

*Явгусішин Б. А., здобувач вищої освіти,  
Антонов Ю. С., канд. фіз.-мат. наук, доцент,  
доцент кафедри інформаційних технологій,  
Донецький національний університет імені Василя Стуса*

## **ПРОЄКТУВАННЯ РЕКОМЕНДАЦІЙНОЇ СИСТЕМИ ДЛЯ ДІАГНОСТИКИ НЕСПРАВНОСТЕЙ ПЕРСОНАЛЬНИХ КОМП'ЮТЕРІВ**

*Анотація. У роботі теоретично обґрунтовано проєктування інтелектуальної рекомендаційної системи для діагностики несправностей персональних комп'ютерів. Проаналізовано підходи до діагностики. Запропоновано архітектуру рекомендаційної системи для діагностики несправностей.*

*Ключові слова: рекомендаційна система, діагностика, несправності, машинне навчання.*

**Вступ.** З розвитком комп'ютерних технологій та зростанням кількості користувачів персональних комп'ютерів (ПК) постає проблема ефективної діагностики програмних та апаратних несправностей. Сучасні системи технічної підтримки часто потребують значних часових і людських ресурсів. Експертні системи та бази знань які вже використовуються у різних сферах [1; 2], можуть допомогти автоматизувати процес ефективної діагностики програмних та апаратних несправностей.

**Основний текст.** Із розвитком операційних систем, графічних інтерфейсів, драйверів і мережевих технологій з'явилися численні програмні або апаратні проблеми, які могли проявлятися у вигляді конфліктів між компонентами, нестабільної роботи ПК тощо. Хоча до цього через високий поріг входу технічних знань і фінансової доступності та малу кількість альтернатив у програмному та апаратному забезпеченні несправності могли бути виправлені самим користувачем [3].

Ефективність процесу діагностики переважно залежить від правильного визначення природи несправності. У ПК несправності умовно поділяють на апаратні, програмні та операційної системи (ОС) [4]. Такий поділ дає змогу систематизувати причини збоїв і полегшує процес діагностики. До апаратних належать усі фізичні проблеми компонентів системи. Апаратні збої найчастіше проявляються у вигляді нестабільної роботи системи, артефактів на екран, шумів або відмови у ввімкненні. Для діагностики таких проблем зазвичай потрібні вимірювальні прилади або тестування компонентів поза системою. До несправностей ОС належать проблеми ядра, драйверів і служб, які безпосередньо забезпечують роботу апаратного та програмного забезпечення. До таких проблем належать пошкодження системних файлів, конфлікти драйверів або неправильні параметри завантаження системи. Вони зазвичай проявляються у формі зависання, синіх екранів або зниження продуктивності ПК. Програмні несправності належать до прикладного рівня. Вони зазвичай встановлюються самим користувачем. Несправності такого виду часто характеризуються зависанням ПК, спотворенням відображення інтерфейсу, помилок під час запуску або некоректного збереження даних. Опишемо далі підходи, що застосовуються на практиці для діагностування несправностей ПК.

*Rule-based підхід.* Базується на правилах, що описують взаємозв'язки між симптомами та можливими причинами несправностей. Кожне правило – формату «якщо, ... то». Наприклад, якщо комп'ютер перезавантажується без попередження і температура головного процесу перевищує 98 градусів Цельсія, то можлива несправність системи охолодження. Системи з таким підходом мають високу інтерпретованість, легкі в налаштуванні та підходять для прототипів рекомендаційних систем, але їх складно масштабувати, оскільки для додавання нових правил потрібна перевірка створених до цього.

*Ймовірнісні методи.* Зокрема, Баєсівські мережі дають змогу оцінювати ймовірність кожної можливої несправності з урахуванням наявних симптомів. Ця мережа будує залежності між симптомами та несправностями у вигляді графу. З переваг варто виділити ефективність у разі неповних або неточних даних, що може бути результатом опису симптомів користувачем у форматі довільного тексту. Водночас є два недоліки, а саме складність у правильності опису залежностей між усіма симптомами і причинами та потреба у початкових ймовірностях подій, що на момент прототипу та початкових етапах розробки відсутні.

*Методи машинного навчання.* Застосовуються для автоматизації процесу діагностики шляхом навчання на основі датасету з симптомами та результатами перевірок і подальшою перевіркою на контрольному датасеті з даними, що були відсутні у датасеті навчання. Типовими представниками алгоритмів є логістична регресія, дерева рішень, нейронні мережі. Основним недоліком, як і ймовірнісних методів, є потреба у достатньому розмірі та якості вибірки для надання точних результатів.

*Гібридні підходи.* На практиці варто об'єднувати підходи для досягнення кращих результатів та нівелювання недоліків кожного з підходів.

Проведена класифікація типів несправностей та аналіз основних підходів діагностики дає змогу сформулювати вимоги до архітектури рекомендаційної системи. Така система повинна мати модульну структуру, забезпечувати адаптивність до нових типів несправностей, підтримування навчання на накопичених даних про несправності. Також вона повинна забезпечувати користувача зручним інтерфейсом для покрокового визначення проблеми, аналізувати введені або зібрані автоматично дані, а також надавати обґрунтовані рекомендації щодо усунення несправностей та подальших дій.

Враховуючи зазначені вимоги, програма має поєднувати в собі засоби збору та обробки діагностичних даних, механізм логічного висновку й аналітичний модуль машинного навчання. Також вона має забезпечувати гнучкість, масштабованість і можливість інтеграції з великими мовними моделями для покращення інтерпретованості рекомендацій. Для реалізації цих можливостей необхідна гнучка, розширювана та кросплатформна технологічна основа.

Для реалізації доцільно обрати такий стек технологій: C#, .NET 8, .NET MAUI, Entity Framework Core та ML.NET [5]. Таке поєднання дає змогу створити єдину екосистему, у якій клієнтський інтерфейс, бізнес-логіка, аналітика та робота з базою даних інтегруються без потреби у використанні сторонніх бібліотек чи скриптів для поєднання з іншими мовами програмування.

Натомість архітектуру системи доцільно побудувати за принципом багат шаровості, де кожен рівень буде мати власну роль і відповідальність:

- Рівень представлення (.NET MAUI). Взаємодія з користувачем, відображення запитань, отримання відповідей, показ результатів діагностики [6].
- Логічний рівень (C#, .NET 8). Реалізація процесу діагностики, логічний висновок, обробка та збір даних і формування рекомендацій.
- Рівень даних (Entity Framework Core). Збереження дій користувача під час діагностики, отриманих даних, симптомів, наданих рекомендацій щодо несправностей.
- Аналітичний рівень (ML.NET). Навчання моделей, аналіз даних, підвищення точності несправностей [5].
- Інтеграційний рівень. Взаємодія з зовнішніми сервісами, зокрема з великими мовними моделями, для спроби технічного спрощення рекомендацій та кращого пояснення.

**Висновки.** Внаслідок проведеного дослідження сформовано багат шарову архітектуру рекомендаційної системи діагностики несправностей персональних комп'ютерів. Запропонована структура поєднує модульність, гнучкість і масштабованість. Реалізація системи за згаданому стеку технологій дає можливість подальшого розвитку та впровадження нових рішень і підходів до діагностики.

#### Список використаних джерел

1. Antonov Y., Smoktii K. Expert Systems Using For Answers Analysis In Automated Knowledge Control Systems. *Herald of Khmelnytskyi National University. Technical Sciences*. 2024. Vol. 339, № 4. P. 323–331.
2. Штовба С. Д., Штовба О. В., Панкевич О. Д. Критерії точності та компактності для оцінювання якості нечітких баз знань в задачах ідентифікації. *Наукові праці Вінницького національного технічного університету*. 2012. № 4. URL: <http://praci.vntu.edu.ua/index.php/praci/article/view/343>
3. Rosenthal M. *The Laptop Repair Workbook: An Introduction to Troubleshooting and Repairing Laptop Computers*. 2008, 195 с.
4. Russinovich M. *Troubleshooting with the Windows Sysinternals Tools (IT Best Practices – Microsoft Press)*. 2016, 688 с.
5. Mahasivabhattu K. *Ultimate Machine Learning with ML.NET: Build, Optimize, and Deploy Powerful Machine Learning Models for Data-Driven Insights with ML.NET, Azure Functions, and Web API*. 2024, 366 с.
6. Liberty J., Juarez R., Montaquila M. *.NET MAUI for C# Developers: Build cross-platform mobile and desktop applications*. 2023, 296 с.