

АНОТАЦІЯ

Шифр наукової роботи: «Oasis».

Тема роботи: Інформаційна технологія прогнозування якості діяльності в e-learning.

Актуальність. E-learning є перспективною технологією навчання в інформаційному суспільстві. Головна проблема навчальних систем – нездатність адаптації до індивідуальних особливостей кожної людини, що навчається. Впровадження інтелектуальних технологій управління діалоговим процесом в e-learning стримується через відсутність моделей взаємодії людей з машинами.

Мета: Розробка інформаційної технології прогнозування якості діяльності в e-learning.

Об'єкт дослідження. Діяльність в технології e-learning.

Предмет дослідження. Інформаційна технологія прогнозування якості діяльності.

Робота складається з вступу, двох розділів, висновку, списку літератури, чотирьох додатків; включає 29 сторінок, 22 рисунків, 14 літературних джерел.

Публікації. За матеріалами дослідження опубліковано 6 наукових робіт. Список робіт та копії публікацій додаються.

Апробації. Результати доповідались на 4 наукових конференціях:

- Міжнародна науково-технічної конференція «Інформатика, Математика, Автоматика ІМА::2018»(Суми, 05-09 лютий 2018);
- XII Міжнародна науково-практична конференція «Цифрові технології в навчанні, науці, суспільстві» (Петрозаводск, 4-6 грудня 2018 року);
- X Міжнародна студентська конференція «Перший крок у науку»;
- Міжнародна науково-технічна конференція «Інформатика, Математика, Автоматика ІМА::2019»(Суми, 23-26 лютий 2019 року).

Впровадження. Результати впроваджено в навчальний процес Сумського державного університету.

Ключові слова: e-learning , нейронна мережа, діалогова взаємодія, електронне навчання, безпомилковість.

**Наукова робота на конкурс за напрямом:
Інформаційні системи і технології**

на тему:

**ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ПРОГНОЗУВАННЯ
ЯКОСТІ ДІЯЛЬНОСТІ В E-LEARNING**

2020

ЗМІСТ

ВСТУП	3
1 АНАЛІЗ ОРГАНІЗАЦІЇ СИСТЕМ ДЛЯ ПРОГНОЗУВАННЯ ЯКОСТІ ДІЯЛЬНОСТІ В E-LEARNING	4
1.1 Інтелектуальний агент управління процесом навчання	4
1.2 Висновки до розділу 1	7
2 РОЗРОБКА ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ПРОГНОЗУВАННЯ ЯКОСТІ ДІЯЛЬНОСТІ В ТЕХНОЛОГІЇ НАВЧАННЯ.....	9
2.1 Вибір програмних засобів для реалізації	9
2.2 Створення нейронних мереж.....	10
2.3 Проектування інтерфейсу	17
2.4 Комп'ютерні експерименти по вибору оптимальних траєкторій навчання.....	23
2.5 Висновки до розділу 2.....	26
ВИСНОВКИ.....	27
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	28
ДОДАТОК А – Типи нейронних мереж Nntool	30
ДОДАТОК Б – Лістинг М-функцій.....	31
ДОДАТОК В – Лістинг програми	32
ДОДАТОК Г – Інструкція користувача	46

ВСТУП

Актуальність роботи. E-learning є перспективною технологією навчання в інформаційному суспільстві. Головна проблема навчальних систем – нездатність адаптації до індивідуальних особливостей кожної людини, що навчається. Впровадження інтелектуальних технологій управління діалоговим процесом в e-learning стримується через відсутність моделей взаємодії людей з машинами.

Об'єкт дослідження. Діяльність в технології e-learning.

Предмет дослідження. Інформаційна технологія прогнозування якості діяльності.

Мета роботи. Розробка інформаційної технології прогнозування якості діяльності в e-learning.

Наукова новизна. На відміну від існуючих апаратних моделей запропоновано невибагливий до технічного забезпечення модель, орієнтована на штучні нейронні мережі для прогнозування якості діяльності людини в технології навчання.

Публікації. Оpubліковано 6 наукових робіт.

Апробація. Результати роботи доповідались на конференціях:

- Міжнародна науково-технічної конференція «Інформатика, Математика, Автоматика ІМА::2018»(Суми, 05-09 лютий 2018);
- XII Міжнародна науково-практична конференція «Цифрові технології в навчанні, науці, суспільстві» (Петрозаводск, 4-6 грудня 2018 року);
- X Міжнародна студентська конференція «Перший крок у науку»;
- Міжнародна науково-технічна конференція «Інформатика, Математика, Автоматика ІМА::2019»(Суми, 23-26 лютий 2019 року).

Впровадження. Результати впроваджені в навчальні процеси Центру дистанційного навчання Сумського державного університету.

1 АНАЛІЗ ОРГАНІЗАЦІЇ СИСТЕМ ДЛЯ ПРОГНОЗУВАННЯ ЯКОСТІ ДІЯЛЬНОСТІ В E-LEARNING

1.1 Інтелектуальний агент управління процесом навчання

Швидкий розвиток суспільства та його інформаційно-технологічної бази сприяли удосконаленню комп'ютерної техніки, технологій програмування та, як наслідок, появи нових засобів зв'язку, розвитку й застосуванню нових інформаційно-комунікаційних технологій (ІКТ), а також засобів навчання [1].

Специфічні якісні властивості електронного навчання [2]:

- гнучкість і адаптивність навчального процесу до потреб і можливостей студентів, які не відвідують регулярних занять, а працюють у зручний (як для викладача, так і для студента) для такої роботи час у зручному місці й зручному темпі;
- модульність побудови навчальних програм;
- нова роль викладача: викладач координує навчально-пізнавальний процес, коригує курс, який викладає, керує навчальними проектами, перевіряє поточні завдання, консультує при складанні індивідуального навчального плану, управляє навчальними групами взаємопідтримки;
- спеціалізовані форми контролю якості навчальних досягнень: традиційні форми контролю якості освіти (співбесіди, практичні, курсові та проектні роботи, екстернат, робота в середовищі комп'ютерних інтелектуальних тестових систем тощо);
- використання спеціалізованих засобів навчання.

В Сумському державному університеті проводяться роботи по впровадженню нових технологій електронного навчання.

Існує низка факторів, що впливають на процес отримання знань. Студенти мають:

- різний рівень підготовки;
- різний рівень мотивації (дехто прагне п'ятірку, а декого влаштовую і трійка);
- різні індивідуальні особливості (мається на увазі різна здатність сприймання нової інформації, що подається);
- різний час, що може бути витрачений на вивчення тієї чи іншої теми;
- різні можливості вивчати складні модулі.

Таким чином, існує велика кількість параметрів, від яких залежить як саме необхідно побудувати систему діалогової взаємодії.

В зв'язку з низьким рівнем інтерактивності і відсутності механізмів адаптації навчального середовища до особливостей людини, проводяться роботи по створенню інтелектуального агента управління процесом навчання, що дозволить враховувати мотивацію, рівень підготовки та інші особливості кожного студента. Вирішення задачі оптимізації якості людино-машинної взаємодії в навчальному середовищі дозволить підвищити ефективність засвоєння матеріалу курсу та мотивацію студентів і викладачів.

Інтелектуальний агент може функціонувати в умовах наявності єдиного інформаційного простору вищих навчальних закладів [3]. До його складу входять наступні системи:

T – розвинена транспортна система доставки навчальних матеріалів та організації діалогової взаємодії;

E – система баз даних електронних навчальних модулів;

Me – система баз даних ергономічних моделей електронних навчальних модулів і моделей можливої діалогової взаємодії з ними;

Mm – система баз даних і знань про характеристики і переваги студентів;

St – система статистичних баз даних про результати взаємодії студентів з електронними навчальними модулями (характеристики випадкових величин часу і показників успішності навчання);

Im – система оперативної ідентифікації та визначення характеристик поточного стану студента;

Is – система оперативної ідентифікації поточного стану середовища.

Розглянемо принцип функціонування інтелектуального агента.

На першому етапі відбувається ідентифікація студента - визначення характеристик моделі студента і середовища (Im, Is).

Далі вибираються з безлічі альтернативних модулів, що відповідають цілі поточного сеансу, ті безлічі модулів, які відповідають вимогам системи переваг (Me, Mm). На цьому етапі використовується модель, заснована на апараті нечіткої логіки.

Після цього видаються рекомендації з організації ефективного діалогу з обраним модулем (в циклі по точках можливого управління діалогом).

Цей етап включає наступне:

- Генерація альтернативних діалогових технологій і формування моделей діалогу, що включають елементи навчальних процедур, самоконтролю, корекції і т.п. (Me). Використовується апарат функціональних мереж (ФМ) [3].
- Формування вихідних даних (для оцінювання показників часу і успішності реалізації процедур навчання) для окремих елементів діалогових процедур (при заданих характеристиках студента, модуля, середовища).
- Оцінка показників альтернативних варіантів організації діалогу. Використовується апарат ФМ [4].
- Рекомендації по вибору варіанта організації діалогу в поточній точці.

На рисунку 1.1 представлені основні функціональні блоки і принцип роботи агента.

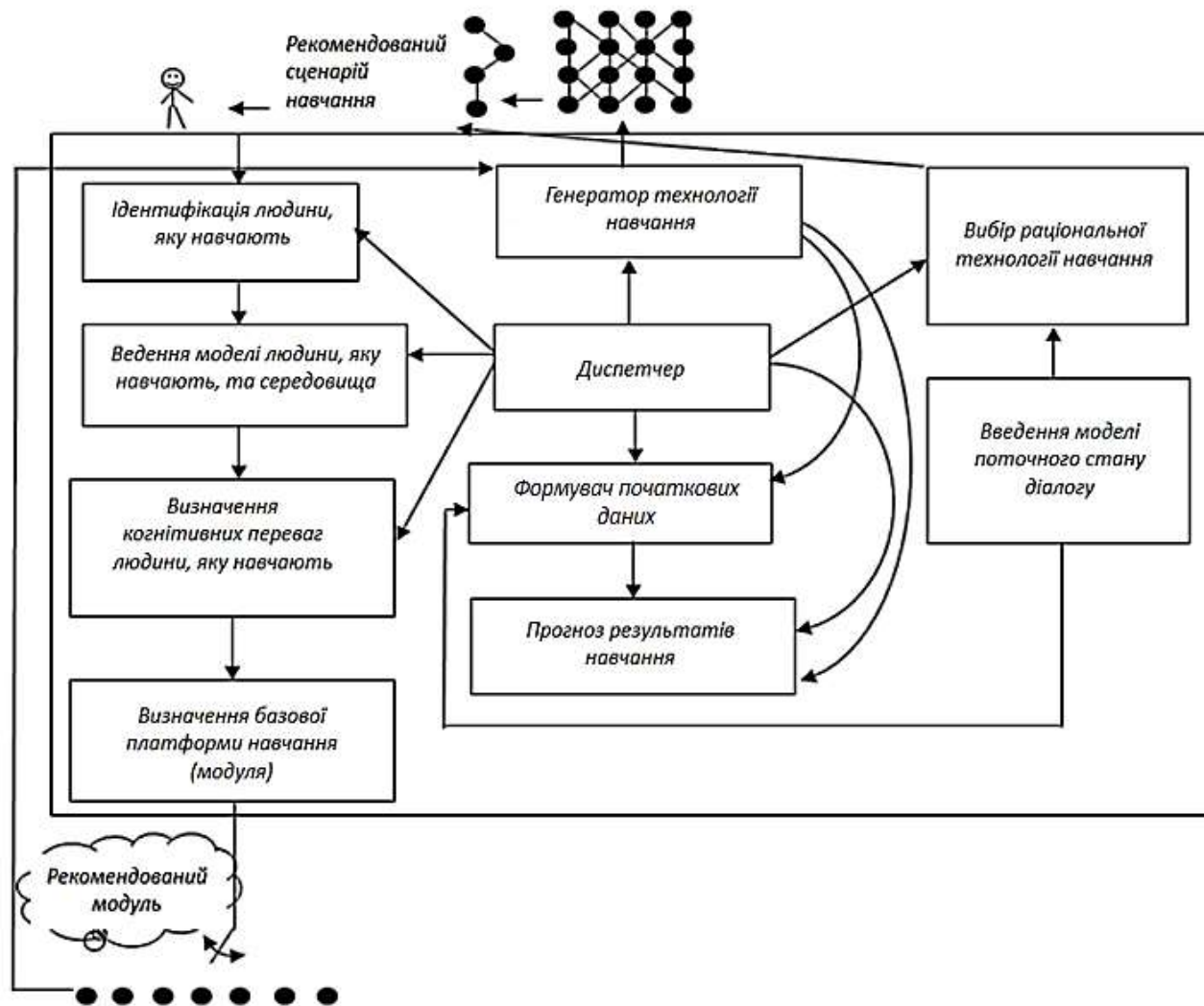


Рисунок. 1.1 – Загальна структура інтелектуального агента

Із зазначених модулів агента, необхідно розробити модуль прогнозування вхідних даних для оптимізації. Також реалізувати їх перенесення в модель інтелектуального агента, використовуючи широкі можливості Matlab – пакет прикладних програм для числового аналізу, а також мови програмування *c#* програмного продукту Microsoft Visual Studio.

1.2 Висновки до розділу 1

Сучасні університети пропонують зручну і доступну форму отримання вищої освіти – використання електронного навчання. Були виявлені наступні недоліки процесу даного навчання:

- Проблеми пристосування до індивідуальних особливостей кожної людини, що навчається, тобто низька адаптивність електронного навчання.
- Слабка пристосованість до роботи з мобільними додатками.
- Важкість прогнозування результатів навчання студента.
- Важкість гнучкого маніпулювання навчанням.

Тому проводяться роботи по впровадженню інтелектуального агента управління процесом навчання. Такий агент дозволить враховувати мотивацію, рівень підготовки та інші особливості студентів.

Впровадження технології інтелектуального агента дозволить визначити оптимальну стратегію людино-машинної взаємодії, а також спрогнозувати результат проходження загального підсумкового контролю.

Необхідно розробити модуль прогнозування вхідних даних для оптимізації та реалізувати їх перенесення в модель інтелектуального агента.

2 РОЗРОБКА ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ПРОГНОЗУВАННЯ ЯКОСТІ ДІЯЛЬНОСТІ В ТЕХНОЛОГІЇ НАВЧАННЯ

2.1 Вибір програмних засобів для реалізації

Для розроблення програмного модуля підготовки даних для оптимізації було обрано середовище Microsoft Visual Studio 2010 та для побудови нейронних мереж – середовище Matlab.

Microsoft Visual Studio - лінійка продуктів компанії Microsoft, що включає інтегроване середовище розробки програмного забезпечення і ряд інших інструментальних засобів. Даний продукт дозволяє розробляти як консольні додатки, так і додатки з графічним інтерфейсом, в тому числі з підтримкою технології Windows Forms. Також є можливість створювати веб-сайти, веб-додатки, веб-служби як в рідному, так і в керованому кодах для всіх платформ, що підтримують Windows, Windows Mobile, Windows CE, .NET Framework, Xbox, Windows Phone .NET Compact Framework, iSilverlight.

Visual Studio дозволяє створювати і підключати сторонні додатки (плагіни) для розширення функціональності практично на кожному рівні, включаючи додавання підтримки систем контролю версій вихідного коду (як, наприклад, Subversion і Visual SourceSafe), додавання нових наборів інструментів (наприклад, для редагування і візуального проектування коду на предметно-орієнтованих мовах програмування) або інструментів для інших аспектів процесу розробки програмного забезпечення (наприклад, клієнт Team Explorer для роботи з Team Foundation Server).

MATLAB (скорочення від англ. «Matrix Laboratory», в українській мові вимовляється як Матлаб) - пакет прикладних програм для вирішення завдань технічних обчислень і однойменна мова програмування, що

використовується в цьому пакеті. MATLAB використовують велика кількість інженерних і наукових працівників, він працює на більшості сучасних операційних систем, включаючи Linux, Mac OS, Solaris.

Мова MATLAB є високорівневою мовою програмування, що включає засновані на матрицях структури даних, широкий спектр функцій, інтегроване середовище розробки, об'єктно-орієнтовані можливості і інтерфейси до програм, написаних на інших мовах програмування.

Програми, написані на MATLAB, бувають двох типів - функції і скрипти. Функції мають вхідні і вихідні аргументи, а також власний робочий простір для зберігання проміжних результатів обчислень і змінних. Скрипти ж використовують загальний робочий простір. Як скрипти, так і функції не компілюються в машинний код і зберігаються у вигляді текстових файлів. Існує також можливість зберігати так звані pre-parsed програми - функції і скрипти, оброблені у вигляді, зручний для машинного виконання. У загальному випадку такі програми виконуються швидше звичайних, особливо якщо функція містить команди побудови графіків.

2.2 Створення нейронних мереж

Рішення задачі прогнозування передбачає отримання ймовірнісно-часових показників якості діяльності студента в електронній системі навчання при різних параметрах учня і середовища (рівні мотивації, втоми, когнітивного комфорту) [5]. Основною метою є визначення ймовірності безпомилкового виконання і математичного сподівання часу виконання блоку навчання (модуля). Отримані дані надходять до програмного інтелектуального агента для розрахунку оптимальної траєкторії навчання.

Для прогнозування даних був обраний метод штучних нейронних мереж (ШНМ). ШНМ — математичні моделі, а також їхня програмна та апаратна реалізація, побудовані за принципом функціонування біологічних

нейронних мереж — мереж нервових клітин живого організму [6]. Системи, архітектура і принцип дії базується на аналогії з мозком живих істот. Ключовим елементом цих систем виступає штучний нейрон як імітаційна модель нервової клітини мозку — біологічного нейрона [7].

Для створення нейронної мережі використовувався пакет прикладних програм для числового аналізу – Matlab R2012b. Дана система включає в себе велику кількість тулбоксів, серед яких був використаний Nntool – графічний інтерфейс для роботи з нейронними мережами. Він дозволяє обирати структуру ШНМ з широкого списку і надає велику кількість алгоритмів для навчання для кожного типу мережі.

Для обрання типу нейронної мережі було розглянуто п'ятнадцять типів мереж, список наведено в додатку А. Для їх порівняння спочатку необхідно провести навчання.

2.2.1 Навчання нейромереж

Навчання мереж – це навчання з учителем – один із способів машинного навчання, в ході якого випробувана система примусово навчається за допомогою наявної множини прикладів «стимул-реакція» з метою визначення «реакції» для «стимулів», які не належать наявній множини прикладів [8].

Для формування навчальної вибірки використовувалися статистичні дані, які були отримані в результаті спостереження за великою кількістю студентів з різними рівнями підготовки. Дані збережені в Excel-файлі, тому для перенесення їх в Matlab використовувалась надбудова Excel Link - дозволяє обмінюватися даними між Matlab і Microsoft Excel, забезпечуючи безпрецедентні можливості для аналізу, обробки і представлення даних.

Розглянемо детальніше вхідні параметри для навчання.

1) Мотивація – це рівень зацікавленості студента в отриманні тієї чи іншої оцінки. Вона приймає значення від 1 до 3, де:

- 1 – низький рівень мотивації,
- 2 – середній рівень мотивації,
- 3 – високий рівень мотивації.

2) Вхідний контроль – оцінка в межах від 0 до 1 вхідної перевірки знать студента.

3) Рівень складності навчання приймає значення від 1 до 3, де:

- 1 – низький рівень складності,
- 2 – середній рівень складності,
- 3 – високий рівень складності.

4) Функціональний стан студента – його самопочуття на момент проходження навчальних курсів, оцінюється в межах від 1 до 3, де:

- 1 – низький рівень функціонального стану,
- 2 – середній рівень функціонального стану,
- 3 – високий рівень функціонального стану.

5) Когнітивний комфорт – показник комфортності середовища, с якому знаходиться студент в момент проходження навчального курсу. Приймає значення від 0 до 1.

Цільовими значеннями є:

- ймовірність безпомилкового виконання підмодулю,
- математичне сподівання часу виконання підмодулю,
- наявність відмови від діяльності.

2.2.2 Вибір типу нейронної мережі

Наступним етапом необхідно обрати тип мережі. Для цього були визначені помилки функціонування кожної з них. Мережі були попередньо навчені, при цьому навчальна вибірка (вхідні дані та цільові значення) для кожної мережі була однаковою. В результаті отримали значення помилок, наведені у таблиці 2.2.

Таблиця 2.2 – Порівняльна таблиця нейронних мереж

Тип мережі	Помилка
Cascade-forward backprop	[0.134]
Generalized regression	[0.09647]
Elman backprop	[0.024776]
Competitive	[0.00034962]
Probalistic	[5.104e-05]
Feed-forward backprop	[2.8276e-09]

Для прогнозування даних була обрана нейронна мережа з прямим розповсюдженням, адже в порівнянні з іншими її помилка виявилась найнижчою. На рисунку 2.1 представлений графік, що демонструє значення помилок попередньо розглянутих мереж.

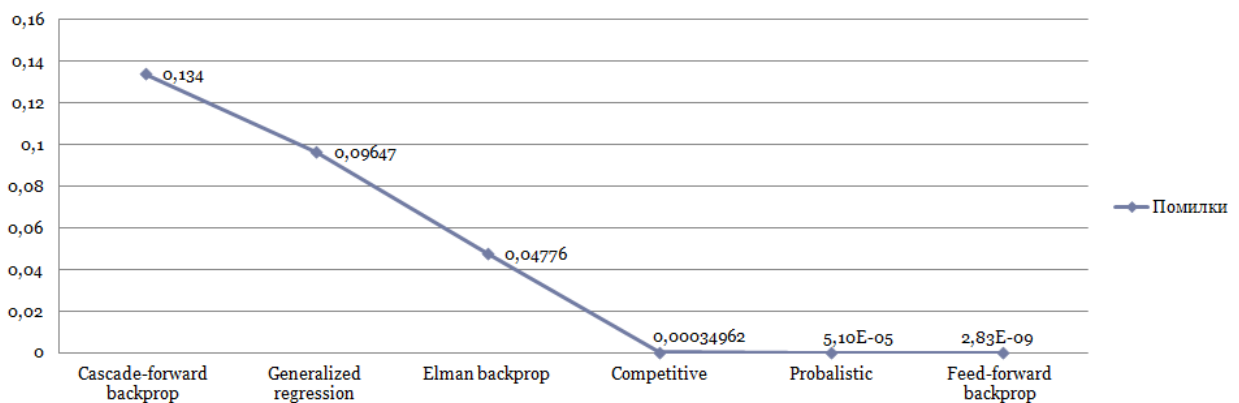


Рисунок 2.1 – Порівняльний графік помилок

Нейронна мережа прямого поширення, нейромережа прямого розповсюдження (англ. Feed-forward neural network) — вид нейронної мережі, в якій сигнали поширюються в одному напрямку, починаючи від вхідного шару нейронів, через приховані шари до вихідного шару і на вихідних нейронах отримується результат опрацювання сигналу [9]. В мережах такого виду немає зворотніх зв'язків.

2.2.3 Вибір функцій для створення нейронної мережі

Для вибору функції налаштування для режиму адаптації були порівняні LEARNGDM - функція налаштування методом градієнтного спуску з обуренням, і LEARNGD - функція налаштування методом градієнтного спуску. Результати наведені в таблиці 2.3.

Таблиця 2.3 - Порівняльна таблиця функцій налаштування для режиму адаптації

Функція налаштування	Помилка
LEARNGDM	[2.8276e-09]
LEARNGD	[0.00014171]

Порівняльна таблиця функцій навчання із них наведена в таблиці 2.4. В результаті порівняння отриманих значень помилок була обрана функція TRAINLM.

Таблиця 2.4 - Порівняльна таблиця функцій навчання

Функція навчання	Помилка
TRAINBR	[0.012337]
TRAINGD	[0.001174]
TRAINLM	[2.8276e-09]
TRAINR	[0.01068]

Останнім етапом було обрання функції оцінки якості навчання. Були розглянуті наступні:

- mse – середньоквадратична помилка;
- msereg - середньоквадратична помилка w/reg;
- sse - сумарна середньоквадратична помилка.

Після порівняння отриманих значень помилок, що наведені в таблиці 2.5, була обрана функція MSE.

Таблиця 2.5 - Порівняльна таблиця функцій оцінки якості навчання

Функція якості навчання	Помилка
MSE	[2.8276e-09]
MSEREG	[0.042725]
SSE	[0.0081011]

2.2.4 Архітектура нейронних мереж

Розглянемо детальніше архітектуру кожної створеної нейронної мережі. На рисунку 2.2 зображена схема першої мережі. На її вхід подаються п'ять параметрів:

- мотивація,
- вхідний контроль,
- рівень складності,
- функціональний стан,
- когнітивний комфорт.

Мережа має два прихованих шари – тобто нейрони, які пов'язані лише з іншими нейронами. В результаті її симуляції на виході отримуємо ймовірність безпомилкового виконання підмодулю.

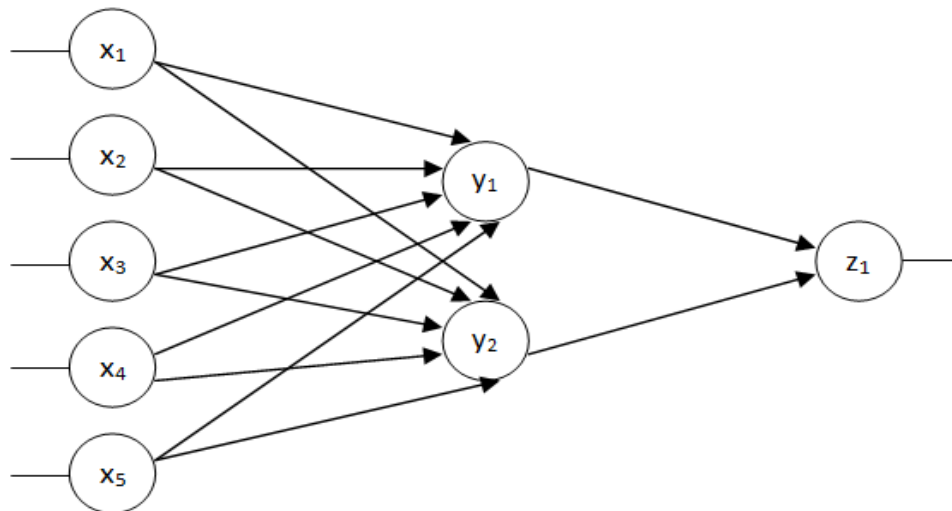


Рисунок 2.2 - Схема першої нейронної мережі:

x_1, x_2, x_3, x_4, x_5 – входи, y_1, y_2 – приховані шари, z_1 – вихід

На рисунку 2.3 зображена архітектура другої нейронної мережі, побудована в тулбоксі Nntool. На її вхід подаються шість параметрів:

- мотивація,
- вхідний контроль,
- рівень складності,
- функціональний стан,
- когнітивний комфорт,
- ймовірність безпомилкового виконання підмодулю.

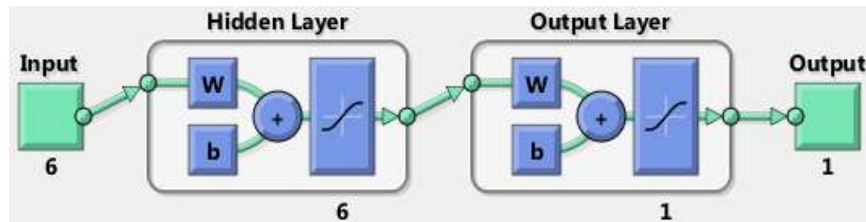


Рисунок 2.3 - Архітектура другої нейронної мережі

Мережа має шість прихованих шарів. В результаті її симуляції на виході отримуємо математичне сподівання часу виконання підмодулю.

Архітектура третьої нейронної мережі представлена на рисунку 2.4. Вона побудована в тулбоксі Nntool. На вхід мережі подаються наступні параметрів:

- мотивація,
- вхідний контроль,
- рівень складності,
- функціональний стан,
- когнітивний комфорт.

На виході отримуємо наявність відмови від діяльності.

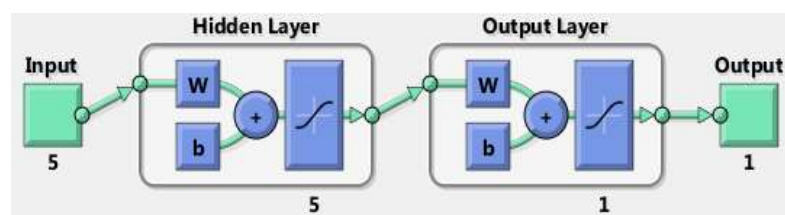


Рисунок 2.4 - Архітектура третьої нейронної мережі

2.3 Проектування інтерфейсу

Після навчання мереж необхідно створити бібліотеки, які потрібні для поєднання можливостей Matlab та Visual Studio. Для цього попередньо був налаштований зовнішній C/C++ компілятор для Matlab.

Також був встановлений Matlab Compiler RunTime (MCR) – середовище виконання компонентів Matlab. Він необхідний для коректної роботи програми, що використовує методи компонентів, що створені на .NET Builder [10].

Для створення .NET компонентів необхідно написати три m-функції, які реалізують передачу входів, виклик нейронних мереж та передачу спрогнозованих даних. Лістинг наведено в додатку Б.

Далі був використаний Deployment Tool - середовище розробки. При створенні компонента потрібно задати ім'я класу і ім'я компонента. Це ім'я компонента також визначає ім'я збірки, яка здійснює компонент. Ім'я класу позначає назву того класу, який інкапсулює функції MATLAB.

Для звернення до методів створеного компонента, були створені екземпляри класів, створених .NET Builder і потім викликалися методи, які інкапсулюють функції MATLAB.

Для проектування інтерфейсу підсистеми формування даних була використана мова програмування c# та програмний продукт Microsoft Visual Studio 2010. Лістинг програми наведено у додатку В.

На рисунку 2.5 представлено головне вікно створеної програми. Меню містить такі пункти:

- Створити новий елемент;
- Редагувати список;
- Розрахувати;
- Результат;
- Відправити в модель агента.

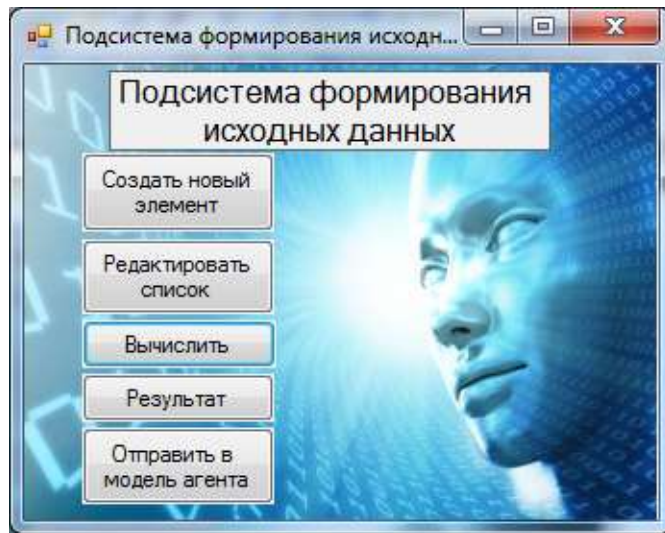


Рисунок 2.5 – Головне вікно створеної програми

Для додавання інформації про нового студента необхідно натиснути на кнопку «Создать новый элемент» головного меню. В результаті відкриється вікно (Рисунок 2.6), де у відповідні поля необхідно внести всю інформацію. Реалізовано обробку виключних ситуацій – якщо користувач натисне на кнопку «Сохранить» до того, як буде введено всю інформацію, на екран буде виведено повідомлення про помилку (Рисунок 2.7).

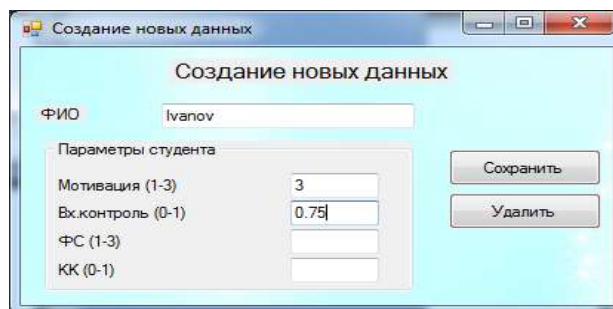


Рисунок 2.6 – Вікно «Создание новых данных»

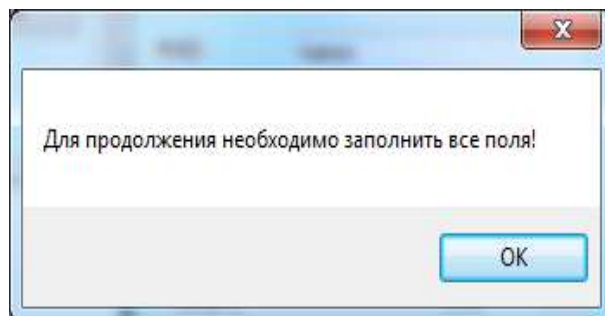


Рисунок 2.7 – Повідомлення про помилку

В результаті коректного введення даних вони зберігаються в текстовий файл, про що свідчить повідомлення для користувача (Рисунок 2.8).

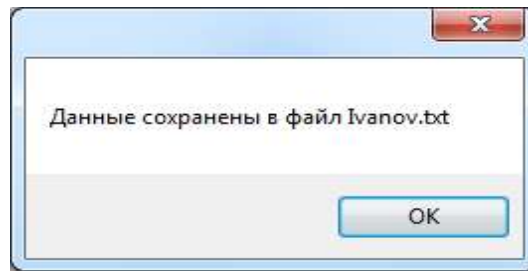


Рисунок 2.8 – Повідомлення про збереження інформації про студента

Для редагування або ж видалення існуючих даних необхідно натиснути на кнопку «Редактировать список» головного меню програми. В результаті відкриється вікно, зображене на рисунку 2.9.

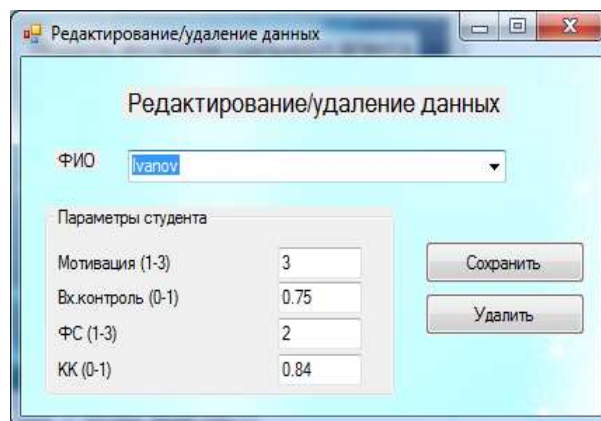


Рисунок 2.9 – Вікно «Редактирование/удаление данных»

Спочатку необхідно обрати із випадаючого списку потрібного студента. Для редагування параметрів - внести всі необхідні зміни та натиснути на кнопку «Сохранить». В результаті текстовий файл буде відредаговано відповідно введеним даним. Для видалення всієї інформації про студента, включаючи і його спрогнозовані результати, достатньо натиснути на кнопку «Удалить». Про успішне видалення свідчить повідомлення, представлене на рисунку 2.10.

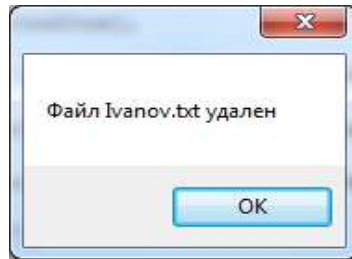


Рисунок 2.10 – Повідомлення про видалення інформації про студента

В результаті вибору користувачем пункту головного меню «Вычислить», відкриється вікно «Прогнозирование входных данных для оптимизации» (Рисунок 2.11).

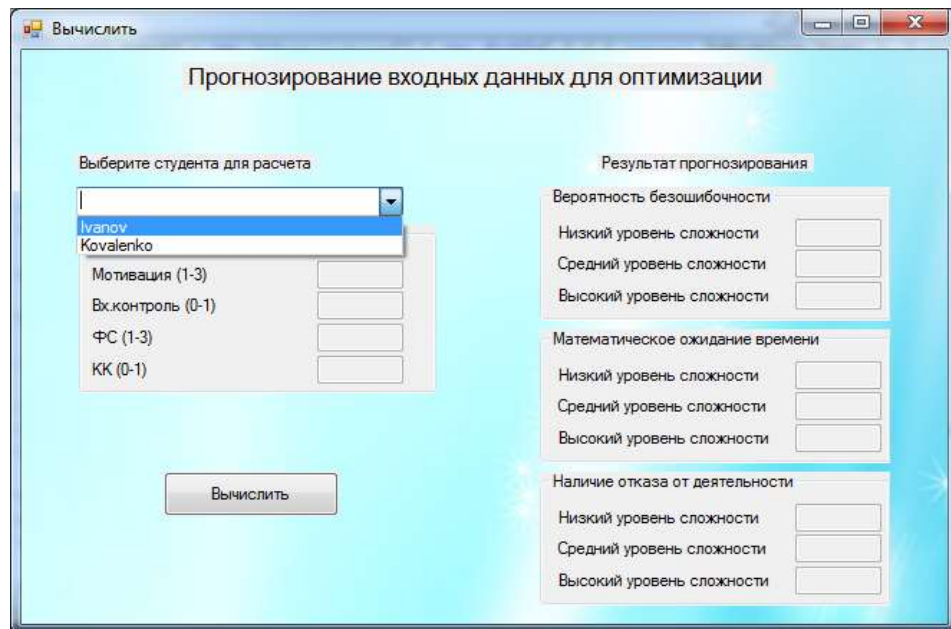


Рисунок 2.11 – Вікно «Прогнозирование входных данных для оптимизации»

Далі користувач повинен обрати потрібного студента, в результаті чого поля з параметрами автоматично заповняться інформацією із раніше збереженого текстового файлу. У випадку, коли кнопку «Вычислить» даної форми була натиснута до вибору студента, користувач отримає повідомлення про помилку (Рисунок 2.12).

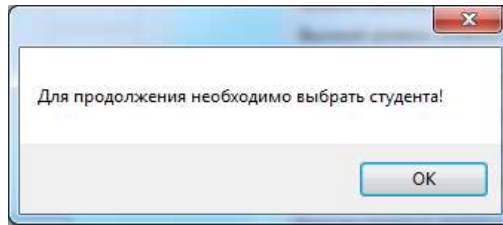


Рисунок 2.12 – Повідомлення про необхідність обрання студента

Для прогнозування безпомилковості виконання підмодуля, математичного сподівання часу виконання та наявності відмови від діяльності, враховуючи рівень складності навчання, користувачу потрібно натиснути на кнопку «Вычислить» даного вікна. При цьому відбувається виклик функцій Matlab. Після симуляції трьох мереж спрогнозовані дані автоматично заносяться у відповідні поля даної форми. Також вони автоматично зберігаються у текстовий файл (Рисунок 2.13).

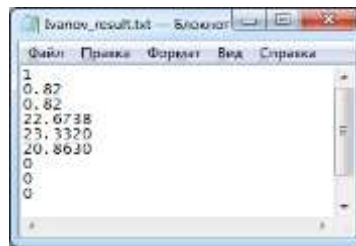


Рисунок 2.13 – Текстовий файл з результатами

Для перегляду раніше спрогнозованих даних для кожного студента необхідно обрати пункт меню «Результат». В результаті з'являється вікно, що зображене на рисунку 2.14.

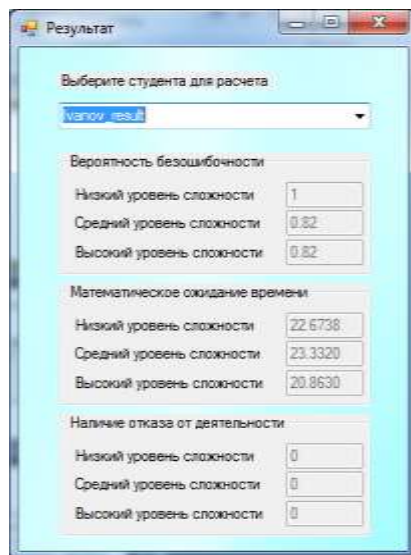


Рисунок 2.14 – Результат

Також маємо змогу перенести спрогнозовані дані в модель агента. Для цього необхідно натиснути на кнопку «Отправить в модель агента» головної форми програми. В результаті відкриється вікно «Перенос данных» (Рисунок 2.15).

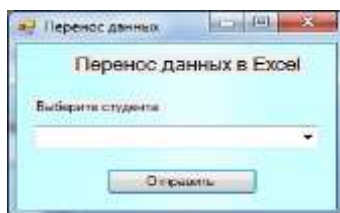


Рисунок 2.15 – Вікно «Перенос данных»

Реалізовано обробку виключних ситуацій – якщо користувач натисне на кнопку «Отправить» до того, як буде обрано студента із випадючого списку, на екран буде виведено повідомлення про помилку (Рисунок 2.16).

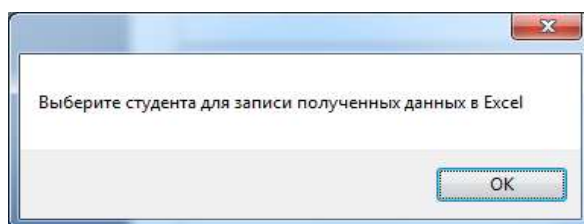


Рисунок 2.16 – Повідомлення про необхідність обрання студента

В результаті коректної роботи даної функції спрогнозовані дані заносяться в модель агента (Рисунок 2.17), після чого їх можна використовувати для оптимізації.

	A	B
1		Ivanov
2	B1n=	1
3	B1s=	0,82
4	B1v=	0,82
7	M(Xp)n=	22,6738
8	M(Xp)s=	23,332
9	M(Xp)v=	20,863

Рисунок 2.17 – Дані, перенесені в модель агента

Інструкція користувача наведена у додатку Г.

2.4 Комп'ютерні експерименти по вибору оптимальних траєкторій навчання

Для перевірки працездатності модулю вибору раціональної технології навчання було проведено серію комп'ютерних експериментів.

В результаті виконання комп'ютерних експериментів була виявлена залежність ймовірності відповіді на всі питання на підсумковому тестуванні від наявного часу на виконання при низькій (Рисунок 2.18), середній (Рисунок 2.19) та високій (Рисунок 2.30) складностях модуля. При цьому:

- T_0 – директивний час виконання;
- T_ϕ – фактичний час виконання;
- V – ймовірність відповіді на всі питання на підсумковому тестуванні.

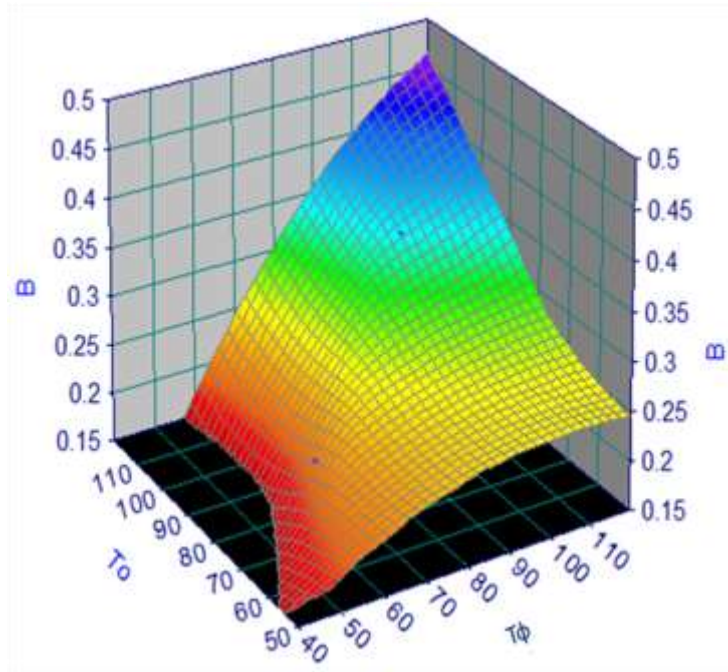


Рисунок 2.18 – Залежність ймовірності відповісти на всі питання на підсумковому тестуванні від наявного часу на виконання при низькій складності підмодуля

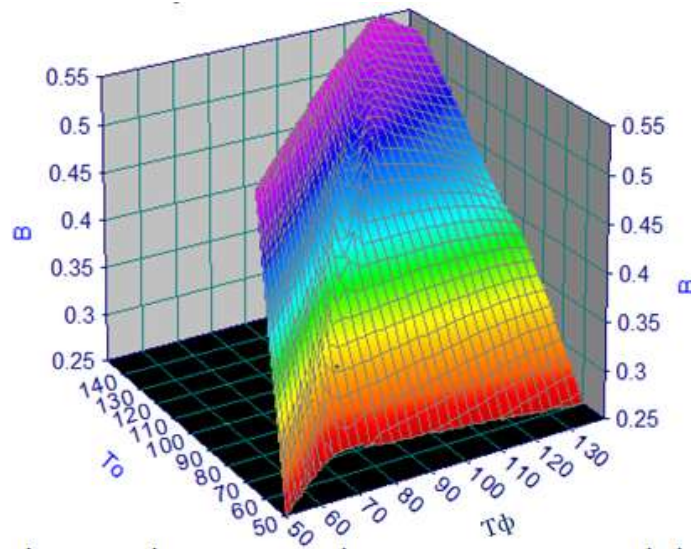


Рисунок 2.19 – Залежність ймовірності відповісти на всі питання на підсумковому тестуванні від наявного часу на виконання при середній складності підмодуля

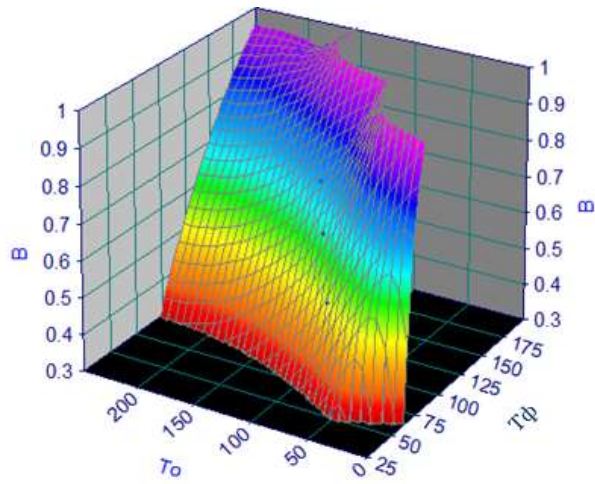


Рисунок 2.20 – Залежність ймовірності відповісти на всі питання на підсумковому тестуванні від наявного часу на виконання при високій складності підмодуля

Також була виявлена залежність ймовірності виконання алгоритму навчання від ймовірності безпомилкового виконання підмодулю та математичного сподівання часу виконання (Рисунок 2.21). При цьому:

- B_i – ймовірність безпомилкового виконання i -го підмодуля;
- M – математичне сподівання часу виконання;
- B – ймовірність виконання алгоритму навчання.

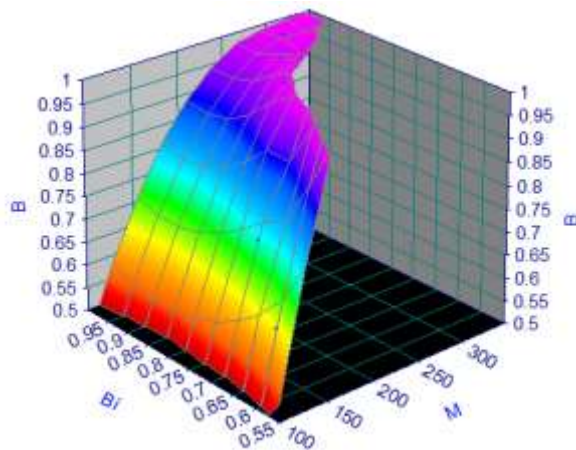


Рисунок 2.21 – Залежність ймовірності відповісти на всі питання на підсумковому тестуванні від ймовірності безпомилкового виконання підмодулю та математичного сподівання часу виконання

2.5 Висновки до розділу 2

Для прогнозування даних був обраний метод штучних нейронних мереж, адже вони доволі точно апроксимують залежність з урахуванням нелінійності.

Для створення нейронної мережі використовувався пакет прикладних програм для числового аналізу – Matlab R2012b. Дана система включає в себе велику кількість тулбоксів, серед яких був використаний Nntool – графічний інтерфейс для роботи з нейронними мережами.

Було розглянуто архітектуру кожної створеної нейронної мережі та особливості їх навчання.

Створена інформаційна технологія дозволяє спрогнозувати наступні вхідні параметри для оптимізації: ймовірність безпомилкового виконання під модулю, математичне сподівання часу виконання, наявність відмови від діяльності.

Для зручного використання мереж був створений інтерфейс на мові *c#* в Microsoft Visual Studio 2010. Створена програма дозволяє наступне: записати дані про нового студента, відредагувати дані існуючих, видалити непотрібні дані, спрогнозувати параметри для оптимізації, переглянути результат, відправити спрогнозовані дані в модель агента.

Також було реалізовано обробку виключних ситуацій.

Зв'язок між попередньо наведеними програмами був реалізований за допомогою середовище виконання компонентів Matlab - Matlab Compiler RunTime. В результаті був створений .NET компонент.

Для перевірки працездатності модулю вибору раціональної технології навчання було проведено серію комп'ютерних експериментів, що демонструють адаптацію мережі до студента в залежності від спрогнозованих даних.

ВИСНОВКИ

Сучасні університети пропонують зручну і доступну форму отримання вищої освіти – використання електронного навчання. Але не дивлячись на досконалу організацію системи дистанційної освіти в університеті, процес даного навчання має низку недоліків.

У результаті роботи був проаналізований об'єкт дослідження, для якого був створений програмний продукт, що реалізує прогнозування якості діяльності в e-learning.

Для прогнозування даних був обраний метод штучних нейронних мереж. Використовувалась нейронна мережа з прямим розповсюдженням, адже в порівнянні з іншими її помилка виявилась найнижчою.

Для розроблення програмного модулю підготовки даних для оптимізації було обрано середовище Microsoft Visual Studio 2010 та для побудови нейронних мереж – пакет прикладних програм – Matlab R2012b.

Створено три нейронні мережі в програмі Matlab для прогнозування безпомилкового виконання модулю, математичного сподівання часу виконання, а також наявності відмови від діяльності. Спроектовано інтерфейс на мові c# за допомогою програми Visual Studio. Реалізований зв'язок між попередньо згаданими програмами, а також перенесення прогнозованих даних в модель інтелектуального агента.

Для перевірки працездатності модулю вибору раціональної технології навчання було проведено серію комп'ютерних експериментів, що демонструють адаптацію мережі до студента в залежності від прогнозованих даних.

В результаті була створена інформаційна технологія підготовки даних для оптимізації, що дозволяє спрогнозувати необхідні дані та перенести їх в модель інтелектуального агента.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Волкова Н. П. Професійно-педагогічна комунікація : навч. посіб. / Н. П. Волкова. - К. : „Академія”, 2006. - 255 с. 2. E-learning или дистанционное обучение. [Електроні дані]. - Режим доступу : <http://e-college.ru/elearning>. 3. Зборовский Г. Самообразование - парадигма XXI века / Г. Зборовский, Е. Шуклина. - 2003. - №5. - С. 25 - 32.
2. Education World: Educators Battle Over Calculator Use [Electronic resource] // Education World. – Mode of access : http://www.educationworld.com/a_curr/curr072.shtml.
3. Лавров, Е.А., Клименко, А.В. Компьютеризация управления вузом [Текст]/ Е.А.Лавров, А.В.Клименко – Сумы: “Довкілля”, 2005. – 307с.
4. Абашин В.Г. Адаптивная математическая модель мультибио-метрической подсистемы определения работоспособности человека-оператора АРМ на основе нечетких множеств. Информационные системы и технологии 2011. - №5(67).С. 90-96.
5. Соколов, Д.Ю. Сравнительный анализ метода применения искусственной нейронной сети в целях решения задачи инженерно-штурманского расчета полета летательного аппарата [Текст] / Д.Ю. Соколов // Вооружение и экономика. – 2012. - №1 (17). С. 50–57.
6. Осовский С. Нейронные сети для обработки информации. – М: Издательство «Финансы и статистика», 2004.
7. Григорьев, Д.Ю. Нейрокомпьютеры в авиации. – М: Издательство «Радиотехника», 2004.
8. В. Дьяконов, В. Круглов. MATLAB. Анализ, идентификация и моделирование систем. Специальный справочник. — СПб.: Питер, 2002.
9. Чен К., Джиблин П., Ирвинг А. 443 MATLAB в математических исследованиях: Пер. с англ.— М.: Мир, 2001.-346с. ил.

10. Н.К. Смоленцев Создание Windows_приложений с использованием математических процедур MATLAB. – М.: ДМК_Пресс, 2008. – 456 с.: ил.
11. Зиборов В.В. Visual C# 2010 на примерах. – Спб.: БХВ-Петербург, 2011. – 432 с.: ил. + CD-ROM.
12. Зиборов В.В. Visual C# 20102 на примерах. – Спб.: БХВ-Петербург, 20113. – 480 с.: ил.
13. Бадд Т. Объектно-ориентированное программирование в действии / Тимоти Бадд ; [пер. с англ.]. – СПб.: «Питер», 1997. – 464 с.
14. Шевчук П. Г. Програмно-технологічні умови використання мови C# для навчання програмування в загальноосвітніх навчальних закладах / П. Г. Шевчук // Комп'ютерно-орієнтовані системи навчання: збірник наукових праць. М-во освіти і науки України, НПУ ім. М. П. Драгоманова; Відп. ред. М. І. Жалдак. – Київ, 2011. – Вип. 17 – С. 80 – 83.

ДОДАТОК А – Типи нейронних мереж Nntool

№	Тип мережі	Назва мережі
1	Competitive	Конкуруюча мережа
2	Cascade-forward backprop	Каскадна мережа з прямим розповсюдженням сигналу і зворотним поширенням помилки
3	Elman backprop	Мережа Елмана зі зворотним поширенням помилки
4	Feed-forward backprop	Мережа з прямим розповсюдженням сигналу і зворотним поширенням помилки
5	Time delay backprop	Мережа з запізненням і зворотним поширенням помилки
6	Generalized regression	Узагальнена регресійна мережа
7	Hopfield	Мережа Хопфілда
8	Linear layer (design)	Лінійний шар (створення)
9	Linear layer (train)	Лінійний шар (навчання)
10	LVQ	Мережа для класифікації входних векторів
11	Perceptron	Персептрон
12	Probabalistic	імовірнісна мережа
13	Radial basis (exact fit)	Радіальна базисна мережа з нульовою помилкою
14	Radial basis (fewer neurons)	Радіальна базисна мережа з мінімальним числом нейронів
15	Self-organizing map	Самоорганізована карта Кохонена

ДОДАТОК Б – Лістинг М-функцій

Лістинг М-функції виклику нейронної мережі, що прогнозує ймовірність
безпомилкового виконання підмодулю

```
function [Results] = MyNet(Input)
%#function network
load C:\Users\Натали\Desktop\DIPLoM\SampleNet.mat;
Results = sim(NET, Input);
end
```

Лістинг М-функції виклику нейронної мережі, що прогнозує математичне
сподівання часу виконання підмодулю

```
function [Results] = MyMNet(Input)
%#function network
load C:\Users\Натали\Desktop\DIPLoM\m\SampleM.mat;
Results = sim(NETM, Input);
end
```

Лістинг М-функції виклику нейронної мережі, що прогнозує наявність
відмови від діяльності

```
function [Results] = MyONet(Input)
%#function network
load C:\Users\Натали\Desktop\DIPLoM\o\SampleO.mat;
Results = sim(NETO, Input);
end
```


ДОДАТОК В – Лістинг програми

Form1 – Головна форма

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using MathWorks.MATLAB.NET.Arrays;
using Test;
namespace Project
{
    public partial class TestNN : Form
    {
        public TestNN()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            Form3 form3 = new Form3();
            form3.Show();
        }

        private void button4_Click(object sender, EventArgs e)
        {
            Form5 form5 = new Form5();
            form5.Show();
        }

        private void button3_Click(object sender, EventArgs e)
        {
            Form4 form4 = new Form4();
            form4.Show();
        }

        private void button2_Click(object sender, EventArgs e)
        {
            Form2 form2 = new Form2();
            form2.Show();
        }

        private void button5_Click(object sender, EventArgs e)
        {
            Form6 form6 = new Form6();
            form6.Show();
        }
    }
}
```

Form2 – Форма створення нових даних

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.IO;    //для чтение файла

namespace Project
{
    public partial class Form2 : Form
    {
        public Form2()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            try
            {
                if (FIO.Text != "" && m.Text != "" && v.Text != "" && fs.Text != ""
&& kk.Text != "")
                {
                    string FileName = FIO.Text + ".txt";
                    FileInfo f = new
FileInfo(System.Windows.Forms.Application.StartupPath + "\\txtData\\" + FileName);
                    StreamWriter w = f.CreateText();
                    w.WriteLine(m.Text);
                    w.WriteLine(v.Text);
                    w.WriteLine(fs.Text);
                    w.WriteLine(kk.Text);
                    w.Write(w.NewLine);
                    MessageBox.Show("Данные сохранены в файл " + FileName);
                    w.Close();
                }
                else
                    MessageBox.Show("Для продолжения необходимо заполнить все
поля!");
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
            }
        }

        private void button2_Click(object sender, EventArgs e)
        {
            try
            {
                string FileName = FIO.Text + ".txt";
                System.IO.File.Delete(System.Windows.Forms.Application.StartupPath +
"\\txtData\\" + FileName);
                MessageBox.Show("Файл " + FileName + " удален");
                FIO.Text = "";
            }
        }
    }
}
```

```

        m.Text = "";
        v.Text = "";
        fs.Text = "";
        kk.Text = "";
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}
}
}
}

```

Form3 –Форма прогнозування

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using MathWorks.MATLAB.NET.Arrays;
using Test;
using TestM;
using TestO;
using System.IO;
namespace Project
{
    public partial class Form3 : Form
    {
        public Form3()
        {
            InitializeComponent();
        }

        private void comboBox1_Click(object sender, EventArgs e)
        {
            try
            {
                comboBox1.Items.Clear();
                string[] filename =
Directory.GetFiles(System.Windows.Forms.Application.StartupPath + "\\txtData\\",
"*.txt");

                int k = 0;
                string res = "result";
                string Names = "";

                //узнаем кол файлов без результатов
                for (int i = 0; i < filename.Length; i++)
                {
                    if (filename[i] != null)
                        Names = Path.GetFileNameWithoutExtension(filename[i]);
                }
                //получить имена файлов без расширения
                if (Names.Contains(res) == false)
                {
                    k++;
                }
            }
        }
    }
}

```

```

    }
}
string[] p = new string[k];
k = 0;
//заполняем массив с именами без результатов
for (int i = 0; i < filename.Length; i++)
{
    if (filename[i] != null)
        Names = Path.GetFileNameWithoutExtension(filename[i]);
//получить имена файлов без расширения
    if (Names.Contains(res) == false)
    {
        p[k] = Names;
        k++;
    }
}
comboBox1.Items.AddRange(p);
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}
}

private void button1_Click(object sender, EventArgs e)
{
    string SelectedFileName = comboBox1.GetItemText(comboBox1.SelectedItem);
    if (SelectedFileName != "")
    {
        string result1 = "";
        string result2 = "";
        string result3 = "";
        string b1 = "";
        string b2 = "";
        string b3 = "";
        string m1 = "";
        string m2 = "";
        string m3 = "";
        string o1 = "";
        string o2 = "";
        string o3 = "";
        string vh = v.Text.Remove(0, 2);
        string k = kk.Text.Remove(0, 2);
        string s = "100";

        //безошибочность
        try
        {
            //для ур.сл. 1
            TestClass testNN1 = new TestClass();
            MWArray[] input1 = new MWNumericArray[] { new double[,] { {
Convert.ToDouble(m.Text) }, { Convert.ToDouble(vh) }, { 1 }, {
Convert.ToDouble(fs.Text) }, { Convert.ToDouble(k) } } } };
            MWArray[] output1 = new MWNumericArray[] { new double[1, 1] };
            testNN1.MyNet(1, ref output1, input1);
            b1 = output1[0].ToString();
            if (b1 != "100")
                b1 = b1.Remove(2);

            result1 = b1.Replace(".", "");
            int indexOfSubstring1 = result1.IndexOf(s);

```

```

        if (indexOfSubstring1 != 0)
            result1 = "0." + result1;
        else
            result1 = "1";
        if (result1.Length > 4)
            result1 = result1.Remove(4);
        textBox1.Text = result1;

        //для ур.сл. 2
        TestClass testNN2 = new TestClass();
        MWArray[] input2 = new MWNumericArray[] { new double[,] { {
Convert.ToDouble(m.Text) }, { Convert.ToDouble(vh) }, { 2 }, {
Convert.ToDouble(fs.Text) }, { Convert.ToDouble(k) } } } };
        MWArray[] output2 = new MWNumericArray[] { new double[1, 1] };

        testNN2.MyNet(1, ref output2, input2);
        b2 = output2[0].ToString();
        if (b2 != "100")
            b2 = b2.Remove(2);
        result2 = b2.Replace(".", "");
        int indexOfSubstring2 = result2.IndexOf(s);
        if (indexOfSubstring2 != 0)
            result2 = "0." + result2;
        else
            result2 = "1";
        if (result2.Length > 4)
            result2 = result2.Remove(4);
        textBox2.Text = result2;

        //для ур.сл. 3
        TestClass testNN3 = new TestClass();
        MWArray[] input3 = new MWNumericArray[] { new double[,] { {
Convert.ToDouble(m.Text) }, { Convert.ToDouble(vh) }, { 3 }, {
Convert.ToDouble(fs.Text) }, { Convert.ToDouble(k) } } } };
        MWArray[] output3 = new MWNumericArray[] { new double[1, 1] };
        testNN3.MyNet(1, ref output3, input3);
        b3 = output3[0].ToString();
        if (b3 != "100")
            b3 = b3.Remove(2);
        result3 = b3.Replace(".", "");
        int indexOfSubstring3 = result3.IndexOf(s);
        if (indexOfSubstring3 != 0)
            result3 = "0." + result3;
        else
            result3 = "1";
        if (result3.Length > 4)
            result3 = result3.Remove(4);
        textBox3.Text = result3;
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

//мат.ожидание
try
{
    //для ур.сл. 1
    ClassMNET testNN1 = new ClassMNET();

```

```

        MWArray[] input1 = new MWNumericArray[] { new double[,] { {
Convert.ToDouble(m.Text) }, { Convert.ToDouble(vh) }, { 1 }, {
Convert.ToDouble(fs.Text) }, { Convert.ToDouble(k) }, { Convert.ToDouble(b1) } } } };
        MWArray[] output1 = new MWNumericArray[] { new double[1, 1] };

        testNN1.МyMNet(1, ref output1, input1);
        m1 = output1[0].ToString();
        textBox4.Text = m1;

        //для ур.сл. 2
        ClassMNET testNN2 = new ClassMNET();
        MWArray[] input2 = new MWNumericArray[] { new double[,] { {
Convert.ToDouble(m.Text) }, { Convert.ToDouble(vh) }, { 2 }, {
Convert.ToDouble(fs.Text) }, { Convert.ToDouble(k) }, { Convert.ToDouble(b2) } } } };
        MWArray[] output2 = new MWNumericArray[] { new double[1, 1] };
        testNN2.МyMNet(1, ref output2, input2);
        m2 = output2[0].ToString();
        textBox5.Text = m2;

        //для ур.сл. 3
        ClassMNET testNN3 = new ClassMNET();
        MWArray[] input3 = new MWNumericArray[] { new double[,] { {
Convert.ToDouble(m.Text) }, { Convert.ToDouble(vh) }, { 3 }, {
Convert.ToDouble(fs.Text) }, { Convert.ToDouble(k) }, { Convert.ToDouble(b3) } } } };
        MWArray[] output3 = new MWNumericArray[] { new double[1, 1] };
        testNN3.МyMNet(1, ref output3, input3);
        m3 = output3[0].ToString();
        textBox6.Text = m3;
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }

    //наличие отказа
    try
    {
        //для ур.сл. 1
        ClassONET testNN1 = new ClassONET();
        MWArray[] input1 = new MWNumericArray[] { new double[,] { {
Convert.ToDouble(m.Text) }, { Convert.ToDouble(vh) }, { 1 }, {
Convert.ToDouble(fs.Text) }, { Convert.ToDouble(k) } } } };
        MWArray[] output1 = new MWNumericArray[] { new double[1, 1] };
        testNN1.МyONet(1, ref output1, input1);
        o1 = output1[0].ToString();
        if (o1.IndexOf('.') != -1)
            o1 = o1.Replace(".", ",");
        double o11 = double.Parse(o1);
        if (o11 >= 0.5)
            o1 = "0";
        else
            o1 = "1";
        textBox7.Text = o1;

        //для ур.сл. 2
        ClassONET testNN2 = new ClassONET();
        MWArray[] input2 = new MWNumericArray[] { new double[,] { {
Convert.ToDouble(m.Text) }, { Convert.ToDouble(vh) }, { 2 }, {
Convert.ToDouble(fs.Text) }, { Convert.ToDouble(k) } } } };
        MWArray[] output2 = new MWNumericArray[] { new double[1, 1] };
        testNN2.МyONet(1, ref output2, input2);

```

```

        o2 = output2[0].ToString();
        if (o2.IndexOf('.') != -1)
            o2 = o2.Replace(".", ",");
        double o22 = double.Parse(o2);
        if (o22 >= 0.5)
            o2 = "1";
        else
            o2 = "0";
        textBox8.Text = o2;

        //для ур.сл. 3
        ClassONET testNN3 = new ClassONET();
        MWArray[] input3 = new MWNumericArray[] { new double[,] { {
Convert.ToDouble(m.Text) }, { Convert.ToDouble(vh) }, { 3 }, {
Convert.ToDouble(fs.Text) }, { Convert.ToDouble(k) } } } };
        MWArray[] output3 = new MWNumericArray[] { new double[1, 1] };
        testNN3.MyONet(1, ref output3, input3);
        o3 = output3[0].ToString();
        if (o3.IndexOf('.') != -1)
            o3 = o3.Replace(".", ",");
        double o33 = double.Parse(o1);
        if (o33 >= 0.5)
            o3 = "1";
        else
            o3 = "0";
        textBox9.Text = o3;
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }

    //запись в файл результата
    string FileName = System.Windows.Forms.Application.StartupPath +
"\\txtData\\" + SelectedFileName + "_result.txt";
    FileInfo f = new FileInfo(FileName);
    StreamWriter w = f.CreateText();
    w.WriteLine(result1);
    w.WriteLine(result2);
    w.WriteLine(result3);
    w.WriteLine(m1);
    w.WriteLine(m2);
    w.WriteLine(m3);
    w.WriteLine(o1);
    w.WriteLine(o2);
    w.WriteLine(o3);
    w.Write(w.NewLine);
    w.Close();
    }
    else
        MessageBox.Show("Для продолжения необходимо выбрать студента!");
}

private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    try
    {
        string SelectedFileName =
comboBox1.GetItemText(comboBox1.SelectedItem);
        //считываем

```

```

        StreamReader stroka = new
StreamReader(System.Windows.Forms.Application.StartupPath + "\\txtData\\" +
SelectedFileName + ".txt");
        // Читаем строки из файла
        m.Text = stroka.ReadLine();
        v.Text = stroka.ReadLine();
        fs.Text = stroka.ReadLine();
        kk.Text = stroka.ReadLine();
        stroka.ReadToEnd();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}
}
}

```

Form4 –Форма для перегляду результату

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.IO;
namespace Project
{
    public partial class Form4 : Form
    {
        public Form4()
        {
            InitializeComponent();
        }

        private void comboBox1_Click(object sender, EventArgs e)
        {
            try
            {
                comboBox1.Items.Clear();
                string[] filename =
Directory.GetFiles(System.Windows.Forms.Application.StartupPath + "\\txtData\\",
"*.txt");

                int k = 0;
                string res = "result";
                string Names = "";

                //узнаем кол файлов без результатов
                for (int i = 0; i < filename.Length; i++)
                {
                    if (filename[i] != null)
                        Names = Path.GetFileNameWithoutExtension(filename[i]);
                    //получить имена файлов без расширения
                    if (Names.Contains(res) == true)
                    {
                        k++;
                    }
                }
            }
            catch { }
        }
    }
}

```



```

    }
}
string[] p = new string[k];
k = 0;
//заполняем массив с именами без результатов
for (int i = 0; i < filename.Length; i++)
{
    if (filename[i] != null)
        Names = Path.GetFileNameWithoutExtension(filename[i]);
//получить имена файлов без расширения
    if (Names.Contains(res) == true)
    {
        p[k] = Names;
        k++;
    }
}
comboBox1.Items.AddRange(p);
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}
}

private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    try
    {
        string SelectedFileName =
comboBox1.GetItemText(comboBox1.SelectedItem);
        //считываем
        StreamReader stroka = new
StreamReader(System.Windows.Forms.Application.StartupPath + "\\txtData\\" +
SelectedFileName + ".txt");
        // Читаем строки из файла
        textBox1.Text = stroka.ReadLine();
        textBox2.Text = stroka.ReadLine();
        textBox3.Text = stroka.ReadLine();
        textBox4.Text = stroka.ReadLine();
        textBox5.Text = stroka.ReadLine();
        textBox6.Text = stroka.ReadLine();
        textBox7.Text = stroka.ReadLine();
        textBox8.Text = stroka.ReadLine();
        textBox9.Text = stroka.ReadLine();
        stroka.ReadToEnd();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}
}
}
}
}
}
}

```

Form5 – Форма для редагування/видалення даних

```

using System;
using System.Collections.Generic;

```

```

using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.IO;
namespace Project
{
    public partial class Form5 : Form
    {
        public Form5()
        {
            InitializeComponent();
        }

        private void comboBox1_Click(object sender, EventArgs e)
        {
            try
            {
                comboBox1.Items.Clear();
                string[] filename =
Directory.GetFiles(@"C:\Users\Натали\Desktop\DIPLOM\Новая
напка\Project\Project\bin\Debug\txtData", "*.txt");
                int k = 0;
                string res = "result";
                string Names = "";

                //узнаем кол файлов без результатов
                for (int i = 0; i < filename.Length; i++)
                {
                    if (filename[i] != null)
                        Names = Path.GetFileNameWithoutExtension(filename[i]);
                }
                //получить имена файлов без расширения
                if (Names.Contains(res) == false)
                {
                    k++;
                }
            }
            string[] p = new string[k];
            k = 0;
            //заполняем массив с именами без результатов
            for (int i = 0; i < filename.Length; i++)
            {
                if (filename[i] != null)
                    Names = Path.GetFileNameWithoutExtension(filename[i]);
                //получить имена файлов без расширения
                if (Names.Contains(res) == false)
                {
                    p[k] = Names;
                    k++;
                }
            }
            comboBox1.Items.AddRange(p);
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message);
        }
    }
}

```

```

private void button1_Click(object sender, EventArgs e)
{
    try
    {
        string SelectedFileName =
comboBox1.GetItemText(comboBox1.SelectedItem);
        string FileName = System.Windows.Forms.Application.StartupPath +
"\\txtData\\" + SelectedFileName + ".txt";
        FileInfo f = new FileInfo(FileName);
        StreamWriter w = f.CreateText();
        w.WriteLine(m.Text);
        w.WriteLine(v.Text);
        w.WriteLine(fs.Text);
        w.WriteLine(kk.Text);
        w.WriteLine(w.NewLine);
        MessageBox.Show("Данные сохранены в файл " + SelectedFileName +
".txt");
        w.Close();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

private void button2_Click(object sender, EventArgs e)
{
    try
    {
        string SelectedFileName =
comboBox1.GetItemText(comboBox1.SelectedItem);

        string FileName = System.Windows.Forms.Application.StartupPath +
"\\txtData\\" + SelectedFileName + ".txt";
        System.IO.File.Delete(FileName);
        FileName = System.Windows.Forms.Application.StartupPath +
"\\txtData\\" + SelectedFileName + "_result.txt";
        System.IO.File.Delete(FileName);
        FileName = System.Windows.Forms.Application.StartupPath +
"\\ExcelData\\" + SelectedFileName + ".xls";
        System.IO.File.Delete(FileName);
        MessageBox.Show("Файл " + SelectedFileName + ".txt" + " удален");
        comboBox1.Text = "";
        m.Text = "";
        v.Text = "";
        fs.Text = "";
        kk.Text = "";
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    try
    {
        string SelectedFileName =
comboBox1.GetItemText(comboBox1.SelectedItem);
        //считываем

```

```

        StreamReader stroka = new
StreamReader(System.Windows.Forms.Application.StartupPath + "\\txtData\\" +
SelectedFileName + ".txt");
        // Читаем строки из файла
        m.Text = stroka.ReadLine();
        v.Text = stroka.ReadLine();
        fs.Text = stroka.ReadLine();
        kk.Text = stroka.ReadLine();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}
}
}
}

```

Form6 – Форма перенесения данных в модель агента

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.IO;
using Excel = Microsoft.Office.Interop.Excel;
namespace Project
{
    public partial class Form6 : Form
    {
        public Form6()
        {
            InitializeComponent();
        }
        private Excel.Application m_objExcel;
        private Excel.Workbooks m_objBooks;
        private Excel.Workbook m_objBook;
        private Excel.Sheets m_objSheets;
        private Excel.Worksheet m_objSheet;
        private Excel.Range m_objRange;
        private object m_objOpt = System.Reflection.Missing.Value;
        private Excel.Font m_objFont;
        private object m_strSampleFolder = Application.StartupPath + "\\ExcelData\\";

        private void comboBox1_Click(object sender, EventArgs e)
        {
            try
            {
                comboBox1.Items.Clear();
                string[] filename =
Directory.GetFiles(System.Windows.Forms.Application.StartupPath + "\\txtData\\",
"*.txt");

                int k = 0;
                string res = "result";
                string Names = "";
            }
        }
    }
}

```

```

        //узнаем кол файлов без результатов
        for (int i = 0; i < filename.Length; i++)
        {
            if (filename[i] != null)
                Names = Path.GetFileNameWithoutExtension(filename[i]);
            //получить имена файлов без расширения
            if (Names.Contains(res) == false)
            {
                k++;
            }
        }
        string[] p = new string[k];
        k = 0;
        //заполняем массив с именами без результатов
        for (int i = 0; i < filename.Length; i++)
        {
            if (filename[i] != null)
                Names = Path.GetFileNameWithoutExtension(filename[i]);
            //получить имена файлов без расширения
            if (Names.Contains(res) == false)
            {
                p[k] = Names;
                k++;
            }
        }
        comboBox1.Items.AddRange(p);
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

private void button1_Click(object sender, EventArgs e)
{
    try
    {
        string SelectedFileName =
comboBox1.GetItemText(comboBox1.SelectedItem);
        if (SelectedFileName != "")
        {
            string fileName = System.Windows.Forms.Application.StartupPath +
"\\ExcelData\\" + "Model" + ".xls";
            string fileName_new =
System.Windows.Forms.Application.StartupPath + "\\ExcelData\\" + SelectedFileName +
".xls";

            StreamReader stroka = new
StreamReader(System.Windows.Forms.Application.StartupPath + "\\txtData\\" +
SelectedFileName + "_result.txt");
            // Читаем строки из файла во временную переменную.
            string bn = stroka.ReadLine();
            string bs = stroka.ReadLine();
            string bv = stroka.ReadLine();
            string mn = stroka.ReadLine();
            string ms = stroka.ReadLine();
            string mv = stroka.ReadLine();
            stroka.ReadToEnd();

            // Приложение Excel.
            m_objExcel = new Excel.Application();
        }
    }
}

```

```
    //Книга
    m_objBooks = m_objExcel.Workbooks;

    m_objExcel.Workbooks.Open(fileName,
                               Type.Missing, Type.Missing, Type.Missing,
Type.Missing,
                               Type.Missing, Type.Missing,
Type.Missing, Type.Missing,
                               Type.Missing, Type.Missing,
Type.Missing, Type.Missing,
                               Type.Missing, Type.Missing);
    m_objBook = m_objBooks[1];

    //Получаем массив ссылок на листы выбранной книги
    m_objSheets = m_objBook.Worksheets;
    //Выбираем лист 1
    m_objSheet = (Excel.Worksheet)(m_objSheets.get_Item(1));
    //Выбираем ячейки
    m_objRange = m_objSheet.get_Range("B1", m_objOpt);
    m_objRange.Value = SelectedFileName;

    m_objRange = m_objSheet.get_Range("B2", m_objOpt);
    m_objRange.Value = bn;
    m_objRange = m_objSheet.get_Range("B3", m_objOpt);
    m_objRange.Value = bs;
    m_objRange = m_objSheet.get_Range("B4", m_objOpt);
    m_objRange.Value = bv;

    m_objRange = m_objSheet.get_Range("B7", m_objOpt);
    m_objRange.Value = mn;
    m_objRange = m_objSheet.get_Range("B8", m_objOpt);
    m_objRange.Value = ms;
    m_objRange = m_objSheet.get_Range("B9", m_objOpt);
    m_objRange.Value = mv;

    // сохраняем
    m_objBook.SaveAs(fileName_new);
    m_objBook.Close(false, m_objOpt, m_objOpt);
    m_objExcel.Quit();
}
else
    MessageBox.Show("Выберите студента для записи полученных данных в
Excel");
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}
}
}
```

ДОДАТОК Г – Інструкція користувача

Щоб розпочати роботи з інформаційною системою підготовки даних для оптимізації достатньо викликати файл «Підсистема підготовки даних.exe», що знаходиться в кореневій папці проекту. Відкриється головна форма програми (Рисунок Г.1).

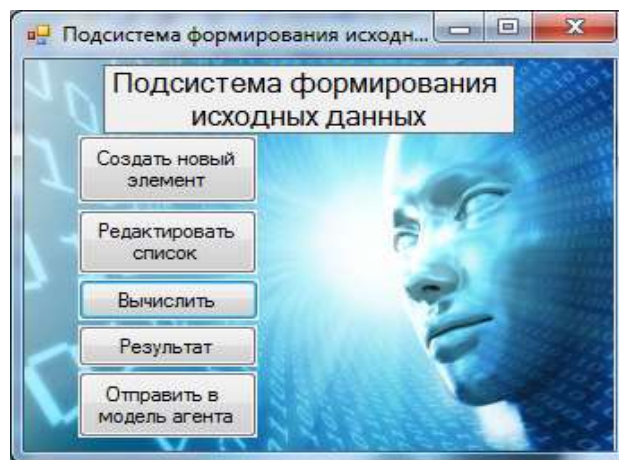


Рисунок Г.1 – Головне вікно створеної програми

Для додавання інформації про нового студента користувачу необхідно натиснути на кнопку «Создать новый элемент» головного меню. В результаті відкриється вікно (Рисунок Г.2), де у відповідні поля необхідно внести всю інформацію про студента.

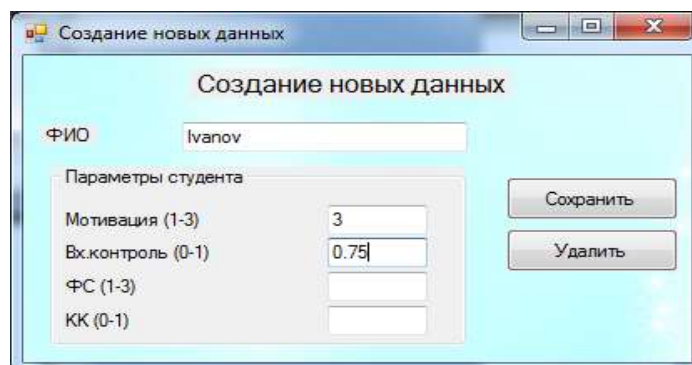


Рисунок Г.2 – Вікно «Создание новых данных»

Продовження додатку Г

В результаті коректного введення даних вони зберігаються в текстовий файл, про що свідчить повідомлення для користувача (Рисунок Г.3).

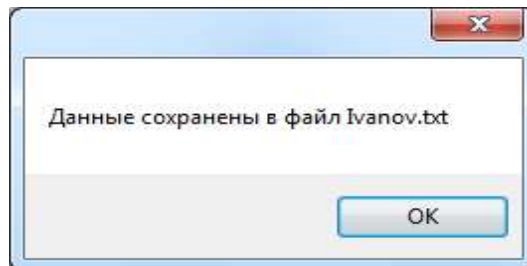


Рисунок Г.3 – Повідомлення про збереження інформації про студента

Щоб відредагувати або видалити існуючі дані необхідно натиснути на кнопку «Редактировать список» головного меню програми. В результаті відкриється вікно, представлене на рисунку Г.4.

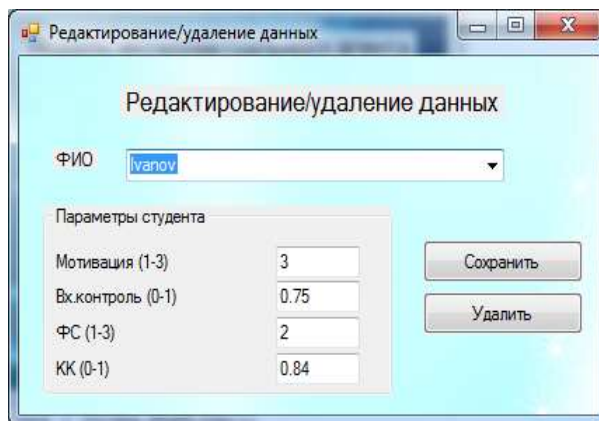


Рисунок Г.4 – Вікно «Редактирование/удаление данных»

Спочатку користувачу необхідно обрати із випадуючого списку потрібного студента. Для редагування параметрів - внести всі необхідні зміни та натиснути на кнопку «Сохранить». В результаті текстовий файл буде відредаговано відповідно введеним даним.

Продовження додатку Г

Для видалення всієї інформації про студента, включаючи і його спрогнозовані результати, користувачу достатньо натиснути на кнопку «Удалить». Про успішне видалення свідчить повідомлення, представлене на рисунку Г.5.

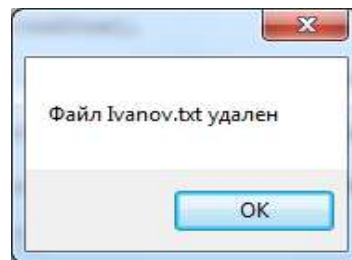


Рисунок Г.5 – Повідомлення про видалення інформації про студента

Для прогнозування даних користувачу необхідно обрати пункт головного меню «Вычислить». В результаті відкриється вікно «Прогнозирование входных данных для оптимизации» (Рисунок Г.6).

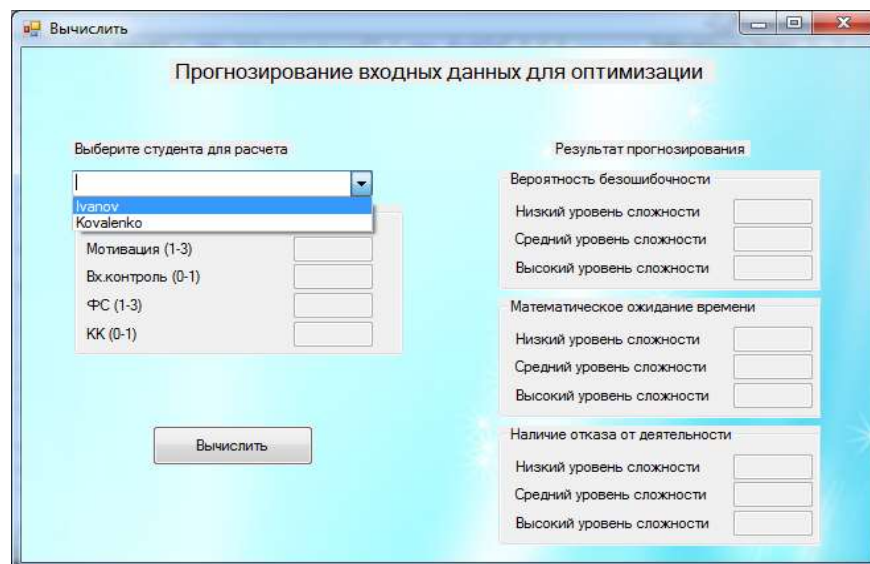


Рисунок Г.6 – Вікно «Прогнозирование входных данных для оптимизации»

Далі користувач повинен обрати потрібного студента, в результаті чого поля з параметрами автоматично заповняться інформацією із раніше
Продовження додатку Г

збереженого текстового файлу. Для прогнозування безпомилковості виконання підмодуля, математичного сподівання часу виконання та наявності відмови від діяльності, враховуючи рівень складності навчання, користувачу потрібно натиснути на кнопку «Вычислить» даного вікна.

Після цього користувач отримує спрогнозовані дані, що автоматично заносяться у відповідні поля даної форми. Також вони автоматично зберігаються у текстовий файл (Рисунок Г.7).

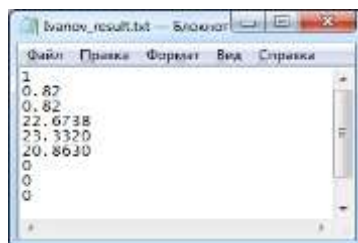


Рисунок Г.7 – Текстовий файл з результатами

Щоб переглянути раніше спрогнозовані дані для кожного студента користувачу необхідно обрати пункт меню «Результат». Після чого з'являється вікно, що представлено на рисунку Г.8.

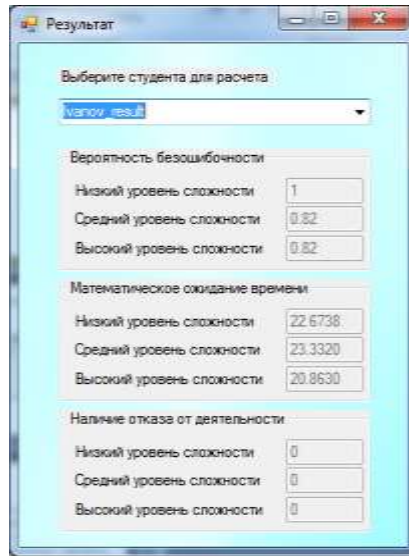


Рисунок Г.8 – Результат

Продовження додатку Г

Для перенесення спрогнозованих даних користувачу необхідно натиснути на кнопку «Отправить в модель агента» головної форми програми. В результаті відкриється вікно «Перенос данных» (Рисунок Г.9).

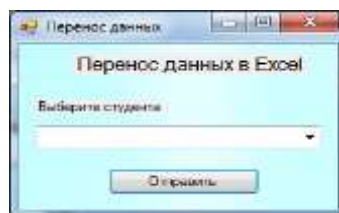


Рисунок Г.9 – Вікно «Перенос данных»

Для коректної роботи даної функції користувачу достатньо обрати потрібного студента та натиснути на кнопку «Отправить» даного вікна. В результаті спрогнозовані дані заносяться в модель агента (Рисунок Г.10), після чого їх можна використовувати для оптимізації.

	A	B
1		Ivanov
2	B1n=	1
3	B1s=	0,82
4	B1v=	0,82
7	M(Xp)n=	22,6738
8	M(Xp)s=	23,332
9	M(Xp)v=	20,863

Рисунок Г.10 – Дані, перенесені в модель агента