але в даній науковій роботі буде розглянуто робочий процес саме інженера спектрального аналізу центральної заводської лабораторії ДП «Завод обважених бурильних та ведучих труб».

Шлях зразка починається зі сталеливарного цеху, де він отримується з ковша з рідкою сталлю. Далі він передається до центральної заводської лабораторії пневматичною поштою – у відділ механічної обробки. Наступним кроком відбувається його охолодження та обробка пласкої частини на шліфувальній машині. Поверхня повинна бути якомога більш рівною для наступного етапу. Зразок переноситься в спектральну лабораторію, де розміщується та закріплюється на спектрометрі (рис. 1.1).





### Рисунок 1.1 – Робоча частина спектрометра із закріпленим зразком

Через комп'ютер на спектрометр дається вказівка почати пропалювання зразка, під час якого визначаються концентрації кожного хімічного елемента в зразку. Результат вимірювання відображається на екрані оператора у вигляді відсоткових значень. Саме ці значення визначають, до якої марки сталі належить зразок. Оператору необхідно вручну звіряти концентрацію кожного елемента з журналом марок сталі, щоб знайти в списку потрібну та порівняти допустимі відхилення за ДСТУ.

Процес класифікації зразка є важливою частиною, помилки в якій затримують виробничий процес та можуть бути причиною бракованої продукції.

Існуюча система підтримки прийняття рішень «Flare» має інструменти для співвіднесення маркувальних та розривних зразків, що пришвидшило роботу оператора [1]. В цій програмі були використані алгоритмічні методи пошуку зразків з необхідністю налаштувань відхилень експертом в залежності від специфіки сталей. Даний спосіб хоч і забезпечує необхідну точність знаходження зразків та полегшує процес, але потребує участі оператора у процесі пошуку, що може бути складним процесом для молодосвідченого спеціаліста. При цьому програмне забезпечення не призначене для класифікації маркувального зразка, а допомагає лише знайти відповідний розривний зразок.

Керівництво підприємства шукає способи вдосконалення виробничого процесу на всіх етапах виробництва. Впровадження автоматизації визначення марки сталі дасть пришвидшення процесу експрес аналізу в цілому та дозволить відловлювати можливі помилки попередньо класифікованих маркувальних зразків.

## 1.2 Аналіз підходів до класифікації

### 1.1.1 Аналіз існуючих систем підтримки прийняття рішень по класифікації зразків

Наразі, можна виділити лише дві актуальні системи підтримки прийняття рішень.

Перша СППР – це «FP-LIMS» розроблена німецькою компанією «Fink and Partners». Попередній продукт цієї компанії називався «DIA2000», який йшов у комплекті поставки зі спектрометрами серії «SpectroLAB» або «SpectroMAXxx» [2]. Дана програма підтримує сучасні операційні системи сімейства Windows, може працювати зі спектрометрами минулого покоління та має модульний підхід до збірки функціоналу, необхідного для користувача.

Функціонал забезпечує візуалізацію даних шляхом графічного представлення статистики зразків (рис.1.2-1.3) та підтримує обмін даними по локальній мережі виробництва [4].

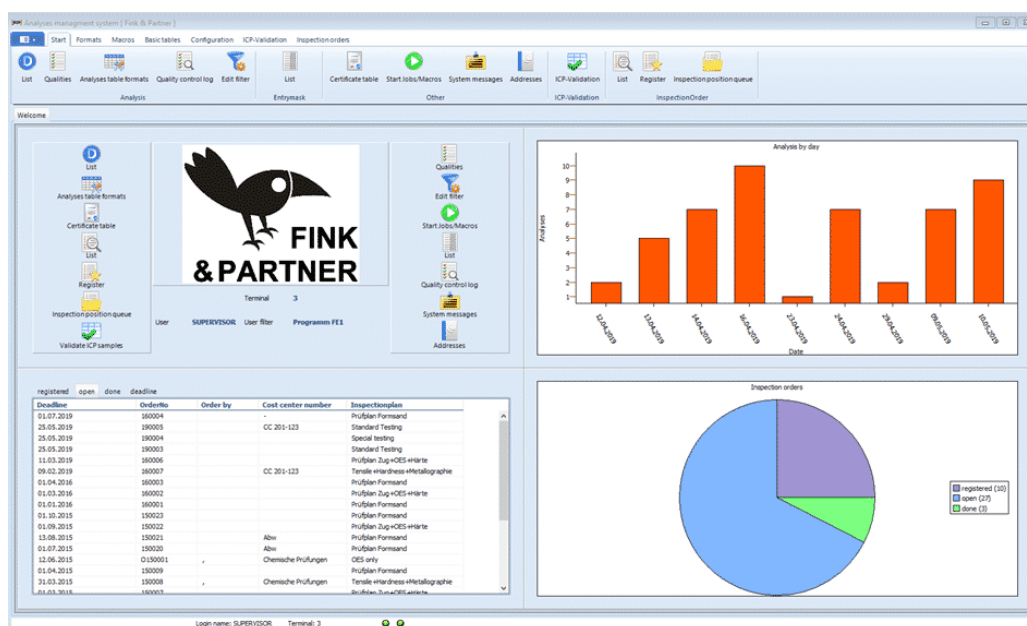


Рисунок 1.2 – Графічне представлення даних FP-LIMS[30]

Datum	Zeit	Typ	Gerät	Bemerkung	Qualitätscharge-Nr.	Probe Nr.	Qu	Kohlen	Si	Stilzum	Mangan	Schwefel	Magnes	Nickel	Molybd	Kupfer	Zinn	De
01.10.2019	13:16:00	Einzel	O1		Q205	2222 180002	1	1.50	2.4556000	0.0250	0	0.0500	0.160	0.00100	0.0250	0.09500		
01.10.2019	12:44:44	Einzel	O1		Q205	2222 180003	1	1.50	2.4402000	0.0250	0.00100	0.0500	0.160	0.00100	0.0250	0.09500		
01.10.2019	12:31:44	Einzel	O1		Q205	2222 180004	1	1.50	2.4670000	0.0300	0.00100	0.0500	0.160	0.00100	0.0250	0.09500		
01.10.2019	12:26:44	Einzel	O1		Q205	2222 180005	1	1.50	2.4600000	0.0250	0.00100	0.0500	0.160	0.00100	0.0250	0.09500		
01.10.2019	12:25:44	Einzel	O1		Q205	2222 180004	1	1.50	2.5000000	0.0500	0.00110	0.0500	0.160	0.00100	0.0250	0.09500		
01.10.2019	12:22:44	Einzel	O1		Q205	2222 180057	1	1.66	1.3950000	0.0230	0.00100	0.0500	0.160	0.00100	0.0250	0.09500		
01.10.2019	12:19:44	Einzel	O1	Kommentare - bis zu 256 Z	Q205	2222 180012	1	1.50	2.2700000	0.0270	0.00100	0.0500	0.160	0.00100	0.0250	0.09500		
01.10.2019	12:11:44	Einzel	O1		Q205	2222 180019	1	1.50	2.4600000	0.00024	0.00100	0.0500	0.160	0.00100	0.0250	0.09500		
01.10.2019	12:07:44	Einzel	O1		Q205	2222 180017	1	1.50	2.3460000	0.0200	0.00100	0.0500	0.160	0.00100	0.0250	0.09500		
01.10.2019	12:00:00	Einzel	O1	Ble berücksichtigt	Q205	2222 180010	1	1.50	2.2030000	0.0250	0.00100	0.0500	0.160	0.00100	0.0250	0.09500		

Рисунок 1.3 – Вікно показу зразків в FP-LIMS

Програмне забезпечення поширюється за платною ліцензією, починаючи від 3000 до 12800 доларів.[3]

Даний продукт має потужні аналітичні інструменти, можливості порядкування зразків та візуальне представлення даних про зразки. Комплекс цих можливостей може допомогти оператору класифікувати зразок до певної групи інших зразків, але це потребуватиме ручного пошуку та вибору правильних фільтрів.

Другим варіантом слугує СППР «Flare», розроблена у 2022 році [1]. Спроектвана спеціально для підприємства «ОБ та ВТ», вона спрямована на класифікацію розривного зразка шляхом знаходження відповідного маркувального, який і визначить марку сталі. Вважається, якщо маркувальний зразок відноситься до однієї і тієї ж плавки, що й розривний (мають однаковий хімічний склад), то вони мають однакову марку сталі. Передбачено ручний режим пошуку за допомогою покрового збільшення інтервалу пошуку.

Програма має мінімалістичний, але зручний інтерфейс, який дозволяє швидко опанувати основні можливості, серед яких:

1. Класифікація за номером блюму та відповідним хімічним складом;
2. Ручна класифікація методом аналізу інтервалів відхилень;

3. Ручний відбір за фільтрами;
4. Можливість додавання в закладки для подальшого порівняння;
5. Налаштування параметрів пошуку відповідного маркувального зразка.

Недоліком поточної версії програми є неможливість визначити марку сталі маркувального зразка, лише розривного. Також, необхідне налаштування параметрів алгоритму, що може бути складним для оператора з малим досвідом.

### 1.1.2 Аналіз підходів до вирішення задач класифікації

Підходи вирішення задач ергономіки є у ряді досліджень проблем автоматизованого управління. [5-14]

Умовно процес інтелектуального аналізу даних можна зобразити наступним чином (рис. 1.4):

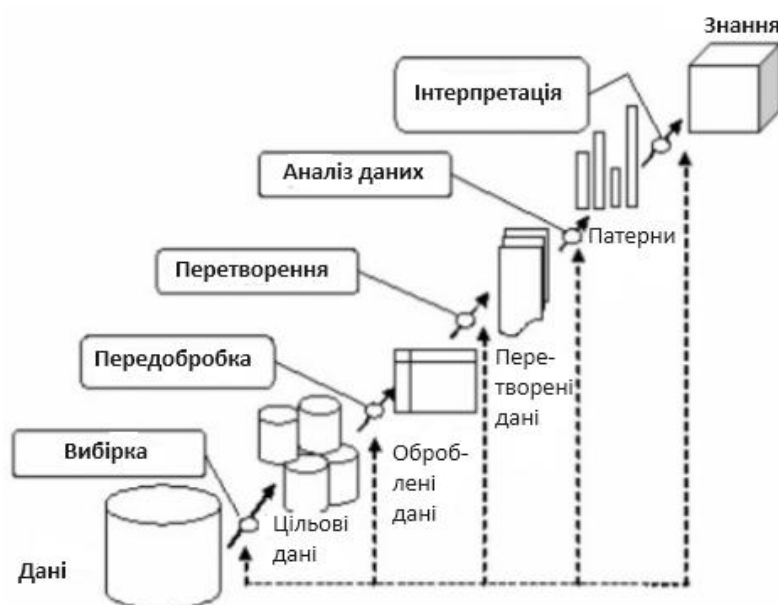


Рисунок 1.4 – Типова схема процесу аналізу даних для систем підтримки прийняття рішень.

Відповідно до принципів вирішення проблем підтримки прийняття рішень побудовано системи забезпечення ергономічної якості автоматизованих систем різного типу [15-36]:

- Системи обробки інформації, контакт-центри, системи забезпечення якості програмного забезпечення;
- Гнучкі виробничі системи;
- Системи хімічних виробництв;
- Системи управління судноплавством, авіацією, магістральними газопроводами;
- Системи електронного навчання;
- Системи спектрального аналізу продукції агропромисловості.

Актуальність проблеми підтримки прийняття рішень у сфері металургійного виробництва також приступає [37]. Проблеми, які було розглянуто у цій сфері виробництва представлено невеликою кількістю робіт, серед яких аналіз мікроструктури металу засобами комп'ютерного зору [38-39, 41] та керування процесами в печі під час плавки металу [40].

Є дослідження, в яких вирішувалась схожа задача класифікації зразків за результатом лазерного (а не іскрового) спектрального аналізу. Концентрації елементів були вхідними даними для нейронної мережі, а вихідними даними є чотири класи: вуглеродні, легкоплавкі, тугоплавкі або нержавіючі сталі [41]. Або більша кількість класів, але які характеризують тип сплаву, а не марку сталі [42-44].

Розглянемо класифікатори, які частіше зустрічалися в дослідженнях.

Задача класифікації полягає в призначенні об'єктам (даним) однієї або декількох категорій або класів на основі їхніх характеристик. Це означає, що ми намагаємося розділити набір даних на певні групи заздалегідь відомих класів або категорій, щоб можна було ефективно класифікувати нові об'єкти, які потрапляють у набір даних. Наприклад, класифікація електронного листа як "спам" або "не спам", класифікація зображення як "собака", "кіт" або "автомобіль" - це приклади задач класифікації.

Можна виділити два типи алгоритмів:

- Класичні алгоритми класифікації - це моделі, які використовуються для розділення даних на категорії або класи на основі певних характеристик. Вони

включають такі методи, як лінійна регресія, наївний баєсівський класифікатор, логістична регресія та метод опорних векторів.

– Ансамблеві алгоритми класифікації - це методи, які поєднують декілька базових моделей для отримання більш точного та стабільного прогнозу. Вони використовуються для підвищення точності класифікації та зменшення перенавчання. Деякі з найвідоміших ансамблевих методів включають в себе випадковий ліс, градієнтний бустінг та стекінг. Дерево прийняття рішень - непараметричний метод, який використовують у машинному навчанні, аналізі даних і статистиці.

Дерево прийняття рішень - це дерево, у листках якого записано номери класів, а в інших вузлах - ознаки, за якими розрізняють випадки. Щоб класифікувати новий випадок, треба спуститися по дереву до листка і видати відповідне значення [46]. Дерево прийняття рішень рекурсивно розбиває простір об'єктів навчальної вибірки таким чином, щоб об'єкти з однаковими мітками класу групувалися в одному аркуші. Приклад дерева зображено нижче (рис. 1.5).

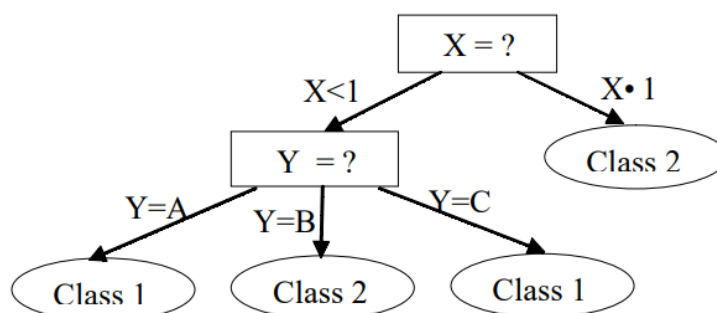


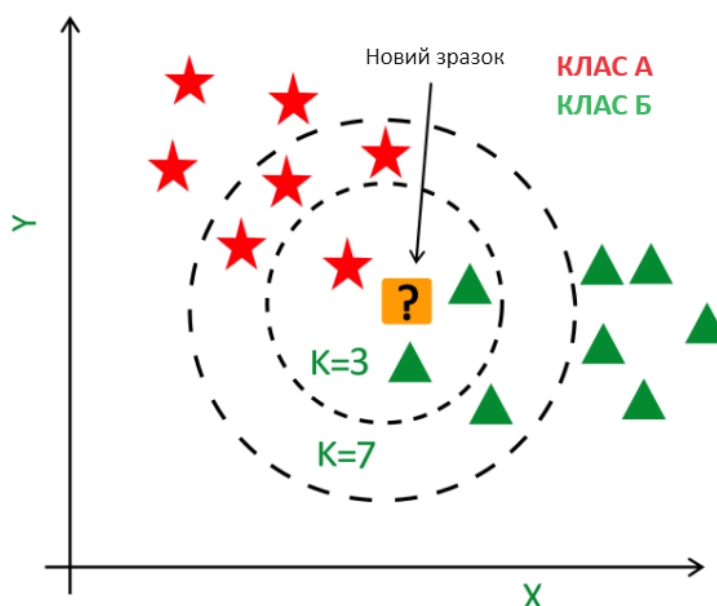
Рисунок 1.5 – Приклад дерева рішень для параметрів X та Y

Даний метод має наступні переваги та недоліки (табл. 1.1):

Таблиця 1.1 – Переваги та недоліки дерев рішень

Переваги	Недоліки
Не треба нормалізувати дані [47]	Можливе перенавчання з надлишковою глибиною дерева
Незалежність від типу змінних	Складність вираження деяких логічних операцій
Робота з великими масивами даних	Прив'язка до структури вхідних даних
Висока точність та швидкість отримання результатів[48,49]	

Метод  $k$  – найближчих сусідів [50] вважається одним з найстаріших непараметричних алгоритмів класифікації. Для класифікації невідомого зразка вимірюється відстань (з використанням деякої міри відстані, наприклад, евклідової) від цього зразка до кожного іншого навчального зразка. Визначається  $k$  найменших відстаней, і найбільш представлений клас серед цих  $k$  класів вважається міткою вихідного класу (рис. 1.6). Значення  $k$  зазвичай визначається за допомогою валідаційної множини або перехресної перевірки. Такий підхід не дає високу точність сам по собі, але може застосовуватися в поєднанні з кластеризацією, як у [51] для поліпшення результатів виділення з трафіку окремих невідомих класів.

Рисунок 1.6 – Визначення класу для зразка методом  $k$  – найближчих сусідів

Метод опорних векторів є одним з найбільш надійних і успішних алгоритмів класифікації [52, 53]. Вони базуються на ідеї максимізації мінімальної відстані від розділювальної гіперплощини до найближчого зразка (рис. 1.7). Початковий метод підтримує лише бінарну класифікацію, але були запропоновані розширення [54-57] для обробки багатокласової класифікації. У цих розширеннях до оптимізаційної задачі додаються додаткові параметри та обмеження, щоб впоратися з розділенням різних класів. Формулювання [52, 58] може призвести до великої оптимізаційної задачі, яка може бути непрактичною для великої кількості класів. З іншого боку, в [56] представлено краще формулювання з більш ефективною реалізацією.

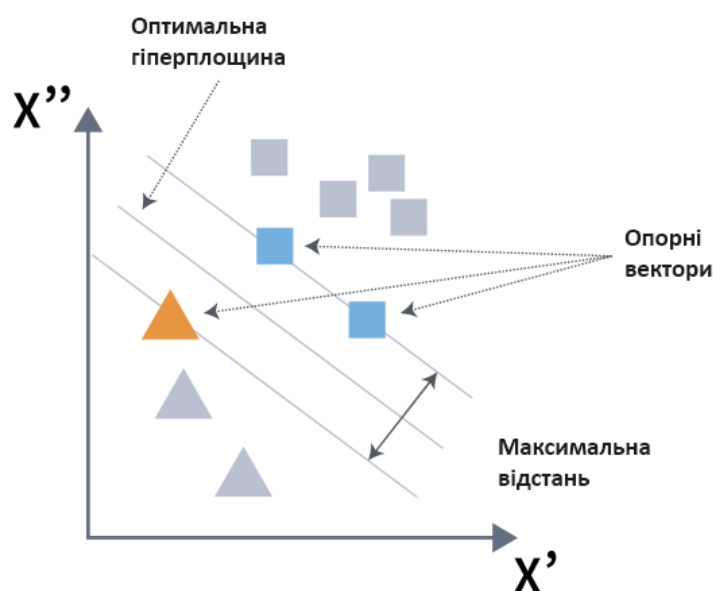


Рисунок 1.7 – Визначення гіперплощини для поділу класів

Випадковий ліс - це спосіб побудови ансамблю з  $m$  дерев прийняття рішень, навчених на різних частинах навчальної вибірки, з метою подолання проблеми перенавчання окремого дерева рішень. Алгоритм побудови:

1. Випадковим чином відбирається 63% об'єктів навчальної вибірки з можливими повторами даних.

2. Будується дерево рішень на відібраній вибірці. При розбитті дерева використовуються  $\sqrt{n}$  випадкових ознак об'єкта з початкового набору з  $n$  ознак.



3. Для кожного дерева на 37% даних, що залишилися, підраховується точність класифікації.

Кроки повторюються  $m$  разів. Після завершення побудови випадковий ліс класифікуватиме об'єкти шляхом вибору того класу, який найчастіше зустрічається серед відповідей дерев [60].

Бустінг (boosting) - спосіб організації класифікаторів, що дає змогу впорядкувати й навчити простіші класифікатори (найчастіше це дерева невеликої глибини, що розв'язують) так, щоб результувальний класифікатор показував найкращі результати. Прикладами алгоритмів бустингу є:

- AdaBoost (Adaptive Boosting) [61, 62] - регулює ваги під час навчання, щоб прикладам, на яких помилився минулий класифікатор, надавали більшого значення;

- XGBoost [63] - помилка мінімізується алгоритмом градієнтного спуску.

За якістю і швидкістю роботи бустинг приблизно можна порівняти з бегінгом (залежить від конкретних даних; один із прикладів такого порівняння можна знайти в [61,64]).

Термін «нейронна мережа» означає схему, що складається з надзвичайно великої кількості оброблювальних елементів, які називаються нейронами [65]. Кожен елемент працює лише з локальною інформацією. Крім того, кожен елемент працює непаралельно, тобто системний годинник відсутній [66]. Основна ідея нейронної мережі полягає в тому, щоб отримати атрибути з лінійних комбінацій вхідних даних, а потім моделювати вихід як нелінійну функцію цих атрибутів. Результатом є одна з найпопулярніших і ефективних форм навчання системи. Нейронні мережі представлені мережевою діаграмою, яка складається з вузлів, з'єднаних спрямованими зв'язками. Вузли вирівнюються по шарах і структура найбільш використовуваної нейронної мережі включає 3 шари: вхідний, прихований і вихідний шар вузлів (рис. 1.8).

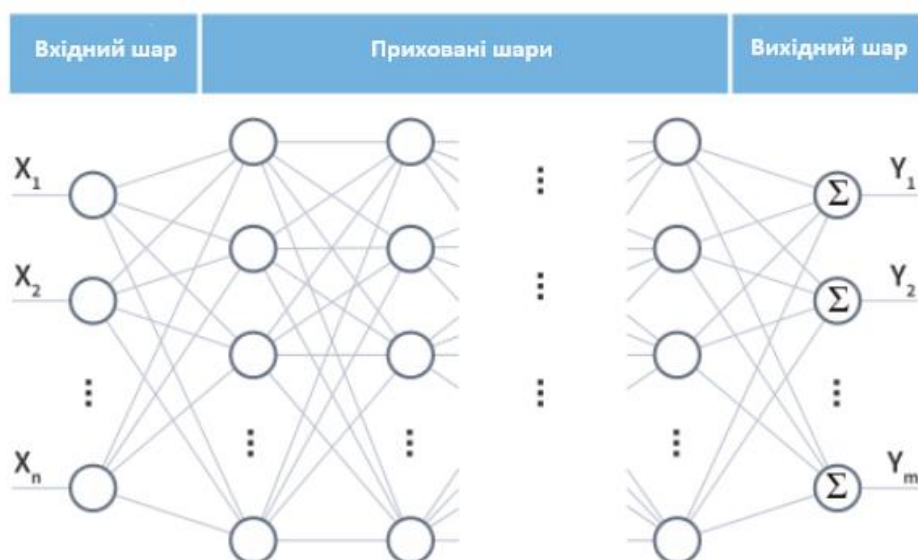


Рисунок 1.8 – Схема багатошарового перцептрону

### 1.3 Висновки по розділу 1

Нажаль, розробки, які використовують методи машинного навчання для класифікації сталі за хімічним складом, не передбачають розпізнавання марок сталі, і призначені до роботи з лазерними спектрометрами. Отже проблема підтримки рішень оператора спектральної лабораторії досі актуальна.

Беручи до уваги попередні дослідження щодо використання класифікаторів доцільним буде обрати декілька видів моделей та навчити їх на однаковому датасеті для вибору найбільш сумісної, яка буде включена в розробку. Планується комбінувати алгоритми СППР «Flare» з розробленою моделлю для забезпечення належної для виробництва точності.

## 2 ПОСТАНОВКА ЗАДАЧІ ТА МЕТОДИ ДОСЛІДЖЕННЯ

### 2.1 Постановка задачі дослідження

На жаль, оператор працює в умовах операційно темпової напруженості, коли виникає черга сигналів, черга запитів на обробку, що викликає емоційне напруження. Для створення оптимальних умов на робочому місці необхідно розробити систему підтримки прийняття рішень, яка унеможливить помилки та створить психологічний комфорт [5-14].

Проаналізовані системи підтримки прийняття рішень, які використовуються в спектральних лабораторіях, не дозволяють виконувати автоматичну класифікацію зразків за хімічним складом, бо мають обмежений функціонал, який залежить від навичок оператора та функціональних можливостей.

Виходячи з цього, підприємством поставлено задачу розробки інтелектуальної системи для операторів спектральної лабораторії.

Для умов конкретного ливарного виробництва визначимо:  $N$  – число діагностичних ознак;  $s_i$  - кількість упорядкованих оцінок складу сталі на шкалі  $i$ -ї діагностичної ознаки;  $X_i = \{X_{1i}, X_{2i}, \dots, X_{s_i}\}$  - безліч оцінок на шкалі  $i$ -ї ознаки;  $Q$  - кількість класів марок сталі ( $P_1, P_2, \dots, P_Q$ ), до яких можуть належати зразки. Задача полягає в розробці технології, орієнтованої на інтелектуальний аналіз даних про зразки ливарної продукції, накопичених в базі даних лабораторії.

Таким чином, метою дослідження є розробити інформаційну технологію інтелектуального аналізу хімічного складу сталі для підтримки прийняття рішень оператора спектральної лабораторії ливарного виробництва.

Можна виділити наступний план дослідження:

- 6) Дослідити діяльність оператора спектральної лабораторії ДП «Завод ОБ та ВТ»;
- 7) Поставити задачу класифікації зразків ливарної продукції;
- 8) Розробити метод класифікації злитків на технології машинного навчання;

9) Розробити вимоги і структуру системи до інформаційної технології.

## **2.2 Опис функціональних вимог до інформаційної технології**

Оператор, отримуючи результат спектрального аналізу, має можливість перенести дані про зразок в сховище програми. Далі він може використовувати як ручний спосіб визначення марки сталі, який використовується в програмному забезпеченні «Flare», так і класифікацію методами машинного навчання. Це дозволить оператору обирати зручний для себе спосіб в конкретному випадку.

Основні вимоги до інформаційної технології наступні:

1. Програмний модуль повинен бути у вигляді інтегрованої системи в програмному забезпеченні «Flare»;
2. Можливість імпортувати дані з вікна вимірювань програмного забезпечення «Spark Analyzer» (постачається зі спектрометром)
3. Точність класифікації інтелектуальною системою для марок сталі, які часто використовуються на підприємстві (30 плавок в місяць) становить 95% і вище.

## 3 МОДЕЛЮВАННЯ ТА ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ

### 3.1 Моделювання інформаційної технології

#### 3.1.1 Імпорт даних

Концентрації хімічних елементів, які отримані зі спектрометра, відображаються у вікні вимірювань програми «Spark Analyzer» у наступному вигляді (рис. 3.1):

Sample No:	Quality:						
Cod_1:	Cod_2:	Cod_3:	Bloom:				
Kristal:	Serial:	Product:	Cod_4:				
	C %	Si %	Mn %	P %	S %	Cr %	Ni %
1	0.0639	0.768	1.949	0.0264	0.0377	0.2157	0.1236
2	0.0887	0.660	1.572	0.0197	0.0303	0.1112	0.0718
3	0.0790	0.639	1.681	0.0342	0.0363	0.1266	0.1009
	Mo %	Cu %	Al %	Ca %	V %	Ti %	Nb %
1	0.0057	0.1145	0.0244	0.0009	0.0020	0.0124	<0.0020
2	<0.0015	0.1503	0.0131	0.0004	0.0014	0.0137	<0.0020
3	<0.0015	0.1398	0.0108	0.0008	0.0025	0.0065	0.0021
	W %	Pb %	Sn %	As %	Zr %	Co %	Sb %
1	<0.0050	<0.0001	0.0059	0.0054	<0.0010	0.0036	<0.0010
2	0.0058	<0.0001	0.0085	0.0071	0.0037	0.0059	<0.0010
3	<0.0050	<0.0001	0.0074	0.0069	0.0030	0.0072	<0.0010
	Se %	Te %	Ta %	B %	N %	Fe %	CEV %
1	<0.0010	<0.0010	0.0963	<0.0001	0.0097	96.5	0.4493
2	<0.0010	0.0020	0.1060	<0.0001	0.0090	97.1	0.3880
3	<0.0010	0.0019	0.0807	<0.0001	0.0090	97.0	0.4010

Рисунок 3.1 – Лист Microsoft Excel з даними маркувальних зразків

З цього вікна, шляхом імпорту даних за допомогою комп'ютерного зору будуть отримані лише концентрації хім. елементів. Дата та час будуть проставлені автоматично в момент виміру, а визначити марку сталі класифікатором оператор зможе натисканням відповідної клавіші.

Перед скануванням знімку екрана алгоритмами розпізнавання тексту виконується попередня обробка зображення наступним чином:

1. Зображення переводиться у відтінки сірого;
2. Виконується розмиття за Гауссом на 5 пікселів;
3. Колір зображення переводиться в бінарний. Де всі значення сірого нижче 250 будуть переведені в чорний (0), а вище – у білий (255);
4. Виконується перекриття всіх чорних елементів розміром менше 15 на 15 пікселів білим кольором. Таким чином на зображенні зникнуть всі цифри;
5. Готова маска поєднується з зображенням у відтінках сірого для перекриття всіх текстових елементів, окрім цифр концентрацій.

Нехай один піксель зображення буде представлений трьома параметрами: червоним, зеленим та синім кольорами. А його розміщення буде представлено номером рядка  $i$  стовпця, отже матимемо:  $p_{i,j}(r), p_{i,j}(g), p_{i,j}(b)$ . Де  $p$  – піксель, параметри  $r, g, b$  – інтенсивність червоного, зеленого та синього кольорів відповідно. Індeksi  $i, j$  – рядок та стовпець пікселя. Якщо  $p$  – піксель вихідного зображення, то пікселями зображення у відтінках сірого та зображення маски будуть  $p_1$  і  $p_2$  відповідно. Маємо наступну формулу (3.1):

$$\begin{cases} p_{i,j}(r) = p_{1,i,j}(r) * p_{2,i,j}(r) \\ p_{i,j}(g) = p_{1,i,j}(g) * p_{2,i,j}(g) \\ p_{i,j}(b) = p_{1,i,j}(b) * p_{2,i,j}(b) \end{cases} \quad (3.1)$$

Підготовлене зображення можна таким чином конвертувати в текст алгоритмами розпізнавання тексту. Розберемо алгоритм на прикладі TesseractOCR. Принцип роботи Tesseract-OCR детально розбираються у відповідній роботі [67].

Коротко процес можна описати так: програма визначає макет сторінки (layout) і групує обриси тексту в об'єкти (blobs). Далі в об'єктах виокремлюють рядки, а

рядки, своєю чергою, розбиваються на слова залежно від довжини проміжків між ними. Це робиться за допомогою як звичайних, так і рекурентних мереж.

Рекурентні нейронні мережі, підвидом яких є LSTM (long short-term memory), - тип нейронних мереж, що задіюють зворотний зв'язок (рис. 3.2). На відміну від нейромереж прямого поширення, у яких інформація передається послідовно від шару до шару, у рекурентних нейронних мережах кожен шар нейронів, окрім вхідних даних, одержує деяку інформацію про попередній стан мережі, що уможливорює аналіз послідовностей даних [68].

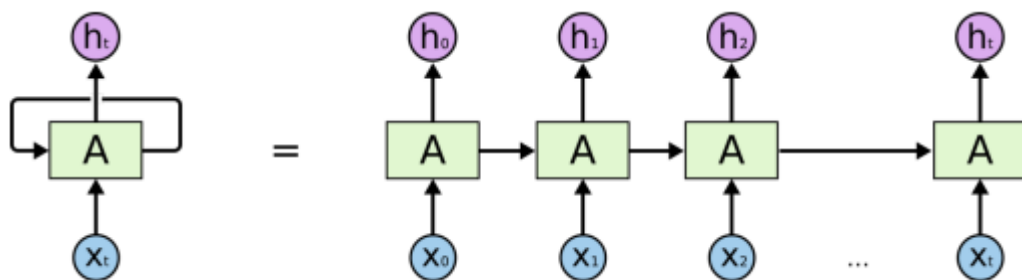


Рисунок 3.2 – Загальний принцип роботи шару нейронів LSTM мережі

Після цього аналізується, який тип шрифту було використано для цієї чи іншої області тексту: моноширинний або пропорційний (fixed pitch text, proportional text). У першому випадку кожен символ має фіксовану ширину, а отже, Tesseract здатний одразу розбити рядок на символи. У другому випадку програма виділяє визначені та невизначені проміжки (definite spaces, fuzzy spaces) між символами: рішення щодо невизначених проміжків ухвалюється в самому кінці.

Розпізнавання проходить у два етапи. Спочатку алгоритм намагається розпізнати кожне слово по черзі і передає кожну успішну спробу класифікатору як навчальну вибірку. Класифікатор використовує ці дані під час подальшого просування сторінкою і на другому етапі розпізнавання, коли алгоритм повторно працює з тими словами, які не вийшло розпізнати першого разу.

Наприкінці Tesseract ухвалює рішення щодо невизначених проміжків і перевіряє, чи не є деякі нерозпізнані малі літери малими великими (small-cap letters).

Таким чином отриманий текст являтиме собою концентрації елементів.

### 3.1.1 Класифікація зразків

Передбачається, що навчальна вибірка буде доповнюватись по мірі того, як оператори спектрального аналізу будуть додавати нові зразки. Це дозволить перенавчати класифікатори для більш точного визначення існуючих марок сталі, які мали мало зразків для точного навчання. Також можна розширити кількість марок сталі, які ще не були виплавлені на підприємстві. Після навчання класифікатора на поточній вибірці, оператор може відібрати той, який на даний момент дає найбільшу точність.

В рамках дослідження буде використано багатошаровий персептрон зворотного поширення помилки з функцією активації типу сигмоїди. Алгоритм навчання зворотнього поширення помилки включає два етапи: прямий та зворотній прохід через всі шари нейронної мережі. На першому кроці вхідний вектор подається на вхідний шар, а потім проходить через усі шари, створюючи набір вихідних сигналів, що відповідає реакції мережі на вхідні дані. У цей час синаптичні ваги залишаються незмінними. При зворотньому проході синаптичні ваги адаптуються з урахуванням різниці між фактичним та бажаним виходом мережі, спрямовуючись на максимальне наближення вихідних сигналів мережі до очікуваних значень. Нехай  $X_i$  – вхідний вектор,  $Y_i$  – вихідний вектор,  $w_{i,j}^k$  –  $i$ -ий ваговий коефіцієнт  $j$ -го нейрону  $k$ -го шару,  $b_i^k$  – поріг  $i$ -го нейрону  $k$ -го шару,  $d_i$  – еталонне вихідне значення  $i$ -го нейрону. Вихідне значення  $j$ -го нейрону  $k$ -го шару можна обчислити наступним чином:

$$Y_j^k = F(\sum w_{i,j}^k Y_i^{k-1} b_j^k) \quad (3.2)$$

Вихідне значення  $j$ -го нейрону вихідного шару визначимо формулою:

$$Y_j = F(\sum w_{i,j} Y_i^{n-1} b_j) \quad (3.3)$$

Функція помилки мережі:



$$E = \frac{1}{2} \sum_j (Y_j - d_j)^2, \quad (3.4)$$

Де  $\gamma_j = Y_j - d_j$  – помилка  $j$ -го нейрону вихідного шару. Помилка  $j$ -го елемента  $k$ -го прихованого шару:

$$\gamma_j^k = \frac{\partial E}{\partial Y_j^k} = \sum_j \frac{\partial E}{\partial Y_j} \frac{\partial Y_j}{\partial S_j} \frac{\partial S_j}{\partial Y_j^k} = \sum_j \frac{\partial E}{\partial Y_j} \frac{\partial Y_j}{\partial S_j} w_{i,j}^k = \sum_j (Y_j - d_j) F'(S_j) w_{i,j}^k \quad (3.5)$$

Градiєнти помилок:

$$\frac{\partial E}{\partial w_{i,j}} = \frac{\partial E}{\partial Y_j} \frac{\partial Y_j}{\partial S_j} \frac{\partial S_j}{\partial w_{i,j}} = \gamma_j F'(S_j) Y_j^k \quad (3.6)$$

$$\frac{\partial E}{\partial b_j} = \frac{\partial E}{\partial Y_j} \frac{\partial Y_j}{\partial S_j} \frac{\partial S_j}{\partial b_j} = -\gamma_j F'(S_j) \quad (3.7)$$

$$\frac{\partial E}{\partial w_{i,j}^k} = \frac{\partial E}{\partial Y_j} \frac{\partial Y_j}{\partial S_j} \frac{\partial S_j}{\partial Y_j^{k-1}} \frac{\partial Y_j^{k-1}}{\partial S_j^{k-1}} \frac{\partial S_j^{k-1}}{\partial w_{i,j}^k} = \gamma_j F'(S_j^k) Y_j^k \quad (3.8)$$

Вагові коефіцієнти та зсуви нейронів визначаються таким чином:

$$w_{i,j}^k(t+1) = w_{i,j}^k - \alpha \gamma_j^k F'(S_j^k) Y_j^k \quad (3.9)$$

$$b_j^k(t+1) = b_j^k + \alpha \gamma_j^k F'(S_j^k) \quad (3.10)$$

## 3.2 Проектування інформаційної технології

### 3.2.1 Функціональне моделювання інформаційної технології в IDEF0

IDEF0 - це методологія, яка допомагає описати, як працюють бізнес-процеси. Вона використовує спеціальні графічні символи, щоб показати, які дії потрібно виконати, щоб досягнути певного результату. Ця методологія зосереджується на взаємозв'язках між діями, а не на тому, коли ці дії відбуваються. IDEF0 також дозволяє розглядати організацію як систему з різних модулів.

Функціональне моделювання інформаційної технології в нотації IDEF0 представлено на рисунку 3.3.



Рисунок 3.3 – Контекстна діаграма IDEF0

Вхідними даними будуть інформація про роботу працівників спектральної лабораторії та знання щодо плавки металів. Результатом роботи є інформаційна технологія. Виконавцями процесу виступають:

- Weka. Бібліотека для аналізу даних та машинного навчання;
- OpenCV. Бібліотека, що має потужні можливості по обробці зображення для подальшого зчитування даних
- Рушій Tesseract OCR. Отримує текст з обробленого зображення та представляє у вигляді строки
- Java. Мова програмування та середовище виконання коду. Необхідна для написання програми та запуску;
- ПЗ «Flare». Система інтегрується в існуючу інформаційну технологію, використовуючи її методи зберігання та обробки даних.

Керівники:

- Технічне завдання. Опис вимог замовника, які повинно бути реалізовано в ПЗ.
- Документація по роботі з API. Супровідна документація до бібліотек OpenCV, Weka та Tesseract OCR.

### 3.2.2 Декомпозиція інформаційної технології за допомогою IDEF0

Декомпозиція в методології IDEF0 - це процес розбиття складного бізнес-процесу на менші та керованіші елементи. Це може включати розбиття процесу на підпроцеси або на більш дрібні функціональні блоки. Декомпозиція дозволяє краще зрозуміти та керувати бізнес-процесом, робить його більш простим для аналізу та оптимізації. У результаті декомпозиції кожен елемент може бути краще описаний та розглянутий окремо, що сприяє покращенню роботи всього процесу. Оскільки програмний модуль інтегрується в існуючу інформаційну технологію, то діаграма відображатиме процеси у її контексті. ПЗ «Flare» з попередньої діаграми тут представлено у вигляді розділених вхідних даних.

Декомпозиція функціональної моделі інформаційної технології представлена на рисунку 3.4.

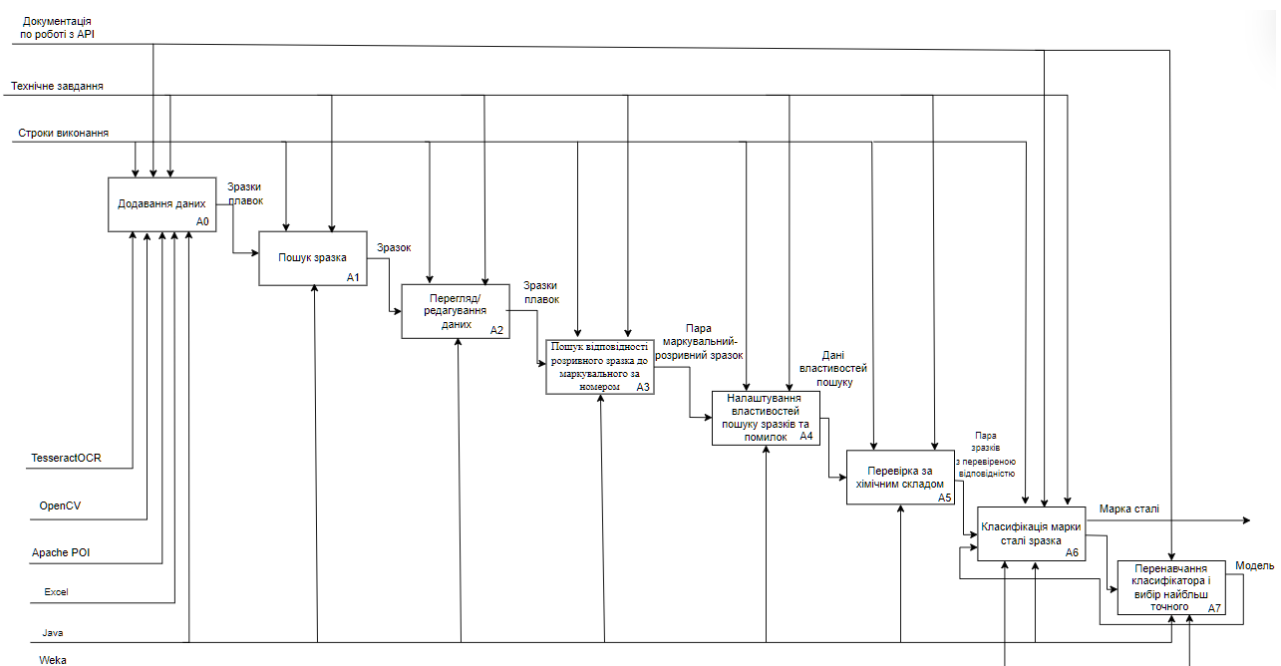


Рисунок 3.4 – Перший рівень декомпозиції інформаційної технології в нотації IDEF0

На діаграмі можна виокремити наступні блоки:

- Імпорт даних з Excel через API Apache POI, імпорт з вікна концентрацій (OpenCV, TesseractOCR) або вручну;
- Ручний пошук зразка за різними фільтрами,;
- Перегляд та редагування даних де можна змінити або видалити зразок;

- Пошук відповідності розривного зразка до маркувального за номером. Пошук за номером маркувального зразка та номером блюма розривного;
- Налаштування властивостей пошуку зразків та помилок. Змінює файл конфігурації, який зберігає відхилення по кожному елементу та назву моделі класифікатора;
- Перевірка за хімічним складом. Пошук методом аналізу інтервалів відхилень. Пошук методом аналізу інтервалів відхилень.
- Класифікація марки сталі зразка. Використання навченої моделі класифікатора для визначення марки сталі за концентраціями хімічних елементів.
- Перенавчання класифікатора та вибір найбільш точного. Можливість перенавчити модель класифікатора для збільшення точності визначення марок сталі. Актуально у випадку, коли підприємство виробляє продукцію з новим видом марок сталі, щоб додати їх клас у модель.

### 3.2.3 Моделювання варіантів використання інформаційної технології

Діаграма варіантів використання (Use Case Diagram) - це графічне зображення, що використовується для моделювання функціональності системи з точки зору користувачів (акторів) та їх взаємодій з системою. На діаграмі варіантів використання зображуються актори (користувачі, зовнішні системи або інші ролі) та варіанти їх взаємодії з системою, які називаються варіантами використання. Варіанти використання описують функціональність системи з точки зору того, як користувачі взаємодіють з нею для досягнення своїх цілей. Діаграма варіантів використання є одним із ключових інструментів у визначенні вимог до системи та в її аналізі.

Діаграма варіантів використання в нотації UML представлена на рис. 3.5.

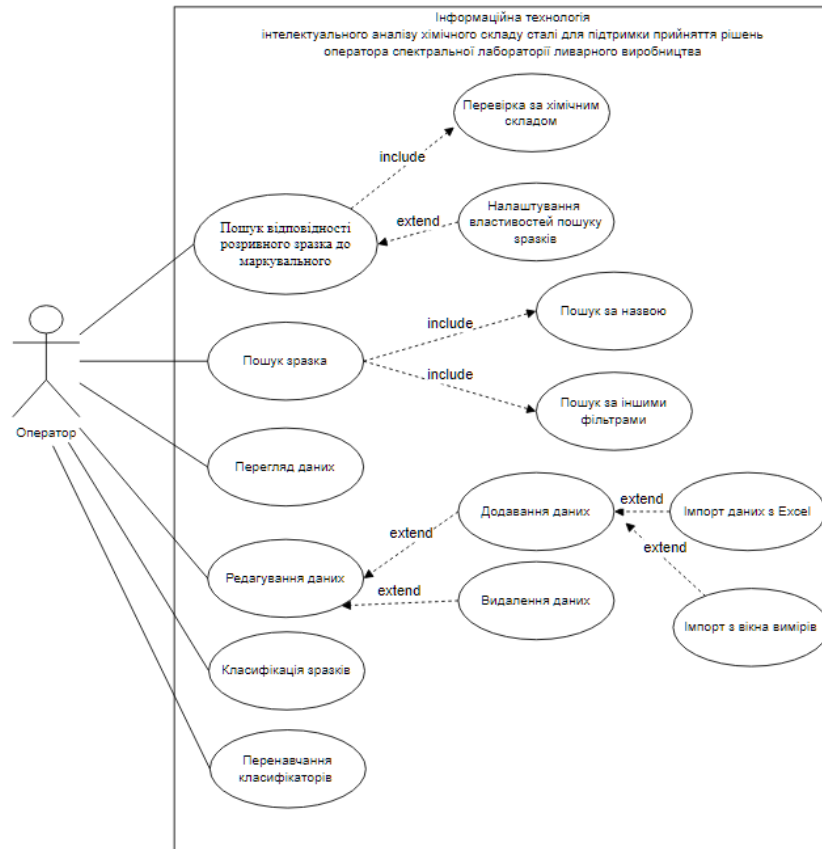


Рисунок 3.5 – UML діаграма варіантів використання інформаційної технології

Оператор може виконувати усі функції. Пошук відповідності розривного зразка до маркувального розширюється налаштуванням властивостей пошуку зразків. Класифікація зразків, перенавчання класифікатора та перегляд даних не мають розширень. Пошук зразків може виконуватися за назвою або за іншими фільтрами. Редагування даних розширюється видаленням та додаванням зразків. Додавання даних розширюється імпортом даних з Excel або за допомогою імпорту з вікна вимірювань.

### 3.2.4 Діаграма послідовності інформаційної технології.

Діаграма послідовності (Sequence Diagram) - це графічний засіб моделювання взаємодії між об'єктами в системі в певний часовий момент. Вона показує

послідовність повідомлень, що передаються між різними об'єктами системи, із зазначенням часу їх відправлення та отримання. На діаграмі послідовності зазвичай зображуються об'єкти (або класи), які взаємодіють, та лінії життя, які представляють часові інтервали, коли об'єкти знаходяться у стані активності. Взаємодії між об'єктами показуються у вигляді повідомлень, які пересилаються вздовж ліній життя. Діаграма послідовності допомагає уявити та зрозуміти порядок виконання операцій в системі, виявити потенційні проблеми взаємодії між об'єктами та сприяє уточненню вимог до системи. У діаграмі послідовності використовуються наступні умовні позначення:

- Актори. Представлені у вигляді прямокутників з назвами акторів (користувачів системи або зовнішніх систем), які взаємодіють з системою.
- Об'єкти. Представлені у вигляді прямокутників з назвами об'єктів, що взаємодіють між собою в системі.
- Лінії життя. Представлені у вигляді вертикальних ліній, що вказують на часовий проміжок існування об'єкта в системі під час виконання діаграми.
- Повідомлення. Представлені стрілками, які з'єднують об'єкти та акторів і вказують на передачу повідомлень між ними. Повідомлення можуть мати написи для уточнення типу та змісту.
- Операції. Представлені внутрішніми повідомленнями, які вказують на виклик методів чи операцій об'єкта. Вони зазвичай з'являються в середині лінії життя. На рисунку 3.6 представлена діаграма послідовності інформаційної технології.

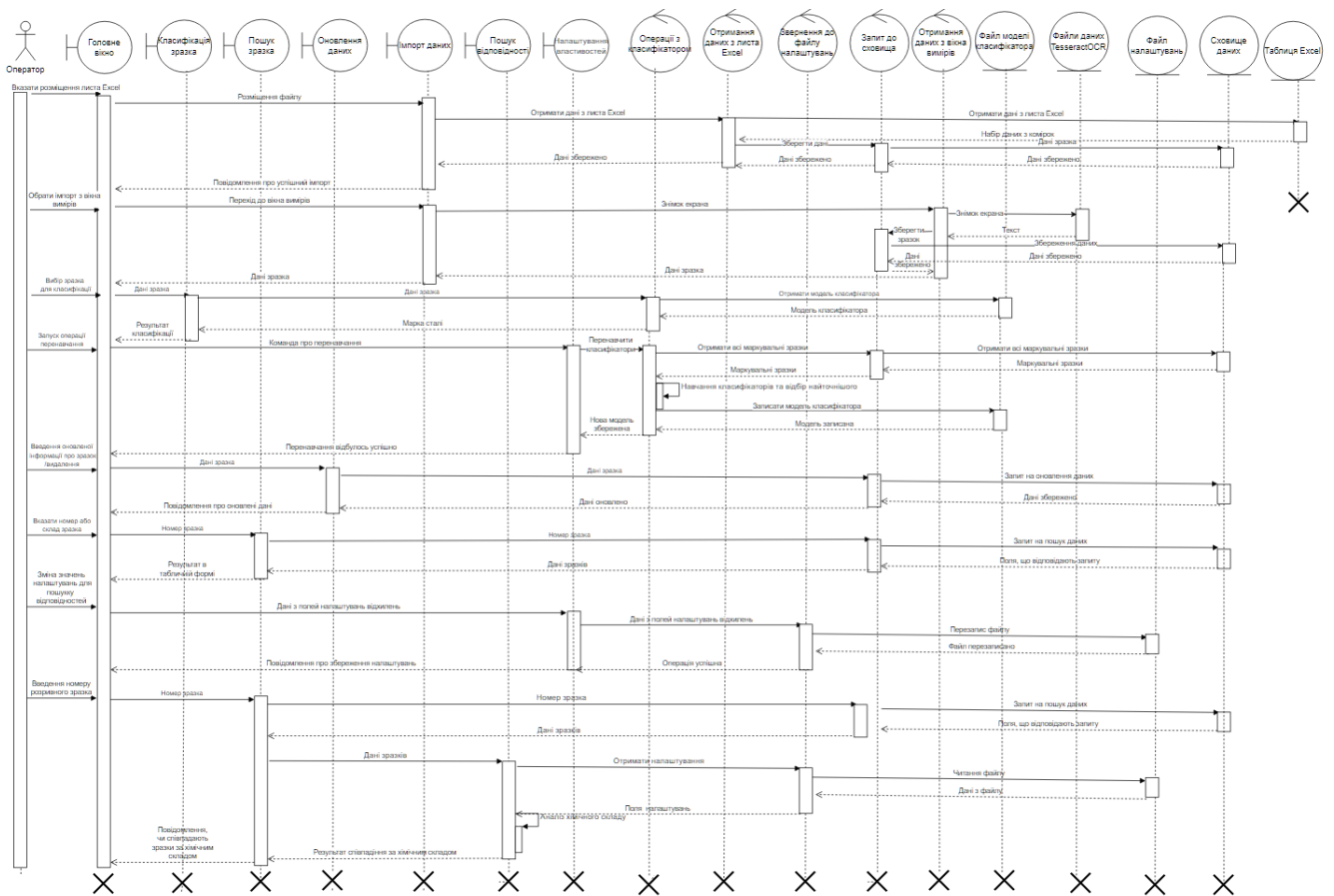


Рисунок 3.6 – Діаграма послідовності інформаційної технології

### 3.2.5 Діаграма комунікацій інформаційної технології.

Діаграма комунікацій (Communication Diagram) - це графічний інструмент для моделювання взаємодії між об'єктами в системі. На цій діаграмі об'єкти системи зображаються у вигляді прямокутників, а комунікаційні канали між ними - у вигляді стрілок або ліній. Кожна стрілка або лінія представляє собою повідомлення або спілкування між об'єктами. Діаграма комунікацій відображає структуру системи та взаємозв'язки між її елементами на високому рівні абстракції. Вона дозволяє легко уявити, які об'єкти спілкуються між собою та яким чином вони обмінюються повідомленнями. Основною метою діаграми комунікацій є уявлення структури взаємодії між об'єктами системи та ілюстрація процесів комунікації. Вона допомагає розуміти потік даних та контролю в системі та може бути використана для аналізу та вдосконалення архітектури програмного забезпечення.

На рисунку 3.7 представлена діаграма комунікацій продукту, де можна побачити класи сутності, та їх взаємодію з іншими елементами.

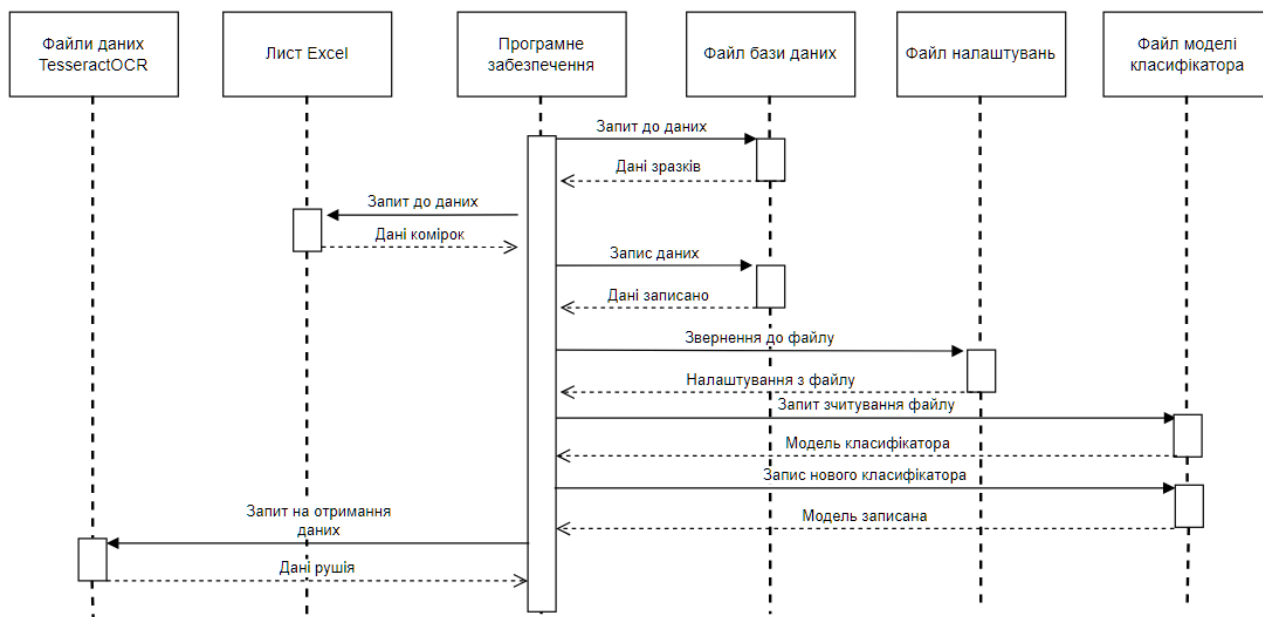


Рисунок 3.7 – Діаграма комунікації інформаційної технології

### 3.2.6 Діаграма компонентів інформаційної технології .

Діаграма компонентів (Component Diagram) - це графічний засіб моделювання, який використовується для показу структури системи та взаємозв'язків між її компонентами. На цій діаграмі компоненти представляють собою фізичні або логічні частини системи, такі як бібліотеки, модулі, пакети програмного забезпечення, або навіть фізичні пристрої. Компоненти зв'язуються між собою за допомогою зв'язків, які показують, які компоненти спілкуються між собою та яким чином це відбувається. Діаграма компонентів також може включати інші елементи, такі як інтерфейси, порти, та залежності між компонентами. Основна мета діаграми компонентів - це показати, як система організована на рівні компонентів та як вони взаємодіють один з одним для досягнення більш великої функціональності. Це дозволяє розуміти архітектуру системи, її розподіл на компоненти, та допомагає в



управлінні складністю проектування. На рисунку 3.8 представлена діаграма компонентів інформаційної технології.

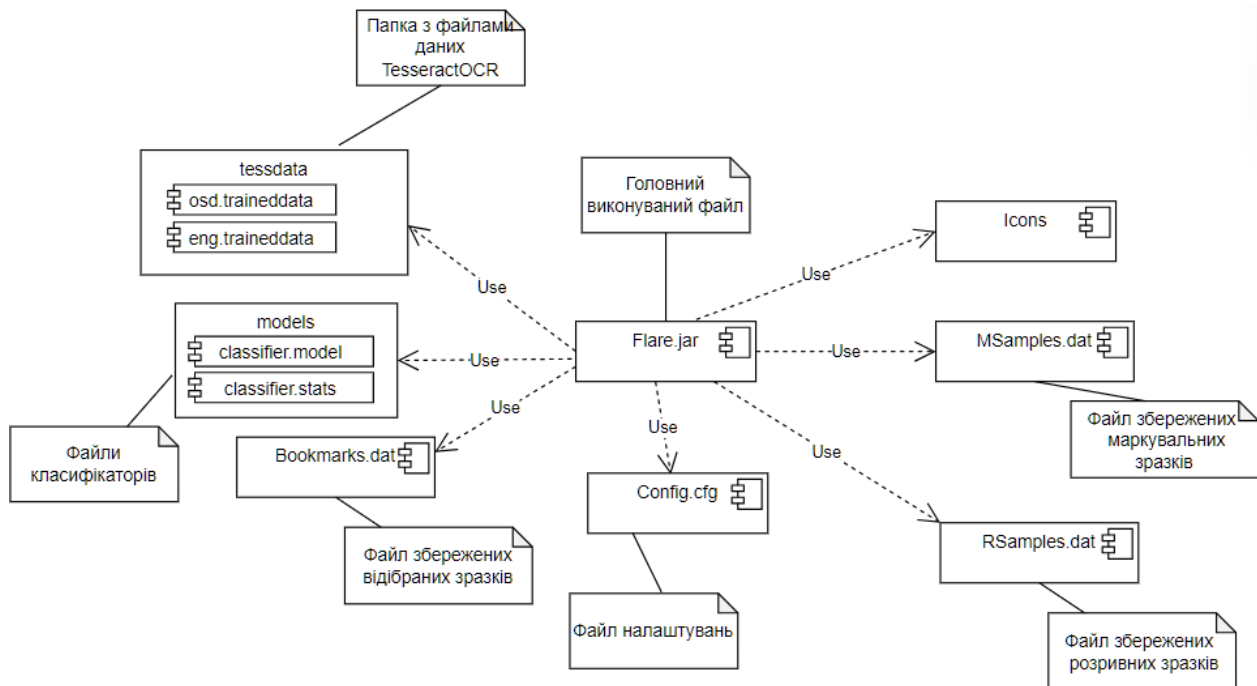


Рисунок 3.8 – Діаграма компонентів інформаційної технології

## 4 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ

### 4.1 Розробка інформаційної технології

Розробка інформаційної технології відбувалась в середовищі розробки IntelliJ IDEA.

Оскільки проект являє собою функціонал, інтегрований в програму Flare, то до інтерфейсу головного вікна, який реалізований в класі Program, в першу чергу було додано кнопки керування функціоналом: кнопка «Определить марку», кнопка налаштувати класифікатора поряд, та кнопка «Скриншот» у вікні додавання зразка (рис 4.1-4.2).

Дата	Время	Номер	Марка	Блюм	Кол-во	Серия	C	Si	Mn	P	S	Cr	Ni	Mo	Cu	Al	V	W	Ti
28.03.2...	14:23:25	0141/4	40X	26714	0		0.361	0.344	0.542	0.017	0.031	0.785	0.114	0.026	0.135	0.028	0.002	0.005	0.003
27.03.2...	12:29:13	0140/4	С120	26711	0		0.204	0.298	0.531	0.023	0.028	0.155	0.134	0.028	0.105	0.037	0.001	0.009	0.003
26.03.2...	18:10:44	0139/8	5XHM	26709	0		0.55	0.169	0.558	0.019	0.017	0.8	1.456	0.169	0.188	0.006	0.004	0.011	0.002
22.03.2...	15:56:56	0138/5	20X2H4A	26707	0		0.169	0.234	0.374	0.01	0.02	1.341	3.328	0.032	0.096	0.02	0.004	0.009	0.004
21.03.2...	12:18:04	0137/6	30X13	26704	0		0.265	0.342	0.444	0.03	0.013	12.02	0.262	0.106	0.085	0.075	0.055	0.038	0.001
20.03.2...	16:50:40	0136/9	08X13	26701	0		0.036	0.362	0.295	0.023	0.018	12.21	0.235	0.167	0.1	0.197	0.052	0.044	0.002
19.03.2...	16:56:56	0135/9	08X13	26698	0		0.042	0.378	0.377	0.026	0.011	12.79	0.227	0.091	0.093	0.248	0.042	0.035	0.001
18.03.2...	18:08:58	0134/8	08X13	26695	0		0.03	0.231	0.365	0.03	0.021	12.69	0.443	0.073	0.117	0.127	0.032	0.036	0.001
15.03.2...	15:11:10	0133/7	08X13	26692	0		0.039	0.2	0.358	0.035	0.007	12.77	0.338	0.084	0.089	0.181	0.048	0.04	0.001
14.03.2...	15:46:31	0132/5	40X	26689	0		0.365	0.252	0.628	0.01	0.013	0.905	0.184	0.032	0.134	0.002	0.003	0.008	0.002
12.03.2...	10:30:12	0131/M	0XНЗМ...	26683	6	1*1	0.378	0.269	0.332	0.009	0.007	1.536	3.33	0.4	0.242	0.005	0.17	0.007	0.003
08.03.2...	15:07:54	0130/7	0XНЗМ...		0		0.348	0.267	0.306	0.009	0.017	1.644	3.134	0.358	0.239	0.024	0.123	0.007	0.004
07.03.2...	16:46:22	0129/5	0XНЗМ...	26677	0		0.379	0.27	0.35	0.009	0.009	1.53	3.037	0.38	0.221	0.01	0.146	0.011	0.004
07.03.2...	16:17:36	0127/M	38XНЗ...	26665	9	1*1	0.363	0.26	0.327	0.013	0.013	1.291	3.035	0.356	0.219	0.006	0.119	0.029	0.003
07.03.2...	16:03:24	0126/M	38XНЗ...	26657	8	2*2	0.354	0.223	0.301	0.01	0.014	1.312	3.131	0.367	0.207	0.004	0.129	0.01	0.003
07.03.2...	15:50:51	0125/M	38XНЗ...	26649	8	1*2	0.341	0.241	0.298	0.014	0.01	1.395	3.059	0.369	0.243	0.011	0.128	0.015	0.004
06.03.2...	19:02:42	0128/6	0XНЗМ...	26674	0		0.373	0.293	0.368	0.008	0.011	1.606	3.142	0.375	0.233	0.015	0.146	0.012	0.004
05.03.2...	13:05:59	0124/M	38XНЗ...	26641	2	2*2	0.377	0.301	0.325	0.012	0.01	1.396	3.17	0.362	0.239	0.015	0.144	0.011	0.004
05.03.2...	13:02:14	0123/M	38XНЗ...	26633	8	1*2	0.351	0.237	0.299	0.01	0.011	1.315	3.069	0.35	0.248	0.011	0.113	0.011	0.003
05.03.2...	12:55:34	0122/M	38XНЗ...	26625	8	2*2	0.359	0.227	0.291	0.011	0.011	1.328	3.119	0.352	0.21	0.006	0.133	0.01	0.003
05.03.2...	12:48:16	0121/M	38XНЗ...	26618	0	1*2	0.353	0.234	0.295	0.014	0.011	1.264	3.108	0.358	0.21	0.007	0.132	0.009	0.003
29.02.2...	16:05:49	0120/4	40X	26615	0		0.385	0.264	0.545	0.011	0.021	0.842	2.008	0.035	0.21	0.044	0.002	0.007	0.003
29.02.2...	13:51:02	0119/6	60C2XFA	26613	0		0.61	1.517	0.459	0.01	0.007	0.913	0.2	0.037	0.177	0.048	0.111	0.006	0.006
29.02.2...	15:07:37	0118/M	EA4T	26606	7	2*2	0.259	0.237	0.598	0.013	0.009	1.119	0.265	0.167	0.164	0.004	0.005	0.006	0.002
29.02.2...	15:05:05	0117/M	EA4T	26596	10	1*2	0.238	0.291	0.526	0.009	0.007	0.913	0.293	0.157	0.185	0.011	0.002	0.005	0.003
29.02.2...	14:53:14	0116/M	SAE41...	26590	1	2*2	0.44	0.298	1.034	0.015	0.011	1.158	0.406	0.224	0.225	0.013	0.006	0.006	0.003
29.02.2...	14:48:45	0115/M	SAE41...	26586	4	1*2	0.446	0.298	0.992	0.016	0.011	1.042	0.346	0.223	0.226	0.011	0.005	0.008	0.003
27.02.2...	12:59:40	0114/M	SAE41...	26581	5	2*2	0.453	0.253	0.953	0.023	0.009	1.081	0.528	0.218	0.217	0.006	0.006	0.014	0.002
27.02.2...	12:56:21	0113/M	SAE41...	26577	4	1*2	0.428	0.231	0.999	0.017	0.018	1.047	0.484	0.218	0.241	0.01	0.005	0.01	0.002
26.02.2...	09:54:20	0112/M	SAE41...	26572	5	2*2	0.434	0.225	1.012	0.02	0.016	1.047	0.531	0.236	0.246	0.008	0.006	0.006	0.002
26.02.2...	09:49:29	0111/M	SAE41...	26568	4	1*2	0.421	0.296	1.012	0.017	0.013	1.034	0.502	0.234	0.207	0.01	0.006	0.007	0.002
23.02.2...	13:47:40	0110/M	SAE41...	26564	4	2*2	0.449	0.236	0.974	0.018	0.017	1.061	0.612	0.242	0.231	0.006	0.007	0.007	0.002
23.02.2...	13:38:42	0109/M	SAE41...	26559	5	1*2	0.43	0.25	0.985	0.02	0.019	1.057	0.48	0.223	0.24	0.004	0.006	0.006	0.002
22.02.2...	10:52:52	0108/M	SAE41...	26554	1	1*1	0.433	0.26	0.885	0.027	0.024	1.004	0.415	0.221	0.262	0.015	0.006	0.012	0.002
21.02.2...	10:47:18	0106/M	SAE41...	26537	14	1*1	0.424	0.232	0.992	0.017	0.008	1.097	0.588	0.235	0.212	0.006	0.007	0.006	0.002
19.02.2...	20:43:09	0107/5	SAE41...	26551	0		0.434	0.262	0.998	0.018	0.02	1.069	0.402	0.235	0.222	0.008	0.007	0.007	0.002

Рисунок 4.1 – Головне вікно з доданими кнопками

Дата	Время	Номер	Марка	Блюм	Кол-во	Серия	C	Si	Mn	P	S	Cr	Ni	Mo	Cu	Al	V	W	Ti	

Рисунок 4.2 – Вікно додавання зразка

Для реалізації отримання інформації з вікна вимірювань програмного забезпечення було створено клас Preprocessor, в якому метод `getProcessedImage()` виконує попередню обробку зображення (рис. 4.3), яке отримано при перемиканні на вікно вимірів після натискання кнопки скріншоту та виклику методу `screenshotDataCapture()` з доповненого класу Controller ПЗ «Flare» [1].

. Далі методом `doTesseract()` викликаємо метод рушія Tesseract для отримання текстової інформації і перетворення її в дані зразка.

0.0639	0.768	1.949	0.0264	0.0377	0.2157	0.1236
0.0887	0.660	1.572	0.0197	0.0303	0.1112	0.0718
0.0790	0.639	1.681	0.0342	0.0363	0.1266	0.1009
0.0057	0.1145	0.0244	0.0009	0.0020	0.0124	<0.0020
<0.0015	0.1503	0.0131	0.0004	0.0014	0.0137	<0.0020
<0.0015	0.1398	0.0108	0.0008	0.0025	0.0065	0.0021
<0.0050	<0.0001	0.0059	0.0054	<0.0010	0.0036	<0.0010
0.0058	<0.0001	0.0085	0.0071	0.0037	0.0059	<0.0010
<0.0050	<0.0001	0.0074	0.0069	0.0030	0.0072	<0.0010
<0.0010	<0.0010	0.0963	<0.0001	0.0097	96.5	0.4493
<0.0010	0.0020	0.1060	<0.0001	0.0090	97.1	0.3880
<0.0010	0.0019	0.0807	<0.0001	0.0090	97.0	0.4010

Рисунок 4.3 – Оброблений скріншот

Щоб класифікувати зразок за хімічним складом, було створено клас Classifier. При виділенні зразка та натисканні кнопки «Определить марку», викликається метод `classifyRow(MSample samples, String classifierName)` класу Controller, який отримує виділений рядок, викликає геттер змінної `modelName` класу Config та викликає метод `classify()` класу Classifier, який повертає список усіх марок сталі та відсоткове значення приналежності зразка до кожної з них. Для передачі даних в

модель треба конвертувати їх в формат .csv. За це відповідає метод `convertToCSV()` з перегрузкою для перетворення одного або списку зразків. Результат виводиться у спливаюче вікно, де можна натисканням відповідної кнопки «Вибрати марку» перенести значення у комірку зразка (рис. 4.4).

The screenshot shows the Flare software interface. The main window displays a table with columns for Date, Time, Number, Brand, Batch, Quantity, and various chemical elements (C, Si, Mn, P, S, Cr, Ni, Mo, Cu, Al, V, W, Ti). A pop-up window titled "Результат определ..." is open, showing a classification table with columns for "Марка стали" (Steel grade) and "Вероятность" (Probability). The table lists several steel grades and their corresponding probabilities. A button labeled "Выбрать марку" (Select brand) is visible in the pop-up window. The main window also has a sidebar with search and auto-search options, and a bottom bar with buttons for "Сохранить", "Импорт из Excel", "Удалить", "Добавить", and "Определить марку".

Рисунок 4.4 – Класифікація зразка

Кнопка налаштувань класифікатора відкриває відповідне вікно, в якому представлена інформація про поточну модель, обрану у спливаючому списку (рис 4.5). Швидкість навчання, імпульс, кількість епох, точність класифікації та кількість унікальних марок сталі, на яких була навчена модель – зберігає файл з розширенням .stats, який необхідний для роботи класифікатора. Інформація отримується через метод `getClassifierProperties(String selectedItem)` класу `Controller`, який шукає файл з такою ж назвою, як класифікатор та десеріалізує клас `ClassifierInfo`, що зберігає цю інформацію.

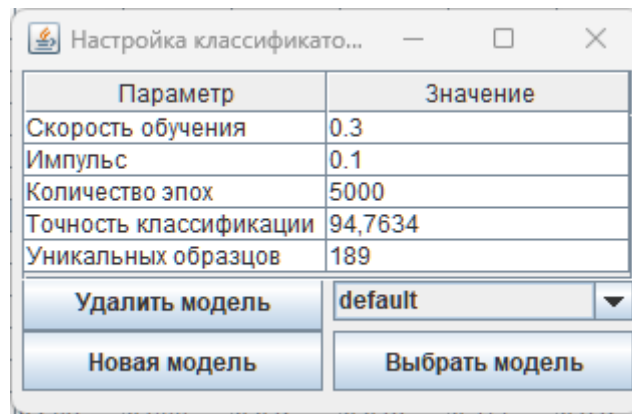


Рисунок 4.5 – Вікно налаштувань класифікатора

При натисканні на «Удалить модель» викликається метод класу `Controller deleteClassifier(String classifierName)`, що видаляє файл `.model` та `.stats`.

При натисканні на «Выбрать модель» викликається сеттер змінної `modelName` класу `Config` та передається назва моделі, обраної у спливаючому списку.

При натисканні на «Новая модель» викликається метод класу `Controller trainNewClassifier(String classifierName, String learningRate, String momentum, String epochs)`, який в свою чергу передає в метод `trainClassifier(List<MSample> mSamples, String classifierName, String learningRate, String momentum, String epochs)` класу `Classifier` налаштування, які вводить користувач перед навчанням класифікатора (рис. 4.6). Якщо користувач залишає поля вводу пустими, то за замовчуванням обираються наступні параметри: швидкість навчання 0.3, імпульс 0.1, кількість епох 1000. Метод викликає `convertToCSV(List<MSample> samples)` для перетворення даних та навчає нейронну мережу на них.

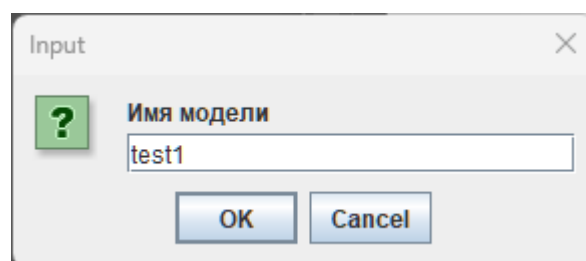


Рисунок 4.6 – Вікно вибору назви

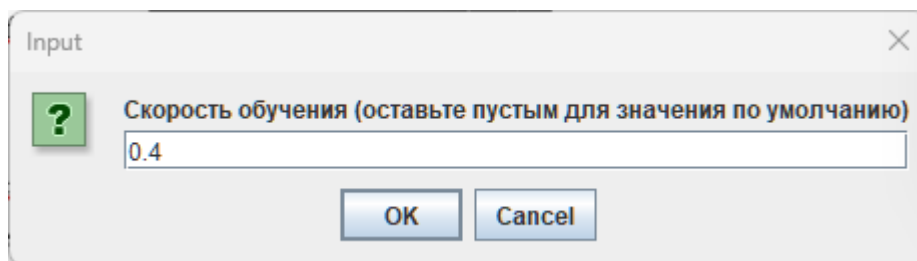


Рисунок 4.7 – Вікно вводу швидкості навчання

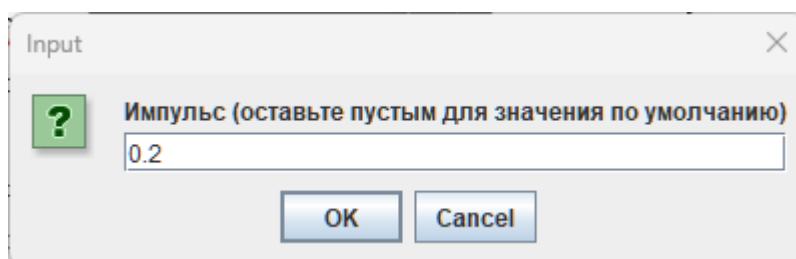


Рисунок 4.8 – Вікно вибору імпульсу (моменту)

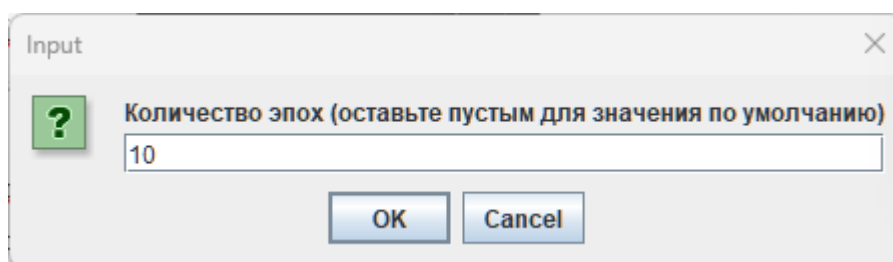


Рисунок 4.9 – Вікно вводу кількості епох

Після навчання з'явиться випадające вікно з повідомленням, якщо навчання пройшло успішно (рис. 4.10) та можна побачити інформацію про модель (рис. 4.11). Після цього нова модель готова до використання (рис.4.12).

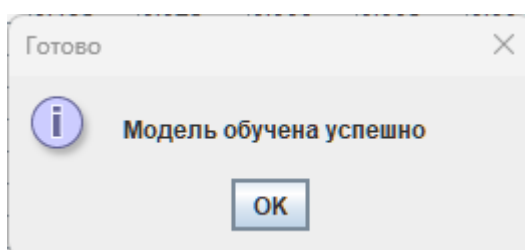


Рисунок 4.10 – Повідомлення про успішне навчання

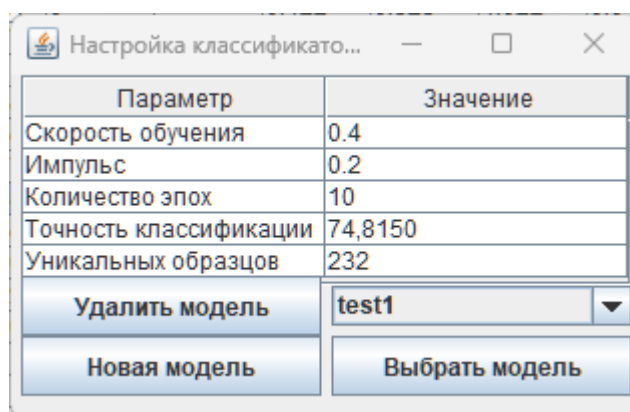


Рисунок 4.11 – Повідомлення про успішне навчання

Дата	Время	Номер	Марка	Бюльм	Кол-во	Серия	C	Si	Mn	P	S	Cr	Ni	Mo	Cu	Al	V	W	Ti
22.10.2.	16:36:58	9816/5	20GL	25249	0		0.164	0.262	1.263	0.016	0.016	0.32	0.271	0.06	0.167	0.009	0.001	0.005	0.001
18.10.2.	16:20:36	9815/3	20GL	25247	0		0.16	0.26	1.236	0.014	0.022	0.279	0.357	0.115	0.213	0.005	0.001	0.005	0.001
18.10.2.	16:20:36	9815/3	20GL	25247	0		0.16	0.26	1.236	0.014	0.022	0.279	0.357	0.115	0.213	0.005	0.001	0.005	0.001
11.10.2.	17:19:18	9814/7	30X13	25244	3		0.27	0.194	0.268	0.039	0.017	12.33	0.468	0.148	0.119	0.018	0.014	0.078	0.001
08.10.2.	16:28:08	9812/4	20GL	25239	0		0.173	0.262	1.33	0.016	0.018	0.18	0.398	0.165	0.207	0.013	0.001	0.005	0.001
08.10.2.	19:39:07	9813/4	40XN	25241	3		0.41	0.268	0.65	0.014	0.015	0.561	1.178	0.125	0.18	0.022	0.001	0.005	0.001
07.10.2.	10:26:46	9811/M	42CrM...	25235	4	2*2	0.408	0.3	0.792	0.018	0.029	1.011	0.436	0.167	0.187	0.022	0.001	0.005	0.001
07.10.2.	10:24:57	9810/M	42CrM...	25229	6	1*2	0.399	0.242	0.705	0.011	0.033	0.969	0.455	0.209	0.215	0.011	0.001	0.005	0.001
06.10.2.	14:01:44	9809/6	14X2G...	25226	3		0.127	0.256	1.024	0.018	0.017	1.358	0.318	0.402	0.171	0.006	0.004	0.037	0.002
05.10.2.	18:11:10	9808/8	14X2G...	25223	3		0.131	0.211	1.1	0.019	0.016	1.403	0.354	0.407	0.181	0.014	0.004	0.038	0.002
05.10.2.	14:07:55	9807/5	35XM	25224	3		0.478	0.011	0.016	0.86	0.515	0.206	0.239	0.011	0.002	0.037	0.002		
04.10.2.	18:52:06	9806/3	35XM	25224	3		0.614	0.015	0.008	0.883	0.544	0.203	0.27	0.015	0.003	0.039	0.003		
04.10.2.	11:43:28	9804/M	AiSi414				1.007	0.017	0.016	1.035	0.423	0.195	0.226	0.023	0.004	0.036	0.003		
04.10.2.	16:18:51	9805/4	Ci20				0.466	0.012	0.014	0.151	0.218	0.059	0.161	0.046	0.001	0.036	0.002		
01.10.2.	13:57:18	9803/3	40X	40XN			0.633	0.011	0.022	0.879	0.195	0.048	0.16	0.026	0.002	0.035	0.003		
01.10.2.	08:58:50	9802/M	AiSi414	5XHM			0.982	0.013	0.017	1.035	0.479	0.218	0.246	0.016	0.004	0.038	0.002		
30.09.2.	13:08:34	9800/M	45XGM	39NiCrMo3			0.923	0.016	0.015	0.992	0.509	0.15	0.331	0.005	0.003	0.032	0.003		
30.09.2.	15:05:20	9801/5	S460N	40XN2MA			1.332	0.014	0.012	0.171	0.557	0.046	0.39	0.02	0.002	0.034	0.003		
29.09.2.	15:22:43	9798/M	42CrM...	20XN3A			0.877	0.014	0.026	1.035	0.361	0.272	0.181	0.027	0.005	0.035	0.003		
28.09.2.	12:36:16	9797/M	45XGM				0.975	0.012	0.017	0.998	0.433	0.16	0.179	0.009	0.003	0.032	0.002		
28.09.2.	12:26:04	9796/M	45XGM				0.951	0.015	0.015	1.054	0.464	0.23	0.195	0.007	0.004	0.033	0.003		
28.09.2.	20:09:50	9799/6	C45+N				0.683	0.01	0.019	0.314	0.385	0.087	0.213	0.01	0.002	0.035	0.003		
26.09.2.	17:31:36	9795/4	42CrM...				0.391	0.23	0.674	0.014	0.027	0.992	0.345	0.19	0.168	0.049	0.003	0.032	0.003
26.09.2.	14:12:38	9794/5	42CrM...	25177	3		0.392	0.226	0.67	0.016	0.028	1.009	0.339	0.194	0.168	0.046	0.003	0.032	0.003
25.09.2.	18:42:31	9793/6	X20Cr13	25174	3		0.182	0.251	0.434	0.03	0.01	12.09	0.297	0.105	0.094	0.018	0.022	0.047	0.001
25.09.2.	15:53:55	2792/7	X20Cr13	25171	3		0.2	0.408	0.417	0.03	0.012	12.25	0.561	0.113	0.162	0.05	0.033	0.081	0.001
24.09.2.	14:18:36	9790/4	20GL	25167	0		0.154	0.257	1.322	0.017	0.015	0.101	0.318	0.061	0.19	0.005	0.001	0.029	0.006
24.09.2.	20:07:32	9791/7	20X13	25168	3		0.191	0.428	0.274	0.039	0.018	12.13	0.6	0.212	0.168	0.094	0.007	0.13	0.001
11.09.2.	14:22:02	9788/5	50CrMo4	25161	3		0.49	0.26	0.701	0.012	0.019	0.984	0.324	0.237	0.196	0.017	0.005	0.012	0.003
11.09.2.	16:54:36	9789/4	C45+N	25164	3		0.424	0.235	0.57	0.01	0.017	0.135	0.22	0.075	0.131	0.013	0.001	0.01	0.002
10.09.2.	16:44:22	9787/5	50CrMo4	25158	3		0.481	0.296	0.588	0.011	0.011	0.95	0.356	0.175	0.173	0.024	0.004	0.012	0.003
10.09.2.	13:49:32	9786/3	C45+N	25155	3		0.454	0.3	0.605	0.011	0.012	1.048	0.311	0.064	0.156	0.019	0.001	0.011	0.003
09.09.2.	13:46:53	9784/6	09G2C	25149	3		0.081	0.567	1.461	0.018	0.015	0.099	0.18	0.046	0.138	0.026	0.004	0.012	0.004
09.09.2.	16:33:24	9785/4	15GC	25152	3		0.143	0.896	0.96	0.015	0.013	0.14	0.18	0.041	0.12	0.035	0.002	0.011	0.003
09.09.2.	11:11:53	9779/M	51CrV4...	25120	4	1*1	0.501	0.313	0.753	0.017	0.016	0.974	0.149	0.026	0.214	0.02	0.119	0.013	0.004
08.09.2.	10:56:34	9781/M	30XGC...	25139	4		0.27	0.997	1.027	0.014	0.011	0.937	1.427	0.094	0.173	0.012	0.005	0.012	0.002

Рисунок 4.12 – Класифікація новою моделлю

## 4.2 Аналіз результатів

У результаті експериментів було визначено, що найбільшу точність та швидкість дають параметри швидкості навчання 0.3, імпульсу (моменту) 0.1, та кількість епох 3000. Але для додаткової точності було обрано 5000 епох.

Достовірність результатів забезпечена:

- Коректним використанням методу аналізу відхилень, який був розроблений в ПЗ «Flare»;
- Коректним використанням моделі класифікатора, побудованого на принципах багат шарового перцептрона;
- Використанням накопичених в системі статистичних даних і результатів попередньої діяльності операторів (17034 зразка за 18 років).

Достовірність підтверджена таблицею результатів перевірки похибки навченої моделі класифікатора з результатами класифікації, що відбувалась оператором в ручному режимі, що можна побачити на таблиці нижче (Табл. 4.1).



Таблиця 4.1 – Точність класифікації марок сталі

Точність	Марка сталі	Точність	Марка сталі
0,000	03X17H11M2	0,000	12X18H10T
0,917	20X13	1,000	12X2H4A
0,950	1.2327	1,000	13XGMRB
0,000	03X17H12M2	1,000	14X17H2
0,000	03X17H14M3	1,000	14X2GMP
0,000	03X18H10	1,000	14XGC
0,786	03X18H11	0,963	15GC
0,923	03X18H14M3	0,000	15X1M1F
0,000	05X12H2M	1,000	15X5M
1,000	06X12H3D	1,000	15XM
1,000	07X16H6	0,933	15-Y
0,000	08X13	1,000	16GC
0,000	08X14MF	0,789	17G1C
0,926	08X17H12M2T	1,000	18CrNiMo7-6
0,000	08X17H13M2T	1,000	18NiCr5-4
0,983	08X18H10T	1,000	18NiCrMo5
1,000	08X18H6AG10C	1,000	18XGH2MFB
0,000	08X21H6M2T	1,000	18XGT
0,000	08X22H6T	0,897	20GL
0,000	08X25H20	0,000	20X17H2
1,000	09G2C	0,000	20X1M1F1TP
1,000	09G2CL	0,985	20X2H4A
1,000	0X1H3MFAY	1,000	20XGCA
0,000	1.4104/QT650	1,000	20XGHP
0,000	10X13G12BC2H2D2	1,000	20XH3A
0,789	10X17H13M2T	1,000	20Ych
0,000	10X23H18	1,000	21CrMoV5-11

Продовження таблиці 4.1 – Точність класифікації марок сталі

Точність	Марка сталі	Точність	Марка сталі
0,964	22K	0,957	35XM
0,000	22X3M	0,733	36CrNiMo4
0,950	25CrMo4	1,000	36CrNiMo6+QT
0,000	25X1MF	1,000	36G2C
1,000	25X2M1F	1,000	36NiCrMo16
1,000	25XGT	0,955	36XHM
0,938	25XM-Y	0,857	38X2H2MA
1,000	30CrGS+A	1,000	38X2MUA
0,000	30CrMoV9+QT	0,966	40X
1,000	30CrNiMo8+QT	1,000	38XC
1,000	30CrNiMo8+QT(Y1)	1,000	38XGH
0,922	30X13	1,000	39NiCrMo3
0,846	30X2H2M-Y	0,975	40XH
1,000	30XGCA	1,000	40CrMnMoS8-6
1,000	30XGCH2A	0,000	40X13
0,000	30XH2MFA	1,000	40XFA
0,937	30XMA	0,870	40XGMA
1,000	31CrMoV9+QT	1,000	40XH2MA
1,000	32X2HVMBP	1,000	41CrAlMo7
1,000	34CrAlNi7-10+QT	1,000	42CM
0,865	34CrNiMo6+QT	0,852	42CrMo4
0,800	34X2H2M	0,714	42CrMoS4
0,833	34X2H2M-Y	0,698	42CrMo4-Y
0,545	34X2HMU-Y	0,998	SAE4145H
0,750	34XH1M	0,000	45X2H4MA
0,000	34XH3M	0,167	45-Y
0,909	35XGCA	0,000	46CrSiMoV7+A

Продовження таблиці 4.1 – Точність класифікації марок сталі

Точність	Марка сталі	Точність	Марка сталі
0,000	4X5MFC	0,717	AiSi4130 NQ+T
1,000	50CrMo4	0,440	AiSi4130 Q+T
0,500	50XH	0,667	AiSi4130 Q+T/1
1,000	50XM	0,250	AiSi4130 Q+T/2
0,611	51CrV4	0,500	AiSi4130 NQ+T(1)
0,913	51CrV4+A(1)	0,833	AiSi4140 Hmod Q+
1,000	51CrV4+A(2)	0,531	AiSi4140H
0,981	55NiCrMoV7	0,645	AiSi4140Q+T
1,000	55X	0,211	AiSi4142
0,958	56NiCrMoV7	0,011	AiSi4145H
0,000	56CrNiV-7	0,372	AiSi4145Hmod
0,000	5XB2CF	0,857	AiSi4145HmodQ+T
1,000	5XHM	0,842	AiSi4145Q+T
1,000	60C2XFA	1,000	AiSi4330
1,000	60-Y	1,000	AiSi4340 QT
1,000	65G	1,000	AiSi4340
0,000	6XB2C	1,000	AiSi4715M
1,000	817M40	1,000	AiSi8620
1,000	86CrMoV	1,000	C15
0,889	90MnCrV8+A	0,905	Ct40
1,000	9X1	0,959	Ct45
1,000	9XC	0,000	CCh15
1,000	A-350LF2	0,987	Ct20
0,953	AiSi4130	0,000	Ct25
0,928	AiSi4140	0,600	Ct30
1,000	AiSi1340N	1,000	Ct35
0,950	AiSi1340N/1	0,571	Ct50

Продовження таблиці 4.1 – Точність класифікації марок сталі

Точність	Марка сталі	Точність	Марка сталі
1,000	Ct55	0,803	S355J2
0,800	CtOC	0,833	S355JO
1,000	EA1N /35G/	0,500	S460NL
0,500	EA4T	0,500	S460NL+N
1,000	EN19	0,905	SA-105
1,000	EP517	0,968	SA-694
0,000	F304L	0,000	X17CrNi16-2
0,000	G-X3CrNiMo 13-4	0,000	X30Cr13
0,850	OC	0,000	X38CrMo16
0,552	S355J2G3	1,000	X52
0,868	S355J2G4	0,000	X5CrNiCuNb16-4
1,000	S355NL	0,000	Y10A

Таким чином, досягнута точність 94,7%. Існує деяка похибка, яку оператор має можливість усунути на послідовних етапах аналізу. Але, при тому, час обробки зразка в середньому зменшився на ~20%, що підтверджено серією експериментів в лабораторії, що сприяє зменшенню напруженості діяльності операторів.

## ВИСНОВКИ

В ході дослідження було проаналізовано проблеми операторів спектрального аналізу ДП «Завод ОБ та ВТ». Були зроблені висновки про те, що велика кількість часу витрачається на класифікацію зразків, оператор працює в умовах темпової напруженості. Це впливає на час експрес-аналізу зразка і на те, що на даному етапі може виникнути помилка оператора.

Для вирішення проблеми доцільно створити систему підтримки прийняття рішень, яка б могла класифікувати зразки за хімічним складом та присвоювати їм відповідну марку сталі

Розроблено метод класифікації зразків з використанням моделі класифікатора побудованого за принципом багатосарового персептрона. Модель була перевірена на накопичених даних спектральної лабораторії за 18 років, тобто зразках (17034 зразка), що класифікувались операторами в ручному режимі.

Запропонована технологія може бути основою системи підтримки прийняття рішень оператора спектральної лабораторії і забезпечує точність класифікації 94,76%, скорочує час класифікації в середньому на 20%, та дозволяє зменшити операційно-темпову напруженість, що є прийнятним для використання на підприємстві результатом.

На відміну від існуючих наукових підходів, що використовують алгоритмічний підхід до аналізу марок сталі хімічного складу зразків, запропонована модель основана на моделях інтелектуального аналізу накопичених в сховищах даних лабораторії про визначення марок сталі зразків.

Інформаційну технологію введено в робочий процес ДП «Завод ОБ та ВТ».

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Бельдієв А. С. Інформаційна технологія підтримки діяльності оператора аналізу хімічного складу зразків ливарної продукції : робота на здобуття кваліфікаційного ступеня бакалавра : спец. 122 - комп'ютерні науки / наук. кер. Є. А. Лавров. Суми : Сумський державний університет, 2022. 117 с.
2. Update DIA2000 [Електронний ресурс] – Режим доступу до ресурсу: <https://fp-lims.net/no-es-possible-actualizar-dia2000se-a-windows-10/> (дата звернення: 21.04.2024).
3. LIMS prices and information about the different versions [Електронний ресурс] – Режим доступу до ресурсу: <https://www.fp-lims.com/en/lims-software-prices-and-versions/> (дата звернення: 21.04.2024).
4. LIMS Software [Електронний ресурс] – Режим доступу до ресурсу: <https://fp-lims.net/> (дата звернення: 21.04.2024).
5. E. Lavrov , O. Siryk, “Functional Networks For Ergonomics And Reliability Tasks On The 90th Anniversary Of A. Gubinsky And V. Evgrafov”, *Sigurnost* 64 (3) 245 - 257 (2022) <https://doi.org/10.31306/s.64.3.3>
6. Lavrov, E., Siryk, O., Kirichenko, I., Barchenko, N., Chybiriak, Y. The Methodology of Managed Functional Networks for Organizing Effective and Adaptive Human-Machine Dialogue in Automated Systems CEUR Workshop Proceedings, 2021, 3013, pp. 428–437
7. Lavrov, E., Paderno, P., Siryk, O., Kyzenko, V., Kosianchuk, S., Bondarenko, N., Burkov, E. Functional Networks for Modeling and Optimization Human-Machine Systems (2021) *Advances in Intelligent Systems and Computing*, 1313 AISC, pp. 189-196. <http://www.springer.com/series/11156> ISBN: 978-303066936-2 doi: 10.1007/978-3-030-66937-9\_21
8. Lavrov, E., Pasko, N. Development of Models for Computer Systems of Processing Information and Control for Tasks of Ergonomic Improvements Communications in

- Computer and Information Science, 2018, 920, pp.98–109 DOI 10.1007/978-3-319-99972-2\_8
9. Lavrov, E., Pasko, N., Krivodub, A. Automated analysis of ergonomic measures in discrete control systems (2015) Eastern-European Journal of Enterprise Technologies, 4 (3), pp. 16-22. doi: 10.15587/1729-4061.2015.48050
  10. E. Lavrov, O. Siryk, Y. Chybiriak, L. Danilova, V. Nahorny and S. Vakal, "A Model for the Organization of Adaptive Dialogue Interaction "Man-Computer" Taking into Account the Requirements of Reliability and Efficiency," 2021 IEEE 4th International Conference on Advanced Information and Communication Technologies (AICT), Lviv, Ukraine, 2021, pp. 31-35, doi: 10.1109/AICT52120.2021.9628939
  11. E. Lavrov, O. Siryk, N. Sedova, P. Paderno, E. Burkov and S. Vakal, "Modeling Errors and Failures in Automated Human-Machine Control Systems to Solve Reliability Assessment Tasks," 2021 IEEE 8th International Conference on Problems of Infocommunications, Science and Technology (PIC S&T), Kharkiv, Ukraine, 2021, pp. 399-402, doi: 10.1109/PICST54195.2021.9772122.
  12. E. Lavrov, "Methodology for Reliability Analysis of Human-Machine Interaction in Automated Control Systems," 2021 IEEE 4th International Conference on Advanced Information and Communication Technologies (AICT), Lviv, Ukraine, 2021, pp. 293-297, doi: 10.1109/AICT52120.2021.9628925
  13. E. Lavrov, O. Siryk, B. Rybalov and D. Velykodnyi, "The Method of Adaptive Control of the Process of Human-Machine Interaction to Ensure the Reliability and Efficiency of Automated Control Systems," 2021 IEEE 8th International Conference on Problems of Infocommunications, Science and Technology (PIC S&T), Kharkiv, Ukraine, 2021, pp. 403-407, doi: 10.1109/PICST54195.2021.9772126.
  14. E. Lavrov, "Mathematical Models of Human-Computer Interaction. Analysis of Human Operator Activity in Terms of Error Generation and Problem Elimination," 2022 IEEE 9th International Conference on Problems of Infocommunications, Science and Technology (PIC S&T), Kharkiv, Ukraine, 2022, pp. 325-328, doi: 10.1109/PICST57299.2022.10238654.

15. Lavrov E., Paderno P., Siryk O., Burkov E., Kosianchuk S., Bondarenko N., Kyzenko V. Analysis of Working Conditions and Modeling of Activity Algorithms for Contact-Center Operators Proceedings of the 8th International Ergonomics Conference. March, 2021. Switzerland: Springer, 2021. Pp.121–128. DOI:10.1007/978-3-030-66937-9\_14
16. Lavrov, E., Pasko, N. Automation of assessing the reliability of operator's activities in contact centers that provide access to information resources CEUR Workshop Proceedings, 2018, 2105, pp. 445–448
17. E. Lavrov, P. Paderno, O. Siryk, E. Burkov, N. Pasko and V. Nahorny, "Decision Support in Incident Management Systems. Models of Searching for Ergonomic Reserves to Increase Efficiency," 2020 IEEE International Conference on Problems of Infocommunications. Science and Technology (PIC S&T), Kharkiv, Ukraine, 2020, pp. 653-658, doi: 10.1109/PICST51311.2020.9467991
18. E. Lavrov et al., "Mathematical Models and Decision Support System for the Efficiency and Ergonomic Quality of IT Service Management Systems," 2021 IEEE 8th International Conference on Problems of Infocommunications, Science and Technology (PIC S&T), Kharkiv, Ukraine, 2021, pp. 506-510, doi: 10.1109/PICST54195.2021.9772245.
19. E. Lavrov, A. Tolbatov, N. Pasko and V. Tolbatov, "Ergonomie reserves for improving reliability of data processing in distributed banking systems," 2017 2nd International Conference on Advanced Information and Communication Technologies (AICT), Lviv, Ukraine, 2017, pp. 79-82, doi: 10.1109/AIACT.2017.8020070
20. E. Lavrov and O. Siryk, "Ergonomic Support for the Activities of Software Testers," 2022 IEEE 9th International Conference on Problems of Infocommunications, Science and Technology (PIC S&T), Kharkiv, Ukraine, 2022, pp. 335-340, doi: 10.1109/PICST57299.2022.10238605.
21. Lavrov, E., Siryk, O. (2023). Decision Support for Solving Problems of Ergonomic Provision of Contact Centers. In: Salopek Čubrić, I., Čubrić, G., Jambrošić, K., Jurčević Lulić, T., Sumpor, D. (eds) Proceedings of the 9th International



- Ergonomics Conference. ERGONOMICS 2022. Lecture Notes in Networks and Systems, vol 701. Springer, Cham. [https://doi.org/10.1007/978-3-031-33986-8\\_19](https://doi.org/10.1007/978-3-031-33986-8_19)
22. Lavrov, E., Pasko, N., Siryk, O., Mukoseev, V., Dubovyk, S. Automation of reliability assessment of functional elements of flexible automated production based on functional network methodology CEUR Workshop Proceedings, 2020, 2740, pp. 357–364
23. E. Lavrov, N. Pasko, O. Siryk, P. Paderno and E. Burkov, "Models and Information Technology for Reliable Design of the Functioning Processes of Flexible Production Systems as Complex Human-Machine Systems," 2020 IEEE International Conference on Problems of Infocommunications. Science and Technology (PIC S&T), Kharkiv, Ukraine, 2020, pp. 673-678, doi: 10.1109/PICST51311.2020.9467894.
24. E. Lavrov, N. Pasko, A. Krivodub and A. Tolbatov, "Mathematical models for the distribution of functions between the operators of the computer-integrated flexible manufacturing systems," 2016 13th International Conference on Modern Problems of Radio Engineering, Telecommunications and Computer Science (TCSET), Lviv, Ukraine, 2016, pp. 72-75, doi: 10.1109/TCSET.2016.7451974
25. Lavrov, E., Siryk, O., Volosiuk, A., Zolkin, A., Sedova, N. Sustainability and reliability ensurance models for automated technological systems in chemical industry: systemic ergonomic approach (2021) E3S Web of Conferences, 280, art. no. 02005. doi: 10.1051/e3sconf/202128002005
26. Pinchuk, O., Sokolyuk, O., Burov, O., Lavrov E., Shevchenko, S., Aksakovska, V. ICT for training and evaluation of the solar impact on aviation safety CEUR Workshop Proceedings, 2020, 2732, pp. 786–792
27. Sedova N.A., Sedov V.A., Lavrov E.A., Bazhenov R.I., Gorbunova T.N. Assessing the degree of collision hazard for an autonomous unmanned vessel based on fuzzy logic and neural network technologies // Marine Intelligent Technologies. 2021. Vol. 3. No. 4 (54). pp. 128-134.
28. O. Siryk and E. Lavrov, "Mathematical Models for Assessing the Reliability of Processes of Automated Control of the Main Gas Pipeline, Taking into Account the

- "Human Factor"," 2022 IEEE 9th International Conference on Problems of Infocommunications, Science and Technology (PIC S&T), Kharkiv, Ukraine, 2022, pp. 329-334, doi: 10.1109/PICST57299.2022.10238600.
- 29.Lavrov, E., Logvinenko, V., Siryk, O., Kyzenko, V. Method for assessing the information content of factors forming the cognitive independence of students (2021) Journal of Physics: Conference Series, 1840 (1), art. no. 012066. <http://iopscience.iop.org/journal/1742-6596> doi: 10.1088/1742-6596/1840/1/012066
- 30.Lavrov, E., Siryk, O., Chabanenko, P. A method to ensure the effectiveness and attractiveness of e-learning. Human-oriented systemic ergonomic approach CEUR Workshop Proceedings, 2020, 2732, pp. 572–582
- 31.Lavrov, E., Lavrova, O. Intelligent adaptation method for human-machine interaction in modular E-learning systems CEUR Workshop Proceedings, 2019, 2393, pp. 1000–1010
- 32.Khramova, L.N., Tsakhaeva, A.A., Posokhova, A.V., Lavrov E.A.Litvishkov, V.M., Vilkova, A.V. Modern managers training in the context of competence approach Journal of Social Sciences Research, 2018, 2018 (Special Issue 5), pp. 194–199 DOI 10.32861/jssr.spi5.194.199
- 33.Lavrov, E., Barchenko, N., Pasko, N., Borozenec, I. Development of models for the formalized description of modular e-learning systems for the problems on providing ergonomic quality of human-computer interaction (2017) Eastern-European Journal of Enterprise Technologies, 2 (2-86), pp. 4-13. doi: 10.15587/1729-4061.2017.97718
- 34.Lavrov, E., Kuppenko, O., Lavryk, T., Barchenko, N. Organizational approach to the ergonomic examination of e-learning modules (2013) Informatics in Education, 12 (1), pp. 105-123
- 35.Lavrov, E., Logvinenko, V., Siryk, O., Kyzenko, V. Method for assessing the information content of factors forming the cognitive independence of students (2021) Journal of Physics: Conference Series, 1840 (1), art. no. 012066. <http://iopscience.iop.org/journal/1742-6596> doi: 10.1088/1742-6596/1840/1/012066

36. Krause J. et al. Smartspectrometer—embedded optical spectroscopy for applications in agriculture and industry // *Sensors*. 2021. Vol. 21, № 13.
37. Karatas M., Geçili H. THE ROLE OF DECISION SUPPORT SYSTEMS IN STEEL INDUSTRY // *Engineering Science and Technology, an International Journal*. 2012. Vol. 15. P. 45–52.
38. Azimi S.M. et al. Advanced steel microstructural classification by deep learning methods // *Sci Rep*. 2018. Vol. 8, № 1.
39. Li J.Y. et al. Applications of iron and steel industry with deep learning technologies // *Kang T'ieh/Iron and Steel*. 2021. Vol. 56, № 9.
40. Spirin N.A. et al. Digital Transformation of Pyrometallurgical Technologies. State, Scientific Problems, and Prospects of Development // *Steel in Translation*. 2021. Vol. 51, № 8.
41. Kong H. et al. Selection of spectral data for classification of steels using laser-induced breakdown spectroscopy // *Plasma Science and Technology*. 2015. Vol. 17, № 11.
42. Long Liang, Tianlong Zhang, Kang Wang, Hongsheng Tang, Xiaofeng Yang, Xiaoqin Zhu, Yixiang Duan, and Hua Li, "Classification of steel materials by laser-induced breakdown spectroscopy coupled with support vector machines," *Appl. Opt.* 53, 544-552 (2014)
43. Pavel Pořízka, Jakub Klus, David Prochazka, Erik Képeš, Aleš Hrdlička, Jan Novotný, Karel Novotný, Jozef Kaiser, Laser-Induced Breakdown Spectroscopy coupled with chemometrics for the analysis of steel: The issue of spectral outliers filtering, *Spectrochimica Acta Part B: Atomic Spectroscopy*, Volume 123, 2016, Pages 114-120, ISSN 0584-8547, <https://doi.org/10.1016/j.sab.2016.08.008>.
44. Tianlong Zhang, Donghui Xia, Hongsheng Tang, Xiaofeng Yang, Hua Li, Classification of steel samples by laser-induced breakdown spectroscopy and random forest, *Chemometrics and Intelligent Laboratory Systems*, Volume 157, 2016, Pages 196-201, ISSN 0169-7439, <https://doi.org/10.1016/j.chemolab.2016.07.001>.

45. Boikov A. et al. Synthetic data generation for steel defect detection and classification using deep learning
46. L. Breiman, J. Friedman, R. A. Olshen, and C. J. Stone. Classification and Regression Trees. Chapman and Hall, 1984.
47. Wang W., Zhu M., Wang J., Zeng X., Yang Z. End-to-end encrypted traffic classification with onedimensional convolution neural networks. In Proc. of the IEEE International Conference on Intelligence and Security Informatics (ISI), 2017, pp. 43-48.
48. Aceto G., Ciuonzo D., Montieri A., Pescapé A. Multi-classification approaches for classifying mobile app traffic. Journal of Network and Computer Applications, vol. 103, 2018, pp. 131-145.
49. Li W., Moore A. W. A Machine Learning Approach for Efficient Traffic Classification. In Proc. of the 15th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, 2007, pp. 310-317.
50. Stephen D. Bay. Combining nearest neighbor classifiers through multiple feature subsets. In Proceedings of the 17th International Conference on Machine Learning, pages 37–45, Madison, WI, 1998.
51. Takyi K., Bagga A., Gupta P. A Semi-Supervised QoS-Aware Classification for Wide Area Networks with Limited Resources. International Journal of Innovative Technology and Exploring Engineering, vol. 8, issue 11, 2019, pp. 970-981.
52. Corinna Cortes and Vladimir Vapnik. Support-vector networks. Machine Learning, pages 273–297, 1995.
53. C. J. Burges. A tutorial on support vector machines for pattern recognition. In Data Mining and Knowledge Discovery, pages 1–47, 1998.
54. Erin J. Bredensteiner and Kristin P. Bennett. Multicategory classification by support vector machines. Computational Optimization and Applications, 12:53–79, January 1999.
55. J. Weston and C. Watkins. Multi-class support vector machines. Technical Report CSD-TR-98-04, Department of Computer Science, Royal Holloway, University of London, 1998.

56. Koby Crammer and Yoram Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research*, pages 265–292, 2001.
57. Yoonkyung Lee, Yi Lin, and Grace Wahba. Multicategory support vector machines: Theory and application to the classification of microarray data and satellite radiance data. *Journal of the American Statistical Association*, 99(465):67–81, March 2004.
58. J. Weston and C. Watkins. Support vector machines for multiclass pattern recognition. In *Proceedings of the Seventh European Symposium On Artificial Neural Networks*, 4 1999.
59. Chih-Wei Hsu and Chih-Jen Lin. A comparison of methods for multiclass support vector machines. In *IEEE TRANSACTIONS ON NEURAL NETWORKS*, volume 13, pages 415–425, March 2002.
60. Breiman, L. Random Forests. *Machine Learning* 45, 5–32 (2001). <https://doi.org/10.1023/A:1010933404324>
61. Gómez S.E., Martínez B.C. et al. Ensemble network traffic classification: Algorithm comparison and novel ensemble scheme proposal. *Computer Networks*, vol. 127, 2017, pp. 68-80.
62. Jin Y., Duffield N. et al. A modular machine learning system for flow-level traffic classification in large networks. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 6, issue 1, 2012, pp. 1- 34.
63. Zhao S., Chen S. et al. Identifying Known and Unknown Mobile Application Traffic Using a Multilevel Classifier. *Security and Communication Networks*, vol. 2019, 2019, article id 9595081.
64. Ensemble methods. Scikit-learn. [Электронный ресурс] – Режим доступа до ресурсу: <https://scikit-learn.org/stable/modules/ensemble.html> (дата звернення: 21.04.2024).
65. Shu-Hsien Liao, Pei-Hui Chu, Pei-Yuan Hsiao, “Data mining techniques and applications – A decade review from 2000 to 2011”, Elsevier, pg 11303-11311, 2012.

66. A.S. Ahmad, M.Y. Hassan, M.P. Abdullah, H.A. Rahman, F. Hussin, H. Abdullah and R. Saidur, “A review on applications of ANN and SVM for building electrical energy consumption forecasting”, Elsevier, pg 102-109, 2014.
67. R. Smith. An overview of the Tesseract-OCR engine. // Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on. IEEE, 2007. Vol. 2. P. 629 – 633.
68. C. Olah. Understanding LSTM Networks [Электронный ресурс] – Режим доступа до ресурсу: <http://colah.github.io/posts/2015-08-Understanding-LSTMs/> (дата звернення: 21.04.2024).
69. Burges C. A tutorial on support vector machines for pattern recognition. [Электронный ресурс] – Режим доступа до ресурсу: <http://research.microsoft.com/en-us/um/people/cburges/papers/SVMTutorial.pdf> (дата звернення: 21.04.2024).
70. Guangya Zhang, Huihua Ge. Support vector machine with a Pearson VII function kernel for discriminating halophilic and non-halophilic proteins, Computational Biology and Chemistry, Volume 46, 2013, p.16-22, ISSN 1476-9271, <https://doi.org/10.1016/j.compbiolchem.2013.05.001>.

## ДОДАТОК А. Повний список робіт проекту

З поширенням тотальної автоматизації у столітті, в якому ми живемо, пришвидшується робота та виробництво в усіх сферах діяльності. Підприємства, в яких раніше було виробництво, сконцентроване на ручній роботі, наразі переходить до спрощення процесів саме завдяки інформаційним технологіям.

Розробивши програмне забезпечення для спектральної лабораторії металургічного підприємства, можна значно пришвидшити процес аналізу зразків металу та збільшити кількість плавок на день.

Деталізацію мети проекту виконують за допомогою SMART-методу. SMART-метод дозволяє конкретно сформулювати цілі, призначення, часові обмеження проекту.

Для даного проекту формат постановки SMART-мети такий:

«Розробити інформаційну технологію інтелектуального аналізу хімічного складу сталі для підтримки прийняття рішень оператора спектральної лабораторії ливарного виробництва з можливістю класифікації зразків до кінця 2 курсу магістратури для скорочення часу виробництва на підприємстві».

Результати деталізації мети даного проекту представлено в таблиці А.1.

Таблиця Б.1 – Деталізація мети проекту методом SMART

Specific	Скорочення часу виробництва ливарної продукції
Measurable	Створене програмне забезпечення підтримки прийняття рішень оператора спектральної лабораторії ливарного виробництва
Achievable	Даний проект можна завершити вчасно враховуючи обсяги роботи та технічне завдання
Relevant	Для підвищення швидкості виробництва та зменшення

	кількості помилок оператора
Time-framed	Є конкретний термін – до кінця 2 курсу магістратури (10 травня 2024 р.).

WBS – це графічне відображення цілей проекту у вигляді ієрархії, де йде поділ по групах від глобальних цілей до більш конкретних.

На першому рівні WBS розміщено продукт проекту. На другому рівні розташовано глобальні дії для досягнення мети. На наступних рівнях йде розбиття задач на все більш прості, доки вони не стануть елементарними, тобто такими, які можна сформулювати конкретно із зазначенням часу на виконання та розподіл людей.

Елементарною можна вважати задачу, яку не можна розділити на простішу, або якщо це не потрібно в рамках проекту через зрозумілість цілі вже на даному етапі конкретизації.

Планування у вигляді WBS можна побачити на рисунку Б.1.

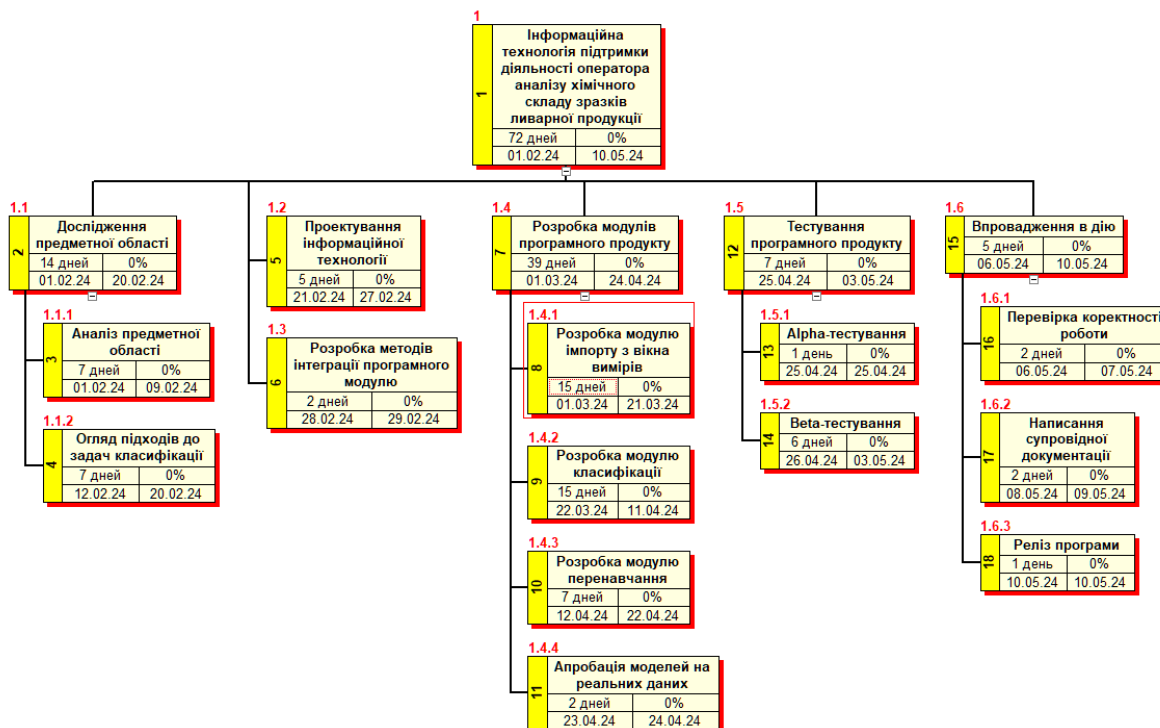


Рисунок А.1 – Планування змісту структури робіт ІТ-проекту WBS

**Планування структури виконавців.** Після визначення цілей треба закріпити їх за учасниками проекту. Це відображається в діаграмі OBS - організаційна структура



планування, де замість назви цілі вписано імена людей, закріплених за проектом. На рисунку А.2 зображено OBS-структуру . Матрицю відповідальності за цілі проекту описано в таблиці А.2.

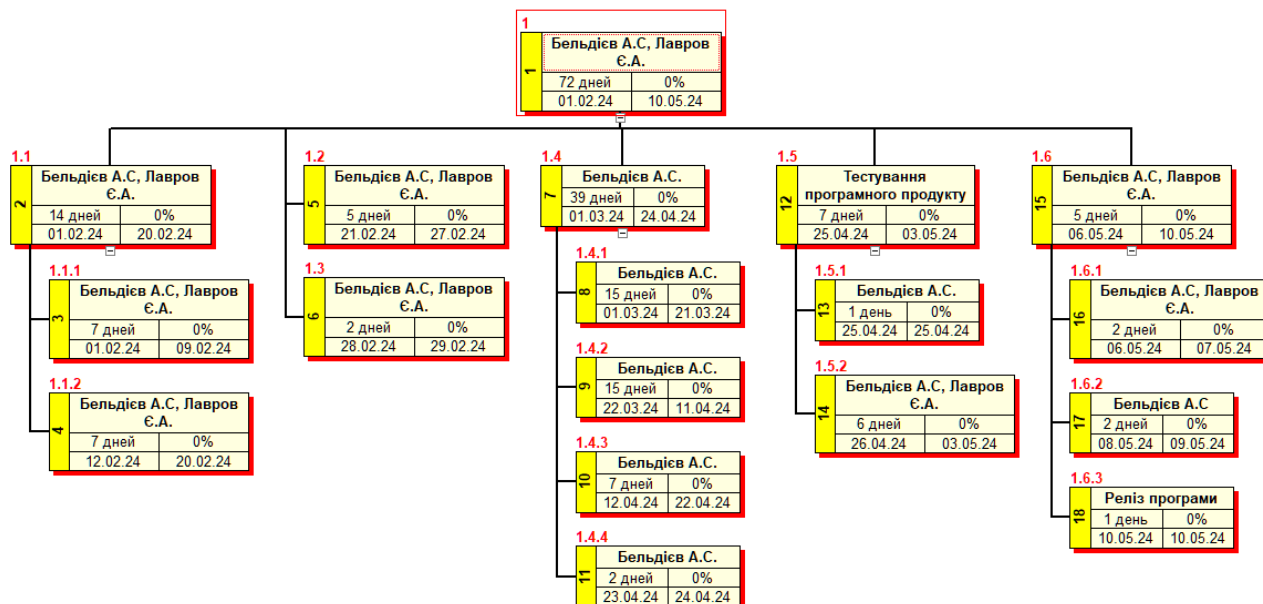


Рисунок А.2 – OBS-структура проекту

Таблиця Б.2 – Матриця відповідальності

WBS/OBS	Бельдієв А.С.	Лавров Є.А.
Інформаційна технологія інтелектуального аналізу хімічного складу сталі для підтримки прийняття рішень оператора спектральної лабораторії ливарного виробництва		
<b>1. Дослідження предметної області</b>		
1.1. Аналіз предметної області	+	+
1.2. Огляд підходів до задач класифікації	+	+
<b>2. Проектування інформаційної технології</b>	+	+

<b>3. Розробка методів інтеграції програмного модулю</b>	+	+
<b>4. Розробка модулів програмного продукту</b>		
4.1. Розробка модулю імпорту з вікна вимірів	+	
4.2. Розробка модулю класифікації	+	
4.3. Розробка модулю перенавчання	+	
4.4. Апробація моделей на реальних даних	+	
<b>5. Тестування програмного продукту</b>		
5.1. Alpha-тестування	+	
5.2. Beta-тестування	+	+
<b>6. Впровадження в дію</b>		
6.1. Перевірка коректності роботи	+	+
6.2. Написання супровідної документації	+	

**Діаграма Ганта.** Потрібна для розробки плану проекту і формування розкладу. Дозволяє отримати графічне представлення об'єму роботи та їх тривалості з урахуванням календарних вихідних та святкових днів. Діаграму Ганта можна побачити на рисунку А.3 нижче.

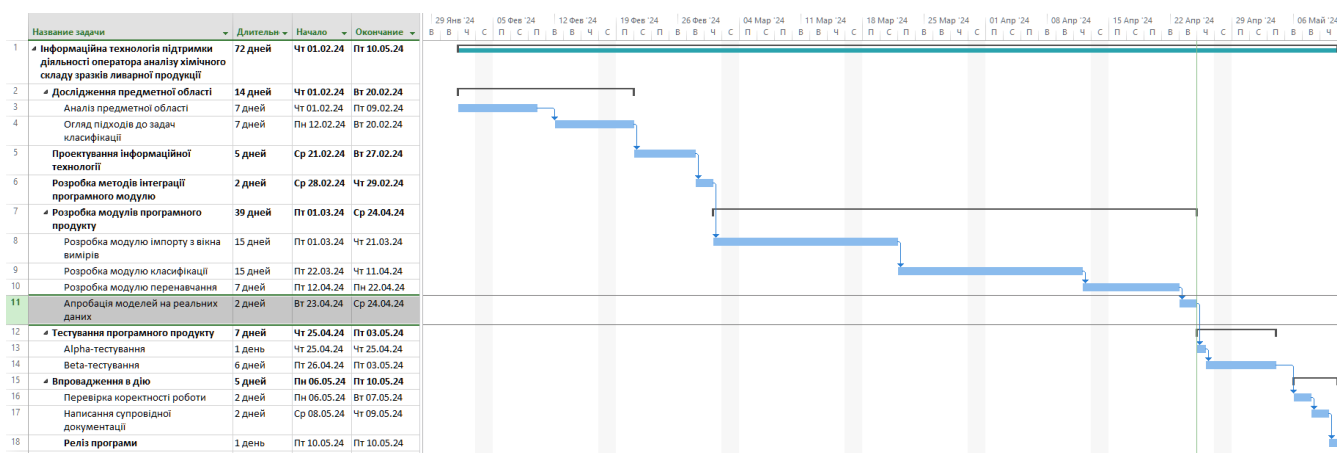


Рисунок А.2 – Діаграма Ганта

**Управління ризиками проекту.** При роботі над проектом слід брати до уваги можливі ризики, щоб розробити стратегію їх усунення. Тип ризику залежить від його критичності та впливає на стратегію реагування на нього. Далі оцінюється кількість та якість ризиків. Нижче можна побачити шкалу оцінювання ризиків за ймовірністю та впливом на проект (таблиця А.3). Саме за нею виконується оцінка наслідків впливу на проект.

Таблиця А.3 – Шкала оцінювання ризиків за ймовірністю виникнення та величиною впливу

Оцінка	Ймовірність виникнення	Вплив ризику	Тип ризику
1	Низька	Низький	Прийнятні
2	Середня	Середній	Виправдані
3	Висока	Високий	Недопустимі

Можна сформувати матрицю ймовірності та впливу, куди розміщуються власне ризики. Помноживши рядок на стовпець отримується ранг, який визначає тип конкретного ризику (таблиця А.4). Зеленим кольором на матриці позначають прийнятні ризики, жовтим – виправдані, а червоним – недопустимі.

Таблиця А.4 – Матриця ймовірності та впливу

Ймовірність ризикау (Й)	Вплив загрози				
	Дуже малий	Малий	Середній	Великий	Дуже великий
	0.05	0.1	0.2	0.4	0.8
0.9					
0.7		R4(0,07)			
0.5					
0.3					
0.1	R6(0,005),		R1(0,02) R3(0,02)		R2(0,08) R5(0,08),

У таблиці А.5 можна побачити розподіл номерів ризиків за рівнем. У таблиці Б.6 розроблено заходи реагування на виявлені ризики проекту на основі попередньо представлених даних.

Таблиця А.5 – Шкала оцінювання за рівнем ризику

№	Назва	Межі	Ризики, які входять(номера)
1	Прийнятні	$0.005 \leq R \leq 0.05$	1,3
2	Виправдані	$0,05 < R \leq 0,14$	4
3	Недопустимі	$0,14 < R \leq 0,72$	2,5

Таблиця А.6 – Ризики та стратегії реагування

ID	Статус ризику	Опис ризику	Ймовірність виникнення	Вплив ризику	Ранг ризику	План А	Тип стратегії реагування	План Б
RS_1	Відкритий	Нестабільна робота на етапі релізу	Низька	Середній	0,02	1. Перевірити роботоздатність на різних комп'ютерах або віртуальних машинах	Зменшення	Опитати працівників щодо дій, які вони виконували перед появою помилки. Знайти несправність у відповідному програмному модулі шляхом відтворення помилки
RS_2	Відкритий	Низька точність класифікації зразків	Низька	Високий	0,08	Змінити параметри класифікаторів, підібрати більш оптимальні. Зробити повторний аналіз і обробку датасету перед тренуванням	Зменшення	Домовитися з замовником про надання додаткових даних про зразки для збільшення датасету

## Продовження таблиці А.6 – Ризики та стратегії реагування

RS_3	Відкритий	Помилки в роботі модулю імпорту з вікна вимірювань	Низька	Середній	0,02	Перевірити роботу на інших комп'ютерах, щоб переконатися, що проблема не в апаратному забезпеченні	Зменшення	У разі неправильного розпізнавання тексту – змінити методи обробки зображення перед отриманням тексту
RS_4	Відкритий	Нестача оперативної пам'яті заводського комп'ютера для перенавчання	Висока	Низька	0,07	Виконати перенавчання на іншому заводському комп'ютері з більшою кількістю оперативної пам'яті	Зменшення	Виконати перенавчання на сервері підприємства
RS_5	Відкритий	Низька швидкість роботи програми на заводському комп'ютері	Низька	Високий	0,08	Використовувати менш трудомісні алгоритми модулів	Ухилення	Оптимізувати програмний код, робити рев'ю
RS_6	Відкритий	Операторам складно опанувати модуль	Низька	Низька	0,005	Написати зрозумілу супровідну операцію	Ухилення	Змінити компоновання кнопок інтерфейсу

## ДОДАТОК Б. Копія публікації

СЕКЦІЯ 2: Інформаційні технології проектування

ІМА :: 2024

### Інформаційна технологія інтелектуального аналізу хімічного складу сталі для спектральної лабораторії ливарного виробництва

Бельдієв А. С., студент ІТ.м-21н; Лавров Є. А., професор

Сумський державний університет, м. Суми, Україна

Вступ. Ливарне виробництво є однією з фундаментальних галузей, яка потребує високої точності контролю за якістю продукції. Одним із критеріїв якості є відповідність хімічного складу сталі до визначених державних та міжнародних стандартів. За визначення складу відповідає спектральна лабораторія, яка за допомогою спектрального аналізатора отримує кількість кожного хімічного елемента у відсотковому значенні. Розроблена нами інформаційна технологія «Flare» [1, 2] підтримки діяльності оператора спектральної лабораторії, на жаль, має ряд обмежень, оскільки орієнтована на традиційні методи аналізу якості типу “порівняння і-го параметру з нормою”, що ускладнює можливість оперативної класифікації зразків, що поступають.

Постановка задачі. Для умов конкретного ливарного виробництва визначимо:  $N$  – число діагностичних ознак;  $s_i$  – кількість упорядкованих оцінок складу сталі на шкалі  $i$ -ї діагностичної ознаки;  $X_i = \{X_{i1}, X_{i2}, \dots, X_{is_i}\}$  – безліч оцінок на шкалі  $i$ -ї ознаки;  $Q$  – кількість класів марок сталі ( $P_1, P_2, \dots, P_Q$ ), до яких можуть належати зразки. Задача полягає в розробці технології, орієнтованої на інтелектуальний аналіз даних про зразки ливарної продукції, накопичених в базі даних лабораторії.

Результати. Інформаційна технологія забезпечує:

- отримання моделі хімічного складу зразка з програмного забезпечення спектрального аналізатора;
- оперативну класифікацію нових зразків за хімічним складом та присвоєння їм марок сталі.

Реалізовано підтримку двох можливих режимів ідентифікації зразків:

Прийняття рішення оператором (за методологією «Flare»)

Класифікація, що проводиться за допомогою попередньо навчених з використанням накопичених даних моделей-класифікаторів, побудованих за методологією машинного навчання.

Принцип роботи інформаційної технології продемонстровано на рис.1.

## ДОДАТОК В. Копія акту впровадження


  
 Затверджую  
 Головний інженер ДП «ОБ та ВТ»  
 Моляков В.І.  
 « 30 09 2024 р.  
 АКТ

### Впровадження в роботу

### Центральної заводської лабораторії ДП «Завод ОБ та ВТ» результатів наукової роботи

студента групи ІТ.м-21н Сумського державного університету

Бельдієва Андрія Сергійовича

на тему

«Інформаційна технологія інтелектуального аналізу хімічного складу сталі для підтримки прийняття рішень оператора спектральної лабораторії ливарного виробництва»

Складений 8 травня 2024 р. комісією у складі:

Голова комісії:

*Начальник центральної заводської лабораторії Пушкар О.О.*

Члени комісії:

1. *Інженер спектрального аналізу центральної заводської лабораторії Бельдієва О.П.*
2. *Інженер спектрального аналізу центральної заводської лабораторії Кумеда О.І.*

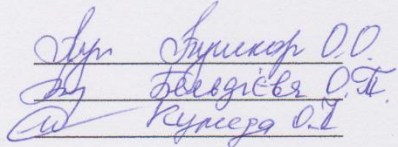
8 травня 2024 р. комісія провела роботу з визначення впровадження результатів Бельдієва А.С. в робочий процес центральної заводської лабораторії.

### Результати роботи комісії

1. У лабораторію спектрального аналізу центральної заводської лабораторії передано оновлену комп'ютерну програму «Flare».
2. Програма використовується для класифікації зразків за маркою сталі, що прискорює виробничий процес та зменшує навантаження на операторів.

Голова комісії

Члени комісії





## ДОДАТОК Г. Лістинг програми

**Файл Config.java:**

```

package Flare.Modules;

import java.awt.*;
import java.io.Serializable;
import lombok.*;

@Data
@Builder
@Getter

public class Config implements Serializable {

    private boolean
dateCheckBox, dateCheckBox1, datePeriodCheckBox, qualityCheckBox, bloo
mCheckBox, seriaCheckBox, samplesPerStepCheckBox;
    private Color chemsButton;
    String
CDev, SiDev, MnDev, PDev, SDev, CrDev, NiDev, MoDev, CuDev, AlDev, VDev, WDev
, TiDev;
    String
CDevAuto, SiDevAuto, MnDevAuto, PDevAuto, SDevAuto, CrDevAuto, NiDevAuto
, MoDevAuto, CuDevAuto, AlDevAuto, VDevAuto, WDevAuto, TiDevAuto;
    String
CStepAuto, SiStepAuto, MnStepAuto, PStepAuto, SStepAuto, CrStepAuto, NiS
tepAuto, MoStepAuto, CuStepAuto, AlStepAuto, VStepAuto, WStepAuto, TiSte
pAuto, samplesPerStep;
    String modelName;
}

```

**Файл MSample.java:**

```

package Flare;
import java.io.Serializable;
import lombok.*;

@Data
@Builder
public class MSample implements Serializable {
    private String date, time, sampleNo, quality, bloom,
codes, seria, C, Si, Mn, P, S, Cr, Ni, Mo, Cu, Al, V, W, Ti;
}

```

**Файл ClassifierInfo.java:**

```

package Flare.Modules;

import java.io.Serializable;
import java.util.List;
public class ClassifierInfo implements Serializable {
    private final String classifierName;
    private final double learningRate;
    private final double momentum;
    private final int epochs;
    private final double percentage;
    private final List<String> qualities;
    ClassifierInfo(String classifierName, double learningRate,
double momentum, int epochs, double percentage, List<String>
qualities) {
        this.classifierName = classifierName;
        this.learningRate = learningRate;
        this.momentum = momentum;
        this.epochs = epochs;
        this.percentage = percentage;
        this.qualities = qualities;
    }
    public String getClassifierName() {
        return classifierName;
    }
    public double getLearningRate() {
        return learningRate;
    }
    public double getMomentum() {
        return momentum;
    }
    public int getEpochs() {
        return epochs;
    }
    public double getPercentage() {
        return percentage;
    }
    public List<String> getQualities() {
        return qualities;
    }
}

```

**Файл Program.java:**

```

package Flare;
import Flare.Modules.ComplexSearch;
import Flare.Modules.Config;
import Flare.Modules.ImportFromExcel;

```

```

import javax.swing.*;
import javax.swing.filechooser.FileNameExtensionFilter;
import java.awt.*;
import java.awt.event.*;
import java.io.File;
import java.io.IOException;

public class Program {

    public static void main(String[] args) {
        String versionInfo = "v. 1.2";
        JFrame frame = new JFrame("Flare");
        JFileChooser fileChooser = new JFileChooser();
        Controller controller = new Controller();
        ComplexSearch complexSearcher = new ComplexSearch();
        JButton OpenFile = new JButton("Импорт из Excel");
        JButton SaveTableButton = new JButton("Сохранить");
        JButton deleteButton = new JButton("Удалить");
        JButton addButton = new JButton("Добавить");
        JButton classifyButton = new JButton("Определить марку");
        JButton classifySettingsButton = new JButton("Определить
марку");

        JButton clearAddingQueueButton = new JButton("Удалить
рядок");
        JButton insertAddingQueueButton = new JButton("Добавить
рядок");
        deleteButton.setToolTipText("Удаляет выделенный элемент (-
ы)");

        Icon searchIcon = new ImageIcon("icons/search.png");
        Icon settingsIcon = new ImageIcon("icons/settings.png");
        Icon refreshIcon = new ImageIcon("icons/refresh.png");
        Icon MrIcon = new ImageIcon("icons/Mr.png");
        Icon mRIcon = new ImageIcon("icons/Rm.png");
        Icon autoSearchIcon = new
ImageIcon("icons/autosearch.png");
        Icon searchNextIcon = new ImageIcon("icons/arrow.png");
        Icon searchStopIcon = new ImageIcon("icons/cross.png");
        Icon autoSearchParamsIcon = new
ImageIcon("icons/autosearchparams.png");
        Icon searchInfoIcon = new ImageIcon("icons/info.png");
        Icon changeStorageIcon = new ImageIcon("icons/clean.png");
        Icon bookmarkIcon = new ImageIcon("icons/bookmarks.png");
        Icon addBookmarkIcon = new
ImageIcon("icons/addbookmark.png");
        Icon deleteBookmarkIcon = new
ImageIcon("icons/deletebookmark.png");
        Icon classifierIcon = new
ImageIcon("icons/classifier.png");
        JButton searchButton = new JButton();
        JButton autoSearchButton = new JButton();
    }
}

```

```

JButton searchNextButton = new JButton();
JButton searchStopButton = new JButton();
JButton searchParamsButton = new JButton();
JButton searchInfoButton = new JButton();
JButton changeStorageButton = new JButton();
JButton addToBookmarkButton = new JButton();
JButton deleteBookmarkButton = new JButton();
JButton bookmarkButton = new JButton();
JLabel versionInfoSubscr = new JLabel(versionInfo);
JLabel autoSearchSubscr = new JLabel("Автопоиск");
JLabel tableOpsSubscr = new JLabel("Таблица");
JLabel manualSearchSubscr = new JLabel("Ручной поиск");
JCheckBox samplesPerStepCheckBox = new JCheckBox("обр. за
шаг");

samplesPerStepCheckBox.setSelected(controller.config.isSamplesPerS
tepCheckBox());
    samplesPerStepCheckBox.setFont(new
Font("Arial", Font.PLAIN, 14));
    autoSearchSubscr.setForeground(Color.GRAY);
    autoSearchSubscr.setFont(new Font("Arial", Font.PLAIN, 14));
    autoSearchSubscr.setForeground(Color.GRAY);
    manualSearchSubscr.setFont(new
Font("Arial", Font.PLAIN, 14));
    manualSearchSubscr.setForeground(Color.GRAY);
    tableOpsSubscr.setFont(new Font("Arial", Font.PLAIN, 14));
    tableOpsSubscr.setForeground(Color.GRAY);
    searchButton.setIcon(searchIcon);
    autoSearchButton.setIcon(autoSearchIcon);
    searchNextButton.setIcon(searchNextIcon);
    searchStopButton.setIcon(searchStopIcon);
    searchParamsButton.setIcon(autoSearchParamsIcon);
    autoSearchButton.setIcon(autoSearchIcon);
    searchInfoButton.setIcon(searchInfoIcon);
    changeStorageButton.setIcon(changeStorageIcon);
    bookmarkButton.setIcon(bookmarkIcon);
    addToBookmarkButton.setIcon(addBookmarkIcon);
    deleteBookmarkButton.setIcon(deleteBookmarkIcon);
    classifySettingsButton.setIcon(classifierIcon);
    changeStorageButton.setToolTipText("Перекидывает образец в
другое хранилище");
    JButton searchSettingsButton = new JButton();
    searchSettingsButton.setIcon(settingsIcon);
    searchSettingsButton.setToolTipText("Расширенные параметры
поиска");
    JTextField searchBox = new JTextField();
    JTextField stepBox = new
JTextField(controller.config.getSamplesPerStep());
    JButton refreshButton = new JButton();
    JButton changeSampleTypeButton = new JButton();
    changeSampleTypeButton.setIcon(MrIcon);
    changeSampleTypeButton.setToolTipText("Для отображения

```

```
выбраны маркировочные образцы. Кликните, чтобы отображать
разрывные");
    refreshButton.setIcon(refreshIcon);
    refreshButton.setToolTipText("Заново вывести данные в
таблицу");
    searchBox.setToolTipText("Введите номер образца для
поиска");
    JScrollPane output = new
JScrollPane(controller.createOutputTable());
    versionInfoSubscr.setBounds(10, 650, 150, 30);
    output.setBounds(0, 0, 1100, 600);
    OpenFile.setBounds(315, 620, 150, 30);
    deleteButton.setBounds(515, 620, 150, 30);
    SaveTableButton.setBounds(115, 620, 150, 30);
    searchBox.setBounds(1105, 40, 136, 30);
    autoSearchSubscr.setBounds(1140, 150, 150, 30);
    manualSearchSubscr.setBounds(1130, 0, 150, 30);
    tableOpsSubscr.setBounds(1140, 330, 150, 30);
    autoSearchButton.setBounds(1105, 190, 30, 30);
    searchButton.setBounds(1105, 70, 30, 30);
    searchSettingsButton.setBounds(1140, 70, 30, 30);
    refreshButton.setBounds(1175, 70, 30, 30);
    addButton.setBounds(715, 620, 150, 30);
    classifyButton.setBounds(915, 620, 150, 30);
    classifySettingsButton.setBounds(1070, 620, 30, 30);

    changeSampleTypeButton.setBounds(1210, 70, 30, 30);
    searchNextButton.setBounds(1140, 190, 60, 30);
    searchStopButton.setBounds(1205, 190, 30, 30);
    searchParamsButton.setBounds(1105, 225, 30, 30);
    changeStorageButton.setBounds(1210, 370, 30, 30);
    bookmarkButton.setBounds(1105, 370, 30, 30);
    addToBookmarkButton.setBounds(1140, 370, 30, 30);
    deleteBookmarkButton.setBounds(1175, 370, 30, 30);
    samplesPerStepCheckBox.setBounds(1100, 275, 100, 30);
    stepBox.setBounds(1210, 275, 20, 30);
    searchInfoButton.setBounds(1205, 225, 30, 30);
    autoSearchButton.setEnabled(false);
    searchNextButton.setEnabled(false);
    searchStopButton.setEnabled(false);
    searchInfoButton.setEnabled(false);
    frame.setSize(1265, 720);
    frame.setResizable(false);
    frame.getContentPane().setLayout(null);
    frame.setLocation(400, 150);
    frame.add(OpenFile);
    frame.add(deleteButton);
    frame.add(SaveTableButton);
    frame.add(output);
    frame.add(searchButton);
    frame.add(searchSettingsButton);
    frame.add(refreshButton);
```

```

frame.add(addButton);
frame.add(classifyButton);
frame.add(classifySettingsButton);

frame.add(changeSampleTypeButton);
frame.add(searchBox);
frame.add(versionInfoSubscr);
frame.add(autoSearchSubscr);
frame.add(manualSearchSubscr);
frame.add(tableOpsSubscr);
frame.add(autoSearchButton);
frame.add(searchNextButton);
frame.add(searchStopButton);
frame.add(searchParamsButton);
frame.add(searchInfoButton);
frame.add(changeStorageButton);
frame.add(samplesPerStepCheckBox);
frame.add(bookmarkButton);
frame.add(addToBookmarkButton);
frame.add(deleteBookmarkButton);
frame.add(stepBox);
frame.setVisible(true);

JDialog addframe = new JDialog(frame, "Очередь на
добавление", true);
JButton screenshotButton = new JButton("Скриншот");
JButton addSampleButton = new JButton();
Icon saveIcon = new ImageIcon("icons/save.png");
addframe.getContentPane().setLayout(null);
JScrollPane input = new
JScrollPane(controller.createInputTable());
addSampleButton.setIcon(saveIcon);
addSampleButton.setToolTipText("Записать образец. Его
принадлежность к разрывным или маркировочным определится
автоматически.");
input.setBounds(20, 0, 1100, 100);
addSampleButton.setBounds(1140, 0, 40, 40);
screenshotButton.setToolTipText("После нажатия на данную
кнопку разверните окно измерений. Flare запишет данные образца в
таблицу выше, где вы сможете его отредактировать и сохранить в
хранилище. Проверьте правильность полученной программой
информации");
screenshotButton.setBounds(20, 110, 150, 30);
clearAddingQueueButton.setBounds(220, 110, 150, 30);
insertAddingQueueButton.setBounds(420, 110, 150, 30);
addframe.setSize(1220, 200);
addframe.add(input);
addframe.add(addSampleButton);
addframe.add(screenshotButton);
addframe.add(clearAddingQueueButton);
addframe.add(insertAddingQueueButton);

```

```

        if(samplesPerStepCheckBox.isSelected()) {
            stepBox.setEnabled(true);
        }
        else
        {
            stepBox.setEnabled(false);
        }
    /*
        ОКНО РАСШИРЕННОГО ПОИСКА
    */
        JDialog Sframe = new JDialog(frame,"Расширенный
поиск",true);
        JButton enableChemsButton = new JButton("Хим. элементы");
        JButton advancedSearchButton = new JButton();
        Icon advancedSearchIcon = new
ImageIcon("icons/hiResSearch.png");
        advancedSearchButton.setIcon(advancedSearchIcon);
        JTextField firstDateBox = new JTextField();
        JTextField secondDateBox = new JTextField();
        JTextField qualityBox = new JTextField();
        JTextField bloomBox = new JTextField();
        JTextField seriaFromBox = new JTextField();
        JTextField seriaToBox = new JTextField();

        JTextField CBox = new JTextField();
        JTextField SiBox = new JTextField();
        JTextField MnBox = new JTextField();
        JTextField PBox = new JTextField();
        JTextField SBox = new JTextField();
        JTextField CrBox = new JTextField();
        JTextField NiBox = new JTextField();
        JTextField MoBox = new JTextField();
        JTextField CuBox = new JTextField();
        JTextField AlBox = new JTextField();
        JTextField VBox = new JTextField();
        JTextField WBox = new JTextField();
        JTextField TiBox = new JTextField();
        JTextField CDevBox = new
JTextField(controller.config.getCDev());
        JTextField SiDevBox = new
JTextField(controller.config.getSiDev());
        JTextField MnDevBox = new
JTextField(controller.config.getMnDev());
        JTextField PDevBox = new
JTextField(controller.config.getPDev());
        JTextField SDevBox = new
JTextField(controller.config.getSDev());
        JTextField CrDevBox = new
JTextField(controller.config.getCrDev());
        JTextField NiDevBox = new
JTextField(controller.config.getNiDev());
        JTextField MoDevBox = new

```

```

JTextField(controller.config.getMoDev());
    JTextField CuDevBox = new
JTextField(controller.config.getCuDev());
    JTextField AlDevBox = new
JTextField(controller.config.getAlDev());
    JTextField VDevBox = new
JTextField(controller.config.getVDev());
    JTextField WDevBox = new
JTextField(controller.config.getWDev());
    JTextField TiDevBox = new
JTextField(controller.config.getTiDev());

    JLabel dateSubscr = new JLabel("Дата:");
    JLabel qualitySubscr = new JLabel("Марка стали:");
    JLabel bloomSubscr = new JLabel("Номер блюма:");
    JLabel seriaSubscr = new JLabel("Серия:");
    JLabel seriaFromSubscr = new JLabel("из");

    JLabel chemSubscr = new JLabel("C          Si
Mn          P          S          Cr          Ni
Mo          Cu          Al          V          W
Ti");
    JLabel chemLineSubscr = new JLabel("значения");
    chemLineSubscr.setToolTipText("Значения хим. элементов");
    JLabel deviationLineSubscr = new JLabel("отклонения");
    deviationLineSubscr.setToolTipText(" +- отклонение от
заданных значений хим. элементов при поиске конкретного образца");
    JCheckBox dateCheckBox = new
JCheckBox((controller.config.isDateCheckBox())?"Вкл.":"Выкл.",
controller.config.isDateCheckBox());//config
    JCheckBox qualityCheckBox = new
JCheckBox(controller.config.isQualityCheckBox()?"Вкл.":"Выкл.",
controller.config.isQualityCheckBox());//config
    JCheckBox bloomCheckBox = new
JCheckBox(controller.config.isBloomCheckBox()?"Вкл.":"Выкл.",
controller.config.isBloomCheckBox());//config
    JCheckBox seriaCheckBox = new
JCheckBox(controller.config.isSeriaCheckBox()?"Вкл.":"Выкл.", contr
oller.config.isSeriaCheckBox());//config
    JCheckBox datePeriodCheckBox = new JCheckBox("Период
(от/до)", controller.config.isDatePeriodCheckBox());//config

    enableChemsButton.setBounds(10, 140, 150, 30);

enableChemsButton.setBackground(controller.config.getChemsButton()
);
    advancedSearchButton.setBounds(760,80,50,50);
    firstDateBox.setBounds(10, 40, 150, 30);
    secondDateBox.setBounds(10, 80, 150, 30);
    qualityBox.setBounds(200, 40, 150, 30);
    bloomBox.setBounds(390, 40, 150, 30);
    seriaFromBox.setBounds(580, 40, 40, 30);

```



```
seriaToBox.setBounds(650, 40, 40, 30);

dateSubscr.setBounds(10,10,70,20);
qualitySubscr.setBounds(200,10,90,20);
bloomSubscr.setBounds(390, 10, 90, 20);
seriaSubscr.setBounds(580, 10, 90, 20);
seriaFromSubscr.setBounds(628, 43, 90, 20);
seriaFromSubscr.setFont(new Font("Arial",Font.PLAIN,12));
chemSubscr.setBounds(43, 190, 900, 20);
chemLineSubscr.setBounds(750, 220, 100, 20);
chemLineSubscr.setFont(new Font("Arial",Font.PLAIN,14));
deviationLineSubscr.setBounds(750, 250, 100, 20);
deviationLineSubscr.setFont(new
Font("Arial",Font.PLAIN,14));
CBox.setBounds(20, 220, 50, 20);
SiBox.setBounds(75, 220, 50, 20);
MnBox.setBounds(130, 220, 50, 20);
PBox.setBounds(185, 220, 50, 20);
SBox.setBounds(240, 220, 50, 20);
CrBox.setBounds(295, 220, 50, 20);
NiBox.setBounds(350, 220, 50, 20);
MoBox.setBounds(405, 220, 50, 20);
CuBox.setBounds(460, 220, 50, 20);
AlBox.setBounds(515, 220, 50, 20);
VBox.setBounds(570, 220, 50, 20);
WBox.setBounds(625, 220, 50, 20);
TiBox.setBounds(680, 220, 50, 20);
CDevBox.setBounds(20, 250, 50, 20);
SiDevBox.setBounds(75, 250, 50, 20);
MnDevBox.setBounds(130, 250, 50, 20);
PDevBox.setBounds(185, 250, 50, 20);
SDevBox.setBounds(240, 250, 50, 20);
CrDevBox.setBounds(295, 250, 50, 20);
NiDevBox.setBounds(350, 250, 50, 20);
MoDevBox.setBounds(405, 250, 50, 20);
CuDevBox.setBounds(460, 250, 50, 20);
AlDevBox.setBounds(515, 250, 50, 20);
VDevBox.setBounds(570, 250, 50, 20);
WDevBox.setBounds(625, 250, 50, 20);
TiDevBox.setBounds(680, 250, 50, 20);

datePeriodCheckBox.setBounds(10, 110, 120, 30);
dateCheckBox.setBounds(100, 5, 100, 30);
qualityCheckBox.setBounds(290, 5, 100, 30);
bloomCheckBox.setBounds(480, 5, 100, 30);
seriaCheckBox.setBounds(630, 5, 100, 30);

secondDateBox.setEnabled(false);
Sframe.setSize(800, 220);
Sframe.setResizable(false);
Sframe.setLocation(400, 150);
```

```
Sframe.add(enableChemsButton);  
Sframe.add(advancedSearchButton);
```

```
Sframe.add(dateSubscr);  
Sframe.add(qualitySubscr);  
Sframe.add(bloomSubscr);  
Sframe.add(seriaSubscr);  
Sframe.add(seriaFromSubscr);  
Sframe.add(chemSubscr);  
Sframe.add(chemLineSubscr);  
Sframe.add(deviationLineSubscr);  
Sframe.add(datePeriodCheckBox);  
Sframe.add(dateCheckBox);  
Sframe.add(qualityCheckBox);  
Sframe.add(bloomCheckBox);  
Sframe.add(seriaCheckBox);
```

```
Sframe.add(firstDateBox);  
Sframe.add(secondDateBox);  
Sframe.add(qualityBox);  
Sframe.add(bloomBox);  
Sframe.add(seriaFromBox);  
Sframe.add(seriaToBox);  
Sframe.add(CBox);  
Sframe.add(SiBox);  
Sframe.add(MnBox);  
Sframe.add(PBox);  
Sframe.add(SBox);  
Sframe.add(CrBox);  
Sframe.add(NiBox);  
Sframe.add(MoBox);  
Sframe.add(CuBox);  
Sframe.add(AlBox);  
Sframe.add(VBox);  
Sframe.add(WBox);  
Sframe.add(TiBox);  
Sframe.add(CDevBox);  
Sframe.add(SiDevBox);  
Sframe.add(MnDevBox);  
Sframe.add(PDevBox);  
Sframe.add(SDevBox);  
Sframe.add(CrDevBox);  
Sframe.add(NiDevBox);  
Sframe.add(MoDevBox);  
Sframe.add(CuDevBox);  
Sframe.add(AlDevBox);  
Sframe.add(VDevBox);  
Sframe.add(WDevBox);  
Sframe.add(TiDevBox);  
JPanel p = new JPanel() {  
    @Override
```

```

        public void paintComponent(Graphics g) {
            Graphics2D g2 = (Graphics2D) g;
            g2.setColor(Color.LIGHT_GRAY);
            Shape rect = new Rectangle(0, 0, 900, 180);
            g2.draw(rect);

        }
    };
    Sframe.getContentPane().add(p);

    JDialog paramsframe = new JDialog(frame, "Параметры
автопоиска", true);
    paramsframe.getContentPane().setLayout(null);
    JLabel chemSubscr1 = new JLabel(chemSubscr.getText());
    JLabel deviationLineSubscr1 = new
JLabel(deviationLineSubscr.getText());
    JLabel stepSubscr = new JLabel("шаг");
    JLabel dateSubscr1 = new JLabel("по");
    JCheckBox dateCheckBox1 = new JCheckBox("В пределах
даты");

    dateCheckBox1.setSelected(controller.config.isDateCheckBox1());
    deviationLineSubscr1.setFont(new
Font("Arial", Font.PLAIN, 14));
    stepSubscr.setFont(new Font("Arial", Font.PLAIN, 14));
    dateSubscr1.setFont(new Font("Arial", Font.PLAIN, 14));
    JTextField CDevBox1 = new
JTextField(controller.config.getCDevAuto());
    JTextField SiDevBox1 = new
JTextField(controller.config.getSiDevAuto());
    JTextField MnDevBox1 = new
JTextField(controller.config.getMnDevAuto());
    JTextField PDevBox1 = new
JTextField(controller.config.getPDevAuto());
    JTextField SDevBox1 = new
JTextField(controller.config.getSDevAuto());
    JTextField CrDevBox1 = new
JTextField(controller.config.getCrDevAuto());
    JTextField NiDevBox1 = new
JTextField(controller.config.getNiDevAuto());
    JTextField MoDevBox1 = new
JTextField(controller.config.getMoDevAuto());
    JTextField CuDevBox1 = new
JTextField(controller.config.getCuDevAuto());
    JTextField AlDevBox1 = new
JTextField(controller.config.getAlDevAuto());
    JTextField VDevBox1 = new
JTextField(controller.config.getVDevAuto());
    JTextField WDevBox1 = new
JTextField(controller.config.getWDevAuto());
    JTextField TiDevBox1 = new
JTextField(controller.config.getTiDevAuto());

```

```
        JTextField CStepBox = new
JTextField(controller.config.getCStepAuto());
        JTextField SiStepBox = new
JTextField(controller.config.getSiStepAuto());
        JTextField MnStepBox = new
JTextField(controller.config.getMnStepAuto());
        JTextField PStepBox = new
JTextField(controller.config.getPStepAuto());
        JTextField SStepBox = new
JTextField(controller.config.getSStepAuto());
        JTextField CrStepBox = new
JTextField(controller.config.getCrStepAuto());
        JTextField NiStepBox = new
JTextField(controller.config.getNiStepAuto());
        JTextField MoStepBox = new
JTextField(controller.config.getMoStepAuto());
        JTextField CuStepBox = new
JTextField(controller.config.getCuStepAuto());
        JTextField AlStepBox = new
JTextField(controller.config.getAlStepAuto());
        JTextField VStepBox = new
JTextField(controller.config.getVStepAuto());
        JTextField WStepBox = new
JTextField(controller.config.getWStepAuto());
        JTextField TiStepBox = new
JTextField(controller.config.getTiStepAuto());

        JTextField firstDateBox1 = new JTextField();
        JTextField secondDateBox1 = new JTextField();

        chemSubscr1.setBounds(43, 10, 900, 20);
        deviationLineSubscr1.setBounds(750, 30, 100, 20);
        stepSubscr.setBounds(750, 73, 900, 20);
        paramsframe.setSize(850, 200);
        CDevBox1.setBounds(20,30 , 50, 20);
        SiDevBox1.setBounds(75, 30, 50, 20);
        MnDevBox1.setBounds(130, 30, 50, 20);
        PDevBox1.setBounds(185, 30, 50, 20);
        SDevBox1.setBounds(240, 30, 50, 20);
        CrDevBox1.setBounds(295, 30, 50, 20);
        NiDevBox1.setBounds(350, 30, 50, 20);
        MoDevBox1.setBounds(405, 30, 50, 20);
        CuDevBox1.setBounds(460, 30, 50, 20);
        AlDevBox1.setBounds(515, 30, 50, 20);
        VDevBox1.setBounds(570, 30, 50, 20);
        WDevBox1.setBounds(625, 30, 50, 20);
        TiDevBox1.setBounds(680, 30, 50, 20);
        CStepBox.setBounds(20,70 , 50, 20);
        SiStepBox.setBounds(75, 70, 50, 20);
        MnStepBox.setBounds(130, 70, 50, 20);
        PStepBox.setBounds(185, 70, 50, 20);
        SStepBox.setBounds(240, 70, 50, 20);
```

```
CrStepBox.setBounds(295, 70, 50, 20);
NiStepBox.setBounds(350, 70, 50, 20);
MoStepBox.setBounds(405, 70, 50, 20);
CuStepBox.setBounds(460, 70, 50, 20);
AlStepBox.setBounds(515, 70, 50, 20);
VStepBox.setBounds(570, 70, 50, 20);
WStepBox.setBounds(625, 70, 50, 20);
TiStepBox.setBounds(680, 70, 50, 20);
firstDateBox1.setBounds(20,120,150,30);
secondDateBox1.setBounds(200,120,150,30);
dateSubscr1.setBounds(175,120,20,30);
dateCheckBox1.setBounds(360,120,200,30);
```

```
paramsframe.add(chemSubscr1);
paramsframe.add(deviationLineSubscr1);
paramsframe.add(stepSubscr);
paramsframe.add(TiDevBox1);
paramsframe.add(CDevBox1);
paramsframe.add(SiDevBox1);
paramsframe.add(MnDevBox1);
paramsframe.add(PDevBox1);
paramsframe.add(SDevBox1);
paramsframe.add(CrDevBox1);
paramsframe.add(NiDevBox1);
paramsframe.add(MoDevBox1);
paramsframe.add(CuDevBox1);
paramsframe.add(AlDevBox1);
paramsframe.add(VDevBox1);
paramsframe.add(WDevBox1);
paramsframe.add(TiDevBox1);
```

```
paramsframe.add(TiStepBox);
paramsframe.add(CStepBox);
paramsframe.add(SiStepBox);
paramsframe.add(MnStepBox);
paramsframe.add(PStepBox);
paramsframe.add(SStepBox);
paramsframe.add(CrStepBox);
paramsframe.add(NiStepBox);
paramsframe.add(MoStepBox);
paramsframe.add(CuStepBox);
paramsframe.add(AlStepBox);
paramsframe.add(VStepBox);
paramsframe.add(WStepBox);
paramsframe.add(TiStepBox);
paramsframe.add(firstDateBox1);
paramsframe.add(secondDateBox1);
paramsframe.add(dateSubscr1);
paramsframe.add(dateCheckBox1);
```

```
JFrame classifyResultsFrame = new JFrame("Результат
определения марки стали");
```

```

        JScrollPane classify_table = new
JScrollPane(controller.createClassifyTable());
        JButton moveQualityToSampleButton = new JButton("Выбрать
марку");
        moveQualityToSampleButton.setToolTipText("Переносит марку
стали в рядок образца");
        classifyResultsFrame.setSize(300,200);
        classifyResultsFrame.getContentPane().setLayout(null);
        classify_table.setBounds(0, 0, 300, 100);
        moveQualityToSampleButton.setBounds(20, 110, 150, 30);
        classifyResultsFrame.add(classify_table);
        classifyResultsFrame.add(moveQualityToSampleButton);

        JFrame classifySettingsFrame = new JFrame("Настройка
классификатора");
        JScrollPane classifierInfo_table = new
JScrollPane(controller.createClassifierInfoTable());
        JButton trainNetButton = new JButton("Новая модель");
        JButton changeClassifierButton = new JButton("Выбрать
модель");
        JButton deleteClassifierButton = new JButton("Удалить
модель");
        JComboBox classifiersBox = new JComboBox();
        trainNetButton.setToolTipText("Переобучает нейронную сеть
на текущем хранилище маркировочных образцов");
        classifySettingsFrame.setSize(330,210);
        classifySettingsFrame.getContentPane().setLayout(null);
        classifierInfo_table.setBounds(5, 0, 305, 105);
        trainNetButton.setBounds(5, 130, 150, 30);
        changeClassifierButton.setBounds(160, 130, 150, 30);
        deleteClassifierButton.setBounds(5, 100, 150, 30);
        classifiersBox.setBounds(160,105,150,20);
        classifySettingsFrame.add(classifierInfo_table);
        classifySettingsFrame.add(trainNetButton);
        classifySettingsFrame.add(changeClassifierButton);
        classifySettingsFrame.add(deleteClassifierButton);
        classifySettingsFrame.add(classifiersBox);

        if(dateCheckBox.isSelected()) {
            dateCheckBox.setForeground(new Color(4, 125, 0));
            dateCheckBox.setText("Вкл.");
        }
        else
        {
            dateCheckBox.setText("Выкл.");
            dateCheckBox.setForeground(Color.GRAY);
        }
        firstDateBox.setEnabled(dateCheckBox.isSelected());
        if(datePeriodCheckBox.isSelected())
            secondDateBox.setEnabled(dateCheckBox.isSelected());
        if(qualityCheckBox.isSelected()) {
            qualityCheckBox.setForeground(new Color(4, 125, 0));

```

```

        qualityCheckBox.setText("Вкл.");
    }
    else
    {
        qualityCheckBox.setText("Выкл.");
        qualityCheckBox.setForeground(Color.GRAY);
    }
    qualityBox.setEnabled(qualityCheckBox.isSelected());
    if(bloomCheckBox.isSelected()) {
        bloomCheckBox.setForeground(new Color(4, 125, 0));
        bloomCheckBox.setText("Вкл.");
    }
    else
    {
        bloomCheckBox.setText("Выкл.");
        bloomCheckBox.setForeground(Color.GRAY);
    }
    bloomBox.setEnabled(bloomCheckBox.isSelected());
    if(seriaCheckBox.isSelected()) {
        seriaCheckBox.setForeground(new Color(4, 125, 0));
        seriaCheckBox.setText("Вкл.");
    }
    else
    {
        seriaCheckBox.setText("Выкл.");
        seriaCheckBox.setForeground(Color.GRAY);
    }
    seriaToBox.setEnabled(seriaCheckBox.isSelected());
    seriaFromBox.setEnabled(seriaCheckBox.isSelected());
    datePeriodCheckBox.addActionListener(actionListener ->
    {
        if(dateCheckBox.isSelected())

secondDateBox.setEnabled(datePeriodCheckBox.isSelected());
    });
    dateCheckBox.addActionListener(actionListener ->
    {
        if(dateCheckBox.isSelected()) {
            dateCheckBox.setForeground(new Color(4, 125, 0));
            dateCheckBox.setText("Вкл.");
        }
        else
        {
            dateCheckBox.setText("Выкл.");
            dateCheckBox.setForeground(Color.GRAY);
        }
        firstDateBox.setEnabled(dateCheckBox.isSelected());
        if(datePeriodCheckBox.isSelected())
            secondDateBox.setEnabled(dateCheckBox.isSelected());
    });
    qualityCheckBox.addActionListener(actionListener ->
    {

```

```

    if(qualityCheckBox.isSelected()) {
        qualityCheckBox.setForeground(new Color(4, 125,
0));
        qualityCheckBox.setText("Вкл.");
    }
    else
    {
        qualityCheckBox.setText("Выкл.");
        qualityCheckBox.setForeground(Color.GRAY);
    }
    qualityBox.setEnabled(qualityCheckBox.isSelected());
});
bloomCheckBox.addActionListener(actionListener ->
{
    if(bloomCheckBox.isSelected()) {
        bloomCheckBox.setForeground(new Color(4, 125, 0));
        bloomCheckBox.setText("Вкл.");
    }
    else
    {
        bloomCheckBox.setText("Выкл.");
        bloomCheckBox.setForeground(Color.GRAY);
    }
    bloomBox.setEnabled(bloomCheckBox.isSelected());
});
seriaCheckBox.addActionListener(actionListener ->
{
    if(seriaCheckBox.isSelected()) {
        seriaCheckBox.setForeground(new Color(4, 125, 0));
        seriaCheckBox.setText("Вкл.");
    }
    else
    {
        seriaCheckBox.setText("Выкл.");
        seriaCheckBox.setForeground(Color.GRAY);
    }
    seriaToBox.setEnabled(seriaCheckBox.isSelected());
    seriaFromBox.setEnabled(seriaCheckBox.isSelected());
});
enableChemsButton.addActionListener(actionListener ->
{
    if (enableChemsButton.getBackground().getGreen() !=
255)
    {
        enableChemsButton.setBackground(new Color(130,
255, 130));
        Sframe.setSize(850, 350);
    }
    else {
        enableChemsButton.setBackground(null);
        Sframe.setSize(850, 220);
    }
}

```



```

});
advancedSearchButton.addActionListener (actionListener ->
{

if (dateCheckBox.isSelected() &&datePeriodCheckBox.isSelected())

controller.print (complexSearcher.searchByDate (firstDateBox.getText
(),secondDateBox.getText(),controller.getTableValue()));
    else if (dateCheckBox.isSelected())

controller.print (complexSearcher.searchByDate (firstDateBox.getText
(),null,controller.getTableValue()));
        if (qualityCheckBox.isSelected())
controller.print (complexSearcher.searchByQuality (qualityBox.getTex
t(),controller.getTableValue()));
            if (bloomCheckBox.isSelected())
controller.print (complexSearcher.searchByBloom (bloomBox.getText(),
controller.getTableValue()));
                if (seriaCheckBox.isSelected())
controller.print (complexSearcher.searchBySeria (seriaFromBox.getTex
t(),seriaToBox.getText(), controller.getTableValue()));
                    if (enableChemsButton.getBackground().getGreen() ==255)
                    {
                        controller.config.setCDev (CDevBox.getText());
                        controller.config.setSiDev (SiDevBox.getText());
                        controller.config.setMnDev (MnDevBox.getText());
                        controller.config.setPDev (PDevBox.getText());
                        controller.config.setSDev (SDevBox.getText());
                        controller.config.setCrDev (CrDevBox.getText());
                        controller.config.setNiDev (NiDevBox.getText());
                        controller.config.setMoDev (MoDevBox.getText());
                        controller.config.setCuDev (CuDevBox.getText());
                        controller.config.setAlDev (AlDevBox.getText());
                        controller.config.setVDev (VDevBox.getText());
                        controller.config.setWDev (WDevBox.getText());
                        controller.config.setTiDev (TiDevBox.getText());
                        controller.print (complexSearcher.searchByChems (
                            CBox.getText(),
                            SiBox.getText(),
                            MnBox.getText(),
                            PBox.getText(),
                            SBox.getText(),
                            CrBox.getText(),
                            NiBox.getText(),
                            MoBox.getText(),
                            CuBox.getText(),
                            AlBox.getText(),
                            VBox.getText(),
                            WBox .getText(),
                            TiBox.getText(),
                            controller.config,controller.getTableValue()));
                    }
}

```

```

        SaveTableButton.setEnabled(false);
        Sframe.dispose();
    });
    samplesPerStepCheckBox.addActionListener(actionListener ->
    {

if(samplesPerStepCheckBox.isSelected()) stepBox.setEnabled(true);
        else stepBox.setEnabled(false);
    });
    autoSearchButton.addActionListener(actionListener ->
    {
    if(controller.getOutput_table().getSelectedRow() != -1)
    {
        controller.startAutosearch();
        searchInfoButton.setEnabled(true);

if(!controller.getSearcher().isFindSuccess()) searchNextButton.setE
nabled(true);
        }
        else JOptionPane.showMessageDialog(frame, "Выберите
разрывной образец", "Ошибка", JOptionPane.ERROR_MESSAGE);
    });
    searchInfoButton.addActionListener(actionListener ->
    {

JOptionPane.showMessageDialog(frame, controller.getSearcher().getRe
port(), "Завершено", JOptionPane.INFORMATION_MESSAGE);
        });
    searchNextButton.addActionListener(actionListener ->
    {

controller.config.setSamplesPerStep(stepBox.getText());

controller.config.setSamplesPerStepCheckBox(samplesPerStepCheckBox
.isSelected());

controller.config.setSamplesPerStep(stepBox.getText());

controller.startStepsearch(firstDateBox1.getText(), secondDateBox1.
getText());
        searchStopButton.setEnabled(true);
    });
    searchStopButton.addActionListener(actionListener ->
    {
        controller.stopStepsearch();
        searchStopButton.setEnabled(false);
        controller.print(controller.pullData("MSamples.dat"));
    });
    OpenFile.addActionListener(actionListener ->
    {
        File ExcelFile;
        FileNameExtensionFilter filter = new

```

```

FileNameExtensionFilter(
    "Файлы Excel (.xls)", "xls");
fileChooser.setFileFilter(filter);
fileChooser.setDialogTitle("Выбор расположения");
if (fileChooser.showOpenDialog(null) ==
JFileChooser.APPROVE_OPTION ) {
    ExcelFile = fileChooser.getSelectedFile();
    try {

controller.pushData(ImportFromExcel.GetExcelData(ExcelFile));

if(changeSampleTypeButton.getIcon().equals(MrIcon))
controller.print(controller.pullData("MSamples.dat"));
        else
controller.print(controller.pullData("RSamples.dat"));
    }
    catch(IOException e)
    {
        JOptionPane.showMessageDialog(frame, "Ошибка
открытия файла. Возможно:\n 1. Файл не имеет нужных колонок, или
он пуст;\n 2. Выбран некорректный
файл.", "Ошибка", JOptionPane.ERROR_MESSAGE);
    }
    }
});
SaveTableButton.addActionListener(actionListener ->
{

if(changeSampleTypeButton.getIcon().equals(MrIcon)) controller.prin
tOutputTable("MSamples.dat");
        else controller.printOutputTable("RSamples.dat");
    });
searchButton.addActionListener(actionListener ->
{
    if(searchBox.getText().length()>0) {
        SaveTableButton.setEnabled(false);
    }

controller.print(complexSearcher.searchByName(searchBox.getText(),
controller.getTableValue()));;

    });
searchSettingsButton.addActionListener(actionListener ->
{
    if(enableChemsButton.getBackground().getGreen() !=
255) Sframe.setSize(850, 220);
        else Sframe.setSize(850,350);

Sframe.setVisible(true);
    });
refreshButton.addActionListener(actionListener ->
{

```

```

        SaveTableButton.setEnabled(true);
        searchInfoButton.setEnabled(false);
        searchNextButton.setEnabled(false);
        searchBox.setText(null);
        if(changeSampleTypeButton.getIcon().equals(MrIcon))
controller.print(controller.pullData("MSamples.dat"));
        else
controller.print(controller.pullData("RSamples.dat"));
    });
    changeSampleTypeButton.addActionListener(actionListener ->
    {
        if(changeSampleTypeButton.getIcon().equals(MrIcon))
        {
            changeSampleTypeButton.setIcon(mrIcon);
            changeSampleTypeButton.setToolTipText("Для
отображения выбраны разрывные образцы. Кликните, чтобы отображать
маркировочные");
        }
        else
        {
            changeSampleTypeButton.setIcon(MrIcon);
            changeSampleTypeButton.setToolTipText("Для
отображения выбраны маркировочные образцы. Кликните, чтобы
отображать разрывные");
        }
    });
    controller.print(controller.pullData("MSamples.dat"));
    autoSearchButton.setEnabled(false);
    searchNextButton.setEnabled(false);
    searchInfoButton.setEnabled(false);
    deleteButton.addActionListener(actionListener ->
    {
        if(changeSampleTypeButton.getIcon().equals(MrIcon))
controller.delete("MSamples.dat");
        else controller.delete("RSamples.dat");
    });
    addButton.addActionListener(actionListener ->
    {
        addframe.setVisible(true);
    });
    classifyButton.addActionListener(actionListener ->
    {
        try {
            controller.classifyRow();
            classifyResultsFrame.setVisible(true);
        } catch (IOException e) {
            JOptionPane.showMessageDialog(addframe,

```

```

e.getMessage(), "Ошибка", JOptionPane.ERROR_MESSAGE);
    }
    });
    moveQualityToSampleButton.addActionListener(actionListener
->
    {
        try {
            controller.moveQualityToSample();
            SaveTableButton.doClick();
        } catch (IOException e) {
            JOptionPane.showMessageDialog(addframe,
e.getMessage(), "Ошибка", JOptionPane.ERROR_MESSAGE);
        }
    });
    classifySettingsButton.addActionListener(actionListener ->
    {
        classifiersBox.removeAllItems();
        for (File item:
            controller.getClassifiersList()) {

classifiersBox.addItem(item.getName().split("\\.")[0]);
        }

classifiersBox.setSelectedItem(controller.config.getModelName());
        try {

controller.getClassifierProperties((String)classifiersBox.getSelectedItem());
        } catch (Exception e) {
            JOptionPane.showMessageDialog(addframe, "Данный
классификатор не существует", "Ошибка",
JOptionPane.ERROR_MESSAGE);
        }
        classifySettingsFrame.setVisible(true);
    });
    classifiersBox.addItemListener(e -> {
        if (e.getStateChange() == ItemEvent.SELECTED) {
            try {

controller.getClassifierProperties((String)e.getItem());
            } catch (Exception ex) {
                JOptionPane.showMessageDialog(addframe,
"Данный классификатор не существует", "Ошибка",
JOptionPane.ERROR_MESSAGE);
            }
        }
    });
    changeClassifierButton.addActionListener(actionListener ->
    {

controller.config.setModelName((String)classifiersBox.getSelectedI
tem());

```

```

    });
    deleteClassifierButton.addActionListener (actionListener ->
    {

controller.deleteClassifier ((String) classifiersBox.getSelectedItem
());

classifiersBox.removeItem(classifiersBox.getSelectedItem());

classifiersBox.setSelectedItem(classifiersBox.getItemAt(0));

controller.config.setModelName ((String) classifiersBox.getSelectedI
tem());
    });
    trainNetButton.addActionListener (actionListener ->
    {
        try {
            String modelName =
JOptionPane.showInputDialog("Имя модели");
            controller.trainNewClassifier(modelName,
                JOptionPane.showInputDialog("Скорость
обучения (оставьте пустым для значения по умолчанию)"),
                JOptionPane.showInputDialog("Импульс
(оставьте пустым для значения по умолчанию)"),
                JOptionPane.showInputDialog("Количество
эпох (оставьте пустым для значения по умолчанию)"));
            classifiersBox.addItem(modelName);
            classifiersBox.setSelectedItem(modelName);
            JOptionPane.showMessageDialog(addframe, "Модель
обучена успешно", "Готово", JOptionPane.INFORMATION_MESSAGE);
        } catch (Exception e)
        {
            JOptionPane.showMessageDialog(addframe, "Произошла
непредвиденная ошибка создания модели", "Ошибка",
JOptionPane.ERROR_MESSAGE);
        }
    });
    WindowAdapter screenshotTrigger = new WindowAdapter() {
        @Override
        public void windowDeactivated(WindowEvent e) {
            screenshotButton.doClick();
            try {
                controller.screenshotDataCapture();
                JOptionPane.showMessageDialog(addframe,
"Снимок получен", "Успех", JOptionPane.INFORMATION_MESSAGE);
            } catch (Exception ex) {
                JOptionPane.showMessageDialog(addframe,
"Ошибка получения снимка экрана. Программа не смогла найти
открытое окно измерений", "Ошибка", JOptionPane.ERROR_MESSAGE);
            }
        }
    };
};

```

```

        screenshotButton.addActionListener (actionListener ->
        {
if (screenshotButton.getBackground().getGreen() != 255) {
                screenshotButton.setBackground(new Color(130,
255, 130));
                addframe.addWindowListener (screenshotTrigger);
            }else {
addframe.removeWindowListener (screenshotTrigger);
                screenshotButton.setBackground (null);
            }
        });
clearAddingQueueButton.addActionListener (actionListener ->
{
        controller.deleteRowManually();
    });
insertAddingQueueButton.addActionListener (actionListener -
>
{
        controller.addRowManually();
    });
addSampleButton.addActionListener (actionListener ->
{
        controller.addSampleManually();
        addframe.dispose();
        JOptionPane.showMessageDialog (frame, "Образец
добавлен", "Добавление", JOptionPane.INFORMATION_MESSAGE);
    });
searchParamsButton.addActionListener (actionListener ->
{
        paramsframe.setVisible (true);
    });
bookmarkButton.addActionListener (actionListener ->
{
controller.print (controller.pullData ("Bookmarks.dat"));
    });
deleteBookmarkButton.addActionListener (actionListener ->
{
        controller.delete ("Bookmarks.dat");
    });
addToBookmarkButton.addActionListener (actionListener ->
{
        controller.copySelectedInto ("Bookmarks.dat");
    });
changeStorageButton.addActionListener (actionListener ->
{
        if (changeSampleTypeButton.getIcon().equals (MrIcon))
        {
            controller.copySelectedInto ("RSamples.dat");
            controller.delete ("MSamples.dat");
        }
    });

```

```
    }
    else
    {
        controller.copySelectedInto("MSamples.dat");
        controller.delete("RSamples.dat");
    }
});
paramsframe.addWindowListener(new WindowAdapter() {
    public void windowClosing(WindowEvent e) {
        controller.config.setCDevAuto(CDevBox1.getText());

controller.config.setSiDevAuto(SiDevBox1.getText());

controller.config.setMnDevAuto(MnDevBox1.getText());
        controller.config.setPDevAuto(PDevBox1.getText());
        controller.config.setSDevAuto(SDevBox1.getText());

controller.config.setCrDevAuto(CrDevBox1.getText());

controller.config.setNiDevAuto(NiDevBox1.getText());

controller.config.setMoDevAuto(MoDevBox1.getText());

controller.config.setCuDevAuto(CuDevBox1.getText());

controller.config.setAlDevAuto(AlDevBox1.getText());
        controller.config.setVDevAuto(VDevBox1.getText());
        controller.config.setWDevAuto(WDevBox1.getText());

controller.config.setTiDevAuto(TiDevBox1.getText());

controller.config.setCStepAuto(CStepBox.getText());

controller.config.setSiStepAuto(SiStepBox.getText());

controller.config.setMnStepAuto(MnStepBox.getText());

controller.config.setPStepAuto(PStepBox.getText());

controller.config.setSStepAuto(SStepBox.getText());

controller.config.setCrStepAuto(CrStepBox.getText());

controller.config.setNiStepAuto(NiStepBox.getText());

controller.config.setMoStepAuto(MoStepBox.getText());

controller.config.setCuStepAuto(CuStepBox.getText());

controller.config.setAlStepAuto(AlStepBox.getText());

controller.config.setVStepAuto(VStepBox.getText());
```



```

controller.config.setWStepAuto(WStepBox.getText());

controller.config.setTiStepAuto(TiStepBox.getText());

controller.config.setDateCheckBox1(dateCheckBox1.isSelected());
    }
});
frame.addWindowListener(new WindowAdapter() {
    public void windowClosing(WindowEvent e) {
        controller.config = Config.builder()
            .dateCheckBox(dateCheckBox.isSelected())

.datePeriodCheckBox(datePeriodCheckBox.isSelected())

.qualityCheckBox(qualityCheckBox.isSelected())
            .bloomCheckBox(bloomCheckBox.isSelected())
            .seriaCheckBox(seriaCheckBox.isSelected())

.chemsButton(enableChemsButton.getBackground())
            .CDev(CDevBox.getText())
            .SiDev(SiDevBox.getText())
            .MnDev(MnDevBox.getText())
            .PDev(PDevBox.getText())
            .SDev(SDevBox.getText())
            .CrDev(CrDevBox.getText())
            .NiDev(NiDevBox.getText())
            .MoDev(MoDevBox.getText())
            .CuDev(CuDevBox.getText())
            .AlDev(AlDevBox.getText())
            .VDev(VDevBox.getText())
            .WDev(WDevBox.getText())
            .TiDev(TiDevBox.getText())
            .CDevAuto(CDevBox1.getText())
            .SiDevAuto(SiDevBox1.getText())
            .MnDevAuto(MnDevBox1.getText())
            .PDevAuto(PDevBox1.getText())
            .SDevAuto(SDevBox1.getText())
            .CrDevAuto(CrDevBox1.getText())
            .NiDevAuto(NiDevBox1.getText())
            .MoDevAuto(MoDevBox1.getText())
            .CuDevAuto(CuDevBox1.getText())
            .AlDevAuto(AlDevBox1.getText())
            .VDevAuto(VDevBox1.getText())
            .WDevAuto(WDevBox1.getText())
            .TiDevAuto(TiDevBox1.getText())
            .CStepAuto(CStepBox.getText())
            .SiStepAuto(SiStepBox.getText())
            .MnStepAuto(MnStepBox.getText())
            .PStepAuto(PStepBox.getText())
            .SStepAuto(SStepBox.getText())
            .CrStepAuto(CrStepBox.getText())

```

```

        .NiStepAuto (NiStepBox.getText ())
        .MoStepAuto (MoStepBox.getText ())
        .CuStepAuto (CuStepBox.getText ())
        .AlStepAuto (AlStepBox.getText ())
        .VStepAuto (VStepBox.getText ())
        .WStepAuto (WStepBox.getText ())
        .TiStepAuto (TiStepBox.getText ())
        .samplesPerStep (stepBox.getText ())

.samplesPerStepCheckBox (samplesPerStepCheckBox.isSelected ())
        .dateCheckBox1 (dateCheckBox1.isSelected ())

.modelName (controller.config.getModelName ())
        .build ();
        controller.setConfig (controller.config);
    }
    });
}
}

```

### Файл Controller.java:

```

package Flare;

import Flare.Modules.*;

import com.sun.jna.platform.WindowUtils;
import lombok.*;

import javax.imageio.ImageIO;
import javax.swing.*;
import javax.swing.table.DefaultTableModel;
import java.awt.*;
import java.awt.image.BufferedImage;
import java.io.*;
import java.text.DecimalFormat;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.*;
import java.util.List;
import java.util.stream.Collectors;

class SortByDate implements Comparator<MSample>
{
    SimpleDateFormat dt = new SimpleDateFormat ("dd.MM.yyyy");
    Date sample1Date;
    Date sample2Date;
    public int compare (MSample a, MSample b)
    {
        try {
            sample1Date = dt.parse (a.getDate ());

```

```

        sample2Date = dt.parse(b.getDate());
    } catch (ParseException e){}
    return sample2Date.compareTo(sample1Date);
}
}
@Getter
public class Controller {
    private final String[] header = new
String[]{"Дата", "Время", "Номер образца", "Марка стали", "Блюм",
"Кол-во блюмов", "Серия",
"C", "Si", "Mn", "P", "S", "Cr", "Ni", "Mo", "Cu", "Al", "V", "W", "Ti"};
    private final String[] classify_header = new String[]{"Марка
стали", "Вероятность"};
    private final String[] classifierInfo_header = new
String[]{"Параметр", "Значение"};
    private JTable
output_table, input_table, classify_table, classifierInfo_table;
    private DefaultTableModel output_table_inner = new
DefaultTableModel(header, 0);
    private DefaultTableModel input_table_inner = new
DefaultTableModel(header, 3);
    private DefaultTableModel classify_table_inner = new
DefaultTableModel(classify_header, 0);
    private DefaultTableModel classifierInfo_table_inner = new
DefaultTableModel(classifierInfo_header, 0);
    private List<MSample> tableList = new ArrayList<>();
    private ComplexSearch searcher = new ComplexSearch();
    Classifier classifier = new Classifier();
    private MSample bufferSample;
    public Config config;

    public Controller()
    {
        getConfig();
    }
    public JTable createOutputTable() {
        print(pullData("MSamples.dat"));
        output_table = new JTable(output_table_inner);
        output_table.putClientProperty("terminateEditOnFocusLost",
true);
        return output_table;
    }
    public JTable createInputTable()
    {
        input_table = new JTable(input_table_inner);
        input_table.putClientProperty("terminateEditOnFocusLost",
true);
        input_table_inner.setRowCount(1);
        return input_table;
    }
    public JTable createClassifyTable()
    {

```

```

        classify_table = new JTable(classify_table_inner);

classify_table.putClientProperty("terminateEditOnFocusLost",
true);
        classify_table_inner.setRowCount(0);
        return classify_table;
    }
    public JTable createClassifierInfoTable()
    {
        classifierInfo_table = new
JTable(classifierInfo_table_inner);
        classifierInfo_table.setDefaultEditor(Object.class, null);
        classifierInfo_table_inner.setRowCount(5);
        classifierInfo_table_inner.setValueAt("Скорость
обучения",0,0);
        classifierInfo_table_inner.setValueAt("Импульс",1,0);
        classifierInfo_table_inner.setValueAt("Количество
эпох",2,0);
        classifierInfo_table_inner.setValueAt("Точность
классификации",3,0);
        classifierInfo_table_inner.setValueAt("Уникальных
образцов",4,0);
        return classifierInfo_table;
    }

    public void print(List<MSample> tableList) {
        output_table_inner.setRowCount(0);
        tableList.forEach(sample ->
        {
            String date = sample.getDate();
            String time = sample.getTime();
            String sample_No = sample.getSampleNo();
            String quality = sample.getQuality();
            String bloom = sample.getBloom();
            String bloomCount = sample.getCodes();
            String seria = sample.getSeria();
            String C = sample.getC();
            String Si = sample.getSi();
            String Mn = sample.getMn();
            String P = sample.getP();
            String S = sample.getS();
            String Cr = sample.getCr();
            String Ni = sample.getNi();
            String Mo = sample.getMo();
            String Cu = sample.getCu();
            String Al = sample.getAl();
            String V = sample.getV();
            String W = sample.getW();
            String Ti = sample.getTi();
            Object[] rowData =
{date,time,sample_No,quality,bloom,bloomCount,seria,C,Si,Mn,P,S,Cr

```

```

,Ni,Mo,Cu,Al,V,W,Ti};
        output_table_inner.addRow(rowData);
    });
}

public List<MSample> pullData(String target) {
    File f = new File(target);

    if(!f.exists()){
        try {
            f.createNewFile();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
    List<MSample> sampleList = new ArrayList<>();
    try (ObjectInputStream fout = new ObjectInputStream(new
FileInputStream(target))) {
        sampleList = ((ArrayList<MSample>) fout.readObject());
    } catch (Exception e) {

    }
    return sampleList;
}

public void pushData(List<MSample> collectedData){
    List<MSample> sampleMList = new ArrayList<>();
    List<MSample> sampleRList = new ArrayList<>();
    List<MSample> oldSampleMList = pullData("MSamples.dat");
    List<MSample> oldSampleRList = pullData("RSamples.dat");
    boolean found = false;
    for (int i = 0; i<collectedData.size();++i)
    {
        if(collectedData.get(i).getSampleNo().contains("M"))
        {
            for(int j = 0; j<oldSampleMList.size();++j)

if(collectedData.get(i).getSampleNo().compareTo(oldSampleMList.get
(j).getSampleNo())==0) found = true;
                }

if(found==false) sampleMList.add(collectedData.get(i));
                found = false;
            }
            else
            {

if(collectedData.get(i).getSampleNo().contains("R") ||

```

```

collectedData.get(i).getSampleNo().contains("P") ||
collectedData.get(i).getSampleNo().contains("T") ||
collectedData.get(i).getSampleNo().contains("C") ||
collectedData.get(i).getSampleNo().contains("U")
    ) {
        for(int j = 0; j<oldSampleRList.size();++j)
        {

if(collectedData.get(i).getSampleNo().compareTo(oldSampleRList.get
(j).getSampleNo())==0) found = true;
        }

if(found==false) sampleRList.add(collectedData.get(i));
        found = false;
    }
    else
    {
        for(int j = 0; j<oldSampleMList.size();++j)
        {

if(collectedData.get(i).getSampleNo().compareTo(oldSampleMList.get
(j).getSampleNo())==0) found = true;
        }

if(found==false) sampleMList.add(collectedData.get(i));
        found = false;
    }
}

sampleMList.addAll(oldSampleMList);
sampleRList.addAll(oldSampleRList);
Collections.sort(sampleMList,new SortByDate());
Collections.sort(sampleRList,new SortByDate());
refreshData(sampleMList,"MSamples.dat");
refreshData(sampleRList,"RSamples.dat");
}

public void refreshData(List<MSample> sample,String target) {
    try (ObjectOutputStream fout = new ObjectOutputStream(new
FileOutputStream(target))) {
        fout.writeObject(sample);
        fout.flush();
    } catch (Exception ex) {
        System.out.println(ex.getMessage());
    }
}

public void addSampleManually()
{

```

```

List<MSample> sampleList = new ArrayList<>();
String[] data = new String[20];
for (int i = 0; i < input_table.getRowCount(); i++) {
    for (int j = 0; j < input_table.getColumnCount(); j++)
    {
        data[j] =
String.valueOf(input_table_inner.getValueAt(i, j));
        if (data[j] == "null") data[j] = null;
    }
    MSample sample = MSample.builder()
        .date(data[0])
        .time(data[1])
        .sampleNo(data[2])
        .quality(data[3])
        .bloom(data[4])
        .codes(data[5])
        .seria(data[6])
        .C(data[7])
        .Si(data[8])
        .Mn(data[9])
        .P(data[10])
        .S(data[11])
        .Cr(data[12])
        .Ni(data[13])
        .Mo(data[14])
        .Cu(data[15])
        .Al(data[16])
        .V(data[17])
        .W(data[18])
        .Ti(data[19])
        .build();
    sampleList.add(sample);
}
pushData(sampleList);
}
public void addRowManually()
{
input_table_inner.setRowCount(input_table_inner.getRowCount()+1);
}
public void deleteRowManually()
{
    int selectedRow[] = input_table.getSelectedRows();
    for(int i = 0; i < selectedRow.length; i++) {
input_table_inner.removeRow(selectedRow[selectedRow.length-1-i]);
    }
}
public void classifyRow() throws IOException {
    int selectedRow[] = output_table.getSelectedRows();
    String[] data = new String[20];
    classify_table_inner.setRowCount(0);

```

```

        if(selectedRow.length>1) throw new IOException("Было
выделено несколько рядков. Для определения выберите один");
        for(int j = 0; j < output_table.getColumnCount(); j++)
    {
        data[j] =
String.valueOf(output_table_inner.getValueAt(selectedRow[0], j));
        if (data[j] == "null") data[j] = null;
    }
    MSample sample = MSample.builder()
        .date(data[0])
        .time(data[1])
        .sampleNo(data[2])
        .quality(data[3])
        .bloom(data[4])
        .codes(data[5])
        .seria(data[6])
        .C(data[7])
        .Si(data[8])
        .Mn(data[9])
        .P(data[10])
        .S(data[11])
        .Cr(data[12])
        .Ni(data[13])
        .Mo(data[14])
        .Cu(data[15])
        .Al(data[16])
        .V(data[17])
        .W(data[18])
        .Ti(data[19])
        .build();

    List<Map.Entry<String,Double>> result =
classifier.classify(sample,"models/"+config.getModelName());
    result.forEach(quality->
    {
        Object[] rowData = {quality.getKey(),new
DecimalFormat("#0.00%").format(quality.getValue().doubleValue())};
        classify_table_inner.addRow(rowData);
    });
}
public void moveQualityToSample() throws IOException {
    int selectedRow[] = output_table.getSelectedRows();
    int qualityRow[] = classify_table.getSelectedRows();
    if(qualityRow.length>1) throw new IOException("Было
выделено несколько марок стали. Для переноса выберите одну");

output_table_inner.setValueAt(classify_table_inner.getValueAt(qual
ityRow[0],0),selectedRow[0],3);

```



```

    }
    public void getClassifierProperties(String selectedItem)
throws Exception {
        ClassifierInfo classifierInfo =
classifier.getClassifierInfo(selectedItem+".stats");

classifierInfo_table_inner.setValueAt(classifierInfo.getLearningRa
te(),0,1);

classifierInfo_table_inner.setValueAt(classifierInfo.getMomentum()
,1,1);

classifierInfo_table_inner.setValueAt(classifierInfo.getEpochs(),2
,1);
        classifierInfo_table_inner.setValueAt(new
DecimalFormat("#0.0000").format(classifierInfo.getPercentage()),3,
1);

classifierInfo_table_inner.setValueAt(classifierInfo.getQualities(
).size(),4,1);
    }
    public void trainNewClassifier(String classifierName, String
learningRate, String momentum, String epochs) throws Exception {

classifier.trainClassifier(pullData("MSamples.dat"),classifierName
,learningRate, momentum, epochs);
    }
    public void deleteClassifier(String classifierName){
        File f = new File("models/"+classifierName+".stats");
        f.delete();
        f = new File("models/"+classifierName+".model");
        f.delete();
    }
    public void screenshotDataCapture() throws Exception
    {
        Thread.sleep(100);
        final Rectangle rect = new Rectangle(0, 0, 0, 0);
        WindowUtils.getAllWindows(true).forEach(desktopWindow -> {
            if (desktopWindow.getTitle().equals("Окно Измерений -
Элементы: Концентрации")) { //"Окно Измерений - Элементы:
Концентрации"
                rect.setRect(desktopWindow.getLocAndSize());
            }
        });
        BufferedImage image = new
Robot().createScreenCapture(rect);
        //BufferedImage image = new
Robot().createScreenCapture(new
Rectangle(Toolkit.getDefaultToolkit().getScreenSize()));
        ImageIO.write(image, "PNG", new
File("screenshot.png"));/**/
        Preprocessor preprocessor = new Preprocessor();

```

```

List<String> sampleList = preprocessor.doTesseract();
input_table_inner.setRowCount(0);
SimpleDateFormat dt = new SimpleDateFormat("dd.MM.yyyy");
SimpleDateFormat tm = new SimpleDateFormat("HH:mm:ss");
sampleList.forEach(sample ->
{
    List<String>sampleElems= Arrays.stream(sample.split("
")).collect(Collectors.toList());

    String date =
dt.format(Calendar.getInstance().getTime());
    String time =
tm.format(Calendar.getInstance().getTime());
    String sample_No = "";
    String quality = "";
    String bloom = "";
    String bloomCount = "";
    String seria = "";
    String C = sampleElems.get(0);
    String Si = sampleElems.get(1);
    String Mn = sampleElems.get(2);
    String P = sampleElems.get(3);
    String S = sampleElems.get(4);
    String Cr = sampleElems.get(5);
    String Ni = sampleElems.get(6);
    String Mo = sampleElems.get(7);
    String Cu = sampleElems.get(8);
    String Al = sampleElems.get(9);
    String V = sampleElems.get(10);
    String W = sampleElems.get(11);
    String Ti = sampleElems.get(12);
    Object[] rowData =
{date,time,sample_No,quality,bloom,bloomCount,seria,C,Si,Mn,P,S,Cr
,Ni,Mo,Cu,Al,V,W,Ti};
    input_table_inner.addRow(rowData);
});
}
public void printOutputTable(String target) {
    List<MSample> sampleList = new ArrayList<>();
    String[] data = new String[20];
    try
    {
        for (int i = 0; i < output_table.getRowCount(); i++)
        {
            for (int j = 0; j < output_table.getColumnCount();
j++) {
                data[j] =
String.valueOf(output_table_inner.getValueAt(i, j));
                if(data[j]=="null") data[j]=null;
            }
            MSample sample = MSample.builder()
                .date(data[0])

```

```

        .time(data[1])
        .sampleNo(data[2])
        .quality(data[3])
        .bloom(data[4])
        .codes(data[5])
        .seria(data[6])
        .C(data[7])
        .Si(data[8])
        .Mn(data[9])
        .P(data[10])
        .S(data[11])
        .Cr(data[12])
        .Ni(data[13])
        .Mo(data[14])
        .Cu(data[15])
        .Al(data[16])
        .V(data[17])
        .W(data[18])
        .Ti(data[19])
        .build();
        sampleList.add(sample);
    }
    refreshData(sampleList, target);
    print(pullData(target));
} catch (IllegalArgumentException e)
{
    JOptionPane.showMessageDialog(null, "Заполните поля
корректно");
    print(pullData(target));
}
}
public List<MSample> getTableValue()
{
    String[] data = new String[20];
    List<MSample> result = new ArrayList<>();
    for(int i = 0; i < output_table.getRowCount(); i++) {
        for (int j = 0; j < output_table.getColumnCount();
j++) {
            data[j] =
String.valueOf(output_table_inner.getValueAt(i, j));
            if (data[j] == "null") data[j] = null;
        }

        MSample sample = MSample.builder()
            .date(data[0])
            .time(data[1])
            .sampleNo(data[2])
            .quality(data[3])
            .bloom(data[4])
            .codes(data[5])
            .seria(data[6])
            .C(data[7])

```

```

        .Si(data[8])
        .Mn(data[9])
        .P(data[10])
        .S(data[11])
        .Cr(data[12])
        .Ni(data[13])
        .Mo(data[14])
        .Cu(data[15])
        .Al(data[16])
        .V(data[17])
        .W(data[18])
        .Ti(data[19])
        .build();
    result.add(sample);
}
return result;
}
public void delete(String target)
{
    int selectedRow[] = output_table.getSelectedRows();
    List<MSample> deleteList = new ArrayList<>();
    List<MSample> sampleList = new
ArrayList<>(pullData(target));
    String[] data = new String[20];
    for(int i = 0; i < selectedRow.length; i++) {
        for(int j = 0; j < output_table.getColumnCount(); j++)
        {
            data[j] =
String.valueOf(output_table_inner.getValueAt(selectedRow[i], j));
            if(data[j]=="null") data[j]=null;
        }

        MSample sample = MSample.builder()
            .date(data[0])
            .time(data[1])
            .sampleNo(data[2])
            .quality(data[3])
            .bloom(data[4])
            .codes(data[5])
            .seria(data[6])
            .C(data[7])
            .Si(data[8])
            .Mn(data[9])
            .P(data[10])
            .S(data[11])
            .Cr(data[12])
            .Ni(data[13])
            .Mo(data[14])
            .Cu(data[15])
            .Al(data[16])
            .V(data[17])
            .W(data[18])

```

```

        .Ti(data[19])
        .build();
        deleteList.add(sample);
    }

    sampleList.removeAll(deleteList);
    refreshData(sampleList,target);
    for(int i = 0; i < selectedRow.length; i++) {

output_table_inner.removeRow(selectedRow[selectedRow.length-1-i]);
    }

}
public void copySelectedInto(String target)
{
    int selectedRow[] = output_table.getSelectedRows();
    List<MSample> addList = new ArrayList<>();
    List<MSample> sampleList = new
ArrayList<>(pullData(target));
    String[] data = new String[20];
    for(int i = 0; i < selectedRow.length; i++) {
        for(int j = 0; j < output_table.getColumnCount(); j++)
        {
            data[j] =
String.valueOf(output_table_inner.getValueAt(selectedRow[i], j));
            if(data[j]=="null") data[j]=null;
        }

        MSample sample = MSample.builder()
            .date(data[0])
            .time(data[1])
            .sampleNo(data[2])
            .quality(data[3])
            .bloom(data[4])
            .codes(data[5])
            .seria(data[6])
            .C(data[7])
            .Si(data[8])
            .Mn(data[9])
            .P(data[10])
            .S(data[11])
            .Cr(data[12])
            .Ni(data[13])
            .Mo(data[14])
            .Cu(data[15])
            .Al(data[16])
            .V(data[17])
            .W(data[18])
            .Ti(data[19])
            .build();
        addList.add(sample);
    }
}

```

```

        sampleList.addAll(addList);
        refreshData(sampleList,target);

    }
    public void startAutosearch()
    {
        int selectedRow = output_table.getSelectedRow();
        String[] data = new String[20];
        for(int j = 0; j < output_table.getColumnCount(); j++)
        {
            data[j] =
String.valueOf(output_table_inner.getValueAt(selectedRow, j));
            if(data[j]=="null") data[j]=null;
        }
        MSample sample = MSample.builder()
            .date(data[0])
            .time(data[1])
            .sampleNo(data[2])
            .quality(data[3])
            .bloom(data[4])
            .codes(data[5])
            .seria(data[6])
            .C(data[7])
            .Si(data[8])
            .Mn(data[9])
            .P(data[10])
            .S(data[11])
            .Cr(data[12])
            .Ni(data[13])
            .Mo(data[14])
            .Cu(data[15])
            .Al(data[16])
            .V(data[17])
            .W(data[18])
            .Ti(data[19])
            .build();
        bufferSample=sample;

print(searcher.autosearch(sample,pullData("MSamples.dat"),config)
;
    }
    public void startStepsearch(String firstDate, String
secondDate)
    {

print(searcher.stepsearch(bufferSample,pullData("MSamples.dat"),co
nfig,firstDate,secondDate));
    }
    public void stopStepsearch()
    {

```

```
        searcher.setStepNo(1);
        searcher.setIterationNo(1);
    }
    public void getConfig()
    {
        File f = new File("config.cfg");
        if(!f.exists()){
            try {
                f.createNewFile();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }

        try (ObjectInputStream fout = new ObjectInputStream(new
FileInputStream("config.cfg"))) {
            config = ((Config) fout.readObject());
        } catch (Exception e) {
            config = Config.builder()
                .dateCheckBox(false)
                .datePeriodCheckBox(false)
                .qualityCheckBox(false)
                .bloomCheckBox(false)
                .chemsButton(null)
                .seriaCheckBox(false)
                .modelName("default.model")
                /*.CDev(null)
                .SiDev(null)
                .MnDev(null)
                .PDev(null)
                .SDev(null)
                .CrDev(null)
                .NiDev(null)
                .MoDev(null)
                .CuDev(null)
                .AlDev(null)
                .VDev(null)
                .WDev(null)
                .TiDev(null)
                .CDevAuto(null)
                .SiDevAuto(null)
                .MnDevAuto(null)
                .PDevAuto(null)
                .SDevAuto(null)
                .CrDevAuto(null)
                .NiDevAuto(null)
                .MoDevAuto(null)
                .CuDevAuto(null)
                .AlDevAuto(null)
                .VDevAuto(null)
                .WDevAuto(null)
                .TiDevAuto(null)
```

```

        */
        .build();
    }
}
public void setConfig(Config config)
{
    try (ObjectOutputStream fout = new ObjectOutputStream(new
FileOutputStream("config.cfg"))) {
        fout.writeObject(config);
        fout.flush();
    } catch (Exception ex) {
        System.out.println(ex.getMessage());
    }
}

public List<File> getClassifiersList() {
    File dir = new File("models"); //path указывает на
директорию
    List<File> lst = new ArrayList<>();
    for ( File file : dir.listFiles() ){
        if ( file.isFile() &&
file.getName().split("\\.")[1].equals("stats"))
            lst.add(file);
    }
    return lst;
}
}

```

### Файл ComplexSearch.java:

```

package Flare.Modules;

import Flare.MSample;
import Flare.Program;

import javax.swing.*;
import java.util.ArrayList;
import java.util.List;
import lombok.*;

@Getter
@Setter

public class ComplexSearch {
    private String report;
    private boolean findSuccess;
    int stepNo=1;
    int iterationNo=1;
    private String

```



```

CStep, SiStep, MnStep, PStep, SStep, CrStep, NiStep, MoStep, CuStep, AlStep
, VStep, WStep, TiStep;
    public List<MSample> searchByName(String name, List<MSample>
searchList) {
        List<MSample> result = new ArrayList<>();
        for (int i = 0; i < searchList.size(); ++i) {
            if (searchList.get(i).getSampleNo().contains(name))
result.add(searchList.get(i));
        }
        return result;
    }
    public List<MSample> searchByDate(String dateFrom, String
dateTo, List<MSample> searchList) {
        List<MSample> result = new ArrayList<>();
        String[] splitFromDate = dateFrom.split("\\.");
        if (dateTo != null) {
            try {
                String[] splitToDate = dateTo.split("\\.");

if (Integer.parseInt(splitToDate[2]) < Integer.parseInt(splitFromDate
[2]) || Integer.parseInt(splitToDate[1]) > 12 ||
Integer.parseInt(splitToDate[1]) < 1 ||
Integer.parseInt(splitToDate[0]) < 1 ||
Integer.parseInt(splitToDate[0]) > 31) throw new Exception();
                for (int i = 0; i < searchList.size(); ++i) {
                    String[] splitDate;
                    splitDate =
searchList.get(i).getDate().split("\\.");
                    if (Integer.parseInt(splitDate[2]) >=
Integer.parseInt(splitFromDate[2]) &&
Integer.parseInt(splitDate[2]) <= Integer.parseInt(splitToDate[2])
&&
                                Integer.parseInt(splitDate[1]) >=
Integer.parseInt(splitFromDate[1]) &&
Integer.parseInt(splitDate[1]) <= Integer.parseInt(splitToDate[1])
&&
                                Integer.parseInt(splitDate[0]) >=
Integer.parseInt(splitFromDate[0]) &&
Integer.parseInt(splitDate[0]) <= Integer.parseInt(splitToDate[0])
) result.add(searchList.get(i));
                }
            }
            catch (Exception e)
            {
                JOptionPane.showMessageDialog(null, "Интервал
или формат даты введен
некорректно", "Ошибка", JOptionPane.ERROR_MESSAGE);
            }
        }
        else
        {
            try {

```

```

        if (Integer.parseInt(splitFromDate[1])>12 ||
Integer.parseInt(splitFromDate[1])<1 ||
Integer.parseInt(splitFromDate[0])<1 ||
Integer.parseInt(splitFromDate[0])>31)
            for (int i = 0; i < searchList.size(); ++i)
                if
(dateFrom.compareTo(searchList.get(i).getDate()) == 0)
result.add(searchList.get(i));
            }
            catch (Exception e)
            {
                JOptionPane.showMessageDialog(null, "Формат
даты введен некорректно", "Ошибка", JOptionPane.ERROR_MESSAGE);
            }
        }

        return result;
    }
    public List<MSample> searchByQuality(String name,
List<MSample> searchList) {
        List<MSample> result = new ArrayList<>();
        boolean errors = false;
        try {
            if(name.length()<1) throw new
IllegalArgumentException();
            for (int i = 0; i < searchList.size(); ++i) {
                try {
                    if
(searchList.get(i).getQuality().compareTo(name) == 0)
result.add(searchList.get(i));
                } catch (Exception e) {
                    errors = true;
                }
            }
            if (errors)
                JOptionPane.showMessageDialog(null, "При поиске
возникли некоторые ошибки. Возможно стоит проверить, все ли ячейки
заполнены", "Ошибка", JOptionPane.ERROR_MESSAGE);
        }
        catch (IllegalArgumentException e)
        {
            JOptionPane.showMessageDialog(null, "Поле марки стали
пустое", "Ошибка", JOptionPane.ERROR_MESSAGE);
        }
        return result;
    }
    public List<MSample> searchByBloom(String name, List<MSample>
searchList)
    {
        List<MSample> result = new ArrayList<>();
        for (int i = 0; i < searchList.size(); ++i) {
            try

```

```

        {
            if (searchList.get(i).getBloom().compareTo(name)
== 0) result.add(searchList.get(i));
        }
        catch (Exception e)
        {
        }
    }
    return result;
}

public List<MSample> searchBySeria(String seriaFrom,String
seriaTo, List<MSample> searchList) {
    List<MSample> result = new ArrayList<>();
    boolean errors = false;
    seriaFrom+="*";
    seriaFrom+=seriaTo;
    for (int i = 0; i < searchList.size(); ++i) {
        try {
            if
(searchList.get(i).getSeria().compareTo(seriaFrom) == 0)
result.add(searchList.get(i));
        }
        catch (Exception e) {}
    }
    return result;
}

public List<MSample> searchByChems(String C,String Si,String
Mn,String P,String S,String Cr,String Ni,String Mo,String
Cu,String Al,String V,String W,String Ti,Config config,
List<MSample> searchList) {
    List<MSample> result = new ArrayList<>();
    boolean errors = false;
    for (int i = 0; i < searchList.size(); ++i) {
        /* try{*/
        if(

Double.parseDouble(searchList.get(i).getC())>=Double.parseDouble(C
)-Double.parseDouble(config.getCDev())&&

Double.parseDouble(searchList.get(i).getC())<=Double.parseDouble(C
)+Double.parseDouble(config.getCDev())&&

Double.parseDouble(searchList.get(i).getSi())>=Double.parseDouble(
Si)-Double.parseDouble(config.getSiDev())&&

Double.parseDouble(searchList.get(i).getSi())<=Double.parseDouble(
Si)+Double.parseDouble(config.getSiDev())&&

Double.parseDouble(searchList.get(i).getMn())>=Double.parseDouble(
Mn)-Double.parseDouble(config.getMnDev())&&

Double.parseDouble(searchList.get(i).getMn())<=Double.parseDouble(

```

```
Mn)+Double.parseDouble(config.getMnDev()) &&
```

```
Double.parseDouble(searchList.get(i).getP())>=Double.parseDouble(P)-Double.parseDouble(config.getPDev()) &&
```

```
Double.parseDouble(searchList.get(i).getP())<=Double.parseDouble(P)+Double.parseDouble(config.getPDev()) &&
```

```
Double.parseDouble(searchList.get(i).getS())>=Double.parseDouble(S)-Double.parseDouble(config.getSDev()) &&
```

```
Double.parseDouble(searchList.get(i).getS())<=Double.parseDouble(S)+Double.parseDouble(config.getSDev()) &&
```

```
Double.parseDouble(searchList.get(i).getCr())>=Double.parseDouble(Cr)-Double.parseDouble(config.getCrDev()) &&
```

```
Double.parseDouble(searchList.get(i).getCr())<=Double.parseDouble(Cr)+Double.parseDouble(config.getCrDev()) &&
```

```
Double.parseDouble(searchList.get(i).getNi())>=Double.parseDouble(Ni)-Double.parseDouble(config.getNiDev()) &&
```

```
Double.parseDouble(searchList.get(i).getNi())<=Double.parseDouble(Ni)+Double.parseDouble(config.getNiDev()) &&
```

```
Double.parseDouble(searchList.get(i).getMo())>=Double.parseDouble(Mo)-Double.parseDouble(config.getMoDev()) &&
```

```
Double.parseDouble(searchList.get(i).getMo())<=Double.parseDouble(Mo)+Double.parseDouble(config.getMoDev()) &&
```

```
Double.parseDouble(searchList.get(i).getCu())>=Double.parseDouble(Cu)-Double.parseDouble(config.getCuDev()) &&
```

```
Double.parseDouble(searchList.get(i).getCu())<=Double.parseDouble(Cu)+Double.parseDouble(config.getCuDev()) &&
```

```
Double.parseDouble(searchList.get(i).getAl())>=Double.parseDouble(Al)-Double.parseDouble(config.getAlDev()) &&
```

```
Double.parseDouble(searchList.get(i).getAl())<=Double.parseDouble(Al)+Double.parseDouble(config.getAlDev()) &&
```

```
Double.parseDouble(searchList.get(i).getV())>=Double.parseDouble(V)-Double.parseDouble(config.getVDev()) &&
```

```
Double.parseDouble(searchList.get(i).getV())<=Double.parseDouble(V)+Double.parseDouble(config.getVDev()) &&
```

```
Double.parseDouble(searchList.get(i).getW())>=Double.parseDouble(W)-Double.parseDouble(config.getWDev()) &&
```

```

Double.parseDouble(searchList.get(i).getW())<=Double.parseDouble(W
)+Double.parseDouble(config.getWDev())&&

Double.parseDouble(searchList.get(i).getTi())>=Double.parseDouble(
Ti)-Double.parseDouble(config.getTiDev())&&

Double.parseDouble(searchList.get(i).getTi())<=Double.parseDouble(
Ti)+Double.parseDouble(config.getTiDev())
    ) result.add(searchList.get(i));
    /*}catch (Exception e)
    {
    errors=true;
    }*/
    }
    if(errors)JOptionPane.showMessageDialog(null,"При поиске
по хим. составу возникли некоторые ошибки. Возможно стоит
проверить, все ли ячейки хим. элементов
заполнены", "Ошибка",JOptionPane.ERROR_MESSAGE);
    return result;
    }
    public List<MSample> autosearch(MSample input,List<MSample>
searchList, Config config) {
    List<MSample> result = new ArrayList<>();
    report = new String();
    findSuccess=false;
    List<MSample> bufferList =new ArrayList<>();
    MSample bufferSample = null;
    Config bufferConfig =Config.builder()
        .CDev(config.getCDevAuto())
        .SiDev(config.getSiDevAuto())
        .MnDev(config.getMnDevAuto())
        .PDev(config.getPDevAuto())
        .SDev(config.getSDevAuto())
        .CrDev(config.getCrDevAuto())
        .NiDev(config.getNiDevAuto())
        .MoDev(config.getMoDevAuto())
        .CuDev(config.getCuDevAuto())
        .AlDev(config.getAlDevAuto())
        .VDev(config.getVDevAuto())
        .WDev(config.getWDevAuto())
        .TiDev(config.getTiDevAuto())
        .build();
    boolean errors = false;
    String splitter[]=input.getSampleNo().split("/");
    try {
        result.add(searchByBloom(splitter[0],
searchList).get(0));
        report+="При поиске по номеру бюлма был найден образец
"+result.get(0).getSampleNo()+", \n";
        bufferList.add(input);

```

```

bufferList=searchByChems(result.get(0).getC(),result.get(0).getSi(
),result.get(0).getMn(),

result.get(0).getP(),result.get(0).getS(),result.get(0).getCr(),re
sult.get(0).getNi(),

result.get(0).getMo(),result.get(0).getCu(),result.get(0).getAl(),
result.get(0).getV(),

result.get(0).getW(),result.get(0).getTi(),bufferConfig,bufferList
);
        if(bufferList.size()!=0) {
            report += " и он совпал по хим. составу с
разрывным. Поиск завершён.";
            findSuccess=true;

JOptionPane.showMessageDialog(null,report,"Завершено",JOptionPane.
INFORMATION_MESSAGE);
            result.add(input);
            return result;
        }
        else {
            report += " \ни он не совпал по хим. составу с
разрывным. Поиск завершён.";
            result.add(input);

JOptionPane.showMessageDialog(null,report,"Завершено",JOptionPane.
INFORMATION_MESSAGE);
            return result;
        }
    }
    catch (IndexOutOfBoundsException e)
    {
        report+="\nПри поиске по номеру бляма образец не
был найден. ";
        for(int i = 0; i<searchList.size();++i)
        {
            try {
                if ((Integer.parseInt(splitter[0]) <=
Integer.parseInt(searchList.get(i).getBloom()+
Integer.parseInt(searchList.get(i).getCodes()-
1)&&(Integer.parseInt(splitter[0]) >=
Integer.parseInt(searchList.get(i).getBloom()))
                    bufferList.add(searchList.get(i));
                }catch (Exception ex){}
            }
            try {

                if
(Integer.parseInt(bufferList.get(0).getBloom()) +
Integer.parseInt(bufferList.get(0).getCodes()-1 >=
Integer.parseInt(splitter[0])) {

```

```

        report += "\nБыл найден маркировочный
образец "+bufferList.get(0).getSampleNo()+" с номером блюда
"+bufferList.get(0).getBloom()+", который является последним
записанным в хранилище блюдом перед искомым и входит в нумерацию
блюдом искомой плавки. ";
        result.clear();
        bufferSample=bufferList.get(0);
        bufferList.clear();
        result.add(bufferSample);
        bufferList.add(input);

bufferList=searchByChems(result.get(0).getC(),result.get(0).getSi(
),result.get(0).getMn(),

result.get(0).getP(),result.get(0).getS(),result.get(0).getCr(),re
sult.get(0).getNi(),

result.get(0).getMo(),result.get(0).getCu(),result.get(0).getAl(),
result.get(0).getV(),

result.get(0).getW(),result.get(0).getTi(),bufferConfig,bufferList
);
        }
        else
        {
                report += "\nБыл найден маркировочный
образец с номером блюда "+bufferSample.getBloom()+", который
является последним записанным в хранилище блюдом перед
искомым,\но не входит в нумерацию блюдом искомой плавки. Поиск не
дал результатов. ";
                result.clear();

JOptionPane.showMessageDialog(null,report,"Завершено",JOptionPane.
INFORMATION_MESSAGE);
                return result;
        }
        if (bufferList.size() !=0)
        {

                report += "\nЭтот образец
("+bufferSample.getSampleNo()+" ) совпал по хим. составу с
разрывным. Поиск завершён.";

JOptionPane.showMessageDialog(null,report,"Завершено",JOptionPane.
INFORMATION_MESSAGE);

                result.add(input);
                findSuccess=true;
                return result;
        }
        else
        {
                report += "\nНо этот маркировочный образец

```

не совпал по хим. составу с разрывным. Поиск завершён.";

```
JOptionPane.showMessageDialog(null, report, "Завершено", JOptionPane.
INFORMATION_MESSAGE);
```

```
        result.add(input);
```

```
        return result;
```

```
    }
```

```
    }catch (Exception ex)
```

```
    {
```

```
        report += "\nПоследних записанных в хранилище
блужомов перед искомым не обнаружено. Поиск не дал результатов";
```

```
        result.clear();
```

```
JOptionPane.showMessageDialog(null, report, "Завершено", JOptionPane.
INFORMATION_MESSAGE);
```

```
        return result;
```

```
    }
```

```
}
```

```
}
```

```
public List<MSample> stepsearch(MSample input, List<MSample>
searchList, Config config, String firstDate, String secondDate)
```

```
{
```

```
    List<MSample> result = new ArrayList<>();
```

```
    report = new String();
```

```
    List<MSample> bufferList= new ArrayList<>();
```

```
    Config bufferConfig = null;
```

```
    result.add(input);
```

```
    MSample bufferSample;
```

```
    if(!config.isSamplesPerStepCheckBox())
```

```
config.setSamplesPerStep("1");
```

```
    if(config.isDateCheckBox1() && firstDate.length() > 6)
```

```
    {
```

```
        searchList =
```

```
searchByDate(firstDate, secondDate, searchList);
```

```
    }
```

```
        for(stepNo=stepNo;
```

```
bufferList.size() < Integer.parseInt(config.getSamplesPerStep()) * ite
rationNo || stepNo > 1000; ++stepNo) {
```

```
            bufferConfig = Config.builder()
```

```
.CDev(String.valueOf(Double.parseDouble(config.getCDevAuto()) + Do
uble.parseDouble(config.getCStepAuto()) * (stepNo)))
```

```
.SiDev(String.valueOf(Double.parseDouble(config.getSiDevAuto()) + Do
uble.parseDouble(config.getSiStepAuto()) * (stepNo)))
```

```
.MnDev(String.valueOf(Double.parseDouble(config.getMnDevAuto()) + Do
uble.parseDouble(config.getMnStepAuto()) * (stepNo)))
```

```
.PDev(String.valueOf(Double.parseDouble(config.getPDevAuto()) + Dou
```



```

le.parseDouble(config.getPStepAuto()) * (stepNo))

.SDev(String.valueOf(Double.parseDouble(config.getSDevAuto())+Double.parseDouble(config.getSStepAuto()) * (stepNo)))

.CrDev(String.valueOf(Double.parseDouble(config.getCrDevAuto())+Double.parseDouble(config.getCrStepAuto()) * (stepNo)))

.NiDev(String.valueOf(Double.parseDouble(config.getNiDevAuto())+Double.parseDouble(config.getNiStepAuto()) * (stepNo)))

.MoDev(String.valueOf(Double.parseDouble(config.getMoDevAuto())+Double.parseDouble(config.getMoStepAuto()) * (stepNo)))

.CuDev(String.valueOf(Double.parseDouble(config.getCuDevAuto())+Double.parseDouble(config.getCuStepAuto()) * (stepNo)))

.AlDev(String.valueOf(Double.parseDouble(config.getAlDevAuto())+Double.parseDouble(config.getAlStepAuto()) * (stepNo)))

.VDev(String.valueOf(Double.parseDouble(config.getVDevAuto())+Double.parseDouble(config.getVStepAuto()) * (stepNo)))

.WDev(String.valueOf(Double.parseDouble(config.getWDevAuto())+Double.parseDouble(config.getWStepAuto()) * (stepNo)))

.TiDev(String.valueOf(Double.parseDouble(config.getTiDevAuto())+Double.parseDouble(config.getTiStepAuto()) * (stepNo)))
        .build();
        bufferList = searchByChems(result.get(0).getC(),
result.get(0).getSi(), result.get(0).getMn(),
        result.get(0).getP(), result.get(0).getS(),
result.get(0).getCr(), result.get(0).getNi(),
        result.get(0).getMo(), result.get(0).getCu(),
result.get(0).getAl(), result.get(0).getV(),
        result.get(0).getW(), result.get(0).getTi(),
bufferConfig, searchList);
    }
    iterationNo++;
    bufferList.add(input);
    bufferSample =MSample.builder()
        .sampleNo("Отклонения")
        .C(bufferConfig.getCDev())
        .Si(bufferConfig.getSiDev())
        .Mn(bufferConfig.getMnDev())
        .P(bufferConfig.getPDev())
        .S(bufferConfig.getSDev())
        .Cr(bufferConfig.getCrDev())
        .Ni(bufferConfig.getNiDev())
        .Mo(bufferConfig.getMoDev())
        .Cu(bufferConfig.getCuDev())
        .Al(bufferConfig.getAlDev())

```

```

        .V(bufferConfig.getVDev())
        .W(bufferConfig.getWDev())
        .Ti(bufferConfig.getTiDev())
        .build();

    bufferList.add(bufferSample);
    return bufferList;
}
}

```

### Файл ImportFromExcel.java:

```

package Flare.Modules;

import java.io.*;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.List;
import java.util.Iterator;

import Flare.MSample;
import org.apache.poi.hssf.usermodel.HSSFCellStyle;
import org.apache.poi.hssf.usermodel.HSSFFont;
import org.apache.poi.hssf.usermodel.HSSFSheet;
import org.apache.poi.hssf.usermodel.HSSFWorkbook;
import org.apache.poi.ss.usermodel.FormulaEvaluator;
import org.apache.poi.ss.usermodel.Cell;
import org.apache.poi.ss.usermodel.CellType;
import org.apache.poi.ss.usermodel.Row;

public class ImportFromExcel {

    private static HSSFCellStyle createStyleForTitle(HSSFWorkbook
workbook) {
        HSSFFont font = workbook.createFont();
        font.setBold(true);
        HSSFCellStyle style = workbook.createCellStyle();
        style.setFont(font);
        return style;
    }

    public static List<MSample> GetExcelData(File ExcelFile)
throws IOException {
        String[] checkList = new
String[]{"Date","Time","Sample_NO", "Quality","Program", "Bloom",
"Cod_1", "Cod_2", "Cod_3", "Cod_4","Kristal","Seria","Product",
"C","Si","Mn","P","S","Cr","Ni","Mo","Cu","Al","As","V","W","Ti"};
        String quality,bloom,cod_1,cod_2,cod_3,cod_4;
        String[] data = new String[24];
        boolean checkHeaderRow = true;
        List<MSample> sampleList = new ArrayList<>();
    }
}

```

```

SimpleDateFormat dt = new SimpleDateFormat("dd.MM.yyyy");
SimpleDateFormat dt1 = new SimpleDateFormat("HH:mm:ss");
// Read XLS file
FileInputStream inputStream = new
FileInputStream(ExcelFile);

// Get the workbook instance for XLS file
HSSFWorkbook workbook = new HSSFWorkbook(inputStream);

// Get first sheet from the workbook
HSSFSheet sheet = workbook.getSheetAt(0);

// Get iterator to all the rows in current sheet
Iterator<Row> rowIterator = sheet.iterator();
//ObjectOutputStream fout = new ObjectOutputStream(new
FileOutputStream("db.dat"));
    Row row = rowIterator.next();
    // Get iterator to all cells of current row
    Iterator<Cell> cellIterator = row.cellIterator();
    for(int i = 0; i< checkList.length;++i) //Проверка
заголовков таблицы на корректность
    {
        Cell cell = cellIterator.next();
        String cellValue = cell.getStringCellValue();
        if(!cellValue.equals(checkList[i]))// Если хоть
один заголовок не соответствует нужному порядку из проверочной
строки, то выбрасываем исключение
        {
            throw new IOException();
        }
    }

while (rowIterator.hasNext()) {
    row = rowIterator.next();
    cellIterator = row.cellIterator();

    //while (cellIterator.hasNext()) {

        /*Cell cell =
sheet.getRow(row.getRowNum()).getCell(0);
        // Change to getCellType() if using POI 4.x
        CellType cellType = cell.getCellTypeEnum();*/

        Cell cell =
sheet.getRow(row.getRowNum()).getCell(0);
        data[0] = dt.format(cell.getDateCellValue()); //
cell = sheet.getRow(row.getRowNum()).getCell(1);
        data[1] = dt1.format(cell.getDateCellValue());
        for(int i = 2; i<=3;++i)
        {
            cell =

```

```

sheet.getRow(row.getRowNum()).getCell(i);
    if(cell == null )
        data[i] = null; // "sampleNo" "Quality",
"Bloom", "Cod_1", "Cod_2", "Cod_3", "Cod_4"
    else if(cell.getCellTypeEnum().name() ==
"STRING")
        data[i] = cell.getStringCellValue();
    else data[i] =
String.valueOf(cell.getNumericCellValue());
}
for(int i = 5; i<=9;++i)
{
    cell =
sheet.getRow(row.getRowNum()).getCell(i);
    if(cell == null )
        data[i-1] = null;
    else if(cell.getCellTypeEnum().name() ==
"STRING")
        data[i-1] = cell.getStringCellValue();
    else data[i-1] =
String.valueOf(cell.getNumericCellValue());
}
cell = sheet.getRow(row.getRowNum()).getCell(11);
if(cell != null) {
    if(cell.getCellTypeEnum().name() == "STRING")
        data[9] = cell.getStringCellValue();
    else
    {
        data[9] =
dt.format(cell.getDateCellValue());
        String splitter[] = data[9].split("\\.");

data[9]=Integer.parseInt(splitter[0])+"*"+Integer.parseInt(splitte
r[1]);
    }
}
else data[9] = null;

for(int i = 13; i<=26;++i)
{
    cell =
sheet.getRow(row.getRowNum()).getCell(i);
    if(cell==null) data[i-3] = null;
    else if(cell.getCellTypeEnum().name() ==
"STRING") {
        data[i-3] = cell.getStringCellValue(); //
        data[i-3]=data[i-3].replace(",",".");
        data[i-3]=data[i-3].replace("<","");
        data[i-3]=data[i-3].replace(">","");
        data[i-3]=data[i-3].replace("~","");
    }
    else data[i-3] =

```

```

String.valueOf(cell.getNumericCellValue());
    }
    System.out.println("");
    if(data[4]!=null)//Убираем .0 после номера блюда
    {
    StringBuffer buffer = new StringBuffer(data[4]);
    buffer.delete(data[4].length()-
2,data[4].length());
    data[4]= buffer.toString();
    }
    int bloomCount = 0;
    String[] bloomDivide;
    for(int i = 0; i<4;++i) {
        try {
            bloomDivide = data[i+5].split("\\*");
            bloomCount +=
Integer.parseInt(bloomDivide[0]);
        } catch (Exception e) {
        }
    }
    MSample sample = MSample.builder()
        .date(data[0])
        .time(data[1])
        .sampleNo(data[2])
        .quality(data[3])
        .bloom(data[4])
        .codes(String.valueOf(bloomCount))
        .seria(data[9])
        .C(data[10])
        .Si(data[11])
        .Mn(data[12])
        .P(data[13])
        .S(data[14])
        .Cr(data[15])
        .Ni(data[16])
        .Mo(data[17])
        .Cu(data[18])
        .Al(data[19])
        .V(data[21])
        .W(data[22])
        .Ti(data[23])
        .build();
    sampleList.add(sample);
    }

    return sampleList;
}
}

```

**Файл Classifier.java:**

```

package Flare.Modules;

import Flare.MSample;
import weka.classifiers.evaluation.Evaluation;
import weka.classifiers.functions.MultilayerPerceptron;
import weka.core.Attribute;
import weka.core.Instances;
import weka.core.WekaPackageManager;
import weka.core.converters.ConverterUtils.DataSource;

import java.io.*;
import java.util.*;
import java.util.stream.Collectors;

public class Classifier {

    public Classifier()
    {
        WekaPackageManager.loadPackages(true);
    }
    public List<Map.Entry<String,Double>> classify(MSample
samples,String classifierName)
    {
        MultilayerPerceptron perceptron;
        Instances example;
        HashMap<String,Double> distribution = new HashMap<>();
        List<Map.Entry<String,Double>> result;
        List<String> qualities;
        double clsLabel[];
        try {
            example = new Instances(new
DataSource(ConvertToCSV(samples).getPath()).getDataSet());
            example.setClassIndex(example.numAttributes() - 1);

            perceptron = (MultilayerPerceptron)
weka.core.SerializationHelper.read(classifierName+".model");

            qualities =((ClassifierInfo)
weka.core.SerializationHelper.read(classifierName+".stats")).getQu
alities();
            for (int i = 0; i < example.numInstances(); i++) {
                clsLabel =
perceptron.distributionForInstance(example.instance(i));
                for (int j = 0; j < qualities.size(); j++) {
                    distribution.put(qualities.get(j), clsLabel[j]);
                    System.out.println(qualities.get(j) + " -> " +
clsLabel[j]);
                }
            }
            result = distribution.entrySet().stream()

```

```

.sorted(Map.Entry.<String,Double>comparingByValue().reversed()).collect(Collectors.toList());
    } catch (Exception e) {
        throw new RuntimeException(e);
    }
    return result;
}

public double trainClassifier(List<MSample> mSamples, String classifierName, String learningRate, String momentum, String epochs) throws Exception {
    MultilayerPerceptron model = new MultilayerPerceptron();

    model.setOptions(weka.core.Utils.splitOptions("-L
"+(learningRate.length()<1?"0.3":learningRate)+" -M
"+(momentum.length()<1?"0.1":momentum)+" -N
"+(epochs.length()<1?"1000":epochs)+" -V 0 -S 0 -E 20 -H a -C -R -I"));

    ConvertToCSV(mSamples);
    DataSource source = new DataSource("converted.csv");
    Instances data = source.getDataSet();

    data.setClassIndex(0);
    model.buildClassifier(data);
    Evaluation eval = new Evaluation(data);
    eval.evaluateModel(model, data);

weka.core.SerializationHelper.write("models/"+classifierName+".model",model);

weka.core.SerializationHelper.write("models/"+classifierName+".stats",new
ClassifierInfo(classifierName,model.getLearningRate(),model.getMomentum(),model.getTrainingTime(),eval.pctCorrect(),(List)Collections.list(data.attribute(0).enumerateValues())));
    return eval.pctCorrect();
}

public ClassifierInfo getClassifierInfo(String classifierName) throws Exception {
    return
(ClassifierInfo)weka.core.SerializationHelper.read("models/"+classifierName);
}

private File ConvertToCSV(MSample samples) throws IOException
{
    File file = new File("converted.csv");
    FileWriter fileWriter;
    fileWriter = new FileWriter(file);

fileWriter.write("Quality,C,Si,Mn,P,S,Cr,Ni,Mo,Cu,Al,V,W,Ti\n");
    fileWriter.write(", "+
        samples.getC()+" "+

```

```

        samples.getSi()+" "+
        samples.getMn()+" "+
        samples.getP()+" "+
        samples.getS()+" "+
        samples.getCr()+" "+
        samples.getNi()+" "+
        samples.getMo()+" "+
        samples.getCu()+" "+
        samples.getAl()+" "+
        samples.getV()+" "+
        samples.getW()+" "+
        samples.getTi()+"\n"

    );
    fileWriter.flush();
    fileWriter.close();
    return file;
}
private File ConvertToCSV(List<MSample> samples) throws
IOException {
    File file = new File("converted.csv");
    FileWriter fileWriter;
    MSample temp;

    fileWriter = new FileWriter(file);

fileWriter.write("Quality,C,Si,Mn,P,S,Cr,Ni,Mo,Cu,Al,V,W,Ti\n");
    for (int i = 0; i < samples.size(); i++) {
        temp = samples.get(i);

        fileWriter.write(
            temp.getQuality()+" "+
            (temp.getC()==null? "":
temp.getC()+" "+
            (temp.getSi()==null? "":
temp.getSi()+" "+
            (temp.getMn()==null? "":
temp.getMn()+" "+
            (temp.getP()==null? "":
temp.getP()+" "+
            (temp.getS()==null? "":
temp.getS()+" "+
            (temp.getCr()==null? "":
temp.getCr()+" "+
            (temp.getNi()==null? "":
temp.getNi()+" "+
            (temp.getMo()==null? "":
temp.getMo()+" "+
            (temp.getCu()==null? "":
temp.getCu()+" "+
            (temp.getAl()==null? "":
temp.getAl()+" "+
            (temp.getV()==null? "":

```



```

temp.getV()+"", "+
                                (temp.getW()==null? "":
temp.getW()+"", "+
                                (temp.getTi()==null? "":
temp.getTi()+"\n"
                                );
                                fileWriter.flush();
                                }
                                fileWriter.close();
                                return file;
                                }
}

```

### Файл Preprocessor.java:

```

package Flare.Modules;

import net.sourceforge.tess4j.Tesseract;
import net.sourceforge.tess4j.TesseractException;
import nu.pattern.OpenCV;
import org.opencv.core.Mat;
import org.opencv.core.MatOfPoint;
import org.opencv.core.Scalar;
import org.opencv.core.Size;
import org.opencv.imgcodecs.Imgcodecs;
import org.opencv.imgproc.Imgproc;

import java.awt.*;
import java.io.File;
import java.util.ArrayList;
import java.util.Collections;
import java.util.List;
import java.util.stream.Collectors;
import java.util.stream.Stream;

public class Preprocessor {
    final String IMAGE_FOLDER = "tesdata/preprocessor/";
    Mat image;

    private File getProcessedImage()
    {
        OpenCV.loadLocally();
        image = Imgcodecs.imread(IMAGE_FOLDER+"screenshot.png");

        Mat imgGray = new Mat();
        Imgproc.cvtColor(image, imgGray, Imgproc.COLOR_BGR2GRAY);
        Imgcodecs.imwrite(IMAGE_FOLDER+"gray.png", imgGray);

        Mat imgGaussianBlur = new Mat();
        Imgproc.GaussianBlur(imgGray, imgGaussianBlur, new Size(5,
3), 0);

```

```

        Imgcodecs.imwrite(IMAGE_FOLDER+"gaussian_blur.png",
imgGaussianBlur);
        /*Mat imgSobel = new Mat();
        Imgproc.Sobel(imgGaussianBlur, imgSobel, -1, 1, 0);
        Imgcodecs.imwrite("sobel.png", imgSobel);*/

        Mat imgThreshold = new Mat();
        Imgproc.threshold(imgGaussianBlur, imgThreshold, 250, 255,
Imgproc.THRESH_BINARY);
        Imgcodecs.imwrite(IMAGE_FOLDER+"binary.png",
imgThreshold);

        Mat imgMorphClose = new Mat();
        Mat element =
Imgproc.getStructuringElement(Imgproc.MORPH_RECT, new Size(15,
15));
        Imgproc.morphologyEx(imgThreshold, imgMorphClose,
Imgproc.MORPH_CLOSE, element);
        Imgcodecs.imwrite(IMAGE_FOLDER+"morph_close.png",
imgMorphClose);

        /*List<MatOfPoint> contours = new ArrayList<>();
        Mat hierarchy = new Mat();

        Imgproc.findContours(imgMorphClose, contours, hierarchy, Imgproc.RETR
_TREE, Imgproc.CHAIN_APPROX_SIMPLE);
        Imgproc.drawContours(image, contours, -1, new Scalar(0, 255,
0), 2, Imgproc.LINE_AA);
        Imgcodecs.imwrite("preprocess.png", image);*/
        image =Imgcodecs.imread(IMAGE_FOLDER+"morph_close.png");
        Mat image2 = new Mat();

        Imgproc.threshold(Imgcodecs.imread(IMAGE_FOLDER+"gray.png"), image2
, 50, 255, Imgproc.THRESH_BINARY);

        for(int i = 0; i<image.height();i++)
        {
            for(int j = 0; j<image.width();j++) {
                double[] pixel = image.get(i, j);
                double[] pixel2 = image2.get(i, j);
                pixel[0] *= pixel2[0];
                pixel[1] *= pixel2[1];
                pixel[2] *= pixel2[2];
                image.put(i, j, pixel);
            }
        }
        Imgcodecs.imwrite(IMAGE_FOLDER+"preprocess.png", image);
        return new File(IMAGE_FOLDER+"preprocess.png");
    }
    public List doTesseract() throws TesseractException
    {
        Tesseract tesseract = new Tesseract();

```

```

String text;
List<String> lines;
List<String> sampleList=new ArrayList<>();
    tesseract.setDatapath("tessdata");
    text = tesseract.doOCR(getProcessedImage());
text=text.replace("<","").replace(">","");
lines =
Stream.of(text.split("\n")).collect(Collectors.toList());
int samples = lines.size()/4; //Ибо строчек хим элементов
4;

for (int i = 0; i < samples; i++) {
    text= "";
    for (int j = 0; j < 4; j++) {
        text+= lines.get((j*samples)+i);
        text+=" ";
    }
    sampleList.add(text);
}

for (int i = 0; i < samples; i++) {
    lines=Stream.of(sampleList.get(i).split("
")).collect(Collectors.toList());
    Collections.swap(lines,10,11);
    Collections.swap(lines,11,14);
    sampleList.add(i,lines.stream()
        .limit(13)
        .map(x->
            {
                if(!x.contains("."))
                {
                    return
x.charAt(0)+"."+x.substring(1);
                }
                else return x;
            }
        ).collect(Collectors.joining(" ")));
    sampleList.remove(i+1);
}
return sampleList;
}
}

```