

Державний університет інфраструктури та технологій  
Міністерство освіти та науки України

Кваліфікаційна наукова  
праця на правах рукопису

**ШЕВЧЕНКО АНТОН ПЕТРОВИЧ**

УДК 681.595:781

**ДИСЕРТАЦІЯ**

**МОДЕЛІ ТА МЕТОДИ ПРОГНОЗУВАННЯ ТЕХНІЧНОГО  
СТАНУ ЗАСОБІВ ВОДНОГО ТРАНСПОРТУ НА ОСНОВІ М'ЯКИХ  
ОБЧИСЛЕНЬ**

Спеціальність 271 – Річковий та морський транспорт

Галузь знань 27 – Транспорт

Подається на здобуття наукового ступеня  
доктора філософії

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

\_\_\_\_\_ А. П. Шевченко

Науковий керівник:  
Тимощук Олена Миколаївна,  
доктор технічних наук, професор

Київ – 2020

## АНОТАЦІЯ

*Шевченко А.П.* Моделі та методи прогнозування технічного стану засобів водного транспорту на основі м'яких обчислень. – Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття наукового ступеня доктора філософії за спеціальністю 271 – Річковий та морський транспорт. – Державний університет інфраструктури та технологій, Київ, 2020.

В сучасних умовах не викликає сумніву те, що саме водний транспорт, як інфраструктурна галузь, має розвиватися швидкими темпами. На думку експертів у 2020-2022 роках, очікується збільшення обсягу перевезення вантажів морськими та річними судами до 2500 млн. тон; переробка вантажів у державних торговельних портах також збільшиться та становитиме близько 240 млн. тон; обсяги пасажирських перевезень становитимуть більш 10900 млн. пасажирів.

Дослідження, яке виконане в межах даної дисертації показує, що рівень безаварійного судноводіння, показники якості та ефективності перевезень пасажирів та вантажів продовжують покращуватися. Особлива увага приділяється саме заходам забезпечення заданого рівня надійності технічних засобів. Значна роль в цьому напрямку – рішення завдання прогнозування технічного стану засобів водного транспорту. Це ще більш підкреслює актуальність наукових досліджень з метою підвищення ефективності технічної експлуатації морських та річних транспортних засобів за рахунок використання інформації про прогноз їх технічного стану.

На теперішній час для забезпечення гарантованого рівня достовірності прогнозування технічного стану засобів водного транспорту активно використовують прогресивні апаратні та програмні рішення. Аналіз закордонного та вітчизняного досвіду розробки та впровадження систем прогнозування технічного стану як комплектуючих підсистем, так і засобів водного транспорту в цілому свідчить про можливість значного підвищення

їх ефективності за рахунок розвитку математичного та алгоритмічного забезпечення. Найбільш актуальним в цьому напрямку є використання моделей та методів штучного інтелекту, а саме, так званих м'яких обчислень. Незважаючи на стрімкий розвиток теорії штучного інтелекту взагалі, актуальним є завдання удосконалення існуючого та розробки нового математичного та програмного забезпечення комп'ютерних систем прогнозування технічного стану засобів водного транспорту на основі м'яких обчислень.

Таким чином, при вирішенні проблематики розвитку транспортної галузі України, водного транспорту зокрема, актуальним є наукове завдання удосконалення існуючих та розробки нових моделей та методів прогнозування технічного стану засобів водного транспорту на основі м'яких обчислень, вирішенню цього завдання і присвячена дана дисертаційна робота.

Наукова новизна одержаних результатів полягає в тому, що:

вперше розроблено концептуальна модель прогнозування технічного стану засобів водного транспорту, яка базується на OLAP-технології з інтелектуальним аналізом даних на м'яких обчисленнях, а саме, комплексному використанні нечіткої кластеризації на основі методів субтрактивної кластеризації (для визначення кількості значень лінгвістичних змінних) та нечітких с-середніх (FCM, Fuzzy C-Means) для побудови функцій приналежностей, а також адаптивної нейронної нечіткої мережі ANFIS. Програмна реалізація моделі значно покращує показники оперативності та достовірності прогнозу за рахунок зменшення кількості операцій та використання нових принципів обробки апіорної інформації та самонавчання;

удосконалено модель формування функції приналежності при прогнозуванні технічного стану ЗВТ, яка на відміну від існуючих використовує нечітку кластеризацію: метод субтрактивної кластеризації для визначення кількості кластерів, яка інтерпретується як ранг базової терм-

множени лінгвістичної змінної та метод нечітких с-середніх для розрахунку значень функції приналежності; це дозволяє значно покращити адекватність процедури фазифікації та точність;

удосконалено метод прогнозу технічного стану ЗВТ на основі нечіткого висновку при застосуванні гібридної мережі, який на відміну від існуючих базується на моделі адаптивної нейронної нечіткої мережевої системи ANFIS (Adaptive Neuro-Fuzzy Inference System), а це надає можливість одночасно використовувати переваги нечіткої логіки та нейронних мереж для підвищення достовірності прогнозування технічного стану.

Практичне значення одержаних результатів в тому, що їх реалізація доцільна в математичному та програмному забезпеченні, як складової технології автоматизації процесу прогнозування технічного стану судна в інтегрованої мостикові системі. Застосування запропонованих моделей і методів дозволяє підвищити достовірність прогнозу технічного стану в деяких випадках до 20-25 %. При цьому, за результатами математичного моделювання, можливе підвищення ймовірності виконання безвідмовного судноводіння до 10 % за рахунок впровадження на практиці.

Результати дисертаційних досліджень реалізовані у Державному підприємстві водних шляхів «Укрводшлях» (ДП «Укрводшлях») при виконанні практичної тренажерної підготовці судноводіїв (акт від 09.06.2020 р. №3/03/2020); а також в навчальному процесі Державного університету інфраструктури та технологій (акт від 03.09.2020 р. №12/а) підчас створення курсу лекцій з дисциплін «Технічні засоби судноводіння» та «Експлуатація засобів водного транспорту».

Наведені нові науково-обґрунтовані практичні рекомендації щодо перспективної комп'ютерної системи прогнозу технічного стану судна дозволяють розглянути можливість використання даних підходів у інших системах транспортних засобів, а також значно покращити систему технічної

експлуатації судна за рахунок планування та своєчасного проведення відповідних заходів.

Ключові слова: прогнозування, нечітка логика, нейронна мережа, відмова, надійність, технічний стан, засіб водного транспорту.

## СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА

*Наукові праці, в яких опубліковані основні наукові результати дисертації:*

1. Шевченко А. П., Герасимов С. В., Тимочко О. І., Тимошук О. М., Трішин В. В. Оптимальний алгоритм обробки навігаційної інформації у системах управління засобами водного транспорту. *Наукоємні технології*. 2018, №4(40). С.450-457.

2. Шевченко А.П., Воробьёв Е.С., Мазур А.М., Коломиец О.М., Демьяненко С.К. Разработка методов сжатия сообщений о воздушных объектах и управления дискретностью их выдачи от источников радиолокационной информации. *Новітні технології*. 2018. Вип.3(7). С.217-230.

3. Шевченко А.П., Трофименко І.В., Мазур А.М. Підвищення точності вимірювань в суднових радіолокаційних системах з врахуванням тропосферного впливу в умовах неоднорідності морського середовища. *Новітні технології*. 2018, Вип.2(6). С.61-68.

4. Шевченко А.П., Михайлова Т.І, Бойко С.О. Спосіб прогнозування відмов агрегатів суднових комплексів за даними експлуатаційних спостережень. *Новітні технології*. 2019. Вип. 1(8). С.52-58.

5. Шевченко А.П., Будолак С.Ю., Ткаченко В.В., Гуменніков Р.В. Метод структурного синтезу системи управління засобів водного транспорту. *Наукоємні технології*. 2019. №1(41). С.101-108.

6. Шевченко А.П., Дробина В.В., Ковтун О. В., Корнєв І.О., Котигора В.А. Особливості створення автоматизованої системи управління судном. *Новітні технології*. 2019. Вип. 2(9). С. 125-130.

7. Шевченко А.П., Пліта Л.Л., Дакі О.А. Особливості організації процесу експлуатації засобів річкового та морського транспорту. *Новітні технології*. 2019. Вип. 3(10). С. 6-12.

8. Шевченко А.П., Штрибець В.В., Трофіменко А.О. Розроблення фільтрових методів спектрального аналізу випадкових сигналів для контролю технічного стану двигунів засобів водного транспорту. *Slovak international scientific journal*. 2019. Vol.1. No.34. pp.30-38.

9. Шевченко А.П., Пліта Л.Л. Аналіз методів прогнозування технічного стану засобів водного транспорту. *Водний транспорт*. 2020. Вип.1(29). С.23–30.

*Наукові праці, які засвідчують апробацію матеріалів дисертації:*

10. Шевченко А.П., Богом'я В.І. Обґрунтування моделі функціонування автоматизованої системи управління засобів водного транспорту. Міжнародна науково-практична конференція «Водний транспорт: сучасний стан та перспективи розвитку»: тези доповідей, 16-17 травня 2019 року. Київ : ДУІТ, 2019. С.379-380.

11. Шевченко А.П., Коломієць О.М., Бойко С.О. Методи автоматизації контролю технічного стану засобів водного транспорту у різноманітних умовах експлуатації. Всеукраїнська інтернет-конференція студентів, аспірантів та молодих вчених «Технічні науки в Україні: сучасні тенденції розвитку»: тези доповідей, 20-21 листопада 2019 року. Київ :ДУІТ, 2019. С.121-123.

12. Шевченко А.П., Богом'я В.І. Моделі та методи прогнозування технічного стану засобів водного транспорту на основі використання м'яких обчислень. II Міжнародна науково-практична морська конференція кафедри СЕУ і ТЕ Навчально-наукового інституту Морського флоту Одеського національного морського університету. Одеса– 2020. С.174–178.

## ANOTATION

*Shevchenko A.P.* Models and methods for forecasting the technical state of soft-transport water transport. - Qualified scientific work on the rights of the manuscript.

Thesis for a Doctor of Philosophy Degree in Specialty 271 - Maritime and Annual Transport. - State University of Infrastructure and Technology, Kyiv, 2020.

In today's context, it is undeniable that water transport, as an infrastructure industry, should be developing at a rapid pace. According to experts, in 2020-2022, it is expected that shipments by sea and annual vessels will increase to 2500 million tons; cargo processing at state-owned commercial ports will also increase to around 240 million tonnes; passenger traffic will amount to more than 10.9 million passengers.

Research done within the framework of this dissertation shows that the level of accident-free navigation, the quality and efficiency of passenger and cargo transportation continue to improve. Particular attention is given to measures to ensure the required level of reliability of technical means. A significant role in this direction is the decision of the task of forecasting the technical condition of the means of water transport. This further underscores the relevance of scientific research to improve the operational efficiency of marine and annual vehicles by utilizing information on the forecast of their technical condition.

At present, progressive hardware and software solutions are actively used to ensure the guaranteed level of reliability of prediction of the technical state of water transport facilities. The analysis of foreign and domestic experience in the development and implementation of systems for predicting the technical condition of both component subsystems and means of water transport in general indicates the possibility of a significant increase in their efficiency due to the development of mathematical and algorithmic support. The most relevant in this area is the use of models and methods of artificial intelligence, namely, the so-called soft computing. Despite the rapid development of artificial intelligence theory in general, the challenge is to improve the existing and develop new mathematical and software computer systems for predicting the technical state of water transport based on soft calculations.

Thus, in solving the problems of development of the transport industry of Ukraine, water transport in particular, the scientific task of improving the existing

and development of new models and methods of forecasting the technical condition of water transport facilities on the basis of soft calculations is urgent, and this dissertation is devoted to solving this problem.

Scientific novelty of the obtained results is that:

for the first time developed a conceptual model for predicting the technical state of water transport, based on OLAP technology with intelligent data analysis on soft calculations, namely, the complex use of fuzzy clustering based on methods of subtractive clustering (to determine the number of values of linguistic variables) and fuzzy variables medium (FCM, Fuzzy C-Means) for building accessory functions, as well as the adaptive neural fuzzy ANFIS network. The software implementation of the model significantly improves the efficiency and reliability of the forecast by reducing the number of operations and the use of new principles of processing a priori information and self-study;

the model of forming the function of membership in the prediction of the technical state is improved, which, unlike the existing ones, uses fuzzy clustering: the method of subtractive clustering to determine the number of clusters that are interpreted as the rank of the base term-set of linguistic variables and the method of fuzzy c-means for the calculation; this greatly improves the adequacy of the phasing procedure and accuracy;

improved method for predicting the technical state of the FTA based on fuzzy inference when using a hybrid network, which, unlike the existing ones, is based on the model of adaptive neural fuzzy network system ANFIS (Adaptive Neuro-Fuzzy Inference System), which provides the opportunity to simultaneously use the benefits of fuzzy logic to improve the accuracy of predicting the technical condition.

The practical significance of the obtained results is that their implementation is expedient in mathematical and software, as a component of technology for automating the process of forecasting the technical condition of a ship in an integrated bridge system. The application of the proposed models and methods allows to increase the accuracy of the forecast of the technical condition in some

cases up to 20-25%. At the same time, according to the results of mathematical modeling, it is possible to increase the probability of failure-free navigation up to 10% due to implementation in practice.

The results of dissertation research were implemented in the State Enterprise of Waterways "Ukrvodshlyakh" (SE "Ukrvodshlyakh") during the practical training of drivers (act of 09.06.2020 №3 / 03/2020); as well as in the educational process of the State University of Infrastructure and Technology (act of 03.09.2020 №12 / a) during the creation of a course of lectures on the disciplines "Technical means of navigation" and "Operation of water transport".

The new science-based practical recommendations for a promising computerized system for predicting the technical state of a vessel allow considering the use of these approaches in other vehicle systems, as well as significantly improving the system of technical operation of the vessel by planning and timely implementation of appropriate measures.

Keywords: forecasting, fuzzy logic, neural network, failure, reliability, technical condition, means of water transport.

## **LIST OF BUILDER PUBLICATIONS**

*Scientific papers, in which the main scientific results of the dissertation are published:*

1. Shevchenko A.P., Gerasimov S.V., Tymochko O.I., Tymoschuk O.M., Trishin V.V. Optimal algorithm for navigation information processing in water transport management systems. Science-Based Technologies. 2018, No. 4 (40). pp.450-457.

2. Shevchenko A.P., Vorobyov E.S., Mazur A.M., Kolomiets O.M. Demyanenko SK Development of methods of compression of messages on air objects and control of discreteness of their delivery from sources of radar information. New technologies. 2018. Issue 3 (7). pp.217-230.

3. Shevchenko A.P., Trofimenko I.V., Mazur A.M. Improvement of measurement accuracy in ship radar systems taking into account tropospheric effects in conditions of heterogeneity of the marine environment. *New technologies*. 2018, Issue 2 (6). pp.61-68.

4. Shevchenko A.P., Mikhailova T.I., Boyko S.O. Method for predicting failure of ship complexes according to operational observations. *New technologies*. 2019. No. 1 (8). pp.52-58.

5. Shevchenko A.P., Budolak S.Y., Tkachenko V.V., Humennikov R.V. Method of structural synthesis of water transport management system. *Science-Based Technologies*. 2019. No. 1 (41). pp.101-108.

6. Shevchenko A.P., Drobina V.V., Kovtun O.V., Kornev I.O., Kotigora V.A. Features of creation of the automated system of management of the vessel. *New technologies*. 2019. No. 2 (9). pp. 125-130.

7. Shevchenko A.P., Plate LL, Daki OA Features of the organization of the process of operation of the means of river and sea transport. *New technologies*. 2019. No. 3 (10). pp. 6-12.

8. Shevchenko A.P., Stribets V.V., Trofimenko A.O. Development of filter methods for spectral analysis of random signals to control the technical condition of engines of water transport vehicles. *Slovak international scientific journal*. 2019. Vol.1. No.34. pp.30-38.

9. Shevchenko AP, Plate LL Analysis of methods for forecasting the technical condition of water transport vehicles. *Water transport*. 2020. Issue 1 (29). P.23–30.

*Scientific works certifying the approbation of the dissertation materials:*

10. Shevchenko A.P., Bogomyya V.I. Substantiation of the model of functioning of the automated system of management of means of water transport. *International Scientific and Practical Conference "Water Transport: Current Status and Prospects for Development"*: Abstracts, May 16-17, 2019. Kiev: DUIT, 2019. pp.379-380.

11. Shevchenko A.P., Kolomiets O.M., Boyko S.O. Methods of automation of control of a technical condition of means of water transport under various operating conditions. All-Ukrainian Internet conference of students, graduate students and young scientists "Technical Sciences in Ukraine: current trends": abstracts, November 20-21, 2019. Kiev: DUIT, 2019. pp.121-123.

12. Shevchenko AP, Bogomya VI Models and methods for forecasting the technical condition of water transport vehicles based on the use of soft calculations. II International Scientific and Practical Maritime Conference of the Department of SEU and TE Educational and Scientific Institute of the Navy of Odessa National Maritime University. Odessa– 2020. P.174–178.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	14
ВСТУП .....	15
РОЗДІЛ 1. АНАЛІЗ ПРАКТИЧНИХ ТА НАУКОВОВИХ МЕТОДІВ ПРОГНОЗУВАННЯ ТЕХНІЧНОГО СТАНУ ЗАСОБІВ ВОДНОГО ТРАНСПОРТУ. ПОСТАНОВКА НАУКОВОГО ЗАВДАННЯ.....	21
1.1. Оцінка технічного стану засобів водного транспорту .....	21
1.2. Дослідження сучасних та перспективних методів прогнозування технічного стану засобів водного транспорту .....	33
1.3. Визначення напрямків підвищення достовірності прогнозування технічного стану суден .....	37
1.4. Постановка наукового завдання .....	43
Висновки до розділу 1 .....	45
РОЗДІЛ 2. РОЗРОБКА МОДЕЛЕЙ ПРОГНОЗУВАННЯ ТЕХНІЧНОГО СТАНУ ЗАСОБІВ ВОДНОГО ТРАНСПОРТУ .....	47
2.1. Концептуальна модель прогнозування технічного стану засобів водного транспорту.....	47
2.2. Розробка моделі формування функції приналежності.....	57
2.3. Моделювання прогнозу технічного стану за допомогою розробленого програмного забезпечення .....	68
Висновки до розділу 2.....	81
РОЗДІЛ 3. РОЗРОБКА ТЕХНОЛОГІЇ НЕЧІТКОГО ВИСНОВКУ НА ОСНОВІ ГІБРИДНОЇ МЕРЕЖІ.....	83
3.1. Метод прогнозування технічного стану ЗВТ на основі нечіткої нейронної мережі .....	83
3.2. Формування навчальної вибірки гібридної мережі ANFIS.....	89
3.3. Комп'ютерне моделювання на базі ANFIS .....	93
Висновки до розділу 3.....	99

РОЗДІЛ 4. РОЗРОБКА РЕКОМЕНДАЦІЙ ЩОДО ПРАКТИЧНОЇ РЕАЛІЗАЦІЇ НАУКОВИХ РЕЗУЛЬТАТІВ. ПЕРЕВІРКА ДОСТОВІРНОСТІ.....	102
4.1. Рекомендації щодо реалізації моделей та методів прогнозування.....	102
4.2. Перевірка достовірності запропонованих наукових результатів ..	148
Висновки до розділу 4.....	150
ВИСНОВКИ.....	153
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	157
Додаток А.....	171
Акти впровадження.....	177
Додаток В.....	179
СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА ...	<b>Ошибка! Закладка не определена.</b>

**ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ**

ЗВТ – засіб водного транспорту;

КТС – контроль технічного стану;

ПЗ – програмне забезпечення;

ТЕ – технічна експлуатація;

ТО – технічне обслуговування;

ТС – технічний стан;

ШІ – штучний інтелект;

ШНМ – штучна нейронна мережа.

## ВСТУП

**Обґрунтування вибору теми дослідження.** В сучасних умовах не викликає сумніву те, що саме водний транспорт, як інфраструктурна галузь, має розвиватися швидкими темпами. За офіційними даними та на думку експертів у 2020-2022 роках, очікується збільшення обсягу перевезення вантажів морськими та річними судами до 2500 млн. тон; переробка вантажів у державних торговельних портах також збільшиться та становитиме близько 240 млн. тон; обсяги пасажирських перевезень становитимуть більш 10900 млн. пасажирів.

Дослідження, яке виконане в межах даної дисертації показує, що рівень безаварійного судноводіння, показники якості та ефективності перевезень пасажирів та вантажів продовжують покращуватися. Особлива увага приділяється саме заходам забезпечення заданого рівня надійності технічних засобів. Значна роль в цьому напрямку – рішення завдання прогнозування технічного стану засобів водного транспорту. Це ще більш підкреслює актуальність наукових досліджень з метою підвищення ефективності технічної експлуатації морських та річних транспортних засобів за рахунок використання інформації про прогноз їх технічного стану.

Проблеми розвитку водного транспорту України, зокрема системи технічної експлуатації розглядали та досліджували в своїх працях такі вчені: В.В. Панін, М.М. Цимбал, Б.В. Васильєв, Ю.Г. Дейнего, О.А. Оніщенко, С.І. Сербін, О.М. Тимошук, В.І. Богом'я, Д.І. Севастєєв, С.Р. Смирнов та інші.

На теперішній час для забезпечення гарантованого рівня достовірності прогнозування технічного стану засобів водного транспорту активно використовують прогресивні апаратні та програмні рішення. Аналіз закордонного та вітчизняного досвіду розробки та впровадження систем прогнозування технічного стану як комплектуючих підсистем, так і засобів водного транспорту в цілому свідчить про можливість значного підвищення

їх ефективності за рахунок розвитку математичного та алгоритмічного забезпечення. Найбільш актуальним в цьому напрямку є використання моделей та методів штучного інтелекту, а саме, так званих м'яких обчислень. Незважаючи на стрімкий розвиток теорії штучного інтелекту взагалі, актуальним є завдання удосконалення існуючого та розробки нового математичного та програмного забезпечення комп'ютерних систем прогнозування технічного стану засобів водного транспорту на основі м'яких обчислень.

Таким чином, при вирішенні проблематики розвитку транспортної галузі України, водного транспорту зокрема, актуальним є *наукове завдання* удосконалення існуючих та розробки нових моделей та методів прогнозування технічного стану засобів водного транспорту на основі м'яких обчислень, вирішенню цього завдання і присвячена дана дисертаційна робота.

#### **Зв'язок роботи з науковими програмами, планами, темами.**

Дисертаційна робота виконана відповідно до Морської доктрини України на період до 2035 року, затвердженої постановою Кабінету Міністрів України від 7 жовтня 2009 р. № 13074 та її нової редакції, затвердженої постановою Кабінету Міністрів України від 18 грудня 2018 р.; Програми економічних реформ з урахуванням основних напрямів розвитку, визначених у Транспортній стратегії України на період до 2020 р., затвердженій розпорядженням КМУ від 20.10.2010 р. № 2174-р. в інтересах науково-дослідної роботи «Розробка комплексного показника якості пасажирських круїзних суден змішаного плавання в системі безпересадкових круїзних перевезень між портами Дніпра, Чорного моря та Дунаю» (номер держреєстрації 0116U03946), яка виконувалася у Київській державній академії водного транспорту імені гетьмана Петра Конашевича-Сагайдачного, у якій автор приймала участь як виконавець. А також у відповідності до плану наукової роботи та науково-технічної діяльності Державного університету інфраструктури та технологій.

**Мета і задачі дослідження.**

Мета дослідження – підвищення достовірності прогнозу технічного стану засобів водного транспорту за рахунок впровадження запропонованих моделей та методів.

Для досягнення поставленої мети визначені наступні **часткові завдання** дослідження:

- проаналізувати технічний стан засобів водного транспорту та визначити напрямки забезпечення заданого рівня надійності;
- дослідити сучасні та перспективні методи прогнозування технічного стану засобів водного транспорту;
- розробити моделі та методи прогнозування технічного стану засобів водного транспорту на основі м'яких обчислень;
- розробити рекомендації щодо застосування запропонованих наукових результатів. Перевірити достовірність розроблених моделей та методів.

*Об'єкт дослідження* – процес прогнозування технічного стану засобів водного транспорту.

*Предмет дослідження* – моделі та методи прогнозування технічного стану засобів водного транспорту.

*Методи досліджень.* Для досягнення поставленої в роботі мети використано методи дослідження на основі системного підходу з застосуванням математичних моделей і методів дискретної математики, теорії ймовірностей, теорії надійності, штучного інтелекту, м'яких обчислень та нечітких нейронних мереж. Для виявлення наукового завдання використано системний підхід. Методика проведення експерименту та перевірки достовірності розроблених наукових положень реалізована на основі методів комп'ютерного моделювання.

**Наукова новизна отриманих результатів** полягає в тому, що в дисертаційній роботі:

вперше розроблено концептуальна модель прогнозування технічного стану засобів водного транспорту, яка базується на OLAP-технології з

інтелектуальним аналізом даних на м'яких обчисленнях, а саме, комплексному використанні нечіткої кластеризації на основі методів субтрактивної кластеризації (для визначення кількості значень лінгвістичних змінних) та нечітких с-середніх (FCM, Fuzzy C-Means) для побудови функцій приналежностей, а також адаптивної нейронної нечіткої мережі ANFIS. Програмна реалізація моделі значно покращує показники оперативності та достовірності прогнозу за рахунок зменшення кількості операцій та використання нових принципів обробки апріорної інформації та самонавчання;

удосконалено модель формування функції приналежності при прогнозуванні технічного стану ЗВТ, яка на відміну від існуючих використовує нечітку кластеризацію: метод субтрактивної кластеризації для визначення кількості кластерів, яка інтерпретується як ранг базової термножени лінгвістичної змінної та метод нечітких с-середніх для розрахунку значень функції приналежності; це дозволяє значно покращити адекватність процедури фазифікації та точність;

удосконалено метод прогнозу технічного стану ЗВТ на основі нечіткого висновку при застосуванні гібридної мережі, який на відміну від існуючих базується на моделі адаптивної нейронної нечіткої мережевої системи ANFIS (Adaptive Neuro-Fuzzy Inference System), а це надає можливість одночасно використовувати переваги нечіткої логіки та нейронних мереж для підвищення достовірності прогнозування технічного стану.

### **Практичне значення одержаних результатів.**

Реалізація одержаних в роботі результатів доцільна в математичному та програмному забезпеченні, як складової технології автоматизації процесу прогнозування технічного стану судна в інтегрованої мостикові системі (Integrated Bridge System). Застосування запропонованих моделей і методів дозволяє підвищити достовірність прогнозу технічного стану в деяких випадках до 8-15 %. При цьому, за результатами математичного

моделювання, можливе підвищення ймовірності виконання безвідмовного судноводіння до 10-12 % за рахунок впровадження на практиці.

Результати дисертаційних досліджень реалізовані у Державному підприємстві водних шляхів «Укрводшлях» (ДП «Укрводшлях») при виконанні практичної тренажерної підготовці судноводіїв (акт від 09.06.2020 р. №3/03/2020); а також в навчальному процесі Державного університету інфраструктури та технологій (акт від 03.09.2020 р. №12/а) підчас створення курсу лекцій з дисциплін «Технічні засоби судноводіння» та «Експлуатація засобів водного транспорту».

Наведені нові науково-обґрунтовані практичні рекомендації щодо перспективної комп'ютерної системи прогнозу технічного стану судна дозволяють розглянути можливість використання даних підходів у інших системах транспортних засобів, а також значно покращити систему технічної експлуатації судна за рахунок планування та своєчасного проведення відповідних заходів.

**Особистий внесок здобувача.** Основні результати дисертаційних досліджень отримані автором особисто. В роботах, опублікованих у співавторстві, автору належить: у [1] наведено аналіз алгоритмів оброблення навігаційної інформації у системах управління засобами водного транспорту; у [2] надається обґрунтування та варіант програмної реалізації алгоритму обробки інформації; у [3] наведено аналіз методів підвищення точності вимірювань в судових радіолокаційних системах; у [4] обґрунтована методика підвищення достовірності прогнозування відмов агрегатів судових комплексів; у [5] наведено особливості розроблення методу структурного синтезу системи управління засобів водного транспорту; у [6] обґрунтовано модель автоматизованої системи управління судном; у [7] наведено особливості організації процесу технічної експлуатації засобів річкового та морського транспорту; у [8] наведено результати аналізу застосування методів спектрального аналізу випадкових сигналів для контролю технічного стану двигунів засобів водного транспорту; у [9]

наведено результати аналізу методів прогнозування технічного стану засобів водного транспорту за допомогою математичного апарату м'яких обчислень; у [10] обґрунтування моделі функціонування автоматизованої системи управління засобів водного транспорту; у [11, 12] наведено результати досліджень щодо можливостей автоматизації контролю технічного стану засобів водного транспорту за допомогою математичного апарату м'яких обчислень.

**Апробація результатів дисертації.** Результати досліджень, які отримані у ході виконання дисертаційної роботи, доповідалися на міжнародних та всеукраїнських науково-технічних конференціях, а також на засіданнях кафедри експлуатації засобів транспорту на внутрішніх водних шляхах ДУІТ: Міжнародна науково-практична конференція «Водний транспорт: сучасний стан та перспективи розвитку», 16-17 травня 2019 року у ДУІТ, м. Київ; Всеукраїнська інтернет-конференція студентів, аспірантів та молодих вчених «Технічні науки в Україні: сучасні тенденції розвитку», 20-21 листопада 2019 року, ДУІТ м. Київ; II Міжнародна науково-практична морська конференція кафедри СЕУ і ТЕ Навчально-наукового інституту Морського флоту Одеського національного морського університету, 2020 року.

**Публікації.** Основні результати дисертаційної роботи відображені у 12 наукових працях, серед них: 9 статей, опублікованих у співавторстві у збірниках наукових праць, що входять до переліку видань, дозволених МОН України для публікацій результатів досліджень з технічних наук (у тому числі 1 у періодичному науковому виданні держави, яке входить до Європейського Союзу), 3 матеріалах науково-технічних конференцій.

**Структура і обсяг дисертації.** Дисертація загальним обсягом 180 сторінок машинописного тексту складається з вступу, чотирьох розділів, висновків, переліку використаних джерел з 124 найменувань. Дисертація містить 42 рисунки. Ілюстрації, таблиці, перелік використаних джерел та додатки займають 28 сторінок тексту.

## РОЗДІЛ 1 АНАЛІЗ ПРАКТИЧНИХ ТА НАУКОВОВИХ МЕТОДІВ ПРОГНОЗУВАННЯ ТЕХНІЧНОГО СТАНУ ЗАСОБІВ ВОДНОГО ТРАНСПОРТУ. ПОСТАНОВКА НАУКОВОГО ЗАВДАННЯ

### 1.1. Оцінка технічного стану засобів водного транспорту

Технічний стан ЗВТ є суттєвим фактором забезпечення безаварійного судноводіння. Аналіз офіційних статистичних даних стосовно аварійності на водному транспорті 2016-2019 р.р.[4,25,69-75] представлений на рис.1.1.

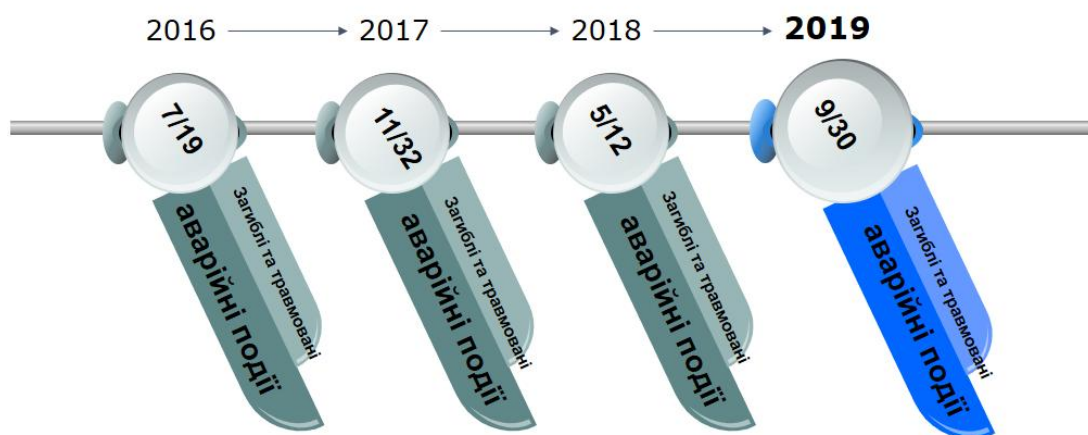


Рис. 1.1. Аналіз аварійності на водному транспорті 2016-2019р.р.

Публічний звіт Департаменту державного нагляду та контролю за безпекою на морському та річковому транспорті [73] дозволяє оцінити аварійність на морському та річковому транспорті за останній час. Цитуємо: «Протягом 1-го півріччя 2019 року на морському і річковому транспорті України, у тому числі з маломірними (малими) суднами сталося 15 аварійних подій, в яких 6 осіб загинуло, 1 особа зникла безвісти та 9 осіб отримали травм. За аналогічний період 2018 року відбулося 5 аварійних подій у яких: 1 людина загинула та 5 було травмовано.

Таким чином, загальна кількість аварійних подій, що сталися протягом 1-го півріччя 2019 року, у порівнянні з 2018 роком збільшилась на 10 аварійних подій (200 %), кількість загиблих та зниклих – збільшилась на 6 осіб (600 %). Кількість травмованих осіб збільшилась на 4 осіб (80 %).».

Актуальність теми дисертації підкреслює її наступна офіційна інформація. За результатами проведених розслідувань аварійних подій, що трапилися за період січень-червень 2019 року, встановлено, що причинами їх виникнення були наступні: рисунок 1.2.

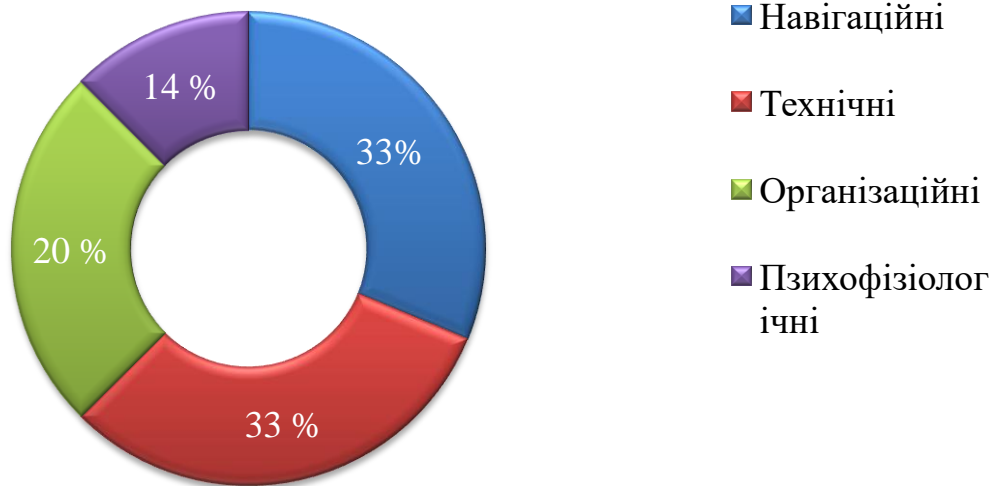


Рис. 1.2. Статистика аварійних подій за причинами виникнення у першому півріччі 2019 року

Матеріали НДР за темою: «Підвищення ефективності операторської діяльності в суднових ергатичних системах на морському транспорті», (Одеська Національна морська академія) (рис. 1.3) також підтверджують, що значна кількість аварійних ситуацій обумовлена так званими технічними причинами.

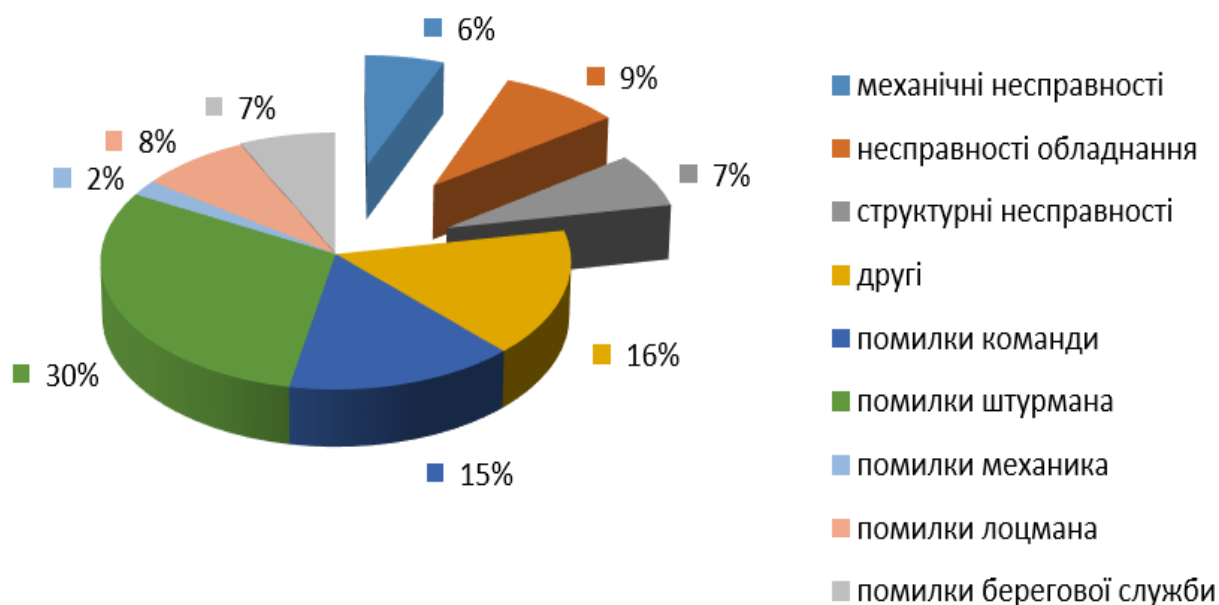


Рис. 1.3. Причини аварійних ситуацій на морському транспорті

Таким чином, аналіз стану безпеки руху засобів морського та водного транспорту України дозволяє зробити висновок про те, що саме «технічні причини» є найбільш значною причиною аварійності на морському та річному флоті за останні роки. Відомо, що на всіх видах транспорту це також актуальне. Це пояснюється складністю транспортних засобів, збільшенням інтенсивності транспортних перевезень, проблемами в системі підготовки фахівців та ін. Тому дослідження, які спрямовані на «...розроблення методів підвищення ефективності ... прогнозування технічного стану засобів транспорту, що забезпечують високу ефективність їх використання та надійність роботи» є актуальними та важливими для безаварійного судноводіння [50,68].

Визначення технічного стану судна є обов'язковою рисою його експлуатації, яке виконуються згідно ДСТУ 2389-94 «Технічне діагностування та контроль технічного стану. Терміни та визначення» [28] та інших керівних документів [29,36 та інші]. Наприклад, «Інструкція з проведення технічного нагляду за суднами рибної промисловості України, що не підлягають нагляду класифікаційного товариства» [36] вимагає: «4.1.

Технічний стан судна визначається за результатами огляду його корпусу, надбудови, пристроїв, обладнання, механічної установки, електрообладнання та перевірки протипожежного захисту».

Для класифікації придатності судна до плавання існують три оцінки технічного стану: «придатне», «обмежено придатне» та «заборонене до експлуатації». Судна, які «обмежено придатні», можуть допущені до плавання за умови обмеження: району плавання та віддалення від берега; допустимої потужності двигуна; вантажопідйомності.

#### *Огляд корпусу і надбудови*

«Оцінка технічного стану корпусу визначається відповідно до нормативно-правових актів і нормативних документів з технічної експлуатації судна, безпеки мореплавства і охорони праці, проектною документацією з метою визначення його зносу, наявності деформацій та інших пошкоджень, які зменшують загальну міцність корпусу та місцеву міцність окремих конструкцій» [36].

В даному випадку необхідно перевірити правильність нанесення ватерлінії або вантажної марки відповідно до існуючих вимог щодо висоти мінімального надводного борту.

Також повинні бути перевірені: зовнішня обшивка, набір корпусу та перебірки; палубні вирізи, настил палуб, комінгси люків і надбудов.

Для дерев'яних суден повинні бути оглянуті: штевні, транцеві дошки, та місця, які найбільш ймовірні для загнивання; також щільність конопатки і стан бортових кріплень.

В [36] визначена мінімальна допустима товщина обшивки корпусу суден: сталевій – бортові листи - 1,5 мм, скулові та днищеві - 2 мм; дюралюмінієвої – по всьому корпусу не менше величини, установлені виробником; дерев'яної – із дощок - не менше 20 мм, із бакелізованої фанери - не менше 6 мм.

Також визначено оцінка технічного стану корпусу як «придатний» у випадку: відсутність водопроникнення та пошкоджень обшивки корпусу і

палуб, розривів обшивки або набору корпусу; відсутність тріщин, корозійного зносу; надійне з'єднання елементів конструкцій корпусу у місцях зварювання та клепання (відсутність тріщин у швах, видимих непроварювань у швах, а також клепок і болтових з'єднань, які випали або послабшали); відсутність дефектів у транцеві дощі та її відповідність потужності встановленого на судні підвісного двигуна; відсутність ознак червоточин, ураження гниллю елементів корпусу дерев'яних суден; відсутність гофр, вм'ятин, а також деформації елементів корпусу, які послабляють загальну міцність корпусу судна; відсутність розшарувань, надрізів, короблень та інших пошкоджень пластикових суден і суден з фанери, які можуть спричинити водопроникнення.

Оцінка технічного стану корпусу – «обмежено придатний» є у випадку наявності будь-якого дефекту, який *не впливає* на загальну міцність корпусу судна.

Оцінка технічного стану корпусу – «заборонений до експлуатації» у випадках невідповідності його вимогам, що *впливають* на загальну міцність корпусу, та відсутність свідоцтва про стабільність і надводний борт судна.

#### *Огляд механізмів судна*

Звертаємо увагу на наступне: «6.1. Огляд механізмів здійснюється відповідно до нормативно-правових актів і нормативних документів з технічної експлуатації судна, безпеки мореплавства і охорони праці з метою визначення рівня готовності механізмів судна до його безпечної експлуатації. Перевірка механізмів здійснюється в дії протягом 0,5 - 1,0 години в різних режимах» [36].

Таким чином, перевіряють: валопроводи та системи двигуна і судна в цілому; робочі параметри двигуна (частота обертання, тиск і температура мастила й охолоджувальної води тощо), які не повинні виходити за визначені межі.

Оцінка технічного стану механічної установки як «придатна» є:

справний стан механічної установки в цілому, що забезпечує штатну роботу її в усіх режимах;

відсутність пошкодження або люфту гребного валу (гвинта), значної димності, перегріву деталей, упорного підшипника, вузлів і поверхонь реверсивного обладнання;

відсутність значної вібрації, яка може призвести до виникнення тріщини у двигуні, фундаменті, у корпусі та системах ЗВТ;

надійне кріплення двигуна (вузлів двигуна) до фундаментів і корпусу ЗВТ;

відсутність дефектів (тріщин, деформацій тощо) у корпусі, фундаменті і навісних агрегатах двигуна, редуктора та гребного валу;

наявність у системі двигуна стоп-кнопки або ключа для аварійної зупинки двигуна;

справність вимірювальних приладів;

наявність на підвісних моторах стопорного пристрою а також страхувального кінця, який попереджає відкидання мотора при реверсуванні на задній хід;

відповідність потужності двигуна корпусу судна;

надійне функціонування реверсивного пристрою;

відгородження моторного відділення водонепроникною перебіркою.

Якщо при оцінці технічного стану механічної установки є невідповідність будь-якій з вимог, зазначених вище, то маємо оцінку – «заборонена до експлуатації».

Також експлуатація двигунів і механізмів забороняється у випадках:

несправність пускового або реверсивного пристроїв;

несправність допоміжних механізмів головного двигуна;

поломка (деформація) лапасте гребного гвинта;

сторонній стуків у двигуні та вібрацій валопроводу;

відхилення робочих параметрів двигуна за межі, установлені заводом-виробником;

тріщини у корпусі двигуна.

Оцінка ТС «обмежено придатна» згідно з вимогами до технічного стану механічної установки не встановлюється.

*Огляд електрообладнання судна*

При огляді повинно бути перевірено щонайменше:

надійність кріплення кабелів; захист кабелів і електропроводів від впливу палива, мастила, підвищених температур, механічних пошкоджень; стан сальників у місцях проходів проводів через непроникні перебірки; стан акумуляторів.

Стан акумуляторів повинні відповідати вимогам «Правил експлуатації суден флоту рибної промисловості, що не підлягають нагляду класифікаційного товариства», затверджених наказом Міністерства аграрної політики та продовольства України від 22 жовтня 2012 року № 642, зареєстрованих у Міністерстві юстиції України 27 грудня 2012 року за № 2200/22512.

Підставою для оцінки технічного стану електрообладнання як «придатне» є:

безперебійне електроживлення споживачів; надійна ізоляція електропроводів; водозахисне виконання електропроводів, сигнально-розпізнавальних вогнів та світильників; установлення вимикачів, штепсельних рознімачів у місцях, захищених від впливу вологи; належний стан і надійне кріплення акумуляторних батарей, наявність вентиляції при розміщенні акумуляторів у закритих ящиках.

Підставою для оцінки технічного стану електрообладнання як «заборонене до експлуатації» є невідповідність одній з вимог, зазначених вище. Оцінка «обмежено придатне» згідно з вимогами до технічного стану електрообладнання судна не встановлюється.

*Перевірка протипожежного захисту судна*

Вимоги до оцінки ТС протипожежного захисту судна як «придатне» [36]:

розташування паливних баків (цистерн) на відстані не менше 800 мм від двигуна і газовихлопу за відсутності перебірок;

розташування вузлів паливної системи стаціонарного двигуна на протилежному від газовихлопу боці двигуна;

компонування і конструкція, що унеможливають витікання палива за борт, а також потрапляння палива в приміщення (відсіки) суден, призначені для розміщення екіпажу та вантажу;

наявність природної або примусової вентиляції відсіків для двигуна і перегородок для розміщення паливних баків (цистерн);

виконання перегородок (відсіків), кожухів з вогнезахисних (самозагасаючих) матеріалів;

наявність теплової ізоляції на деталях механізмів та приладів, що нагріваються, а також на конструкціях, які прилягають до них;

установлення запірною крана (клапана) на паливному трубопроводі від паливного бака до двигуна;

обладнання полум'ягасною сіткою усмоктувального патрубку карбюратора;

розміщення на штатних місцях справного протипожежного інвентарю, стан якого забезпечує його негайне використання.

Підставою для оцінки технічного стану протипожежного захисту судна як «заборонене до експлуатації» є невідповідність одній з зазначених вимог. Оцінка «обмежено придатне» згідно з вимогами до технічного стану протипожежного захисту судна не встановлюється.

#### *Перевірка дотримання вимог правил охорони праці*

Повинно бути перевірено наступне [36]:

1) кут відчинення кришок світлових люків та інших люків, передбачених для руху людей, аварійних виходів, їх фіксацію пристроями у відчиненому положенні. Кут відчинення має становити не менше 90°;

2) наявність пристроїв, які утримують кришки люків і двері у відчиненому положенні і запобігають їх непередбачуваному відчиненню і зачиненню;

3) кут і правильність відчинення стулчастих ілюмінаторів, наявність стопорів для утримання штормових кришок і рам у відчиненому та зачиненому положеннях. Ілюмінатори повинні відчинятися усередину приміщень на кут не менше  $90^\circ$ ;

4) наявність сходинок шириною не менше 150 мм і довжиною, яка дорівнює ширині дверей, розташованих на половині висоти комінгса. Комінгси дверей висотою більше ніж 350 мм повинні бути обладнані сходинками з обох боків;

5) висоту комінгсів дверей житлових, побутових та службових приміщень, розташованих усередині корпусу, і надбудови судна. Висота комінгсів повинна бути не менше ніж 100 мм;

б) кут відчинення дверей приміщень, розташованих усередині корпусу і надбудови.

Двері у відчиненому положенні не повинні перекривати проходи та зменшувати встановлені габарити шляхів сполучення, вони повинні відчинятися всередину цих приміщень не менше ніж на  $90^\circ$ .

Двері, які ведуть з відкритих палуб у приміщення, повинні відчинятися назовні. Кут відчинення має бути не менше ніж  $170^\circ$ . У технічно обґрунтованих випадках на суднах довжиною менше 24 м допускається зменшення кута відчинення зовнішніх дверей. При цьому ширина проходу в приміщення у разі повного відчинення зовнішніх дверей повинна бути не менше 500 мм;

7) наявність пристроїв, що фіксують двері у відчиненому положенні.

Цими пристроями повинні бути обладнані вхідні двері житлових, службових приміщень і всі зовнішні двері, що ведуть з відкритих палуб до внутрішніх приміщень;

8) наявність поверхонь, що запобігають ковзанню;

9) припасування і кріплення плит настилів машинних відділень, наявність отворів або інших пристосувань для їх зняття й установки;

10) наявність страхувальних пристроїв (римів, обухів, кілець, канатів тощо) для страхувальних кінців запобіжних поясів та штормових леєрів у місцях, де можлива небезпека змиву чи падіння людини за борт або з висоти;

11) наявність щитів, пайолів на днищевому наборі безпалубних суден, стан проходів.

Проходи повинні мати рівну поверхню без конструкцій, що виступають і перехрещуються (балок, трубопроводів, римів, приводів арматури тощо).

Наявність штатного закриття усіх отворів у проходах;

12) ширину проходу з боку поста управління головними механізмами.

На палубних суднах ширина вказаного проходу повинна бути не менше 700 мм, при дистанційному управлінні - не менше 400 мм, на безпалубних суднах - не менше 350 мм;

13) наявність і облаштування штормових перехідних леєрів, які повинні встановлюватися на висоті 1200 мм від палуби, бути оснащені талрепами і спеціальними рухомими ручками зі скобами;

14) наявність і облаштування штормових поручнів.

На палубних суднах леєри повинні бути передбачені на відкритих палубах, у коридорах, на стінках (перебірках) на одному рівні від палуби й бажано на висоті 1200 мм. Штормові поручні на безпалубних суднах повинні бути установлені вздовж шляхів сполучення на механізмах і пристроях.

Відстань між поручнями і поверхнями перебірок та інших конструкцій судна повинна бути не менше ніж 60 мм;

15) освітлення палуб та проходів, що ведуть до робочих місць, житлових або службових приміщень;

16) розміщення рятувальних кругів з лінями і наявність стаціонарного освітлення або пристроїв для закріплення переносних освітлювальних пристроїв у місцях установки сходнів, штормтрапів та інших зовнішніх трапів;

17) ширину зовнішніх і внутрішніх трапів, що мають нахил, яка повинна бути не менше 600 мм. На суднах довжиною менше 24 м у місцях, де неможливе влаштування трапів шириною 600 мм, допускається застосування трапів шириною 500 мм. Нижня сходинка вертикальних трапів повинна розташовуватись не вище 350 мм від палуби (майданчика), з якої здійснюється вхід на трап.

Крок сходинки повинен бути 180-255 мм (залежно від кута нахилу трапа);

18) поверхню сходинок металевих трапів.

Поверхня повинна бути з рельєфним візерунком (рифлення тощо). Сходинки, виготовлені з матеріалів з гладкою поверхнею (крім виготовлених з дерева) або покриті такими матеріалами, на відкритій ширині повинні мати рифлені планки для запобігання ковзанню;

19) наявність огороження або поручнів з обох боків трапів.

Висота огороження внутрішніх і зовнішніх трапів повинна бути не менше 1200 мм. Леєрне огороження трапів, крім поручня огороження, повинно мати не менше 2 проміжних леєрів. Леєрні стійки огороження повинні встановлюватися перпендикулярно до основи трапа.

#### *Сучасні методи оцінки технічного стану обладнання ЗВТ*

В даний час особливо актуальні питання оцінки стану обладнання ЗВТ. Це багато в чому пов'язано з тим, що істотна частина комплектуючого обладнання виробила ресурс або термін експлуатації, визначений нормативними документами. Для оцінки стану обладнання ЗВТ використовується моніторинг за допомогою комплексу апаратних і програмних засобів. Такий комплекс призначений для реєстрації різної інформації про параметри устаткування і його складових елементів, умов їх експлуатації та ін., а також для виконання аналізу стану обладнання на основі отриманих даних [11-16].

Слід зазначити, що вартість системи моніторингу основного обладнання становить близько 5-8% від вартості об'єкта контролю.

З огляду на скорочення витрат, пов'язане з плановими і аварійними ремонтами обладнання, термін окупності таких систем не перевищує п'яти років.

Однак навіть при таких термінах окупності і досить помірною вартості економічна (а іноді і технічна) можливість установки системи моніторингу на кожному ЗВТ часто відсутня. У таких випадках хорошою альтернативою підсистемі моніторингу, пов'язаної з оцінкою технічного стану обладнання, може служити система аналізу даних технічної діагностики та випробувань і / або будь-якої доступної інформації про ЗВТ, отриманої в ході його експлуатації [66].

Існують різні визначення діагностики (процесу діагностування) і моніторингу ЗВТ. У контексті цієї дисертаційної роботи доречно скористатися визначенням діагностики як інтелектуального угруповання зв'язкових даних, що включають статистику значень і тренди, які згодом обробляються експертною системою для видачі конкретної інформації про стан обладнання ЗВТ і рекомендованих заходів. Визначити моніторинг можна як накопичення базисних даних [67].

Основною метою технічного діагностування в першу чергу є розпізнавання стану об'єкта в умовах обмеженої інформації і, як наслідок, оцінка залишкового ресурсу системи (обладнання) та підвищення її надійності [1,2,10]. Різні технічні системи мають різну структуру і призначення, тому не можна до всіх системам застосовувати один і той же вид технічного діагностування. У роботі під системою моніторингу мається на увазі саме оцінка технічного стану ЗВТ.

Безумовно, система моніторингу в порівнянні з системою аналізу даних технічного діагностування має низку технічних переваг [75]:

- 1) можливість діагностування в процесі експлуатації;
- 2) визначення фактичного технічного стану в темпі процесу;
- 3) висока достовірність вихідних даних;

4) можливість зберігання вихідних даних (параметрів і характеристик обладнання), що дозволяє виявити залежності і взаємозв'язку між подіями.

У плані оцінки стану ЗВТ для визначення, наприклад, інтегральної оцінки стану ЗВТ, система моніторингу має ряд недоліків:

1) система моніторингу дуже рідко включає в себе вичерпний набір даних по всьому устаткуванню ЗВТ. Вона спрямована на моніторинг основних підсистем;

2) вихідна інформація являє собою набір даних, аналіз і обробка яких складають окрему задачу, рішення якої вимагає відповідного математичного апарату з його програмною реалізацією.

Крім того, для установки системи моніторингу на ЗВТ повинні існувати певні умови:

- обладнання повинно мати порівняно невисоку вартість;
- передбачуваний збиток при відмові такого обладнання повинен бути значним.

## **1.2. Дослідження сучасних та перспективних методів прогнозування технічного стану засобів водного транспорту**

Варте звернути увагу на те, що в процесі створення технічних систем судна, їх зберігання та експлуатації важливо вміти прогнозувати технічний стан ЗВТ за інформацією про ТС комплектуючих систем [40,49,57]. Іншими словами – знати якими характеристиками вона володіє в даний момент часу та в майбутньому. Відомо, що стан системи оцінюється по конкретно обраним значенням вихідних параметрів, що визначають її працездатність. Ці параметри під впливом різних факторів змінюються випадковим чином. Процес такої зміни полягає в тому, що вплив зазначених чинників сприймається елементами системи, внаслідок чого змінюються ці параметри. Так як вихідні параметри системи пов'язані деяким оператором з параметрами елементів, то при зміні останніх змінюються і вихідні

параметри, тобто змінюється технічний стан системи. Таким чином, для того щоб прогнозувати стан системи необхідно визначати значення її параметрів. Це завдання вирішується засобами звичайного технічного контролю, що дозволяє отримати дані про параметри в момент їх вимірювання.

Ефективне прогнозування зміни параметрів систем ЗВТ дозволяє вирішувати завдання запобігання їх відмов і оптимізувати організацію технічного обслуговування. Відомо що ефективною системою технічного обслуговування є система обслуговування по фактичному стану. Інформаційною основою, на базі якої будуються обслуговування за станом, є деякі апріорні відомості про закономірності процесів дрейфу параметрів систем ЗВТ [75].

Обслуговування за фактичним станом засноване на оцінці технічного стану досліджуваного об'єкта за даними контролю його параметрам і прогнозування дрейфу параметрів. Індивідуальне прогнозування стану дозволяє експлуатувати контрольований об'єкт до виявлення ознак відмови, а саме, виключити передчасне втручання в його роботу і виконання трудомістких операцій з обслуговування.

Дрейф параметрах системи обумовлений дією ряду факторів, у тому числі, зазвичай виділяють технологічні, виробничі та експлуатаційні. До виробничих чинників відносять неоднорідність матеріалів, похибки приладів і інше. Дія виробничих факторів викликає технологічний розкид параметрів, а також деяка відмінність фізико-хімічних структур елементів входять до системи. У процесі зберігання і експлуатації в елементах, що складають систему, протікають фізико-хімічні процеси, які викликають незворотні зміни параметрів. Це явище називають старінням. Причинами старіння є дифузія речовин і інші фактори. Крім старіння слід зазначити знос елементів комплектуючих системи [60-64].

В літературе існує декілька класифікацій методів та принципів прогнозування. Наприклад, їх розділяють на три основні групи [1,2,76]:

- методи експертних оцінок, (узагальнення, статистична обробка й аналіз думок фахівців);
- методи моделювання (формування моделі об'єкта дослідження, проведення модельних експериментів і перерахування добутих значень із моделі на натуральний об'єкт);
- статистичні методи (наприклад, метод екстраполяції та ін.), у основі яких лежать закономірності зміни прогнозованих параметрів у часі. Для опису цих закономірностей використовують аналітичну функцію з мінімальною кількістю змінних. Найбільше розповсюджені методи статистичного моделювання.

Доцільне підкреслити, що у багатьох наукових роботах [1,46,77 та ін] теоретичною основою досліджень прогнозування технічного стану є імовірнісний підхід. Завдання прогнозування стану і надійності системи зводиться до прогнозування нестационарного векторного випадкового процесу. Це завдання тісно пов'язана з проблемою екстраполяції випадкових функцій. У ряді наукових робіт завдання планування експлуатації за даними прогнозу параметрів вирішується як завдання управління апріорним випадковим процесом для неконтрольованих об'єктів, а також як задача управління апостеріорним або умовним випадковим процесом при індивідуальному технічному обслуговуванні контрольованих об'єктів. Алгоритми рішення задач прогнозування технічного стану при повній визначеності вихідних статистичних даних добре описані в літературі. Зазвичай їх основу складають класичні методи математичної статистики: метод найменших квадратів, максимального правдоподібності і інші. Частиною таких алгоритмів є і оптимальні фільтри. Дійсно, з точки зору прогнозування технічного стану становлять значний інтерес оптимальні фільтри. Оптимальна фільтрація, як частина теорії управління, має в своєму розпорядженні добре апробований математичний апарат і детально розроблена для вирішення завдань управління [99]. Ці підходи використовують і при вирішенні задач оцінки технічного стану [100].

Дані моделі, методи і алгоритми прогнозування надійності мають істотний дефект, суть якого в тому, що має бути точно відома модель випадкового процесу дрейфу параметрів системи. В даному випадку, прогнозування надійності в умовах невизначеності вихідних даних здійснюється на основі так званих робастних підходів. Однак, робастні або грубі оцінки не завжди забезпечують задану достовірність оцінки технічного стану. Наступні принципи і методи оцінки технічного стану, а також його прогнозування об'єднуються терміном "евристичні". Надалі евристичні моделі і методи отримали узагальнення і формалізацію через моделі і методи штучного інтелекту.

Так у роботі [76] наведено результати розробки методу оцінки, прогнозування технічного стану функціонально-взаємопов'язаних і взаємодіючих підсистем суднових технічних систем при екстремальних умовах. Доведені когнітивні імітаційні, нечіткі моделі оцінки та прогнозування технічного стану підсистем ЗВТ, також удосконалене метод інформаційного забезпечення підтримки прийняття рішень при пошуку причин відмов елементів суднових систем.

Варте проаналізувати сучасні системи оцінки технічного стану будь-яких систем, не тільки ЗВТ.

Сучасні системи оцінки технічного стану будь-яких систем є автоматизовані експертні системи, спрямовані, в основному, на вирішення двох видів завдань – визначення стану обладнання з метою виявлення дефектів і несправностей і оптимізація управляючих впливів на комплектуючі підсистеми та елементи з метою підвищення надійності роботи і продовження терміну експлуатації об'єктів. Деякі європейські системи, на відміну від інших, не ставлять собі завданням продовження терміну служби як комплектуючих, так і системи в цілому, оскільки зарубіжна практика передбачає заміну після закінчення його терміну служби. Також існують досить великі відмінності в нормативній документації по обслуговуванню, діагностиці, випробувань, складом обладнання та його

експлуатації, які не дозволяють використовувати закордонні системи оцінки технічного стану обладнання для вітчизняних будь-яких систем.

У Україні подібних експертних систем, використовуваних на реальних об'єктах будь-якої техніки, вкрай мало і більшість з них спрямовані в основному на рішення задачі оцінки стану окремих одиниць обладнання.

У сучасній практиці в залежності від обсягу і повноти вихідних даних, а також області застосування для аналізу даних технічної діагностики використовуються математичні методи, більш детально описані в [1]: логічні методи розпізнавання кривих; методи поділу в просторі ознак; методи, засновані на визначенні цінності ознак; методи, засновані на теорії ймовірності; методи, засновані на теорії інформації; статистичні методи розпізнавання; експертні методи та ін.

Кожен з методів має як достоїнствами, так і недоліками. Однак використання більшості з них в реальних системах важко, так як при комплексній оцінці стану всіх типів комплектуючих підсистем потрібно вирішувати по суті багатокритеріальну задачу прийняття рішень про подальшу експлуатацію системи на основі існуючої, часто різномірної і неповної, інформації.

У системах оцінки технічного стану, існуючих на сьогоднішній день, з точки зору математичної основи використовуються або прості моделі на основі звичайних правил продукції, або складніші, наприклад, експертна система оцінки технічного стану заснована на методі Байєса. Аналіз функціональних можливостей сучасних експертних систем і принципи їх побудови для діагностики стану розглянуті в [75].

### **1.3. Визначення напрямків підвищення достовірності прогнозування технічного стану суден**

Дослідження показали те, що сучасні засоби діагностики та прогнозування технічного стану ЗВТ базуються на використанні ознак та

характеристик різноманітних фізичних явищ, реалізацію складних технологій збору, обробки, зберігання, передавання та представлення інформації. Тому достовірною оцінкою технічного стану ЗВТ як на стадії експлуатації, так і в загалі передбачає використання відповідних діагностичних апаратних та програмних засобів, складовими частинами якого є технологічне, апаратне, математичне, алгоритмічне та програмне забезпечення [14]. У зв'язку з цим основні напрямки підвищення достовірності прогнозування технічного стану суден спрямовані на розвиток математичних методів виділення і розпізнавання діагностичної інформації, створення ефективних вимірювальних комплексних систем збору, обробки та представлення інформації, аналіз умов експлуатації судових машин і механізмів, специфічних особливостей конструкцій, вивчення типових руйнуючих процесів і їх можливих наслідків, виділення вузлів, які суттєво впливають на безвідмовність і довговічність обладнання, пошук діагностичних ознак деградації технічного стану, синтез алгоритмів прогнозування, орієнтованих на застосування тих чи інших методів і засобів прогнозування.

Слід зазначити, що розширене впровадження засобів прогнозування технічного стану ЗВТ, з одного боку, стимулюється неухильно зростаючими вимогами до надійності і безпеки експлуатації суден і кораблів, але з іншого боку, пов'язано з необхідністю вирішення цілого ряду проблем, пов'язаних з недостатньою вивченістю алгоритмів експлуатаційних відмов та пошкоджень найбільш навантажених вузлів і деталей, відсутністю необхідної кількості статистичних даних про характеристики експлуатаційних відмов, складністю отримання адекватних але повних та достовірних моделей оцінки та прогнозування технічного стану комплектуючих ЗВТ, недостатньою вивченістю характеру впливу можливих відмов (несправностей) на параметри, які супроводжують використання обладнання та судна в цілому за прямим призначенням, складністю адаптації традиційно застосовуваних методів неруйнівного контролю до умов безрозбірної діагностики обладнання в судових умовах, недостатньо розвиненою базою

математичного і програмного забезпечення, нестачею спеціальних датчиків і вимірювальних приладів, відсутністю спеціального моделюючого обладнання для відпрацювання та випробування діагностичних методів, відсутністю системи підготовки кваліфікованих фахівців в області прогнозування технічного стану ЗВТ, низьким рівнем фінансового і матеріального забезпечення.

З іншого боку на ринку представлена найширша номенклатура діагностичних приладів і систем, (як вітчизняних, так і імпортних) що розрізняються особливостями методичного, математичного, алгоритмічного, програмного та апаратного забезпечення, та мають різні принципи дії, функціональні можливості, метрологічні характеристики, області застосування, масогабаритні параметри, елементну базу, співвідношення ціни та ін. На даний час, наприклад, в області вібро діагностики можна нарахувати досить багато виробників, що випускають множину різних датчиків для вимірювання параметрів вібрації. аналогічно ї з іншими напрямками [66].

На думку експертів це призвело до того, що на зміну проблеми дефіциту діагностичних методик і приладів прийшла нова проблема – проблема вибору компонування діагностичного забезпечення, яке здатне забезпечити потрібний рівень достовірності прогнозування та діагностування. Рішення цієї проблеми вимагає, як правило, проведення комплексу теоретичних і експериментальних досліджень, спрямованих на аналіз специфічних умов експлуатації ЗВТ, конструктивних особливостей обладнання, аналіз умов навантаження і характерних експлуатаційних пошкоджень найбільш відповідальних вузлів і деталей, визначення можливих діагностичних ознак, розробки діагностичних методів і приладів, здатних функціонувати в судових умовах експлуатації, застосування методик порівняльного багатофакторного аналізу і критеріальної оцінки діагностичних можливостей різних варіантів компонування діагностичного обладнання [67].

Таким чином, головною тенденцією еволюції систем контролю та прогнозування технічного стану як судового обладнання, так і ЗВТ в цілому слід вважати перехід від екстенсивного розвитку систем централізованого контролю до якісної зміни їх функціональних можливостей за рахунок застосування спеціальних діагностичних методик, алгоритмів і апаратури, адаптованих до специфічних особливостей об'єктів діагностування, тобто мова йде, по суті, про необхідність впровадження в практику експлуатації суден принципово нового класу інформаційних систем, а саме, інтелектуальних та інтелектуалізованих систем [5,12,26,75]. Тут слід зауважити, що практична реалізація цього завдання за своєю складністю і багатоплановості не поступається завдання забезпечення високого рівня автоматизації судових технічних засобів.

Варте звернути увагу на те, що сучасний високо достовірний процес прогнозування ТС безумовне пов'язаний саме з комп'ютерною обробкою інформації з застосуванням передових технологій. Тому цей напрям – широкий спектр питань штучного інтелекту [10,19,30]. Дійсне, сучасний етап розвитку суспільства та технологій характеризується необхідністю вирішення задач, які значно ускладнилися. Труднощі вирішення цих задач пояснюються тим, що об'єктам і процесам і, які досліджуються, окрім всіх основних рис, якими володіють складні системи, мають такі особливості [39]:

- велика кількість слабоформалізованих і часто суперечливих цілей функціонування, їх швидка змінність у часі;
- конфліктний і багатоаспектний характер взаємовідносин як між елементами всередині об'єкта (процесу), так і з зовнішніми об'єктами (процесами) при сильному впливі людського фактору.

За наявності перерахованих властивостей неможливо побудувати жорстку формальну модель об'єкта, яка орієнтована на дані, і рішення задач припадає шукати у класі інших методів, зокрема у тих, що розробляються у теорії штучного інтелекту (ШІ). ШІ відносно молодий науковий напрямок, який в роботі розглядається в контексті розробки програмних засобів, які

імітують розумову діяльність людини при вирішенні певного класу задач, що не можуть бути формалізованими.

Під штучним інтелектом розуміють комплексний науковий напрямок, метою якого є створення і застосування програмно-апаратних засобів, що дозволяють моделювати процес людського мислення (окремі функції творчої діяльності) і забезпечувати діалог з комп'ютером мовою, природною для людини [31-33 та інші]. ШІ – це метафорична назва нових інформаційних технологій обробки інформації, створених фахівцями різноманітних галузей науки і техніки (математиками, інженерами, психологами, лінгвістами, військовими вченими та ін.).

Основними напрямками застосування систем ШІ, з точки зору їхнього використання є: навчальні програми; логіко-лінгвістичні моделі; гібридні експертні системи; системи підтримки прийняття рішень (СППР); обробка зображень; розпізнавання образів (ситуацій) [83].

Термін "м'які обчислення" (Soft Computing) був запроваджено в 1994 р. основоположником теорії нечітких множин Л.Заде [121] (Zadeh, Lotfi A., «Fuzzy Logic, Neural Networks, and Soft Computing», Communications of the ACM, March 1994, Vol. 37 No. 3, pp 77-84.). Традиційний термін "штучний інтелект" був розширений до термін "обчислювальний інтелект" , який пов'язують з концепцією м'яких обчислень і м'яких знань. Дослідження багатьох наукових праць [31,32,101-121] показало те, що термін "м'які обчислення" в математичному та алгоритмічному сенсі базується на наступних моделях та методах:

- Теорія можливостей.
- Нечіткі множини, нечітка логика, нечітке управління і споріднені з ними формалізми для моделювання невизначеності, наприклад, суб'єктивне-ймовірності моделі; наближені міркування.
- Можливісте-імовірнісні моделі та методи прийняття рішень.
- Еволюційне моделювання та генетичні алгоритми.
- Системи з хаотичною динамікою.

Таким чином, перспективні напрямки підвищення достовірності прогнозування ТС ЗВТ в алгоритмічному сенсі, безумовно базуються та так званих м'яких обчисленнях (рис.1.4).



Рис.1.4. Напрямки підвищення достовірності прогнозування ТС ЗВТ

Створення ефективного інтелектуалізованого діагностичного забезпечення передбачає, перш за все проведення системного аналізу об'єктів діагностування та прогнозування з метою виділення і типізації характерних процесів деградації технічного стану найбільш навантажених і відповідальних вузлів і деталей, що лімітують безвідмовність і довговічність обладнання, визначення раціональної глибини діагностування, систематизації діагностичних завдань і подальшого аналізу і вибору методів і засобів діагностування, а також визначення вихідних передумов для дослідження і розробки принципово нових діагностичних методик [94].

#### **1.4. Постановка наукового завдання**

В сучасних умовах економічних, соціальних та політичних змін країни саме водний транспорт, як інфраструктурна галузь, має розвиватися швидкими темпами. На думку експертів у 2020-2022 роках, очікується збільшення обсягу перевезення вантажів морськими та річними ЗВТ до 2500 млн. тон; переробка вантажів у державних торговельних портах також збільшиться та становитиме близько 240 млн. тон; обсяги пасажирських перевезень становитимуть більш 10900 млн. пасажирів [4-8,34].

Дослідження, яке виконане в межах дисертації показує, що рівень безаварійного судноводіння, показники якості та ефективності перевезень пасажирів та вантажів продовжують покращуватися. Особлива увага приділяється саме заходам забезпечення заданого рівня надійності. Значна роль в цьому напрямку – рішення завдання прогнозування технічного стану засобів водного транспорту. Це ще більш підкреслює актуальність наукових досліджень в галузі ефективної технічної експлуатації морських та річних транспортних засобів за рахунок використання інформації про прогноз ТС ЗВТ.

Варто підкреслити те, що для забезпечення гарантованого рівня достовірності прогнозування технічного стану засобів водного транспорту активне використовують прогресивні як апаратні, так і програмні рішення. Аналіз закордонного та вітчизняного досвіду впровадження даних систем свідчить про можливість значного підвищення їх ефективності за рахунок розвитку математичного та алгоритмічного забезпечення, як складової штучного інтелекту. Незважаючи на стрімкий розвиток теорії штучного інтелекту взагалі, актуальним є наукове завдання удосконалення існуючих та розробки нових моделей та методів прогнозування технічного стану засобів водного транспорту на основі м'яких обчислень, вирішенню цього завдання і присвячена дана робота.

Дамо символічну формалізацію наукового завдання.

Потрібно визначити кортеж

$$\langle \beta, T, X, G, M \rangle, \quad (1.1)$$

де  $\beta$  – лінгвістична змінна технічного стану ЗВТ;

$T$  – множина термів лінгвістичної змінної  $\beta$ ;  $T = \{\text{«придатне»}, \text{«обмежено придатне»}, \text{«заборонене до експлуатації»}\}$ ;

$X$  – універсум (область значень лінгвістичної змінної технічного стану ЗВТ);

$G$  – синтаксична процедура, яка описує процес генерування з множини  $T$  нових значень лінгвістичної змінної  $\beta$ ;

$M$  – семантична процедура, яка визначає відповідність кожному новому значенню лінгвістичної змінної  $\beta$  (яке отримане за допомогою  $G$ ) елементам деякої нечіткої множини.

Структурне-логічна схема постановки наукового завдання представлена на рис.1.5.



Рис. 1.5. Структурне-логічна схема постановки наукового завдання

Таким чином, надана формальна постановка актуального нового наукового завдання *удосконалення існуючих та розробки нових моделей та*

*методів прогнозування технічного стану засобів водного транспорту на основі м'яких обчислень, що безпосередньо впливає на підвищення достовірності прогнозу ТС за рахунок автоматизації та комп'ютеризації. Доцільно підкреслити, що данні моделі та методи реалізуються саме в математичному, алгоритмічному та програмному забезпеченні бортової комп'ютерної системи, наприклад, в так званій інтегрованої мостікової системі (Integrated Bridge System) [5,20].*

### **Висновки до розділу 1**

1. На даний час водний транспорт України у цілому характеризується покращенням рівня безпеки судноводіння, показників якості та ефективності перевезень пасажирів та вантажів. Доведено те, що водний транспорт з метою сприяння швидкому економічному, соціальному та політичному розвитку країни має розвиватися випереджальними темпами. Очікується, що у 2020-2022 роках обсяги перевезення вантажів збільшаться в порівнянні з 2010 роком на 40 %, переробка вантажів у державних портах – на 45 %, обсяги пасажирських перевезень – на 30 %.

2. На підставі результатів аналізу досвіду експлуатації та розвитку транспортної галузі України та провідних країн світу встановлено, що в сучасних умовах технічні фактори, які впливають на аварії водного транспорту складають до 30% від загальної кількості. А це безумовно, підкреслює необхідність забезпечення заданого рівня надійності ЗВТ.

3. На думку фахівців загальна проблема надійності засобів водного транспорту має достатню кількість складових, з яких завдання прогнозування технічного стану ЗВТ є важливим на актуальним. В сучасних умовах існує багато напрямків вирішення цього завдання, але дослідження показали, що на даний час існує необхідність пошуку нових підходів, методів та моделей. Це пов'язано з тим що в більшості ситуацій відсутня статистична або будь-яка апріорна інформація про можливі зовнішні впливи та дестабілізуючи

фактори. Забезпечення високого рівня достовірності прогнозу потребує ефективних підходів, з яких найбільш доцільним є той, що заснований на штучному інтелекті.

4. Аналіз існуючих науково-обґрунтованих підходів підвищення ефективності складних технічних систем, до яких повною мірою відноситься й інтелектуальна транспортна система, дозволив зробити висновок про формування за останні роки нового пріоритетного підходу, пов'язаного із реалізацією принципів теорії штучного інтелекту – м'яких обчислень. Таким чином, загальне наукове завдання щодо розвитку моделей, методів, методик та алгоритмів прогнозу технічного стану ЗВТ на основі м'яких обчислень є важливим та актуальним для науки та практики.

5. Аналіз світового досвіду впровадження інтелектуальних технологій на водному транспорті свідчить о можливості значного підвищення ефективності, якості та достовірності прогнозу технічного стану ЗВТ. Таким чином, незважаючи на стрімкий розвиток теорії штучного інтелекту та м'яких обчислень взагалі, актуальним є наукове завдання якому присвячена дана дисертаційна робота.

## РОЗДІЛ 2. РОЗРОБКА МОДЕЛЕЙ ПРОГНОЗУВАННЯ ТЕХНІЧНОГО СТАНУ ЗАСОБІВ ВОДНОГО ТРАНСПОРТУ

### 2.1. Концептуальна модель прогнозування технічного стану засобів водного транспорту

Для системи, яка вирішує завдання прогнозування технічного стану ЗВТ представлено:

1. Структурну схему (рис. 2.1).
2. Текстовий опис призначення складових системи.
3. Математичну модель прогнозу технічного стану ЗВТ у вигляді навантаженого орграфу. Описані матриці суміжності та ваг ребер.

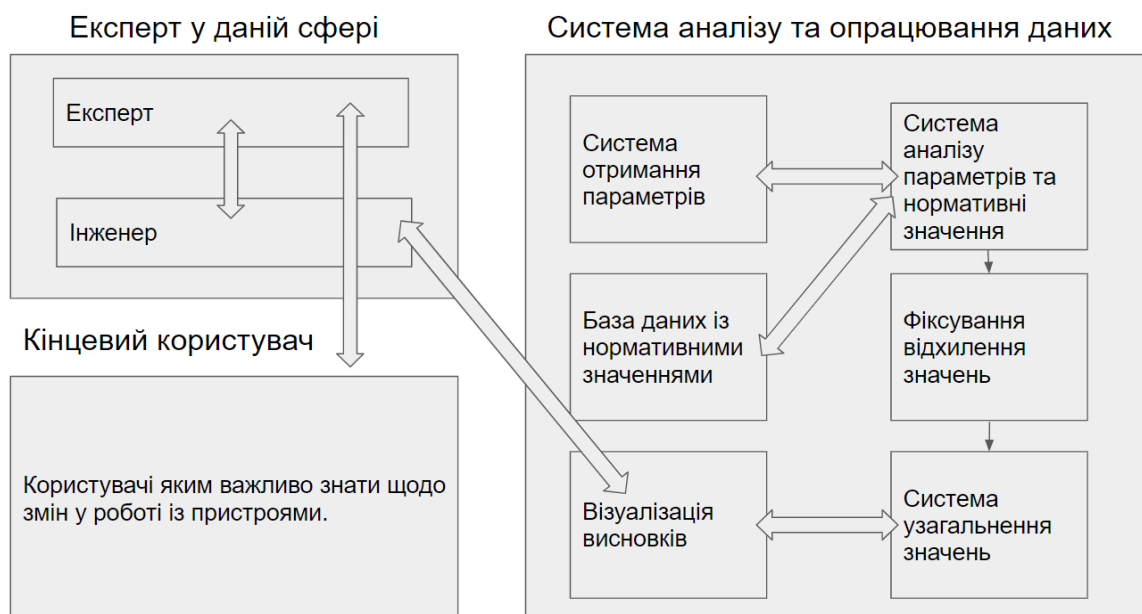


Рис. 2.1. Структурна схема системи опрацювання і аналізу даних

#### Призначення складових системи опрацювання і аналізу даних.

– Система отримання параметрів. Контролюються параметри і ознаки зміни технічного стану обладнання [98].

- База даних із нормативними значеннями. Нормуються значення діагностичних параметрів.
- Система аналізу параметрів та нормативні значення. Порівнюються діагностичні параметри з величинами їх нормативних значень.
- Фіксування відхилень значень. Фіксується момент, абсолютне значення і тривалість виходу діагностичних параметрів обладнання за допустимі межі.
- Система узагальнення значень. Виконуються накопичення, відображення і реєстрація оброблюваної інформації.
- Візуалізація висновків. Візуалізуються поточні та перспективні значення узагальнених ресурсних показників технічного стану контрольованого обладнання.
- Інженер. Тестує систему на правильність виданої інформації.
- Експерт. Виявляє одиниці обладнання, технічний стан яких значно зміниться у наступний період експлуатації;
- Кінцевий користувач. Отримує інформацію про технічний стан і рекомендації про доцільні зміни процесу експлуатації кожної одиниці обладнання.

**Математична модель прогнозу технічного стану ЗВТ у вигляді навантаженого орграфу.**

Природне, що найбільш наглядно процес прогнозу ТС бажано надати у вигляді навантаженого орграфу  $G(F, S)$  [45] (рис. 2.2). Де  $F$  – множина вершин графа чи множина часткових завдань (кроків алгоритму прогнозу), а  $S$  – множина ребер. Практична інтерпретація ребер це час виконання часткового завдання. Напрямок ребра відповідає послідовності дій алгоритму.

Кожна вершина характеризується вектором її ваги  $Z_j$

$$Z_j = (z_{1j}, z_{2j}, \dots, z_{rj}); \quad j = \overline{1, m}. \quad (2.1)$$

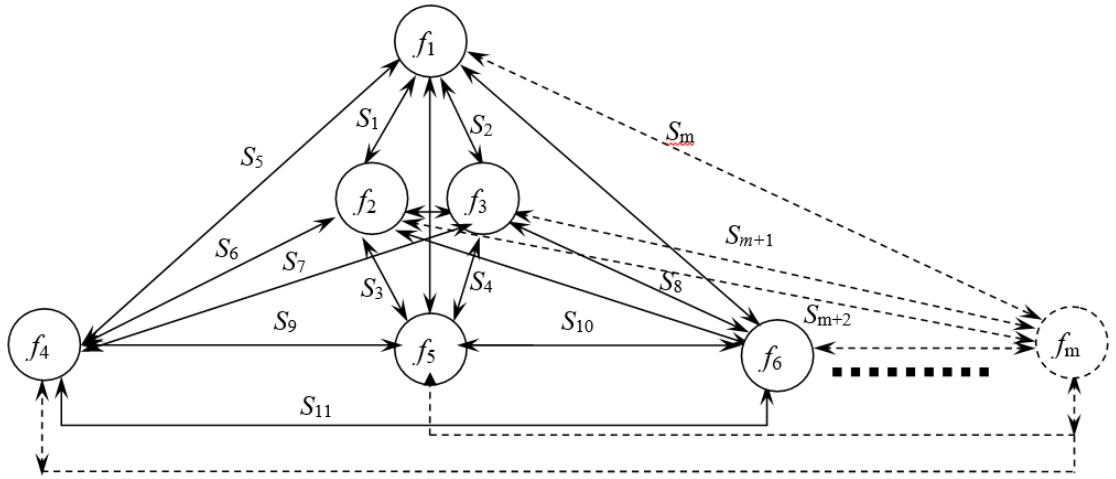


Рис. 2.2. Модель процесу прогнозу ТС

Елементи вектора ваги – це завдання підетапів прогнозу (кроків алгоритму прогнозу):  $z_{ij} = \{0,1\}$ , тобто якщо  $z_{ij} = 0$ , то завдання не вирішується, а якщо  $z_{ij} = 1$ , то відповідне завдання вирішує.

Аналітичний вигляд така модель буде мати як матриця суміжності графа структури (2.2) та матриці ваги вершин і ребер (2.3) та (2.4). Тоді матриця суміжності такого орграфу буде мати такий вигляд:

$$\begin{pmatrix}
 a_{11} & a_{12} & \dots & a_{1n} & a_{1(n+1)} & a_{1(n+2)} & \dots & a_{1m} \\
 a_{21} & a_{22} & \dots & a_{2n} & a_{2(n+1)} & a_{2(n+2)} & \dots & a_{2m} \\
 \vdots & & \dots & & & & \dots & \vdots \\
 a_{n1} & a_{n2} & \dots & a_{nn} & a_{n(n+1)} & a_{n(n+2)} & \dots & a_{nm} \\
 a_{(n+1)1} & a_{(n+1)2} & \dots & a_{(n+1)n} & a_{(n+1)(n+1)} & a_{(n+1)(n+2)} & \dots & a_{(n+1)m} \\
 a_{(n+2)1} & a_{(n+2)2} & \dots & a_{(n+2)n} & a_{(n+2)(n+1)} & a_{(n+2)(n+2)} & \dots & a_{(n+2)m} \\
 \vdots & & \dots & & & & \dots & \vdots \\
 a_{m1} & a_{m2} & \dots & a_{mn} & a_{m(n+1)} & a_{m(n+2)} & \dots & a_{mm}
 \end{pmatrix}, \quad (2.2)$$

де  $a_{ii} = 0; i = \overline{1, m}$ .

Елементи матриці (2.2), що окреслені пунктиром  $\{a_{i(n+2)} \dots a_{mm}\}$  подані як невідомі, які необхідно винайти, на рис. 2.2 це  $\{f_i \dots f_m\}$ .

Складний процес прогнозу ТС ЗВТ описати одним навантаженим оргграфом неможливо тому, що побудова системи має включати не тільки структуру, а й урахування завдань, покладених на етапи, та алгоритми взаємодії етапів у процесі функціонування. Отже, для визначення завдань підетапів запропоновано кожному вершину  $f_j$ ,  $j = \overline{1, m}$  орграфу  $G$  характеризувати вектором її ваги  $Z_j$ . Вагу кожного елемента орієнтованого графа  $G$  відповідно до всієї системи з аспекту сукупності завдань, покладених на неї, подано матрицею  $Z_j$ . Як вершини орграфу  $G$ , так і кожне ребро графа  $S_k$  множини  $S$  мають особисту вагу  $B_k$ , що описує алгоритм дій між вершинами графа (підетапами, що взаємодіють). Така взаємодія також описується оргграфом (рис. 2.3) та матрицею суміжності [51] (2.3).

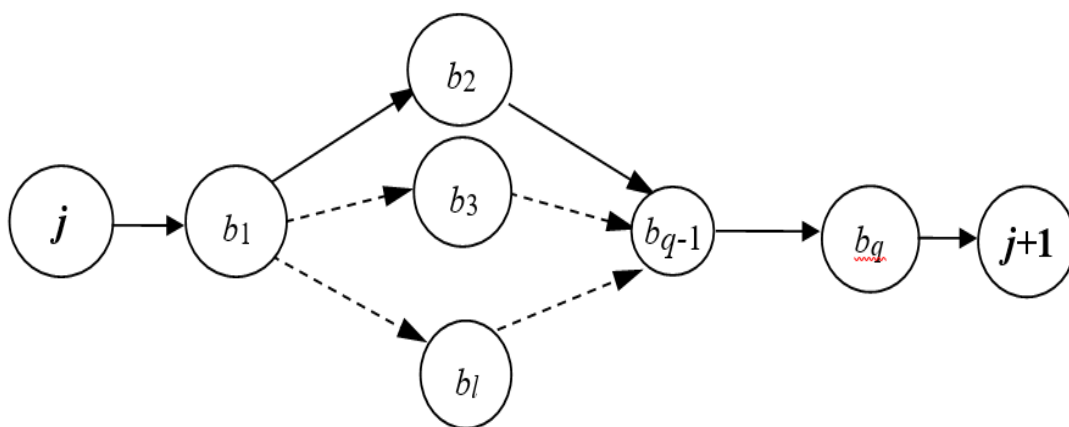


Рис. 2.3. Оргграф  $S_k$  ребра графа  $G(F, S)$  (алгоритму взаємодії підетапів)

Практична інтерпретація вершин такого орграфу (рис. 2.3) – це кроки алгоритму взаємодії, які він виконує під час реалізації своїх завдань, а ребра графа вказують на послідовність зазначених дій, кроків алгоритму у процесі взаємодії між іншими підетапами.

$$B_k = \begin{pmatrix} 0 & b_{12k} & \dots & b_{1ck} \\ b_{21k} & 0 & \dots & b_{2ck} \\ \vdots & \vdots & \ddots & \vdots \\ b_{l1k} & b_{l2k} & \dots & 0 \end{pmatrix}, \quad \text{де } b_{ijk} = \{0,1\}. \quad (2.3)$$

Щоб деталізувати математичну модель прогнозу технічного стану ЗВТ скористаємося **методом розпізнавання образів** [17].

**Метод розпізнавання образів (статистичної класифікації).** Цей метод полягає в тому, що прогнозування можна починати з моменту здійснення одноразового контролю діагностичного обладнання. В результаті прогнозу контрольований об'єкт відносять до того чи іншого класу технічного стану, який встановлюють заздалегідь за критерієм працездатності або довговічності і приймають за еталон (образ). Потім, виходячи з закономірності зміни параметрів даного класу, вирішують, як буде змінюватися даний параметр в майбутньому [24].

При прогнозуванні технічного стану обладнання виконуються та вирішуються наступні завдання:

1. Контролюються параметри і ознаки зміни технічного стану обладнання.
2. Нормуються значення діагностичних параметрів.
3. Порівнюються діагностичні параметри з величинами їх нормативних значень.
4. Фіксується момент, абсолютне значення і тривалість виходу діагностичних параметрів обладнання за допустимі межі.
5. Виконуються накопичення, відображення і реєстрація оброблюваної інформації.
6. Обчислюються поточні та перспективні значення узагальнених ресурсних показників технічного стану контрольованого обладнання;
7. Виявляються одиниці обладнання, технічний стан яких значно зміниться у наступний період експлуатації.

8. Видається інформація персоналу про технічний стан і рекомендації про доцільні зміни процесу експлуатації кожної одиниці обладнання.
9. З урахуванням цих завдань на рисунку 2.4 пропонується модель у вигляді навантаженого орграфу (вага ребра залежить від часу виконання завдань).

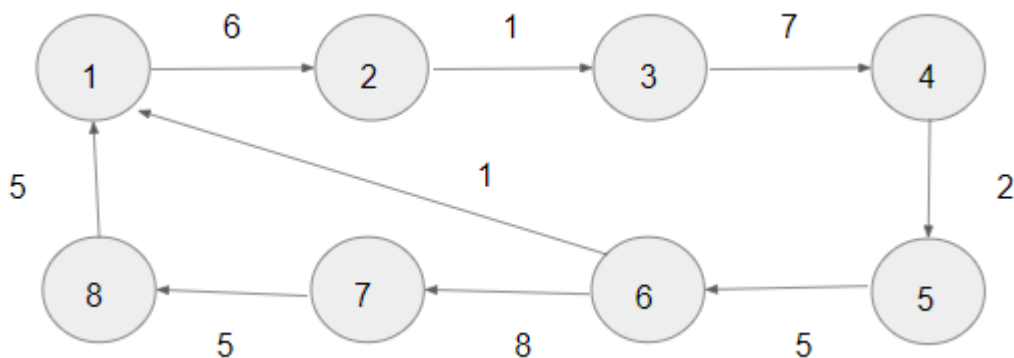


Рис. 2.4. Модель прогнозу технічного стану ЗВТ у вигляді навантаженого орграфу

**Модель даних** в концептуальній моделі прогнозу ТС ЗВТ займає особливу роль [96]. Це пов'язане, безумовно с тим, що процедура будь-якого прогнозування в першу чергу ї є цілеспрямована обробка даних про минулії та теперешній стан з метою визначення закономірностей, які дадуть достовірний прогноз стану в майбутньому. В ході експлуатації і обслуговування комплектуючих ЗВТ накопичуються великі обсяги даних різної природи, тому такого роду системи, що базуються на їх аналізі, припускають розробку моделі цих даних.

Модель даних [19,91] призначена для опису структури інформаційних об'єктів, їх функціональних взаємозв'язків, необхідних для виконання заданих дій. В літературі зустрічаються три основні (за типом) моделі даних: ієрархічні, мережеві і реляційні, а також ряд додаткових моделей – постреляційні, об'єктно-орієнтовані, багатовимірні, комбіновані та ін. Характерна відмінність моделей між собою – метод визначення зв'язків між даними.

Ієрархічна модель даних має деревоподібну структуру, в якій вузли - елементи даних - з'єднані створені за допомогою вертикальних зв'язків. В ієрархічній моделі дані жорстко підпорядковані один одному. Використання ієрархічної моделі для даних зі складними логічними зв'язками ускладнює доступ нижнім рівнями ієрархії, а також робить її громіздкою для обробки інформації. У мережевих моделях між вузлами існують не тільки вертикальні, але і горизонтальні зв'язки. Мережева модель є розширеною ієрархічною моделлю, тому як і ієрархічна модель має ті ж недоліки, а саме, складність і жорсткість моделі.

У реляційних моделях сукупність даних реалізується у вигляді наборів двовимірних таблиць. Дана модель є простою і досить ефективною, але і вона має низку деяких недоліків: строгість структур робить реляційну модель негнучкою, а сильна залежність моделі від складу і структури даних в разі їх зміни змушує змінювати структуру самої моделі.

Концепція реляційної моделі, що розроблена Е.Ф. Кодом у 1970 р., полягає у наступному [38]:

- дані в моделі слід зв'язувати відповідно до їх внутрішніх логічних взаємовідносин, а не до фізичних показників. Завдяки цьому користувачі зможуть комбінувати дані з різних джерел;
- дані подаються у вигляді множини таблиць, зв'язаних між собою згідно певних правил;
- робота з даними здійснюється на основі операції реляційної алгебри, які дозволили створити ряд мов запитів, більш доступних для користувачів, які не є спеціалістами з комп'ютерних технологій. Такими мовами є SQL (Structured Query Language – мова структурованих запитів), Quel (Query Language – мова запитів) і QBE (Query-by-Example – запити за зразком).

Постреляційна модель є розширеною версією реляційної і дає можливість усунути обмеження неподільності даних в таблицях. Формування сукупності пов'язаних реляційних таблиць здійснюється через одну постреляційну таблицю, що дозволяє зробити інформацію наочною і

підвищити ефективність її обробки, при цьому виникають проблеми цілісності і несуперечності збережених даних.

Об'єктно-орієнтовані моделі дають можливість ідентифікувати окремі об'єкти бази, але ці моделі мають багато суттєвих недоліків: відсутність суворої математичної моделі та більш складні механізми пошуку і взаємодії.

Аналіз показав, що саме багатовимірні моделі використовуються при аналізі великих багатовимірних масивів даних (гіперкубів) з метою підтримки прийняття рішення. На відміну від реляційної багатовимірної моделі є більш наочною та інформативною. Тому для вирішення поставленого завдання - побудови моделі даних для системи прогнозу ТС ЗВТ оптимальною є багатовимірна модель даних, так як вона дозволяє аналізувати великі масиви даних про стан комплектуючих ЗВТ для підтримки прийняття рішень про заходи стосовно забезпечення заданого рівня надійності.

У дисертаційній роботі багатовимірна модель даних представлена у вигляді інтелектуального аналізу кубів OLAP (On-line Analytic Processing) – технології, що має на увазі аналіз багатовимірних даних для оперативної обробки інформації, включаючи динамічну побудову звітів в різних розрізах, аналіз даних, моніторинг та прогнозування ключових показників. З іншого боку, OLAP-куби можна розглядати як розширення двовимірних масивів електронних таблиць [92].

Комп'ютерна реалізація розроблених моделей та методів прогнозу за допомогою OLAP-технологій дозволить інтегрувати дані різних об'єктів (від елементарних до комплексних), проектувати видачу діагностичної інформації без участі як програмістів, так і операторів, аналізувати дані по будь-яким категоріям і показниками в реальному часі на будь-якому рівні деталізації, робити моніторинг і прогнозування ключових показників працездатності об'єктів довільного рівня деталізації та ієрархії.

Варте підкреслити, що вибір виду моделі даних у вигляді OLAP-кубів обумовлений можливістю формування прогнозу ТС в різних розрізах і з

довільною глибиною деталізації. В якості методу інтелектуального аналізу обраний метод нейро-нечіткого логічного висновку.

На завершення викладу матеріалу стосовно розробленої в даній роботі концептуальної моделі прогнозування ТС ЗВТ, яка в дисертації заявлена науковою новизною, надамо структури методики її алгоритмічної реалізації (рис. 2.5) та схему технології нечіткого висновку (рис.2.6).



Рис. 2.5. Схема методики алгоритмічної реалізації моделі прогнозу

Таким чином, концептуальна модель прогнозування технічного стану засобів водного транспорту [97], в алгоритмічному сенсі, поєднує OLAP-технологію інтелектуального аналізу даних на м'яких обчисленнях. Пропонується застосовувати метод нечіткої субтрактивної кластеризації для визначення кількості значень лінгвістичних змінних (рангу терм-множини) та методу нечітких с-середніх (Fuzzy C-Means) для побудови функцій приналежностей, а також гібридної мережі – так званої адаптивної нейронної нечіткої мережі ANFIS. Аналіз наукових робіт в даній галузі

[3,22,23,37,41,47,48,53,93,82] дозволяє зробити висновок про високу ефективність саме такого комплексного застосування добре апробованих підходів.

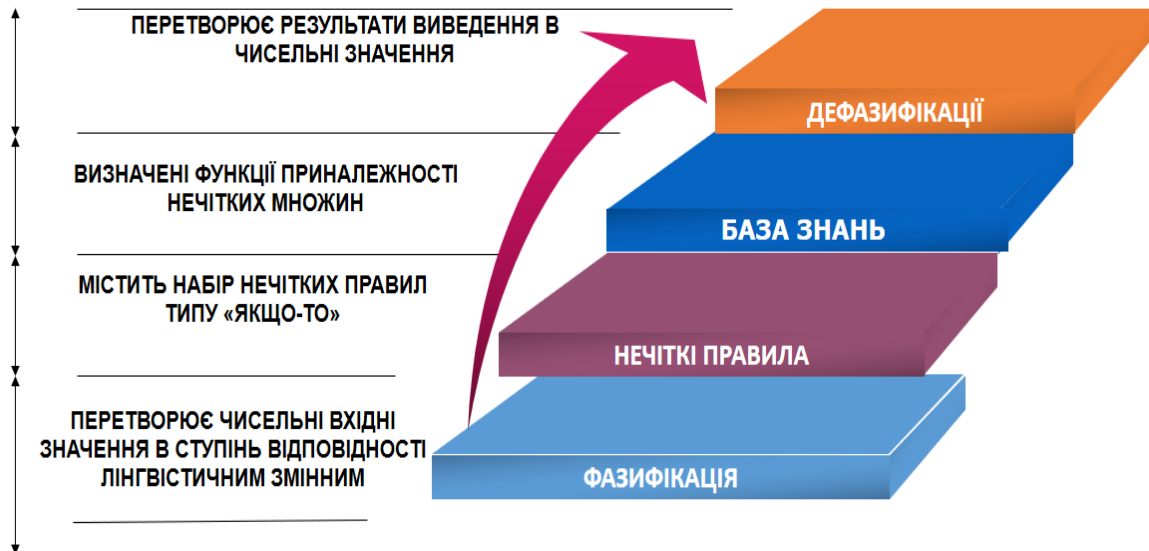


Рис.2.6. Схема технології нечіткого висновку

В роботі використана наступна термінологія [53,65].

Фазифікації – процес перетворення вхідних даних в нечіткі змінні. Забезпечує за допомогою функцій приналежності визначення відповідності вхідних даних термам лінгвістичних змінних.

Дефазифікація – процес переходу від функції приналежності вихідної лінгвістичної змінної до її чіткому числовому значенню.

Агрегування – визначення ступеня істинності умов для кожного правила в системі нейро-нечіткого висновку – мінімізація значень всіх підумов.

Акумуляція – це процес визначення функції приналежності для кожної з змінних. Акумуляція виконується з метою об'єднання всіх ступенів істинності підзаключень для визначення функції приналежності кожної з вихідних змінних.

Система нечіткого логічного висновку – це процес отримання нечітких висновків про об'єкт на основі даних про його поточний стан, у вигляді нечітких умов або передумов.

## **2.2. Розробка моделі формування функції приналежності**

Визначення структури моделі прогнозу ТС ЗВТ має за мету аналіз можливих шляхів вирішення даної проблеми, адаптацію існуючих методів або пошук та виявлення нових. Використання стандартних математичних методів на жорсткій алгоритмічній основі для розв'язання задачі прогнозу ТС ЗВТ та прийняття рішень про їх подальшої експлуатації може дати незадовільні результати. Це пов'язано зі складністю аналізованих об'єктів і тим, що при вирішенні поставлених задач доводиться оперувати великим обсягом вхідних даних не тільки числового формату, а й лінгвістичного, для яких характерні невизначеність, неповнота і відсутність можливості формальної структуризації [95].

Для вирішення поставлених завдань необхідні методи, які здатні реалізувати таку функцію людського інтелекту, як вибір оптимального рішення на основі раніше отриманого досвіду і раціонального аналізу всієї доступної інформації про об'єкт дослідження. Існують різні інтелектуальні системи, але найбільш поширеними вважають наступні:

штучні нейронні мережі (ШНМ) – математична модель біологічних нейронних нейронів і мереж людського мозку, що описує принципи їх організації та функціонування [79,89];

генетичні алгоритми (системи евристичного пошуку) – це алгоритми, використовувані для вирішення завдань оптимізації та моделювання, засновані на генетичних процесах, аналогічних для біологічних організмів [80];

експертні системи та системи, які засновані на знаннях – ті які побудовані на правилах, що зберігаються в базі знань, за допомогою яких на

основі сукупності вихідних фактів здійснюється пошук рішень і висновків з цих фактів [9,54,58].

Серед проаналізованих методів найбільший інтерес для вирішення наукового завдання дисертації являють системи, засновані на знаннях, оскільки мова йде про необхідність висновків з вихідних даних та фактів і пошуки рішень для прогнозу ТС ЗВТ.

Системи логічних висновків умовно можна розділити на два основних напрямки – чіткі і нечіткі висновки. Чіткі логічні висновки побудовані на математичній логіки та теорії чітких множин. Нечіткі логічні висновки (нейро-нечіткі логічні висновки) побудовані на теорії нечіткої логіки, і являють собою процеси отримання нечітких висновків за допомогою нечітких умов або передумов про об'єкт дослідження на основі інформації про його поточний стан. Нечітка множина – це математична формалізація нечіткої інформації. Вона представлена сукупністю елементів, що володіють загальною властивістю різного ступеня і належать до даної множини з різним ступенем.

Нейро-нечіткий логічний висновок – один з небагатьох методів, який допускає неточність, невизначеність і неповну істинність оброблюваних даних і реалізується на основі узагальненого використання методів нечіткої логіки і методів штучних нейронних мереж (ШНМ). Застосування методів нечіткої логіки обумовлено прикладним характером завдання прогнозу ТС ЗВТ, коли необхідно оперувати не тільки чисельними даними, але і лінгвістичними про технічний стан. Класифікація станів, їх кількість і уявлення можуть відрізнятися в залежності від призначення. Крім того, використання апарату нечітких множин дозволяє адекватно формалізувати знання експертів, які можуть бути неповними та суб'єктивними.

Використання методів ШНМ обумовлено їх основною перевагою – можливістю навчання (самонавчання), що дозволяє шляхом об'єднання в навчальній вибірці експертних знань і аналітичних залежностей при прогнозу

стану ідентифікувати неоднозначні параметри комплектуючих ЗВТ. Така структура дозволяє скористатися перевагами обох методів.

Застосування методів ШНМ і нечіткої логіки при прогнозі ТС ЗВТ досить нове та прогресивне. Тим часом використання нейро-нечіткого логічного висновку для вирішення даної задачі є досить обґрунтованим і актуальним. В даний час найбільш поширені наступні види нейро-нечіткого логічного висновку: Мамдані, Сугено (Такагі-Сугено), Цукамото, і Ларсена. Алгоритми цих нейро-нечітких логічних висновків розрізняються видом використовуваних правил, методом дефазифікації і формою результату [3,53].

Цілковито логічне те, що для моделювання багатофакторних задач, до яких відноситься і задача прогнозу ТС ЗВТ, доцільно використовувати алгоритм Сугено. Саме цей алгоритм дозволяє об'єднати в моделі експертні знання та аналітичні залежності при формуванні прогнозу. Експертна підсистема забезпечує змістовну інтерпретацію моделі, а аналітичні залежності «входи - вихід» роблять її компактною. Також загальноновизнано, що при великій кількості вхідних параметрів алгоритм Сугено є універсальним апроксиматором функцій.

В науковій літературі по темі нечітких множин методи побудови функцій приналежності поділяють на два види: прямі і непрямі [53]. *Прямі методи* реалізуються на основі експертного завдання виду функцій приналежності. До прямих методів можна віднести метод статистичної обробки експертної інформації. Істотним недоліком такого методу є його орієнтація на переважно суб'єктивну інформацію про процесі або явище і обов'язкове прийняття допущень при перетворенні даної інформації в ступені приналежності нечітких множин.

*Непрямі методи* передбачають вибір функцій приналежності, заздалегідь задовольняють сформульованим умовам. Як правило, непрямі методи використовуються в тих випадках, коли відсутня можливість виміру конкретних значень параметрів властивості, які можуть бути використані для

побудови нечітких моделей розглянутої предметної області. Слід зазначити, що експертна інформація має статус первинної, що передбачає подальшу обробку, в тому числі з урахуванням додаткових умов. Додаткові умови можуть накладатися як на вид одержуваної інформації, так і на процедури її обробки.

Найбільш поширеним серед непрямих є метод парних порівнянь, у результаті застосування якого формується матриця парних порівнянь  $A$  [81]. Основним недоліком даного методу є складність виконання порівнянь у разі оцінювання великої кількості варіантів.

На практиці буває простіше спочатку побудувати матрицю  $A$  в припущенні, що її діагональні елементи повинні бути рівні 1, а симетричні відносно головної діагоналі елементи мають бути взаємно зворотними, тобто  $a_{ij} = \frac{1}{a_{ji}}$ . Остання умова означає, якщо ступінь приналежності одного з

елементів оцінюється в  $a$  разів сильніше ступеня приналежності іншого, то ступінь приналежності другого елемента повинна бути в  $1/a$  разів сильніше ступеня приналежності першого елемента.

В цьому випадку задача побудови функції приналежності зводиться до знаходження такого вектора  $w$ , яке є рішенням наступного рівняння

$$Aw = \lambda_{\max} w, \quad (2.5)$$

де  $\lambda_{\max}$  — найбільше власне значення матриці  $A$ . Оскільки всі значення елементів матриці  $A$  позитивні, рішення даного рівняння існує і є позитивним.

Власне процес попарного порівняння елементів може бути заснований на суб'єктивній інтуїції або на виконання деякої послідовності алгоритмічних або логічних дій. При цьому окремі елементи універсуму можуть використовуватися в якості еталонів або всі елементи можуть бути розділені на групи з наступним порівнянням цих груп між собою.

Крім методу парних порівнянь у ряді випадків представляє інтерес метод ідентифікації функцій належності на базі визначення за експериментальними даними нечітких моделей типу «входи - вихід». Такий метод дозволяє вирішити задачу формалізації вихідної інформації, хоча пред'являє підвищені вимоги до навчальної вибірки і моделі типу «вхід-вихід» [53,101].

В окрему групу можна виділити методи визначення функцій належності з допомогою *кластеризації*. У загальному випадку кластеризація це об'єднання об'єктів в групи (кластери) на основі однотипності будь-яких ознак об'єктів для однієї групи і відмінностей даних ознак між групами. Більшість існуючих алгоритмів кластеризації не вимагає прийняття припущень, властивих, наприклад, статистичним методам.

Задача кластеризації полягає у визначенні природного розбиття даних на класи незалежно від суб'єктивного судження експерта. Найважливішою особливістю методів кластеризації є їх використання в умовах відсутності якої-небудь інформації про закони розподілу даних [3]. Слід зазначити, що кластеризація може бути виконана для об'єктів як за якісними, так і кількісними або навіть змішаними показниками.

Методи кластеризації діляться на чіткі та нечіткі. Завдання чітких методів кластеризації полягає в розбитті множини  $X$  вихідних об'єктів на декілька непересічних підмножин. Слід зазначити, що кожен об'єкт множини  $X$  належить тільки одному кластеру. У свою чергу нечіткі методи кластеризації допускають належність одного і того ж об'єкта одночасно до кількох кластерам з відповідними ступенями приналежності на інтервалі  $[0,1]$ . В деяких випадках використання методів нечіткої кластеризації більш обґрунтовано, ніж чіткої, особливо при аналізі об'єктів, знаходяться поблизу межі кластерів. Наприклад, в задачі прогнозу технічного стану ЗВТ нечітка кластеризація допускає, що стан будь-якого діагностичної об'єкта дослідження одночасно належить всім кластерам (ТС ЗВТ), але з різним ступенем приналежності. Якщо діагностуємий об'єкт описується однією

ознакою, то функція приналежності нечіткого кластера повинна відповідати функції приналежності нечіткої множини. За умовами нормальності і опуклості нечіткі множини, які визначаються підлягають апроксимації параметричними функціями приналежності [56].

В рамках дисертаційного дослідження було адаптовані до задачі прогнозу ТС ЗВТ наступні методи визначення функцій належності з допомогою двох найбільш часто використовуваних методів кластеризації – методу потенціалів (**субтрактивної кластеризації**) та методу нечіткої кластеризації (**метод нечітких с-середніх**).

### **Нечітка субтрактивна кластеризація**

В основу методу потенціалів покладені ідеї методу гірської кластеризації (методу пікового групування), який не вимагає апріорного завдання певної кількості кластерів. Даний метод був запропонований Д. Филевым і Р. Ягером в 1993 р. Слід зазначити, що кластеризація по гірському методу не є нечіткою, але найчастіше її використовують в якості інструменту для визначення нечітких правил на основі апріорної інформації [119].

Ідея методу полягає в тому, що спочатку визначають точки, які можуть бути центрами кластерів. Далі для кожної такої точки розраховується значення потенціалу, що показує можливість формування кластера в її околиці. Чим щільніше розташовані об'єкти в околиці потенційного центра кластера, тим вище значення його потенціалу. Після цього ітераційне вибираються центри кластерів серед точок з максимальними потенціалами. Таким чином, ключове поняття методу гірської кластеризації – «потенціал точки», який є числовим значенням, що показують щільність вхідних експериментальних даних в її околиці. Потенціал точки характеризує її близькість до центру кластера. Визначення потенціалу точки  $y_i$ ;  $i = \overline{1, n}$  виконується у відповідності з виразом

$$P_i = \sum_{j=1}^n \exp(-4\alpha^2(y_i - y_j)^2), \quad (2.6)$$

де  $\alpha > 0$  – коефіцієнт, що визначає ступінь компактності кластеру. Важливо те, що при використанні (2.6) експериментальні дані слід нормалізувати на інтервалі  $[0,1]$ . Ступінь приналежності нечіткій множині  $y$  визначаються на базі величин потенціалів відповідно до виразу

$$\mu_y(y_i) = \frac{P_i}{\max P_j} . \quad (2.7)$$

Послідовність дій алгоритму представлена рисунку 2.7.

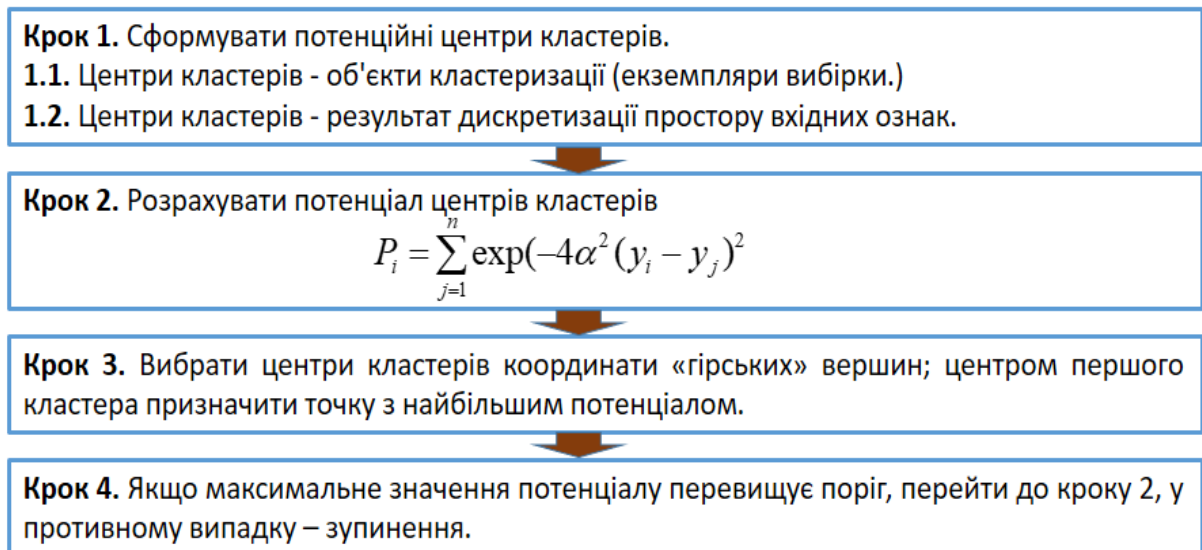


Рис.2.7. Схема субтрактивної кластеризації

Основою програмної реалізації в роботі обране Fuzzy Logic Toolbox програми Matlab. Але звичайне можливе використання або розробка іншого програмного забезпечення, наприклад, Python (інтерпретована об'єктно-орієнтована мова програмування високого рівня). Для знаходження центрів кластерів, кількість яких розгадається в якості рангу відповідної термножени в Matlab програмне реалізована функція `subclust`, яка саме її має призначення виконувати нечітку субтрактивну кластеризацію. Функція `subclust` має наступний синтаксис [53]

[centers, sigmas]=subclust(X, radii, xBounds, ..., options).

Функція subclust визначає центри кластерів даних за алгоритмом субтрактивної кластеризації, який описане раніше та формалізовано в виразах (2.6,2.7). Кількість кластерів визначається під час ітераційного обчислення алгоритму. Функція subclust має до чотирьох вхідних аргументів. Перші два з них – обов’язкові:

X – данні для кластерного аналізу). Кожний рядок матриці X задає один об’єкт кластеризації.

radii – вектор розміру кластерів по кожній координаті. Координати вектора radii мають бути в діапазоні [0, 1] у зв’язку із тим, що при кластеризації данні X масштабуються на одиничний гіперкуб. Інструкція рекомендує надавати значення radii в діапазоні [0.2, 0.5]. Для рівнозначних координат radii задається скаляром.

xBounds – матриця діапазонів зміни вхідних даних для участі в масштабованні матриці X на одиничному гіперкубі. Кожний її рядок задає з однієї координати зміну даних. У випадку коли цей аргумент не задано, то відповідні діапазони розраховують за фактичними значеннями матриці X.

options – вектор параметрів кластерного аналізу.

Синтаксис:

opt.SquashFactor = options(1);

opt.AcceptRatio = options(2);

opt.RejectRatio = options(3).

options(1) – коефіцієнт подавлення. Чим більше його значення, тим більше сусідніх елементів буде належати даному кластерові. За умовчанням дорівнює 1,25;

options(2) – коефіцієнт прийняття. Чим більше його значення, тим більше кластерів буде визначено. За умовчанням дорівнює 0,5;

options(3) – коефіцієнт відторгнення. Чим менше його значення, тим більше віддалених один від одного кластерів може бути знайдено. Значення

цього коефіцієнту повинно бути менше значення коефіцієнта прийняття. За умовчанням дорівнює 0,15;

options(4) – управління виводом на екран поточних результатів кластерного аналізу. Якщо встановлено нульове значення, то вивід поточних результатів подавлено.

Функція subclust може мати до двох вихідних аргументів: centers – матриця центрів кластерів, що знайдені (кожний рядок матриці є координати центру одного кластеру); sigmas – вектор радіусів кластерів.

За результатами кластеризації синтез нечітких правил виконується наступним чином. Якщо об'єкти кластеризації мають дві ознаки, тоді результати нечіткого розбиття зручне представити тривимірною поверхнею. Побудова такої поверхні вимагає для кожного об'єкту відкласти по осях абсцис і ординат значення ознак, а по осі аплікату – міру приналежності об'єкта нечіткому кластеру. Число кластерів дорівнює кількості поверхонь.

В даному випадку є можливість синтезувати нечіткі правила різних баз знань: сингтонної, Мамдані, Сугено. Функції приналежності термів в посилках правила отримують проектуванням ступеня приналежності відповідного кластера на осі вхідних змінних. Наступним кроком отримані множини ступенів приналежності апроксимують відповідними параметричними функціями приналежності. Як висновок правила сингтонної бази знань вибирають координату центру кластера. Висновки правил бази знань Мамдані знаходять так же як і функції приналежності термів вхідних змінних. Висновки правил бази знань Сугено знаходять по методу найменших квадратів. При кластеризації з використанням норми Махалобіса як висновки правил типу Сугено можуть бути вибрані рівняння довгих осей гіпереліпсоїдів.

Вони інтерпретуються функціями приналежності термів, за наступними нечіткими правилами:

ЯКЩО  $x$  = «дуже низький», ТО  $y$  = «дуже низький»

ЯКЩО  $x$  = «зовсім високий», ТО  $y$  = «зовсім високий».

При застосуванні Fuzzy Logic Toolbox програми Matlab для синтезу системи нечіткого висновку – Fuzzy Inference System (FIS) з даних, що отримані в наслідок субтрактивної кластеризації, призначена функція `genfis2`. Функція генерує із даних систему нечіткого висновку типу Сугено з використанням субтрактивної кластеризації. Варте підкреслити, що у зв'язку з тим, що в даній постановці при моделюванні маємо данні з однією вихідною змінною, тому результат виконання функції доцільне розглядати як нечітку систему ANFIS. Екстракція нечітких правил здійснюється в два етапи. Спочатку функція `subclust` знаходить посилки нечітких правил. Потім, за методом найменших квадратів обчислюються висновки правил. В наслідок чого отримуємо систему нечіткого висновку з базою правил, що охоплює завдання прогнозу ТС.

### **Метод нечітких с-середніх (FCM, Fuzzy C-Means)**

Метод нечітких с-середніх (в деякій літературі k-середніх) для розв'язування задачі нечіткої кластеризації був розроблене Дж.К.Данном у 1974 р. Метод попередньо мав назву `fuzzyISODATA`. Але 1980 році Дж.К. Беджек виконав значне теоретичне удосконалення цього методу та дав йому назву нечітких середніх FCM, Fuzzy-C-Means. Програмне метод Fuzzy-C-Means гарне представлений в MATLAB [53].

FCM має зміст послідовного (ітераційного) покращення деякого початкового нечіткого розбиття, яке задається дослідником або розраховується за деяким евристичним правилом автоматично. На кожній ітерації визначають значення функцій приналежності кластерів. Алгоритм методу FCM завершує роботу при умові, коли відбудеться наперед задане число ітерацій, або, коли мінімальна абсолютна різниця між значеннями функцій приналежності на двох послідовних ітераціях не стане менше деякого наперед заданого значення. Метод добре описаний в науковій літературі, тому представимо його лише структурну схему у вигляді послідовності кроків (рис. 2.8).

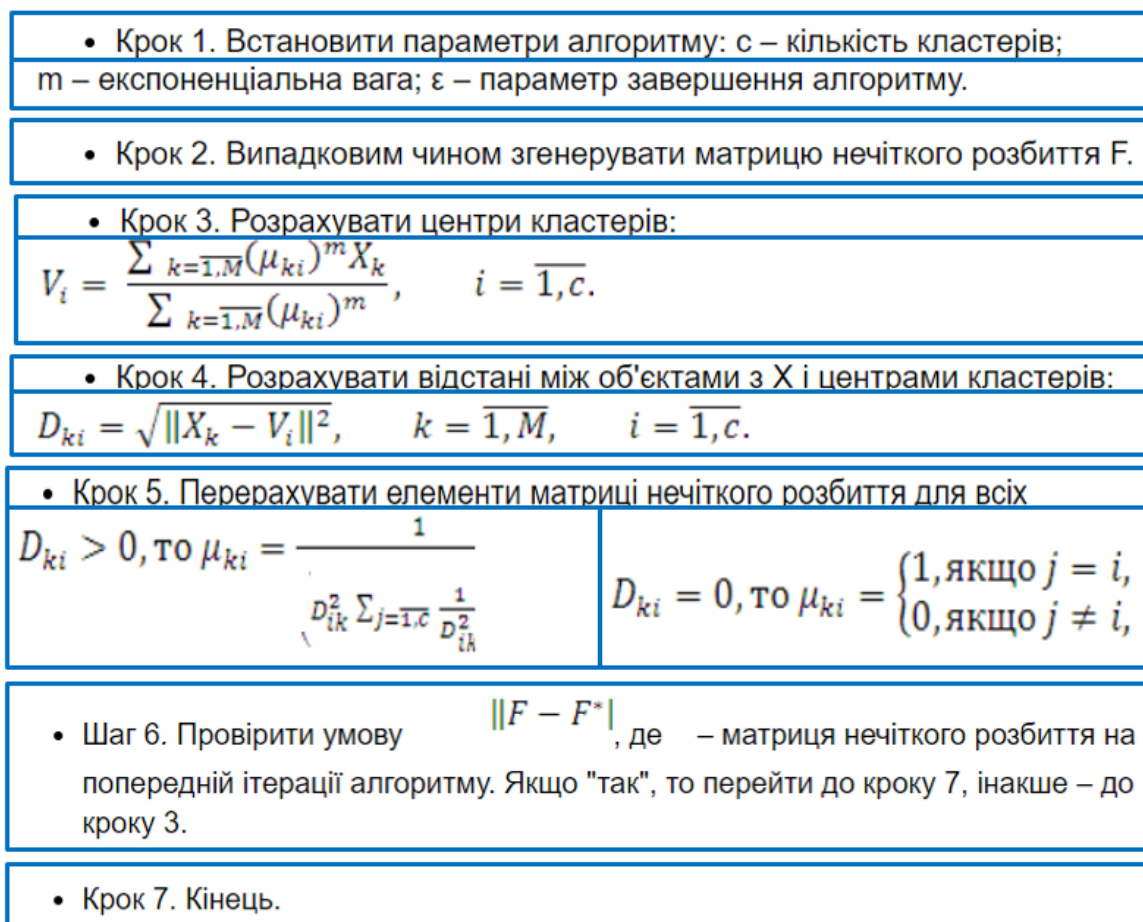


Рис. 2.8. Схема методу нечітких середніх FCM (Fuzzy-C-Means)

Алгоритм методу FCM виконує наступні кроки [3,53].

Крок 1 – задають: кількість нечітких кластерів; максимальну кількість ітерацій; параметр збіжності методу; експоненційну вагу цільової функції і центрів кластерів.

Крок 2 – на першій ітерації для матриці даних в якості початкового розбиття задають деяке нечітку множину для непустих нечітких кластерів, яка описується сукупністю функцій приналежності.

Крок 3 – для поточного нечіткого розбиття розраховуються центри нечітких кластерів і значення цільової функції, яка показує відстань між точною та центром кластеру.

Крок 4 – формують нове нечітке розбиття множини об’єктів кластеризації, для якого розраховуються центри нечітких кластерів і значення цільової функції.

Кроки 5, 6 – у випадку коли кількість виконаних ітерацій більше заданої або модуль різниці між попереднім і новим значенням цільової функції менше заданого значення, тоді в якості результату прийняти попереднє нечітке розбиття і завершити виконання алгоритму. Інакше, вважати поточним розбиттям те яке є і перейти на крок 3, збільшивши на 1 кількість виконаних ітерацій.

В результаті виконання алгоритм метод FCM надає відповідь про деяке локально-оптимальне розбиття, яке описується множиною функцій приналежності та значеннями центрів нечітких класів.

### **2.3. Моделювання прогнозу технічного стану за допомогою розробленого програмного забезпечення**

Для наочності нагадаємо, що в моделях м’яких обчислень використане продукційну систему представлення знань, тобто так звану систему правил продукцій (узгоджену множину окремих продукцій) вигляду:

$$(name): S; P; A \Rightarrow B; C, \quad (2.8)$$

де  $(name)$  – ім’я продукції;  $S$  – сфера застосування продукції;  $P$  – умова застосування ядра продукції;  $A \Rightarrow B$  – ядро продукції, у якому  $A$  – антецедент (умова ядра),  $B$  – консеквент (заключення ядра);  $\Rightarrow$  – знак логічної секвенції (“if A to B”);  $C$  – післяумова продукції.

Ім’я продукції  $(name)$  – сукупністю букв або символів, яка дозволяє єдиним образом визначати продукцію у системі. Найбільш часто продукція задається за допомогою деякого номеру.

Умова застосування ядра продукції  $P$  – це є логічний вираз, наприклад, предикат, за допомогою якого активізується ядро продукції : якщо  $P$  – істина, то ядро активізується, у протилежному випадку – не активізується. У багатьох випадках  $P$  відсутня (інколи об'єднується з ядром продукції).

Ядро продукції  $A \Rightarrow B$  – центральний компонент продукції та має вигляд речення “if  $A$  to  $B$ ”, де  $A$ ,  $B$  – логічні вирази. Знак логічної секвенції має зміст впливання  $B$  із істинного  $A$ . Якщо  $A$  не істинно, то про істинність  $B$  не потрібно робити ніяких висновків. У базах знань інтелектуальних систем умова ядра  $A$  виступає також як деякий зразок – логічний вираз, за яким здійснюється пошук у базі знань. Заключення ядра  $B$  – це дія або процедура, яку потрібно виконати при успішному завершенні пошуку.

Післяумова продукції  $C$  описує процедуру, яку необхідно виконати у випадку реалізації ядра продукції, тобто при умові  $B$  – істино. В нечітких продукційних системах представлення знань, які використані в дисертації кожне з правил продукцій додатково має параметризовану кількісну оцінку ступеня істинності правила, яка формально знаходиться в  $C$ . Приклади побудови нечітких продукційних систем розроблене та представлене далі у наступних розділах рукопису за допомогою пакету програм Fuzzy Logic Toolbox середовища MATLAB.

### *Перший модельний приклад*

Виконаємо оцінку технічно стану обшивки судна на основі вимог [36] та системи знань нечіткого висновку, яка основана на використанні алгоритму Мамдані. Маємо наступні кроки:

**Крок 1.** Формування бази правил. Кожне правило має вигляді

if <умова> to <заключення> .

Для алгоритму Мамдані

<умова> та <заключення>,

виглядають як логічні зв'язки наступних записів

<нечітка змінна> = < значення >.

**Крок 2.** Введення правил в алгоритм висновку (систему висновку).

**Крок 3.** Використання алгоритму висновку для обробки вхідної інформації (конкретних значень вхідних нечітких змінних). Цей крок, в свою чергу, розкладається на наступні кроки:

1. Введення значень вхідних змінних – фактів, які є на 100% істинними.
2. Фазифікація вхідних змінних – встановлення відповідності між конкретним значенням вхідних змінних і значенням її терму, разом з функцією належності.
3. Агрегування складних умов, які стоять в правилах після if, тобто визначення ступеню істинності всіх умов в усіх правилах, якщо умови надаються за допомогою складних логічних виразів. Правило активується, якщо істинність його умови більша за нуль. Агрегування виконується за допомогою нечітких логічних операцій – нечіткої диз'юнкції, нечіткої кон'юнкції та ін.
4. Активація підзаключень – процес визначення ступеню істинності змінних, тобто належності до відповідних термів, які стоять в закінченнях активних правил, за формулою:  $s_k = b_k \Theta_k$ , де  $s_k$  – ступень істинності закінчення правила  $k$ ,  $b_k$  – ступень істинності його умови,  $\Theta_k$  – ступень істинності самого правила (ваговий коефіцієнт  $k$ -того правила). Після визначення вектору  $S = (s_1, \dots, s_q)$  визначаються функції приналежності для кожного із підзаключень кожної вихідної лінгвістичної змінної. Якщо відповідний терм вихідної лінгвістичної змінної визначається функцією приналежності  $\mu(y)$ , тоді після процедури активації отримуємо поновлену функцію

приналежності відповідного терму (підзаключення)  $\mu'(y)$  за одним із методів нечіткої композиції [3]:

1) prod-активізація

$$\mu'(y) = s_i \mu(y);$$

2) min – активізація

$$\mu'(y) = \min\{s_i, \mu(y)\};$$

3) average-активізація

$$\mu'(y) = 0,5(s_i + \mu(y)).$$

Відзначимо, що різні правила підзаключень можуть містити однакові терми лінгвістичних змінних. У цьому випадку для кожного терму визначається множина функцій приналежності, які обчислюються за одним із правил нечіткої композиції по кожному правилу продукції. Остаточна функція приналежності для цього терму визначається у наступному кроці.

5. Акумуляція заключень – визначення значення функцій приналежності для термів всіх вихідних змінних. Якщо для одного терму визначена множина функцій приналежності  $\mu'_1(y), \dots, \mu'_p$ , то акумуляція виконується за одним із правил об'єднання нечітких множин [3]:

1) об'єднання

$$\mu'(y) = \max\{\mu'_1(y), \mu'_2\};$$

2) алгебраїчне об'єднання

$$\mu'(y) = \mu'_1(y) + \mu'_2(y) - \mu'_1(y)\mu'_2(y);$$

3) граничне об'єднання

$$\mu'(y) = \max\{\mu_1'(y) + \mu_2'(y) - 1, 0\};$$

4) драстичне об'єднання

$$\mu'(y) = \begin{cases} \mu_1'(y), & \text{if } \mu_2'(y) = 0, \\ \mu_2'(y), & \text{if } \mu_1'(y) = 0, \\ 1, & \text{else,} \end{cases}$$

5) операція  $\lambda$  - суми

$$\mu'(y) = \lambda\mu_1'(y) + (1 - \lambda)\mu_2'(y), \quad \lambda \in [0,1].$$

6. Дефазифікація – визначення конкретних значень вихідних змінних за функціями належності термів. Виконується методом центру ваги для неперервних та дискретних нечітких множин за наступними виразами:

$$z = \frac{\sum_{i=1}^n y_i \mu'(y_i)}{\sum_{i=1}^n \mu'(y_i)}, \quad z = \frac{\int_{y_{\min}}^{y_{\max}} y \mu'(y) dy}{\int_{y_{\min}}^{y_{\max}} \mu'(y) dy}. \quad (2.9)$$

Розглянемо принципи побудови та роботи системи нечіткого виводу на прикладі задачі оцінки технічного стану обшивки судна. В [36] визначено те, що

мінімальна допустима товщина сталеві обшивки корпусу суден:

- бортові листи - 1,5 мм,
- скулові та днищеві - 2 мм.

Тоді  $T$  – множина термів лінгвістичної змінної  $\beta$ ;  $X$  – універсум (область значень лінгвістичної змінної).

$\beta_1$  – товщина бортового листа;

$T_1 = \{\text{«норма»}, \text{«нище норми»}\};$

$X_1 = [0, 4];$

$\beta_2$  – товщина скулового та днищевого листа;

$T_2 = \{\text{«норма»}, \text{«нище норми»}\};$

$X_2 = [0, 6];$

$\beta_3$  – технічний стан обшивки судна;

$T_3 = \{\text{«придатне»}, \text{«обмежено придатне»}, \text{«заборонене до експлуатації»}\}.$

$X_3 = [0, 100];$

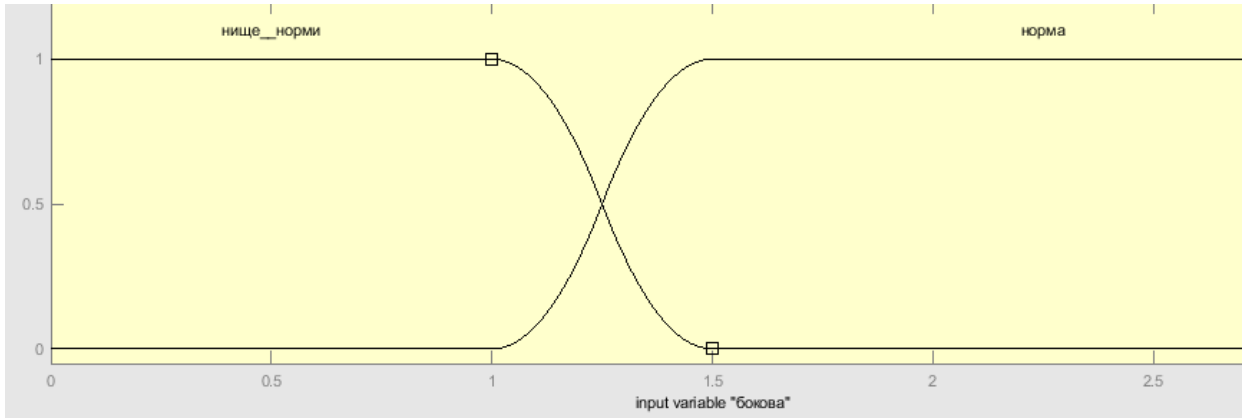


Рис. 2.9. Функції приналежності для  $\beta_1$  (товщина бортового листа)

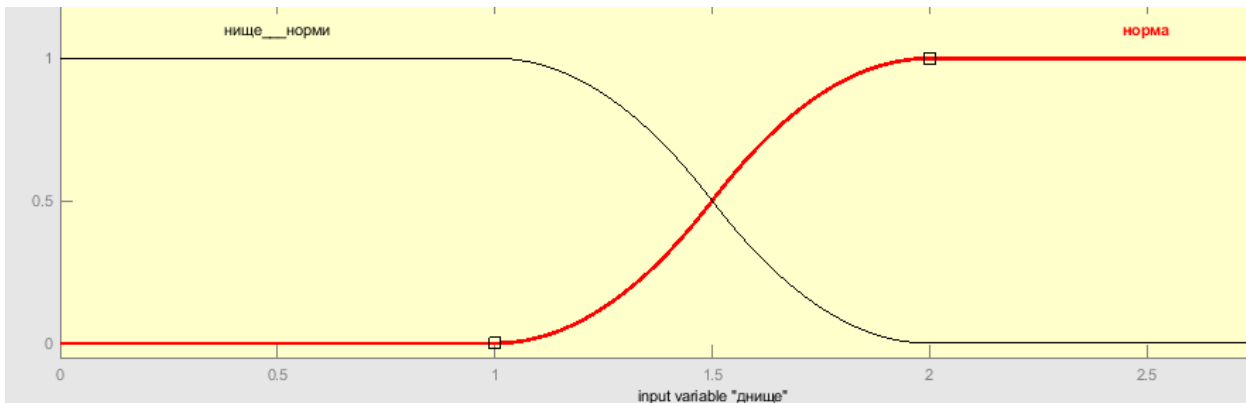


Рис. 2.10. Функції приналежності для  $\beta_2$  (товщина скулового/днищевого листа)

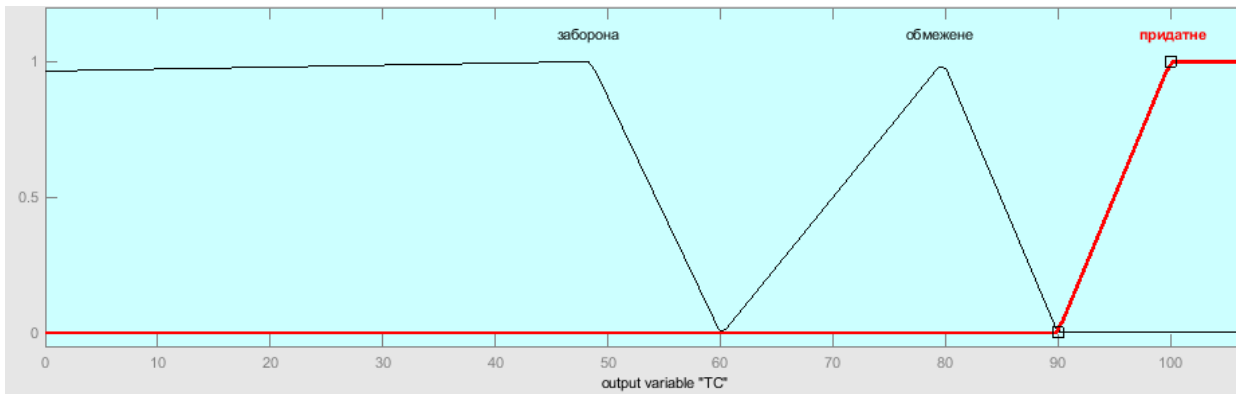


Рис. 2.11. Функції приналежності для  $\beta_3$

1. If (бокова is норма) and (днище is норма) then (ТС is придатне) (1)
2. If (бокова is норма) and (днище is нище\_\_норми) then (ТС is обмежене) (1)
3. If (бокова is норма) and (днище is норма) then (ТС is придатне) (1)
4. If (бокова is нище\_\_норми) or (днище is нище\_\_норми) then (ТС is заборона) (1)
5. If (бокова is нище\_\_норми) and (днище is нище\_\_норми) then (ТС is заборона) (1)

Рис. 2.12. Привила нечіткого висновку

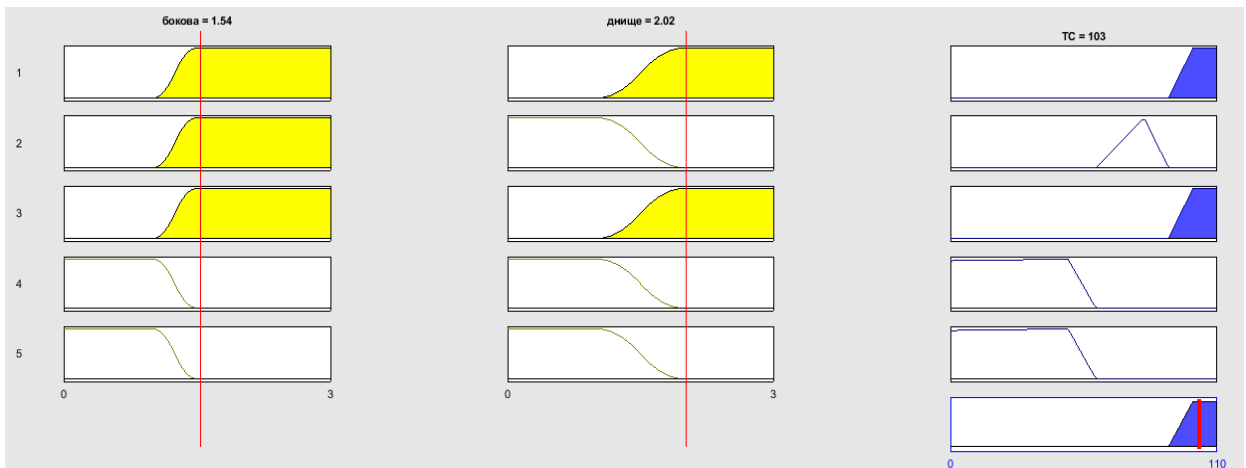


Рис. 2.13. Приклад моделювання правил нечіткого висновку: при  $X_1=1,54$  мм,  $X_2=2,02$  мм обшивка придатна на 100%

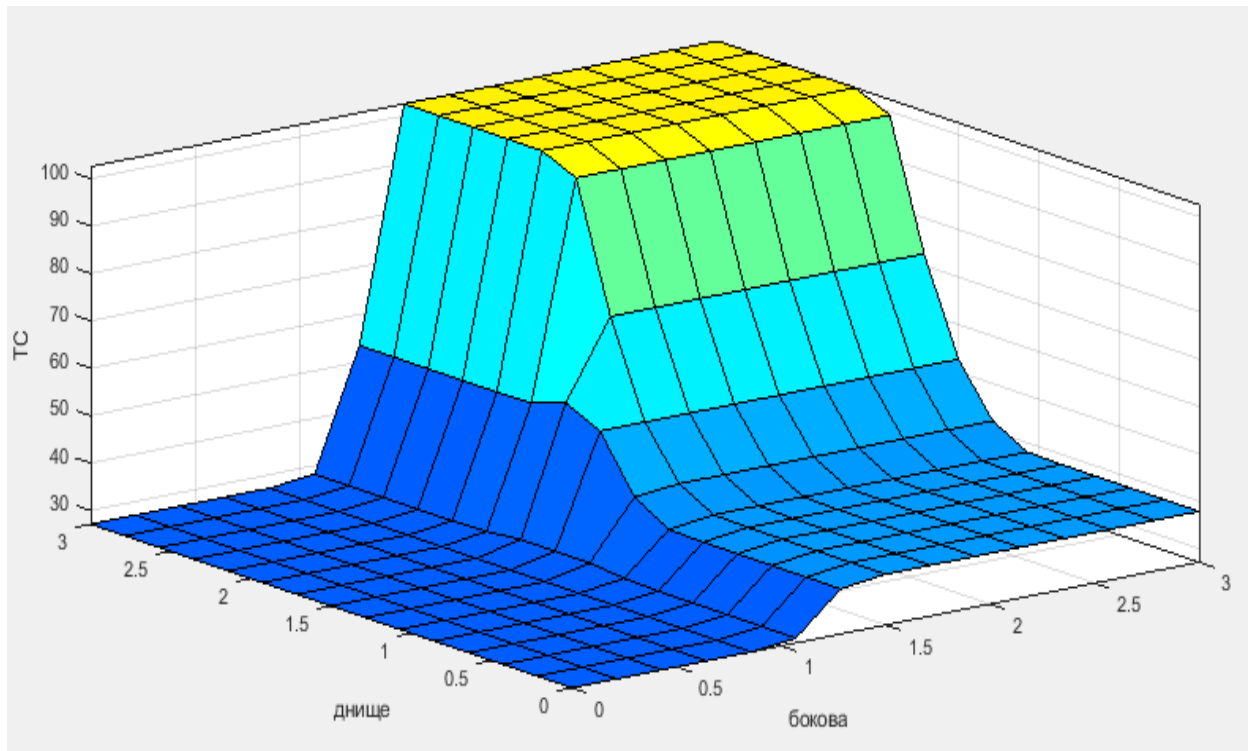


Рис. 2.14. Поверхня нечіткого висновку (алгоритму Мамдані)

Лістинг даної програми надавано в Додатку А (Лістинг А.1).

Відмінність висновку за алгоритмом Сугено саме у формуванні вихідних змінних. База правил систем нечіткого виводу має наступний формат:

Правило 1:

**if** «змінна 1=значення А» **and** «змінна 2=В» **to**  $y = k_1A + k_2B + k_0$ ,

або

Правило 2:

**if** «змінна 1 = значення А» **and** «змінна 2 = В» **to**  $y = C$ .

Результати моделювання представлене на рис. 2.15, 2.16.

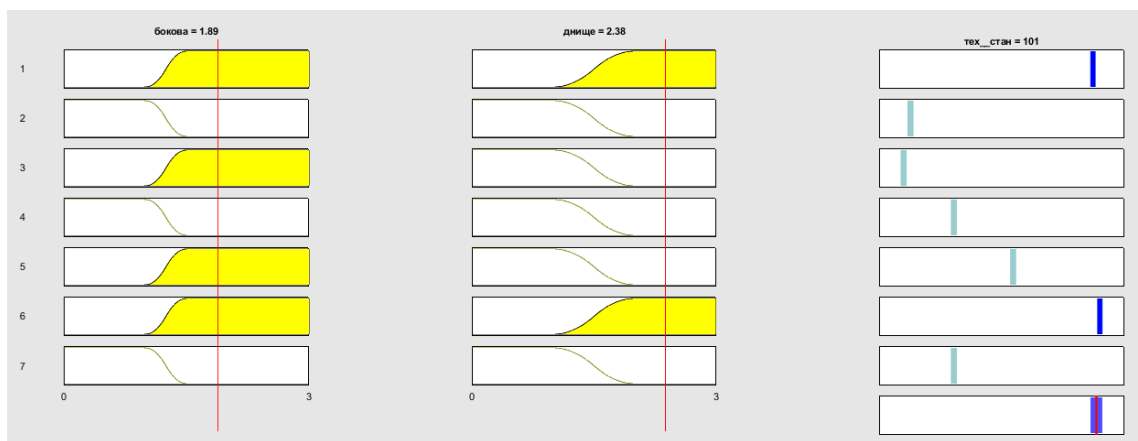


Рис. 2.15. Інтерфейс правил (алгоритм Сугено)

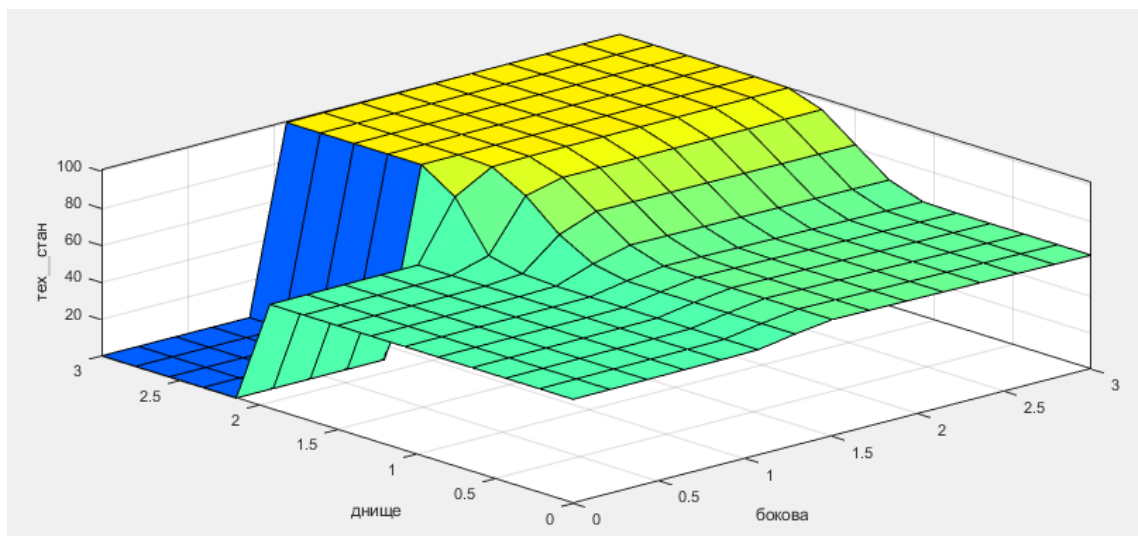


Рис. 2.16. Поверхня нечіткого висновку (алгоритм Сугено)

Як що порівняти цю поверхню і поверхню на рис. 2.14, то можливо зробити висновок про те, що нечіткі правила достатньо добре описують складну залежність технічного стану обшивки судна. При цьому, модель типу Сугено більш точна, але перевага алгоритму Мамдані полягає у тому, що правила бази знань є більш прозорі і інтуїтивно зрозумілі, тоді як для моделей типу Сугено не завжди ясно які лінійні залежності «входи-вихід» необхідно використовувати.

### *Другий модельний приклад*

Розглянемо рішення задачі визначення кількості кластерів (**нечітка субтрактивна кластеризація**) для множини вихідних даних (Додаток А,

таблиця А.1). Ці дані є матрицею даних  $X$  розмірністю (140x2), яка міститься у файлі `fcmdata1.dat`. У цьому випадку виклик функції субтрактивної кластеризації може бути реалізований таким чином:

```
load fcmdata1.dat
```

```
[C,S] = subclust(fcmdata1, [0.5 0.5], [], [1.25 0.5 0.15 1])
```

Оскільки для множини даних з таблиці А.1 (Додаток А), використовується 2 ознаки (матриця `datain` має 5 стовпців), то по кожній з ознак задаються радіуси околиць 0.5 і 0.5 відповідно. Коефіцієнти шкал для відображення множини даних в одиничний гіперкуб знаходять на основі мінімального і максимального значень даних.

Аргумент `squashFactor = 1.25` вказує на те, що необхідно визначити кластери, недалеко розташовані один від одного. Аргумент `acceptRatio=0.5` вказує на те, що для знаходження центрів кластерів дуже високий потенціал не є необхідним. Аргумент `rejectRatio=0.15` не виключає з розгляду точки даних, що не володіють високим потенціалом.

Нарешті, останній аргумент `verbose = 1` дозволяє виведення інформації про виконання процесу кластеризації на екран монітора.

В результаті виконання цього фрагмента команд будуть отримані наступні значення матриці центрів кластерів і вектора  $s$ -значень:

```
Found cluster 1, potential = 1
```

```
Found cluster 2, potential = 0.592871
```

```
C =
```

```
0.6962 0.3893
```

```
0.2661 0.7364
```

```
S =
```

```
0.1580 0.1538
```

```
[C,S] = subclust(fcmdata1, [0.8 0.8], [], [1.5 0.5 0.15 1])
```

```
Normalizing data...
```

```
Computing potential for each data point...
```

```
Found cluster 1, potential = 1
```

```
Found cluster 2, potential = 0.225714
```

```
C =
```

```
0.6334 0.4284
```

```
0.1577 0.8386
```

$$S = \begin{matrix} 0.2529 & 0.2461 \end{matrix}$$

Графічний інтерфейс представлено на рис. 2.17.

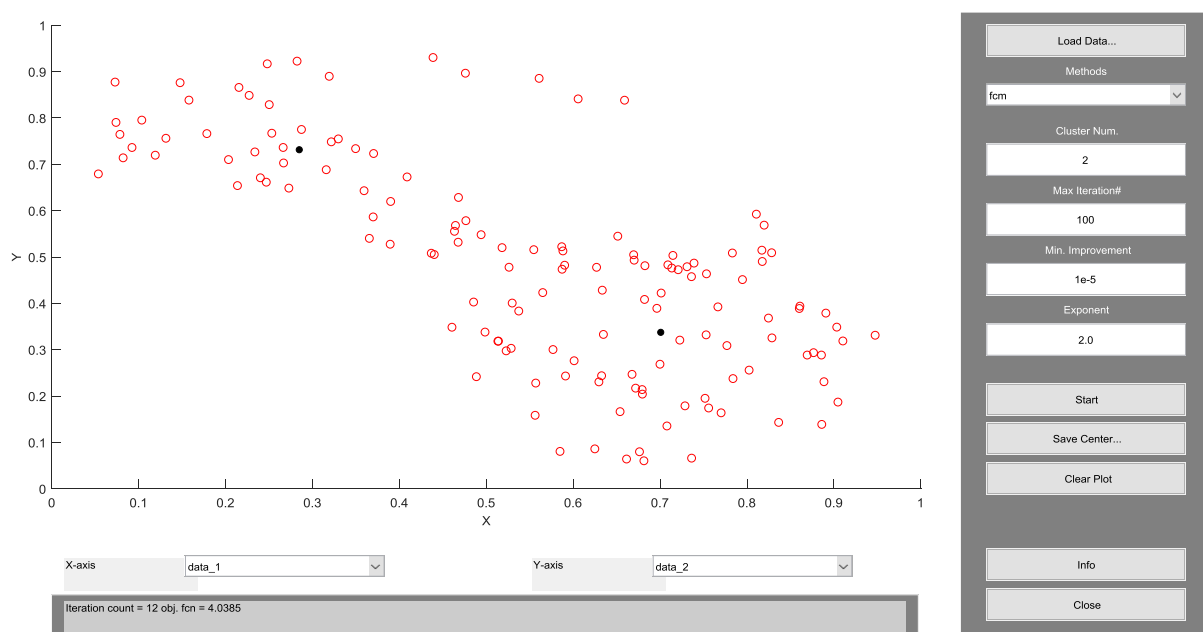


Рис. 2.17. Графічний інтерфейс нечіткої субтрактивної кластеризації

### Моделивання на основі методу нечітких *c*-середніх FCM (Fuzzy-C-Mean)

У Додатку А (Лістинг А.2.) представлено лістинг програми та результати розрахунків.

Формат функція fcm:

$$[center, U, obj\_fcn] = fcm(data, cluster\_n, options)$$

або

$$[center, U, obj\_fcn] = fcm(data, cluster\_n).$$

Аргументи fcm функції:

*data* – матриця початкових даних *D*, *i*-тий рядок якої являє собою інформацію про об'єкт нечіткої кластеризації  $a_i \in A$  у формі

$$\text{вектора } x_i = \{x_i^1, x_i^2, \dots, x_i^q\};$$

*cluster\_n* – кількість кластерів,  $n > 1, n \in N$ .

Значення функції fcm:

*center* – матриця центрів кластерів, кожний рядок якої являє собою координати центру одного з нечітких кластерів в формі вектора

$$v_k = (v_k^1, v_k^2, \dots, v_k^q);$$

нагадаємо, що для кожного кластеру вводяться так звані типові представники або центри  $v_k$  шуканих нечітких кластерів, які розраховуються за наступною формулою [53]

$$v_j^k = \frac{\sum_{i=1}^n (\mu_{A_k}(a_i))^m x_i^j}{\sum_{i=1}^n (\mu_{A_k}(a_i))^m}, (\forall k \in \{2, \dots, c\}, \forall p_j \in P),$$

де  $m$  – деякий параметр, який має назву експоненційна вага і дорівнює деякому дійсному числу ( $m > 1$ );

$P = \{p_1, p_2, \dots, p_q\}$  – множина атрибутів об'єктів кожний елемент з якої є деяка характеристика елементів множини  $A$ . Кожний з центрів кластерів є вектором  $v_k = (v_k^1, v_k^2, \dots, v_k^q)$  в деякому  $q$ -вимірному нормованому просторі ознак, який ізоморфний  $R^q$ , якщо всі ознаки виміряні по шкалі відношень;

$U$  – матриця значень функцій приналежності нечіткого розбиття  $\mu_k(a_i), \forall k \in \{2, \dots, n\}, \forall a_i \in A$ ;

*obj\_fun* – значення цільової функції на кожній з ітерацій роботи алгоритму. Цільова функція  $F(A_k, v_k^j)$  це сума квадратів зважених відхилень координат об'єктів кластеризації від центрів нечітких кластерів

$$F(A_k, v_k^j) = \sum_{i=1}^n \sum_{k=1}^c (\mu_{A_k}(a_i))^m \sum_{j=1}^q (x_i^j - v_k^j)^2.$$

Чим більше елементів містить множина  $A$ , тим менше значення слід вибирати для  $m > 1$ . Де,  $A = \{a_1, a_2, \dots, a_n\}$  – початкова (досліджувана) сукупність даних – скінчена множина або множиною об'єктів кластеризації.

Функція  $fcm()$  викликана з додатковими аргументами  $options$  для управління процесом кластеризації, а також для зміни критерію останова роботи алгоритму.

Ці додаткові аргументи мають наступні значення [53]:

- option (1): експоненційна вага  $m$  для розрахунків матриці нечіткого розбиття  $U$ ;
- option (2): максимальне число ітерацій  $s$ ;
- option (3): параметр збіжності алгоритму  $\varepsilon = 0.00001$ ;
- option (4): інформація про поточну ітерацію.

Результати моделювання надані в Додатку А та на рис.2.17, 2.18.

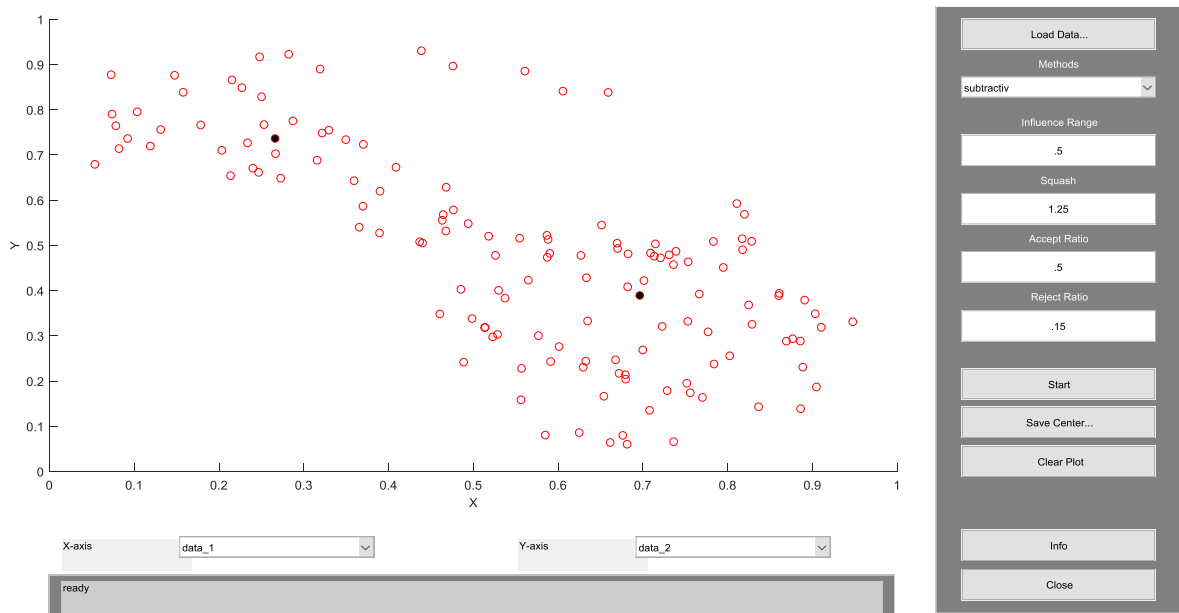


Рис. 2.18. Графічний інтерфейс методу нечітких  $c$ -середніх FCM

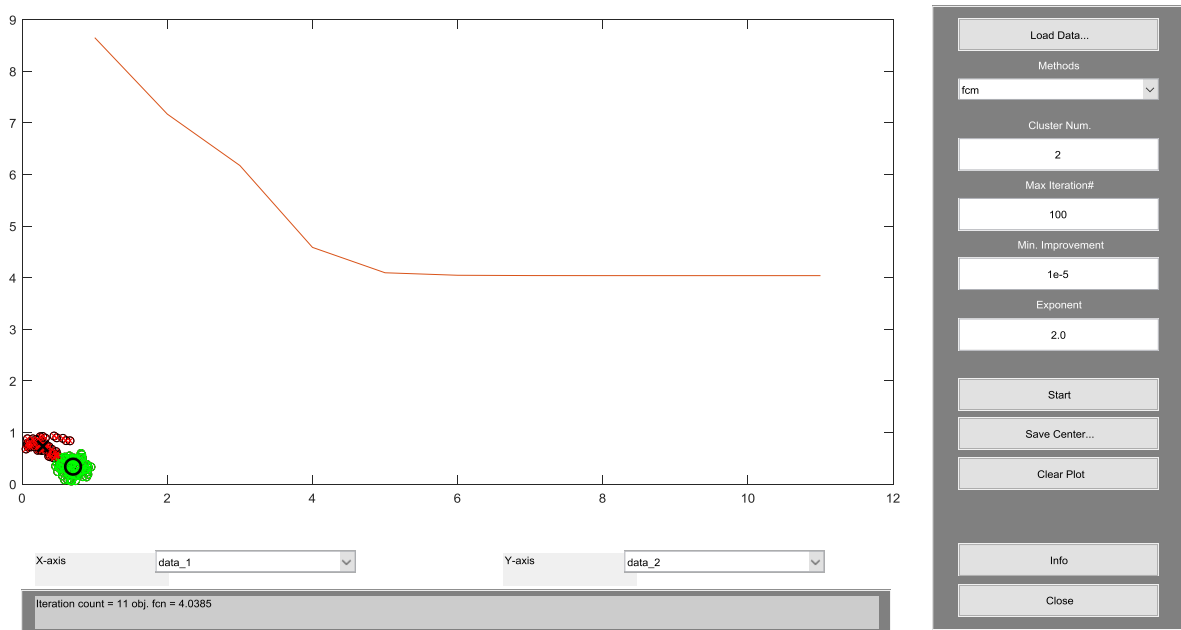


Рис. 2.19. Результати моделювання методом нечітких с-середніх FCM

За результатами моделювання зроблене висновок про те, що пакет прикладних програм для числового аналізу MATLAB, а також мова програмування, що використовується в даному пакеті дозволяє розв'язувати задачі нечіткої кластеризації як за допомогою функцій командного рядка, так і на основі графічного інтерфейсу. Але результати моделювання мають наближений характер тому їх доцільне використовувати для попередньої структуризації вхідних даних про систему, що вивчається.

## Висновки до розділу 2

1. Дослідження сучасних та перспективних принципів та методів прогнозу технічного стану складних систем визначило тенденцію, яку доцільне використовувати в якості загальної стратегії модернізації та створення подібних алгоритмів для засобів водного транспорту, а саме, використання так званих м'яких обчислень.

2. Найбільш ефективною є стратегія удосконалення концептуальної моделі прогнозування технічного стану засобів водного транспорту, в

алгоритмічному сенсі, яка поєднує OLAP-технологію інтелектуального аналізу даних; алгоритм нечіткої субтрактивної кластеризації для визначення кількості значень лінгвістичних змінних (рангу терм-множини); метод нечітких с-середніх (Fuzzy C-Means) для побудови функцій приналежностей, а також гібридні мережі, так звані адаптивні нейроні нечіткі мережі ANFIS для самоналаштування нечіткого висновку на вхідні умови. Аналіз наукових робіт в даній галузі дозволяє зробити висновок про високу ефективність саме такого комплексного застосування добре апробованих підходів.

3. Доведено те, що при удосконаленні методу формування функції приналежності в моделі прогнозування технічного стану доцільне використовувати нечітку кластеризацію, а саме: метод субтрактивної кластеризації для визначення кількості кластерів, яка інтерпретується як ранг базової терм-множини лінгвістичної змінної та методу нечітких с-середніх для розрахунку значень функції приналежності; це дозволяє значно покращити адекватність процедури фазифікації та точність прогнозу.

4. Моделювання прогнозу технічного стану ЗВТ на основі використаних моделей нечіткої кластеризації свідчить про те, що розробка є функціональною, а математичне та програмне забезпечення задовольняє усім потребам користувача. Моделі є адекватними, а дане програмне забезпечення є досить надійним. Інтерфейс програми зручний та ергономічний. Завдяки використанню сучасних алгоритмів для розрахунків та відображення результату програмне забезпечення досить ефективне. Але програмне забезпечення є цілком не портативним, тому що воно розроблялось тільки для роботи на основі пакету MATLAB.

5. Доведене, те що пакет прикладних програм MATLAB та мова програмування, що використовується в даному пакеті дозволяє розв'язувати задачі нечіткої кластеризації за допомогою функцій командного рядка та на основі графічного інтерфейсу. Результати моделювання мають наближений характер тому їх доцільне використовувати для попередньої структуризації вхідних даних.

## РОЗДІЛ 3. РОЗРОБКА ТЕХНОЛОГІЇ НЕЧІТКОГО ВИСНОВКУ НА ОСНОВІ ГІБРИДНОЇ МЕРЕЖІ

### 3.1. Метод прогнозування технічного стану ЗВТ на основі нечіткої нейронної мережі

#### *Розробка нейро-нечіткого логічного висновку*

Нейро-нечіткий логічний висновок формується на етапі прогнозу технічного стану кожної комплектуючої підсистеми ЗВТ. На рис. 3.1 представлена узагальнена структура нейро-нечіткого логічного висновку для прогнозу технічного стану ЗВТ. В якості відомого та добре апробованого вченими наукового методу висновку в дисертація обране так зване ANFIS.

ANFIS (Adaptive Neuro-Fuzzy Inference System) – адаптивна мережева нечітка система виведення, навчається класичними градієнтними алгоритмами. Була запропонована Янгом в 1993р. (ANFIS: Adaptive-Network-Based Fuzzy Inference System) [111]. Ця гібридна мережа є універсальним аппроксиматором та є п'ятишаровою ШНМ прямого поширення сигналу.

На вхід гібридної мережі надходить вектор параметрів  $X = (x_1, x_2, \dots, x_n)$  аналізованого підоб'єкту, який визначається будь-яким методом отримання. Одночасно формується база правил системи нечіткого логічного висновку на основі нечітких правил продукції. Згідно з розробленою технологією для кожної групи параметрів підбирається колекція функцій приналежності. У відповідності до системи нейро-нечіткого виводу використовується алгоритм Сугено (у деякій літературі цей алгоритм має назву Такагі-Сугено, так як від одночасно був запропонований обома вченими) [53]. Мережа буде складатися з 5 шарів так, як представлено на рис. 3.1.

### Призначення та завдання шарів ANFIS

На першому шарі виконується фазифікації вхідних параметрів комплектуючих ЗВТ – на вхід подаються значення і обчислюються значення функцій належності вхідних змінних, тобто перетворення вхідних даних в нечіткі змінні за допомогою функцій приналежності для визначення відповідності вхідних даних термам лінгвістичних змінних.

Другий шар виконує агрегування значень передумов для кожного правила відповідно до обраної Т-норми; тобто – визначення ступеня істинності умов для кожного правила в системі нейро-нечіткого виводу –  $\min$  значень всіх підумови. Також визначається результуюче значення коефіцієнта приналежності характеристичної функції та визначають вагові коефіцієнти  $w$ .

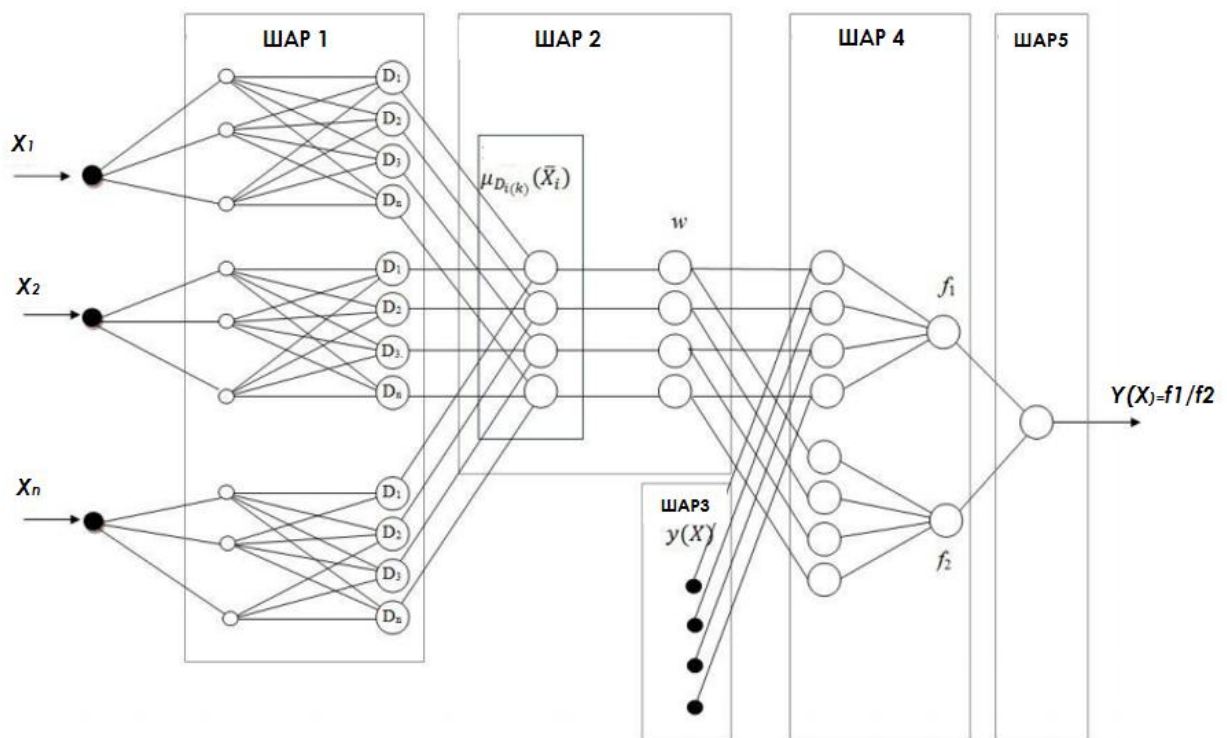


Рис. 3.1. Гібридна мережа ANFIS

Третій шар містить алгоритм навчання, в ході якого коригуються ваги полінома. У даній моделі використовується метод гібридного навчання, що поєднує метод зворотного поширення помилки з методом найменших

квадратів. У методі зворотного поширення помилки відбувається поширення помилки від виходів мережі до входів, а саме, в напрямку, протилежному поширенню сигналів у нормальному режимі роботи.

Відоме, що метод найменших квадратів – математичний метод, який застосовують для вирішення різних завдань. Він заснований на мінімізації відхилень суми квадратів деяких функцій від змінних, які потрібно визначити. Також на третьому шарі визначається вихідна агрегована функція  $y(X)$ , що характеризує множину параметрів  $X$ . В ході дії алгоритму функція  $y(X)$  підлягає коригуванню за допомогою зміни вагових коефіцієнтів  $w$ , визначених на попередньому шарі.

*Четвертий шар* складається з двох нейронів-суматорів: перший відповідає за активацію або композицію підзаключень (перехід від умов до підзаключень і визначення ступеня істинності кожного з підзаключень), другий – забезпечує акумулювання нечітких правил продукції, а це саме є процес знаходження нейро-нечіткої множини або їх об'єднання для кожної з вхідних змінних всіх ступенів істинності підзаключень для отримання функції приналежності кожної з вхідних змінних.

*На п'ятому шарі* виконується дефазифікація вихідних параметрів – перехід від функції приналежності вихідної лінгвістичної змінної до її чіткого (чисельного) значення. Іншими словами, підсумковий результат агрегується шляхом розрахунку середньозважених функцій  $y(x)$  з урахуванням множини ваги  $w$ . Відповідно до виразу [53]

$$y(x) = \frac{f_1}{f_2} = \frac{\sum_{i=1}^N \omega_i y_i(x)}{\sum_{i=1}^N \omega_i}, \quad (3.1)$$

в результаті чого і формується оцінка технічного стану комплектуючого ЗВТ.

Варте підкреслити, що на першому шарі системи нейро-нечіткого виводу на основі алгоритму Такагі-Сугено для кожної вхідної змінної

задається діапазон значень функції належності (фазифікації вхідних параметрів). У нечіткій логіці функція приналежності характеризує ступінь приналежності кожного члена простору міркування до даного нечіткій множині, таким чином, в даному випадку кожна змінна  $X$  визначається функцією приналежності. В залежності від типу об'єкта, виду граничних значень (односторонній або двосторонній діапазон), методи діагностики і т. п. тип і кількість функцій належності можуть бути різними. Визначення оптимального виду і числа функцій приналежності – окремі завдання оцінки технічного стану ЗВТ.

### ***Визначення нечітких правил продукції***

База нечітких продукційних правил системи нечіткого виводу – це система нечітких продукційних правил, що відображає знання експертів про методи управління об'єктом в різних ситуаціях, характер його функціонування в різних умовах тощо, тобто містить формалізовані людські знання.

Найпростіший варіант правила нечіткої продукції, який найбільш часто використовується в системах нечіткого виведення, може бути записаний у формі

якщо " $\beta_1 \in \alpha$ ", тоді " $\beta_2 \in \alpha$ ",

де нечітке вислів " $\beta_1 \in \alpha$ " являє собою умову даного правила нечіткої продукції, а нечітке вислів " $\beta_2 \in \alpha$ " – нечітке укладення даного правила. При цьому вважається, що  $\beta_1 \neq \beta_2$ .

На практиці зручно використовувати ті функції приналежності, які допускають аналітичне представлення у вигляді деякої простої математичної функції. Це спрощує не тільки відповідні чисельні розрахунки, але і скорочує обчислювальні ресурси, необхідні для зберігання окремих значень цих функцій приналежності.

### ***Визначення кількості функцій приналежності***

Для кожного параметра, що описує стан аналізованого ЗВТ (його комплектуючих підсистем), важливо визначити кількість функцій приналежності (універсум), таку що найкращим чином характеризують даний параметр функціонування. Для цього необхідно виходити з розглянутої задачі і необхідної точності опису.

### ***Визначення виду функцій приналежності***

Для кожного параметра, що описує стан аналізованого об'єкта, також важливо визначити вид функції приналежності, найкращим чином характеризує даний параметр.

Розглянемо функції приналежності, які доцільне використовувати в роботі. Данні функції приналежності легко застосовні до вирішення більшості завдань. Введемо наступні функції приналежності: кусково-лінійні - набір відрізків прямих ліній, що утворюють безперервну або кусково-неперервну функцію; гауссових – утворені на основі гауссового розподілу; сигмоидну криву та інші (рис.3.2-3.7). Аналітичні вирази функцій викладене в [44,53,101].

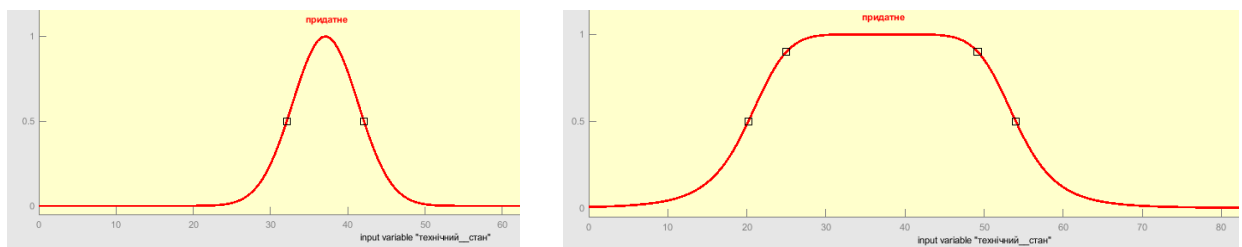


Рис. 3.2. Функції приналежності типу «dbellmf» та «gaussmf»

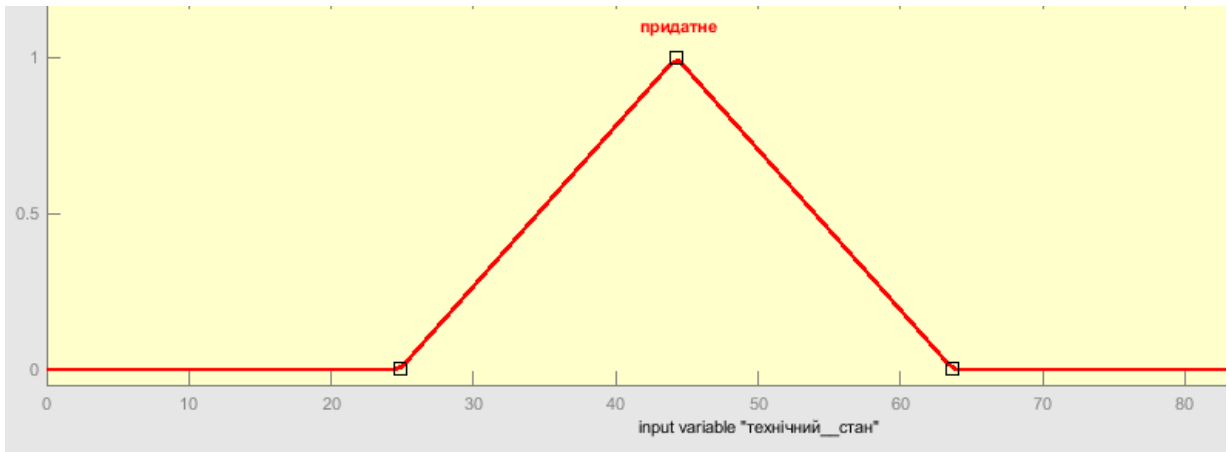


Рис. 3.3. Функція приналежності типу «trimf»

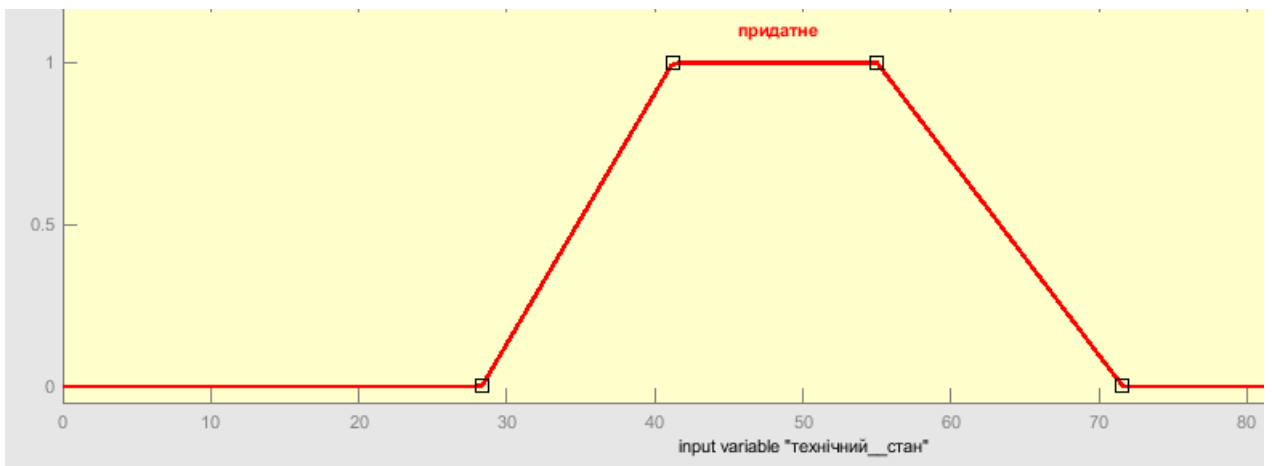


Рис. 3.4. Функція приналежності типу «trapezmf»

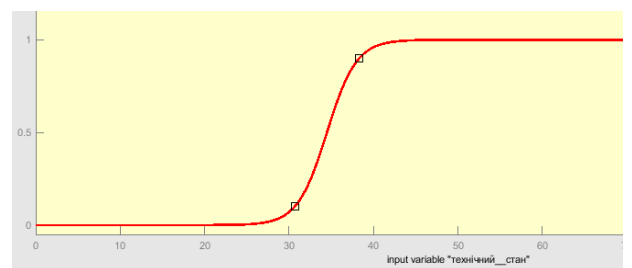
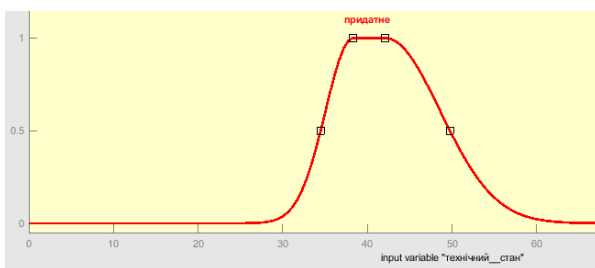


Рис. 3.5. Функції приналежності типу «gauss2mf» та «sigmf»

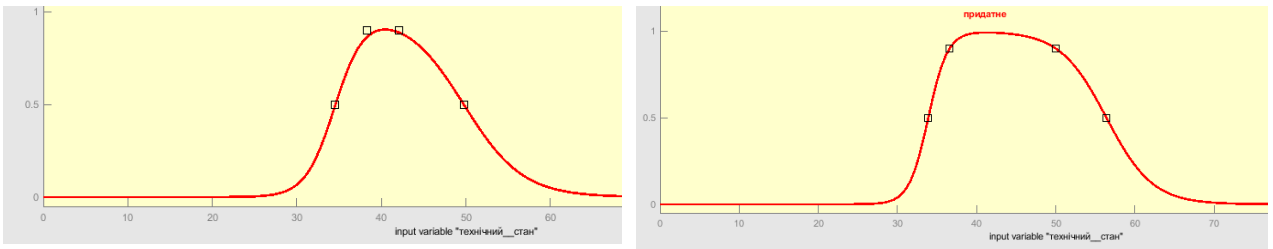


Рис. 3.6. Функції приналежності типу «dsigmf» та «psigmf»

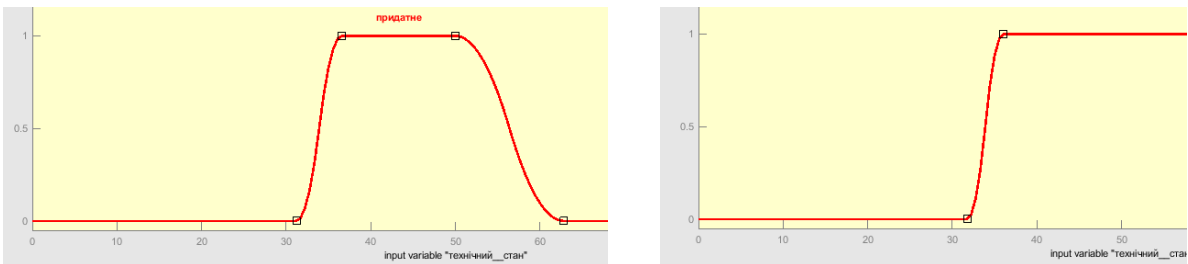


Рис. 3.7. Функції приналежності типу «pmf» та «simf»

В 2 розділі дисертації, в якості одного з наукових результатів розроблена та представлена технологія формування функції приналежності, яка в загальній класифікації відноситься до так званих непрямих методів.

### 3.2. Формування навчальної вибірки гібридної мережі ANFIS

Одним з переваг штучних нейронних мереж є можливість адаптації, так званого, навчання. Для цього в системі нейро-нечіткого висновку формується навчальна вибірка. Репрезентативність або структурованість цієї вибірки також впливає на результати роботи системи, тому, перш ніж перейти до використання адаптивної системи нейро-нечіткого виводу, необхідно виконати попередній аналіз навчальної вибірки.

Формування навчальної вибірки для будь-якого завдання, в тому числі і завдання прогнозу технічного стану ЗВТ, зводиться до оптимізації цієї вибірки. Оптимальна навчальна вибірка повинна володіти трьома основними властивостями [54]:

бути достатньою – кількість навчальних прикладів має бути достатнім для навчання;

бути різноманітною – мати велике число різноманітних комбінацій «вхід-вихід» при обліку, що всі класи, які характерні для вихідної множини, будуть представлені;

бути рівномірною – приклади повинні бути представлені приблизно в однакових пропорціях.

Розглянемо задачу формування навчальної вибірки на прикладі використання діагностичних даних про стан колінвалу трюмного насосу. В цьому випадку навчальна вибірка – це сукупність даних, отриманих в ході експлуатації. Для будь-якого виду комплектуючих існує три варіанти формування навчальної вибірки:

1. У разі відсутності достатньої ретроспективної інформації у вигляді діагностичних даних по даному об'єкту (неповна та недостовірна інформація) навчальна вибірка формується на основі діагностичних даних по аналогічним об'єктам порівнянної потужності і однотипної конструкції.

2. При наявності діагностичних даних по даному об'єкту (неповна але достовірна інформація) навчальна вибірка формується на основі наявних діагностичних даних з використанням діагностичних даних для аналогічних об'єктів порівнянної потужності і однотипної конструкції.

3. Якщо інформація подається не діагностичними даними, а даними вимірів без постановки діагнозів, наприклад з датчиків вібрації без алгоритмів контролю параметрів (повна недостовірна інформація), то таку інформацію необхідно попередньо перевірити на наявність явних випадкових помилок.

При обробці інформації, отриманої з різних вимірювальних приладів, часто доводиться мати справу з так званими «викидами» – вимірами, які різко виділяються серед всієї сукупності і не відповідають нормальному розподілу. Природа таких викидів може бути будь-яка – неточність приладів, неправильний спосіб вимірювання та ін. Спостерігаються односторонні і двосторонні відхилення, відповідно в меншу і більшу сторони [17,24].

Кількість даних у вибірці заздалегідь невідомо. Залежно від об'єкта вона може варіюватися від десятків до сотень, тому аналіз впливу даних навчальної вибірки на результат роботи системи виконувався на основі декількох найбільш часто використовуваних критеріїв [24]:

1. Критерій Романовського.
2. Критерій Ірвіна.
3. Критерій Колмогорова-Смирнова.
4. Критерій «три сигма».

Критерій "трьох сигм" застосовується для результатів вимірювань, розподілених за нормальним законом.

З метою обрання раціонального критерію роботи оцінювався вплив кожного з критеріїв на результат роботи системи нейро-нечіткого логічного висновку. Для аналізу були обрані дані, отримані розміру шийок колінвалу трюмного насосу та компресора силової установки типу 8-74VT2BF160, що складаються з 50 вимірювань [60]. В ході аналізу бралися до уваги наступні показники:

кількість викидів в залежності від критерію;

число пар випадків у вибірці після виключення викидів (навчальна вибірка);

число елементів в тестовій вибірці;

помилка навчання і помилка тестування.

Тестова вибірка складалася з 40 даних.

Крім перерахованих критеріїв розглядалася також первісна вибірка вихідних значень. У розділі 2 даної роботи вже був виконаний аналіз функцій приналежності для представленої моделі. В ході практичних розрахунків було визначено, що оптимальними функціями приналежності для структури нейро-нечіткого логічного висновку на основі експлуатаційних даних є пі подібні функції приналежності.

Аналіз впливу різних критеріїв на оцінку фактичного стану шийок колінвалу трюмного насосу головної силової установки типу 8-74VT2BF160

показує, що кращі результати при налаштуванні системи (найменша помилка навчання) і при перевірці тестової вибірки спостерігаються при використанні критерію Романовського: похибка навчання – 0,00006%; похибка тестування – 0,83%. Найгірші результати отримані при використанні вихідної вибірки без застосування будь-яких критеріїв: похибка навчання – 0,0556%; похибка тестування – 2,37%. Що ще раз підкреслює необхідність попередньої роботи з даними перед використанням в системі оцінки фактичного стану.

Аналіз впливу різних критеріїв на результати роботи системи нейро-нечіткого логічного висновку виявив найбільшу кількість «помилкових» вимірювань в навчальній вибірці при використанні критерію Романовського. Тому, незважаючи на найменші помилки тестування і навчання при розрахунку, даний критерій не є оптимальним. При виключенні «помилкових» вимірювань і аналізі отриманої навчальної вибірки стало ясно, що вона не задовольняє вимогам рівномірності і показності. Наступними значеннями, що дають задовільні результати за критерієм найменших помилок навчання і тестування, є значення, отримані в ході використання критеріїв Смирнова і Зδ.

Аналіз навчальних вибірок, отриманих після виключення «помилкових» вимірювань, наведений для обох критеріїв, виявив, що тільки при використанні критерію «Зδ» вибірка задовольняє всім трьома вимогам – показності, рівномірності і несуперечності. Таким чином, для подальших розрахунків приймався критерій «Зδ». Варте підкреслити, що властивість рівномірності характеризує рівномірність розподілу даних в навчальній вибірці за діагнозами, властивість суперечливості – кількість однакових об'єктів, що належать до різних класів, властивість показності є не тільки критерієм наявності достатнього числа даних по кожному з діагнозів, але і критерієм відсутності повторюваних даних.

Таким чином, не можна використовувати в якості критерію найкращого рішення тільки помилки тестування і навчання нейро-нечіткого логічного висновку. Додатково необхідно проаналізувати, які саме вимірювання

система розпізнає як «помилкові» і чи задовольняє отримана вибірка (після виключення «викидів») всім перерахованим вимогам. Видалення даних після обробки навчальної вибірки виконується тільки для даних третього типу, які відносяться до класу «повна недостовірність» інформація, так як ця інформація є сукупністю даних, які не мають конкретного діагнозу (висновку) по кожному набору даних. Два інших типи інформації не потребують перевірки на наявність викидів, так як на практиці ця інформація вже оброблена експертами і викиди вже були видалені раніше, а кожного набору даних присвоєні свої діагнози.

Представлений принцип формування навчальної вибірки застосуємо для будь-якої доступної інформації про аналізованому об'єкті.

### **3.3. Комп'ютерне моделювання на базі ANFIS**

Для перевірки адекватності та достовірності розроблених та представлених в дисертації моделей та методів прогнозування ТС ЗВТ на основі м'яких обчислень використані дані спостереження експлуатації комплектуючих силової установки типу 8-74VT2BF160, які опубліковані в роботі [60]. Вимірювання виробки робочих поверхонь та зазорів здійснювалось через 4-5 тис. годин роботи. Результати контролю заносились до формулярів обладнання. Елементи циліндре-поршневої групи реєструвались окремо за кожним з циліндрів поелементно.

Виконаємо прогноз технічного стану.

#### ***Третій модельний приклад***

Розглядаються вкладиші корінних та шатунних підшипників головного дизельного двигуна типу 8-74VT2BF160, шатунні шийки колінвалів трюмного насоса та компресора. Вимірювання товщини вкладишів корінних та шатунних підшипників здійснюються мікрометром гладким зі сферою типу МК-102-1 ГОСТ 6507–78 або спеціальною важільною скобою МР-2161, кінцевими мірами довжини 5 розряду ГОСТ 9038–73 у трьох площинах:

вертикальній та зі зсувом на  $45^\circ$  в обидві сторони. Припустимі виміри для шатунних та корінних підшипників наведено у таблиці 3.1.

Таблиця 3.1

### Розміри шатунних та корінних підшипників

Номінальний розмір, мм	Розмір після ремонту, мм	
	Капітальний ремонт	Середній ремонт
$9,45 \pm 0,01$	$9,45 \pm 0,01$	$9,42 \pm 0,01$
	$9,575 \pm 0,01$ (1 ремонт)	
	$9,70 \pm 0,01$ (2 ремонт)	

В [60] наведена інформація про зміни розміру шатунного підшипника (вкладиша) на основі статистичних даних при періодичному контролі через кожні 4000 год. роботи двигуна. Виміри зазору у даному вузлу дозволяють зробити висновок про нелінійний та нерівномірний (для різних екземплярів) характер зносу підшипника. Діапазону часу безвідмовної роботи складає від 16 до 22 тис. год. У такому випадку, за класичними підходами, міжремонтний ресурс для зазначеного типу було встановлено на рівні 18 тис. год.

### Методика моделювання

- 1) Формування навчальних даних (Training) здійснено за інформацією з роботи [60] та згідно п.3.2. Створено файл podship1.dat (рис. 3.8).



Файл	Редагування	Формат	Вигляд	Довідка
9,44	9,45	9,4	9,38	26000
9,44	9,44	9,39	9,37	25000
9,43	9,43	9,38	9,35	24000
9,43	9,42	9,37	9,32	22000
9,42	9,41	9,36	9,3	20000
9,42	9,4	9,32	9,27	18000
9,41	9,38	9,3	9,25	17000
9,4	9,36	9,28	9,2	16000

Рис. 3.8. Навчальні дані (текстовий файл)

- 2) Завантаження навчальних даних. Інтерфейс зображено на рис.3.9.

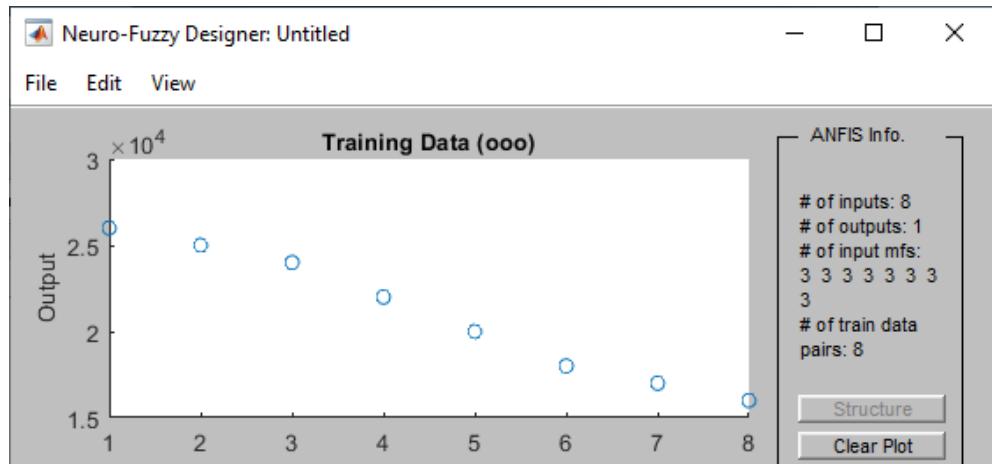


Рис.3.9. Навчальні дані (інтерфейс ANFIS)

- 3) Налаштування нечіткої кластеризації. Інтерфейс вхідних даних на рис. 3.10.

Parameter	Value
Range of influence	0.5
Squash factor	1.25
Accept ratio	0.5
Reject ratio	0.15

Рис. 3.10. Інтерфейс даних для кластеризації

- 4) Формування гібридної мережі (рис.3.11).

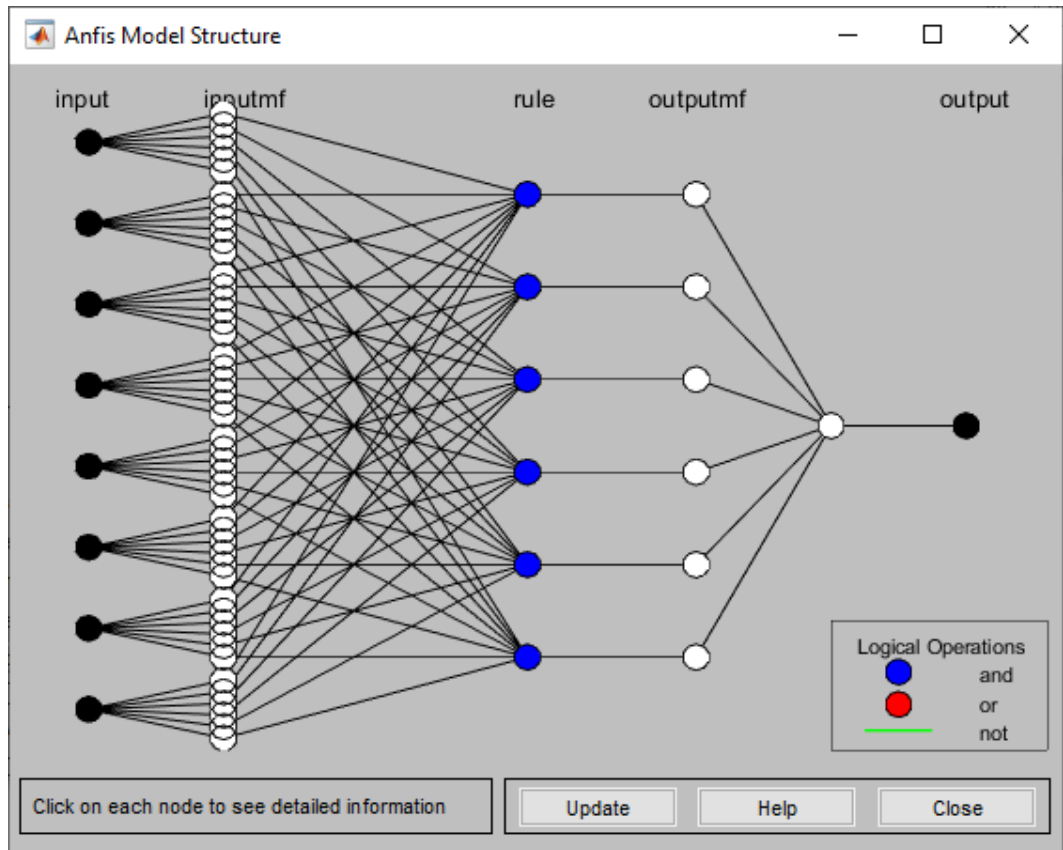


Рис. 3.11. Гібридна мережа

## 5) Навчання мережі гібридним методом (рис. 3.12)

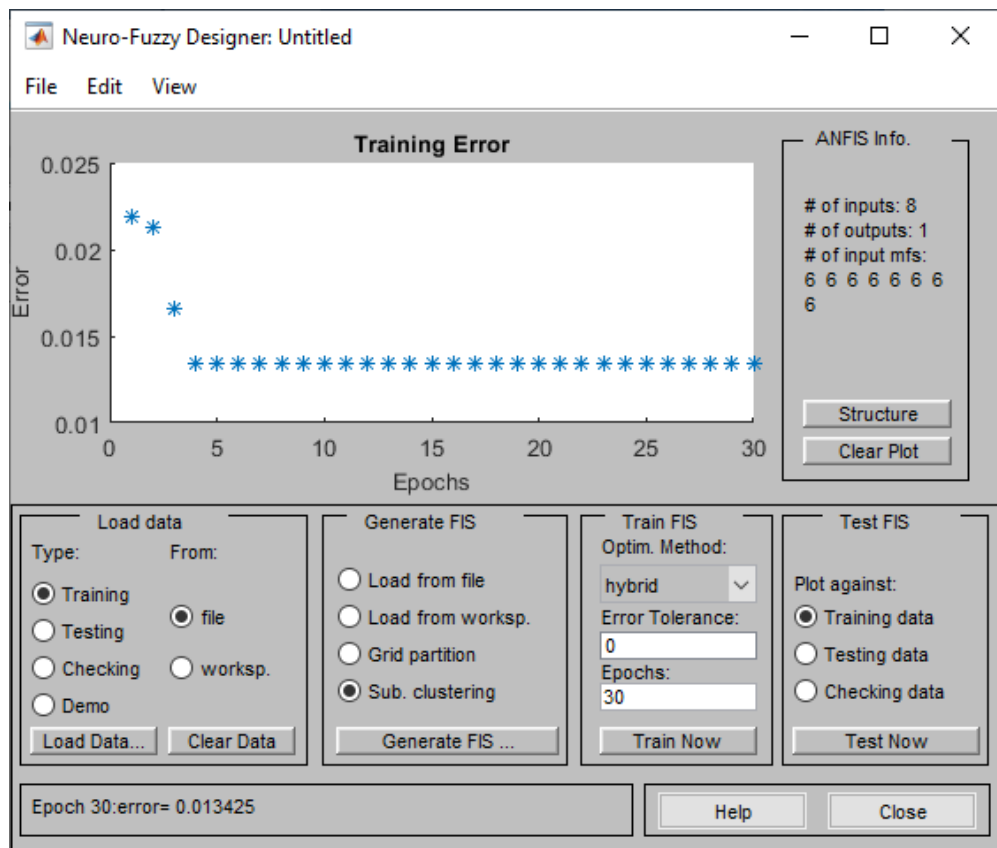


Рис.3.12. Навчання мережі

## 6) Тестування мережі (рис.3.13)



Рис.3.13. Тестування мережі

## 7) Формування правил (рис. 3.14)

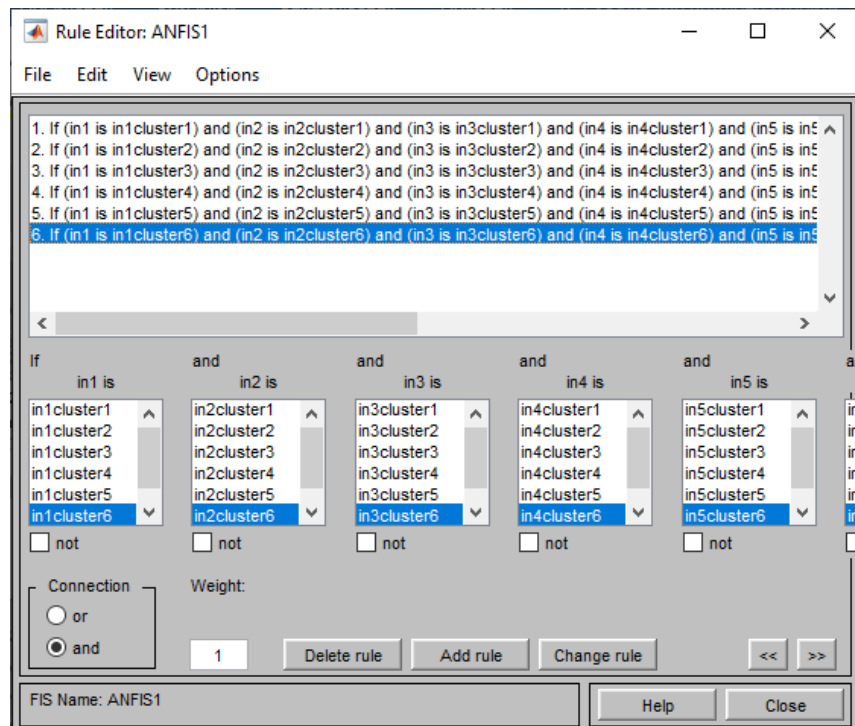


Рис. 3.14. Правила висновків

Лістинг програми надане у Додатку А (Лістинг А3).

На рис. 3.15 надана модель FIS для гібридної мережі, яка сформована для модельного прикладу.

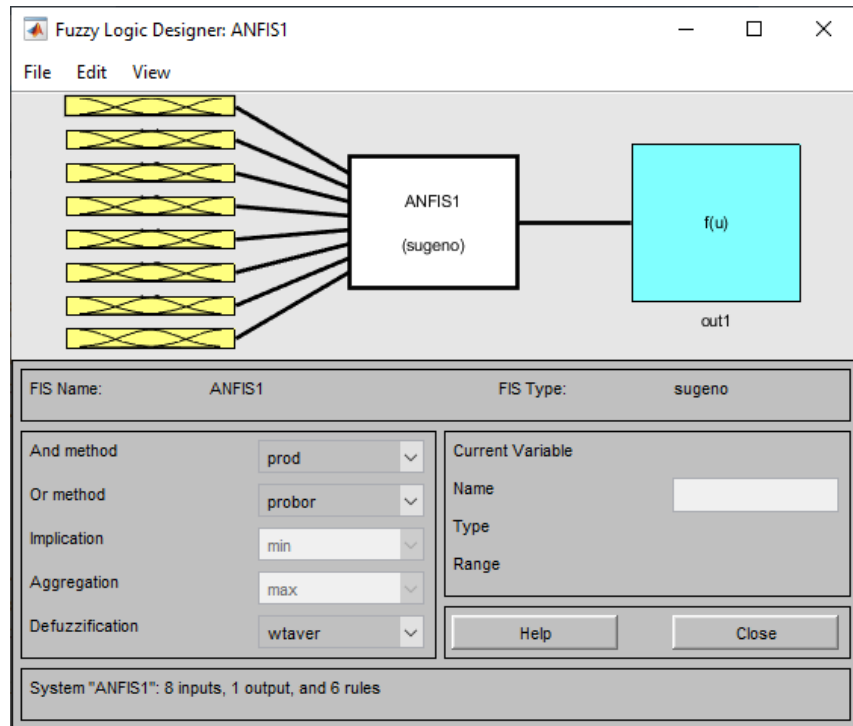


Рис. 3.15. Модель нечіткого висновку за алгоритмом Сугено  
Приклади функцій приналежності надано на рис. 3.16, 3.17.

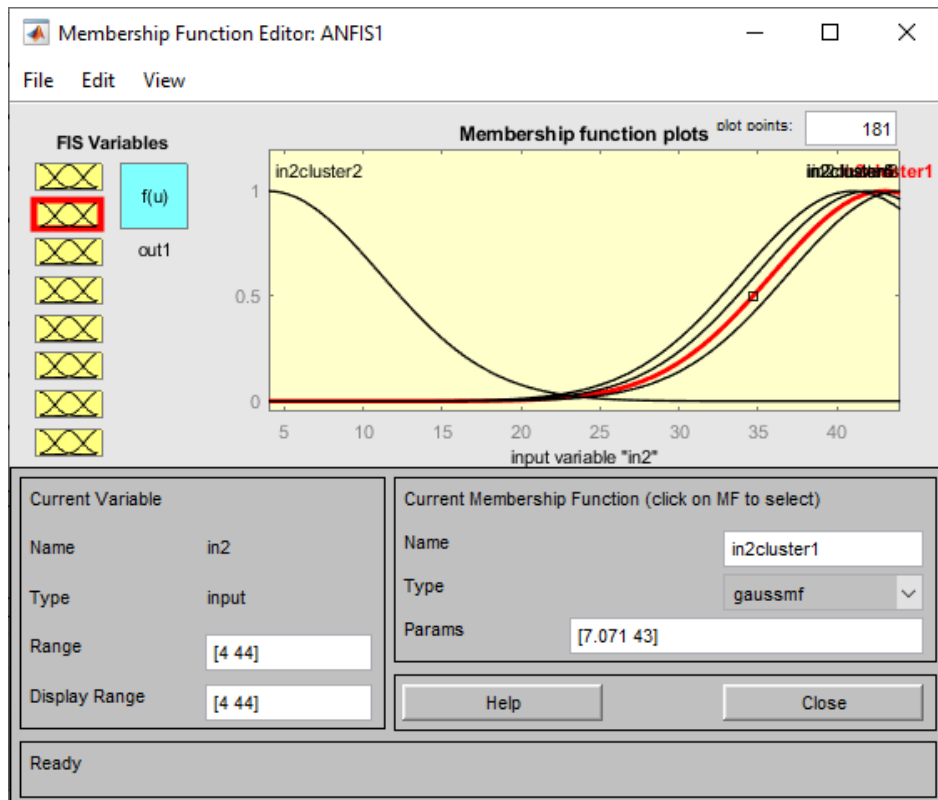


Рис. 3.16. Функція приналежності

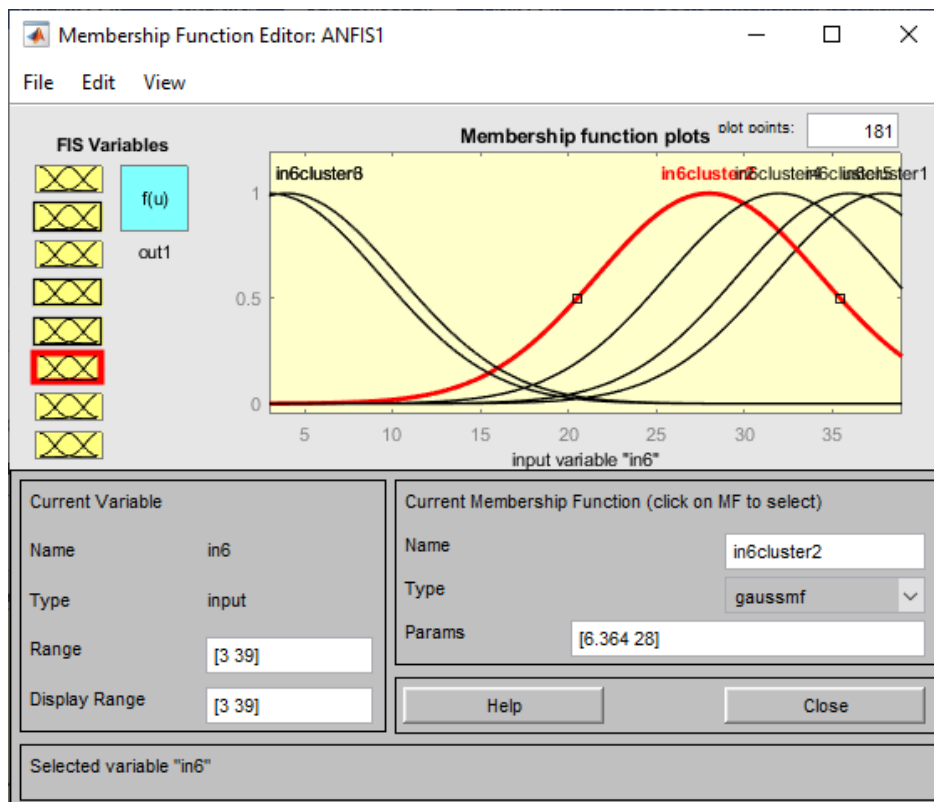


Рис. 3.17. Функція приналежності

Результати моделювання прогнозу ТС ЗВТ на прикладі комплектуючих головної силової установки дизельного двигуна типу 8-74VT2BF160 свідчать повний збіг даних в результаті прогнозу ТЗ з реальними значеннями, які були задокументовані в процесі експлуатації при використанні оптимальної кількості вхідних даних в навчальній вибірці.

### Висновки до розділу 3

1. Аналіз впливу різних критеріїв обрання кількості даних у вибірці на оцінку фактичного стану шийок колінвалу трюмного насоса силової установки типу 8-74VT2BF160 показує, що кращі результати при налаштуванні системи, а саме, найменша похибка навчання спостерігаються при використанні критерію Романовського: похибка навчання – 0,00006%; похибка тестування – 0,83%. Найгірші результати отримані при

використанні вихідної вибірки без застосування будь-яких критеріїв: похибка навчання – 0,0556%; похибка тестування – 2,37%.

2. Дослідження впливу різних критеріїв формування навчальної вибірки на результати роботи системи нейро-нечіткого логічного висновку виявили найбільшу кількість «помилкових» вимірювань в навчальній вибірці при використанні критерію Романовського. Тому, незважаючи на найменші помилки тестування і навчання при розрахунку, даний критерій не є оптимальним. При виключенні «помилкових» вимірювань і аналізі отриманої навчальної вибірки визначене те, що вона не задовольняє вимогам рівномірності і показності. Наступними значеннями, що дають задовільні результати за критерієм найменших похибок навчання і тестування, є значення, отримані в ході використання критеріїв Смирнова і «Зδ».

3. Аналіз навчальних вибірок, отриманих після виключення «помилкових» вимірювань, наведений для обох критеріїв (Смирнова і «Зδ»), виявив, що тільки при використанні критерію «Зδ» вибірка задовольняє всім трьом вимогам – показності, рівномірності і несуперечності. Таким чином, для подальших розрахунків приймався критерій «Зδ». Варте підкреслити, що властивість рівномірності характеризує рівномірність розподілу даних в навчальній вибірці за діагнозами, властивість суперечливості – кількість однакових об'єктів, що належать до різних класів, властивість показності є не тільки критерієм наявності достатнього числа даних по кожному з діагнозів, але і критерієм відсутності повторюваних даних.

4. Таким чином, не доцільне використовувати в якості критерію найкращого рішення тільки помилки тестування і навчання нейро-нечіткого логічного висновку. Додатково необхідно проаналізувати, які саме вимірювання система розпізнає як «помилкові» і чи задовольняє отримана вибірка (після виключення «викидів») всім перерахованим вимогам. Видалення даних після обробки навчальної вибірки виконується тільки для даних третього типу, які відносяться до класу «повна недостовірність» інформація, так як ця інформація є сукупністю даних, які не мають

конкретного діагнозу (висновку) по кожному набору даних. Два інших типи інформації не потребують перевірки на наявність викидів, так як на практиці ця інформація вже оброблена експертами і викиди вже були видалені раніше, а кожного набору даних присвоєні свої діагнози.

5. Метод прогнозу технічного стану ЗВТ на основі нечіткого висновку при застосуванні гібридної мережі, доцільне будувати на моделі адаптивної нейронної нечіткої мережевої системи ANFIS (Adaptive Neuro-Fuzzy Inference System), так як це надає можливість одночасно використовувати переваги нечіткої логіки та нейронних мереж для підвищення достовірності прогнозування технічного стану ЗВТ.

## **РОЗДІЛ 4. РОЗРОБКА РЕКОМЕНДАЦІЙ ЩОДО ПРАКТИЧНОЇ РЕАЛІЗАЦІЇ НАУКОВИХ РЕЗУЛЬТАТІВ. ПЕРЕВІРКА ДОСТОВІРНОСТІ**

### **4.1. Рекомендації щодо реалізації моделей та методів прогнозування**

Дослідження показали, що при реалізації запропонованих в дисертація моделей та методів прогнозу в програмному забезпеченні доцільне використовувати технології об'єктно-орієнтованих програмування. При цьому розробники зустрічаються з цілим комплексом труднощів: необхідно підібрати підходящі об'єкти, віднести їх до різноманітних класів, визначити інтерфейси класів та ієрархію спадкування і встановити відношення між класами, при цьому зберігати розумну ступінь деталізації. З одної сторони, дизайн повинен відповідати вирішуваній задачі, а з іншої – бути універсальним, щоб можна було врахувати потреби, що можуть виникнути у майбутньому. Також, потрібно уникати ситуацій, коли виникає необхідність у перепроєктуванні програми, так як це значно підвищує рівень роботи, яку необхідно виконати розробнику.

Раніше програмне забезпечення розроблялося, написанням сотні рядків програмного коду безпосередньо в інтерфейсі продукту. Іноді використовувалися сервіси і менеджери для роботи з даними і тоді рішення виходило з використанням паттерну Document-View. Підтримка такого коду вимагала колосальних затрат, так як нового розробника треба навчити, який код за що в продукті відповідає, і ні про яке модульне тестування і мови не йшло. З часом розроблювані додатки ставали більше і складніше, з однієї згуртованої команди розробників утворилося багато різних команд розробників, архітекторів, дизайнерів і проектних менеджерів. Вартість

розробки ПЗ зростає в багато разів. Такий підхід до розробки вимагає наявності стійкої архітектури, яка б синхронізувала різні функціональні області продукту між собою. Для полегшення поставлених задач доцільне використання архітектурних шаблонів проектування.

### **Аналіз архітектурних шаблонів проектування програмного забезпечення**

Враховуючи складність проектування програмного забезпечення, з'явилась необхідність у розробці методів зменшення трудових затрат в даних процесах. Концепція вирішення цієї проблеми має назву архітектурного паттерну (шаблону). Нагадаємо елементи, з яких складається будь-який паттерн:

1. *Ім'я*. З його допомогою можна зразу описати проблему проектування, її рішення та наслідки. Присвоєння паттернам імен дозволяє проектувати на більш високому рівні абстракції. Також воно вносить спільну термінологію.

2. *Задача*. Опис того, коли варто застосовувати паттерн. Необхідно сформулювати задачу та її контекст. Тут може описуватися конкретна проблема проектування, наприклад спосіб представлення алгоритмів у виді об'єктів. Іноді відзначається, які структури класів та об'єктів свідчать про негнучкий дизайн. Також може включатися перелік умов, при виконанні яких слушно застосовувати даний паттерн.

3. *Рішення*. Опис елементів дизайну, відношень між ними, функцій кожного елементу. Мається на увазі не конкретний дизайн чи реалізація, оскільки паттерн – це шаблон, що застосовується в різноманітних ситуаціях. Дається абстрактний опис задачі проектування і того, як вона може бути вирішена за допомогою деякого узагальненого поєднання елементів (класів та об'єктів).

4. *Результати.* Це наслідки застосування паттерну та різного роду компроміси. Хоча при описі проектних рішень про наслідки часто не згадують, знати про них необхідно, щоб можна було вибрати між різними варіантами і оцінити переваги та недоліки даного паттерну. Тут йде мова про вибір мови та реалізації. Оскільки в об'єктно-орієнтованому проектуванні повторне використання зазвичай являється важливим фактором, то до результатів слід також відносити й вплив на ступінь гнучкості, розширюваності та переносимості системи. Перелік всіх наслідків може допомогти зрозуміти і оцінити їх роль.

Таким чином, паттерн проектування іменує, абстрагує та ідентифікує ключові аспекти структури загального рішення, що дозволяють застосувати його для створення повторно використовованого дизайну. Він виокремлює задіяні класи та екземпляри, їх роль і відношення, а також функції. При описі кожного паттерну увага акцентується на конкретній задачі об'єктно-орієнтованого проектування. Аналізується, коли слід застосовувати паттерн, чи можна його використовувати з врахуванням інших проектних обмежень, які будуть наслідки застосування даного методу.

Таким чином, паттерн (англ. шаблон) – в області програмної інженерії, повторювана архітектурна конструкція, що описує взаємодію об'єктів та класів, адаптованих для рішення спільної задачі проектування в конкретному контексті.

Архітектурні шаблони виражають фундаментальну схему структурної організації певної програмної системи. Така схема складається із визначених наперед підсистем, а також точно визначає їхні сфери відповідальності та взаємовідносини. На відміну від шаблонів проектування, архітектурні шаблони мають ширший масштаб. Основними елементами структури архітектурних шаблонів є не класи й об'єкти, а умовні частини (підсистеми) програмної системи. В області програмної архітектури концепція паттерну покликана вирішувати такі проблеми як обмеженість потужностей апаратної частини системи, високої доступності та мінімізація бізнес-ризиків.

Одним із найбільш розповсюджених і базових архітектурних шаблонів є Model-View-Controller [122]. MVC – це фундаментальний паттерн, який знайшов застосування в багатьох технологіях, дав розвиток нових технологій. Мета шаблону – гнучкий дизайн програмного забезпечення, який повинен полегшувати подальші зміни чи розширення програм, а також надавати можливість повторного використання окремих компонентів програми. Крім того використання цього шаблону у великих системах призводить до певної впорядкованості їх структури і робить їх зрозумілішими завдяки зменшенню складності. Вперше паттерн MVC з'явився в мові Smalltalk. Розробники повинні були придумати архітектурне рішення, яке дозволяло б відокремити графічний інтерфейс від бізнес логіки, а бізнес логіку від даних. Таким чином, у класичному варіанті, MVC складається з трьох частин, які й дали йому назву.

### **Аналіз існуючих архітектурних шаблонів на базі MVC-паттерну**

При проектуванні інтерактивних додатків, так само як і інших програм, забезпечення модульності його компонентів має достатньо багато переваг. Ізоляція функціональних одиниць одна від одної, настільки, наскільки це можливо дає можливість дизайнеру додатку простіше розуміти та модифікувати кожен окремий модуль. Практика розробки програмного забезпечення показала, що найбільшу ефективність забезпечує розділення програмних компонентів на три частини. Таке розбиття додатку передбачає відділення частин, що представляють модель логіки на якій ґрунтується додаток від того, яким чином ця логіка представляється користувачу і від того, як користувач з цим взаємодіє.

Архітектурний шаблон Модель-Вид-Контролер (MVC) являється реалізацією такого потрібного структурування, де об'єкти різних класів відповідають за операції, що відносяться до доменної логіки програми

(модель), відображення стану додатку (вид) і взаємодії користувача з моделлю та видом (контролер).

Методологія MVC дозволяє програмістам прописувати модель додатку визначивши спочатку нові класи, що будуть втілювати специфічну, для предметної області додатку, інформацію. Потім, вони матимуть змогу створювати користувацький інтерфейс для цієї моделі складаючи комплексні представлення для неї (вікна), за допомогою вбудовування екземплярів попередньо визначених класів інтерфейсу. Такий підхід добре підходить не тільки для представлення, але й також для реалізації контролю (редагування). Перевагою такого підходу є можливість легкої заміни способу, яким модель представляються користувачеві та яким чином він нею керує.

### ***Model-View-Controller***

Дана реалізація шаблону являється первиною та є базою для похідних, більш сучасних та вдосконалених паттернів. Тому, в останній час, в чистому вигляді цей підхід не використовується. Але всі модифікації даного шаблону, в тому чи іншому вигляді реалізують три основні елементи паттерну. Тому, розглянемо кожен з них окремо.

**Модель** – частина, що містить в собі функціональну бізнес-логіку програми. Модель повинна бути повністю незалежна від інших частин продукту. Модельний шар нічого не повинен знати про елементи дизайну, і яким чином він буде відображатися. Досягається результат, що дозволяє міняти представлення даних, те як вони відображаються, не чіпаючи саму Модель.

**Бізнес-логіка** або **логіка предметної області** (англ. *domain logic*) – в розробці інформаційних систем – сукупність правил, принципів, залежностей поведінки об'єктів предметної області (галузі людської діяльності, яку система підтримує). Інакше можна сказати, що бізнес-логіка – це реалізація правил і обмежень операцій, що автоматизуються.

**Представлення або Вид (View).** В обов'язки Представлення входить відображення даних отриманих від Моделі. Однак, представлення не може напряму впливати на модель. Можна говорити, що подання має доступом «тільки для читання» до даних.

Представлення має такі ознаки:

- у представленні реалізується відображення даних, які виходять від моделі будь-яким способом;
- у деяких випадках, представлення може мати код, який реалізує деяку бізнес-логіку.

Приклади представлення: HTML-сторінка, WPF форма, форма Windows Forms.

**Контролер** – це інтерфейс між моделлю, видом, що з нею асоційований та пристроями вводу. Контролери також відповідає за взаємодію з іншими «вид-контролер», вони відслідковують зв'язані з ними дані та реалізують систему повідомлень для передачі цих даних іншим представленням, коли це необхідно. В такий спосіб один вид може виступати, наприклад меню, може виступати як пристрій вводу для іншого.

Ознаки контролера:

- контролер визначає, які представлення мають бути відображені в даний момент;
- події представлення можуть вплинути тільки на контролер. контролер може вплинути на модель і визначити інше представлення;
- можливо декілька представлень тільки для одного контролера.

Контролер перехоплює подію ззовні і відповідно з закладеною в нього логікою, реагує на цю подію змінюючи Модель, за допомогою виклику відповідного методу. Після зміни Модель використовує подію про те що вона змінилася, і всі підписані на це події Представлення, отримавши його, звертаються до Моделі за оновленими даними, після чого відображають їх.

Прикладом використання MVC є такі програмні фреймворки як Spring, для платформи Java та ASP.NET для платформи Microsoft .NET. MVC лежить в основі їх ядра і визначає всі базові особливості цих технологій.

### ***Model-View-Presenter***

Даний підхід дозволяє створювати абстракцію представлення. Для цього необхідно виділити інтерфейс представлення з певним набором властивостей і методів. Презентер, в свою чергу, отримує посилання на реалізацію інтерфейсу, підписується на події вистави і за запитом змінює модель. Кожне представлення має реалізовувати відповідний інтерфейс. Інтерфейс подання визначає набір функцій і подій, необхідних для взаємодії з користувачем. Презентер повинен мати посилання на реалізацію відповідного інтерфейсу, яку зазвичай передають в конструкторі. Логіка представлення повинна мати посилання на примірник презентера. Усі події подання передаються для обробки в презентер і практично ніколи не обробляються логікою представлення. Приклад використання: Windows Forms. Дане розширення архітектурного паттерна MVC дозволяє вирішити деякі проблеми масштабованості програм, що мають класичну MVC-архітектуру.

Згідно з парадигмою HMVC, кожна окрема MVC тріада використовується в якості шару в ієрархічній структурі. При цьому, кожна тріада в цій ієрархії незалежна від інших, і може звернутися до контролера іншої тріади. Такий підхід істотно полегшує і прискорює розробку великих програм, полегшує їх подальшу підтримку та масштабування, сприяє повторному використанню коду.

### ***Model-View-ViewModel***

Дозволяє зв'язувати елементи представлення з властивостями і подіями View-моделі. Можна стверджувати, що кожен шар цього паттерну не знає про існування іншого шару.

MVVM полегшує відокремлення розробки графічного інтерфейсу від розробки бізнес логіки (бек-енд логіки), тобто моделі. Модель представлення являє собою частину, яка відповідає за перетворення даних для їх подальшої підтримки і використання. З цієї точки зору, модель представлення більше схожа на модель, ніж на представлення і оброблює більшість, якщо не всю, логіку відображення даних. Модель представлення може також реалізовувати паттерн медіатор, організовуючи доступ до бек-енд логіки навколо множини правил використання, які підтримуються представленням.

Ознаки View-моделі:

- двостороння комунікація з представленням;
- view-модель – це абстракція представлення та означає, що властивості представлення збігаються з властивостями view-моделі / моделі;
- view-модель не має посилання на інтерфейс представлення (iView). зміна стану view-моделі автоматично змінює представлення і навпаки, оскільки використовується механізм зв'язування даних (bindings).
- один екземпляр view-моделі пов'язаний з одним відображенням.

При використанні цього паттерну, представлення не реалізує відповідний інтерфейс (IView). Представлення повинно мати посилання на джерело даних (DataContext), яким в даному випадку є View-модель. Елементи представлення пов'язані (Bind) з відповідними властивостями і подіями View-моделі. В свою чергу, View-модель реалізує спеціальний інтерфейс, який використовується для автоматичного оновлення елементів представлення. Прикладом такого інтерфейсу в WPF може бути INotifyPropertyChanged.

Даний паттерн широко використовується разом при розробці додатків для Silverlight, Windows Phone, WPF і навіть починають з'являтися JavaScript бібліотеки, що дозволяють використовувати цей шаблон в HTML додатках.

Реалізація MVVM і MVP-паттернів, на перший погляд, виглядає досить простою та подібною. Однак, для MVVM зв'язування представлення з View-моделлю здійснюється автоматично, а для MVP – необхідно програмувати

контролер, що мабуть, дає більше можливостей з управління представленням. Тому, потрібно визначити деякі правила щодо вибору архітектурного паттерну.

MVVM використовується в ситуації, коли можливе зв'язування даних без необхідності введення спеціальних інтерфейсів представлення (тобто відсутня необхідність реалізовувати `IView`). Прикладом є технологія WPF.

MVP використовується в ситуації, коли неможливо зв'язування даних (не можна використовувати `Binding`). Прикладом – використання Windows Forms.

HMVC доцільне використовувати, коли велике значення мають характеристики масштабування розроблюваної системи.

MVC використовується в ситуації, коли зв'язок між представленням та іншими частинами додатку неможливий.

Згідно з викладеними вище рекомендаціями, для реалізації наукових результатів в програмному забезпеченні можна зробити **висновок**, що на цей час серед всіх різновидів паттерну Model-View-Controller найбільш рекомендованим для застосування виявляється Model-View-ViewModel. Такий вибір зумовлений зручністю використання методів зв'язування даних, тому що вони усувають необхідність створення спеціальних інтерфейсів для взаємодії окремих частин системи. В наслідок цього значно зменшується кількість елементів, що входять до складу системи, що значно її спрощує. Також важливою властивістю даного паттерну є те, що всі три його компоненти можуть функціонувати автономне, тобто незалежно один від одного. Завдяки цим особливостям підвищуючи показники таких характеристик системи, як надійність та відмовостійкість.

Через це, останні роки шаблон MVVM став стандартом для проектування більшості програмних продуктів, що мають складний графічний інтерфейс. Але, не зважаючи на велику актуальність та популярність даного паттерну, він являються відносно новою концепцією тому до сих пір не було вироблено чітких правил і підходів щодо його

реалізації. Як наслідок, існує достатня кількість прийомів, за допомогою яких досягаються принципи MVVM в розроблюваній системі. Всі вони мають свої недоліки та переваги по відношенню один до одного та часто бувають, або несумісні, або взаємовиключними. Тому, можна зробити висновок, що існує необхідність у вивченні цих методів та відношень між ними.

### **Аналіз існуючих підходів щодо побудови програмного забезпечення на базі архітектурного шаблону Model-View-ViewModel**

Отже, для аналізу підходів за допомогою яких реалізується шаблон, спочатку звернемо більш детальну увагу на теоретичні аспекти MVVM, цілі які переслідує дана концепція, та які результати вона приносить. Відомо, яку користь може принести розділення даних, якими оперує програма від її інтерфейсу. Але, часто лише такого простого розділення виявляється недостатньо, особливо при масштабуванні програмної системи.

В більшості проектів Види містять багато виділеного коду, який обробляє та передає дані у Вид. Щоб працювати в такий спосіб, код повинен мати детальну інформацію про сам Вид. Але, можна отримати багато користі виділивши логіку поведінки Представлення в окремий від нього клас. Таким чином, такий клас буде повністю незалежним від Виду і виступатиме для нього, як простий постачальник даних та операцій. Іншими словами, чітке розмежування між візуальною частиною Виду та його поведінкою, дає змогу отримати кращий дизайн. Це мета створення паттерну MVVM.

Одна із найбільших проблем, які виникають коли вся чи більшість логіки поведінки Представлення прописані в його виділеному коді, це те, що ця логіка просто не піддається тестуванню. Переплітання логіки та візуального представлення означає, що вони мають широкі знання один про одного. Такий двосторонній зв'язок робить тестування поведінки Виду доволі важкою, чи навіть, неможливою задачею.

Навіть якщо в проекті не використовуються юніт-тести або й зовсім не підіймається питання щодо тестування, впровадження архітектури MVVM в додатку, все одно привносить численні корисні якості:

- дає змогу створювати комбіновані проекти, включаючи можливість легко забезпечувати види даними на етапі проектування. Таким чином, дизайнер може бачити як виглядає інтерфейс заповнений даними без необхідності компіляції та запуску проекту;

- покращує спільний робочий процес дизайнера та програміста, зменшуючи ймовірність ситуацій, коли обидві сторони працюють одночасно над візуальним виглядом, поведінкою і логікою представлення, що може породжувати конфлікти;

- дає потенційну можливість повторного використання логіки представлення в його різноманітних реалізаціях;

- дозволяє застосовувати різні види для однієї моделі-виду, тобто можна мати декілька варіантів візуального відображення елемента, які будуть мати однакову логіку поведінки.

- навіть якщо в проекті не застосовуються юніт-тести, реалізація MVVM дає змогу їх впровадження на більш пізніх етапах, без необхідності значної реструктуризації, чи й зовсім без неї.

Отже, архітектурний паттерн MVVM складається з трьох фундаментальних рівнів:

- Модель.
- Вид (Представлення).
- Модель-виду.

Кожен рівень служить для вирішення задач, що надходять з інтерфейсу користувача. «Вид» технічно повинен бути обізнаний лише про Модель-виду, а Модель-виду повинна знати лише про Модель (Моделі).

**Вид (View).** Вид складається з двох частин: опису компонентів за допомогою деякої мови розмітки та виділеного коду, який ними керує.

Реалізація візуального відображення Виду не впливає на реалізацію самого паттерну, тому ця задача повністю покладається на дизайнера.

В термінах шаблону MVVM, можна говорити про Вид, як про споживача даних та операцій, що постачає Модель-виду. Теоретично, при використанні MVVM, обсяг класу виділеного коду Виду повинен бути незначним. Замість цього, логіка і поведінка Виду повинна розташовуватися в класі Моделі-виду, залишаючи виділеному коду виконання лише його базових завдань. Такий підхід називають «нульовим виділенням кодом». Однак, нульовий виділений код не являється метою шаблону MVVM і зазвичай цього нереально досягти. Деякі задачі просто легше виконувати у виділеному коді, особливо, якщо ці задачі потребують прямої взаємодії з Видом. Рішення перенести виконання такого типу задач в клас на рівень Моделі-виду принесе збільшення складності всього коду і зменшення його здатність до корегування, що в результаті суперечить тим цілям, заради яких реалізується паттерн MVVM. Таким чином, при проектуванні відповідно до шаблону MVVM, слід намагатися мінімізувати використання виділеного коду, але не уникати його повністю.

**Модель** – це об'єкт, що зазвичай містить дані отримані з серверу, бази даних чи введені користувачем, які Вид може відобразити. Вона також може містити такі елементи, як валідація вводу, відслідковування змін в даних, обробка правил за якими функціонує додаток та інші операції, зв'язані з даними системи.

**Модель-виду** – використовується для підтримки стану Виду та має два головних призначення: постачати дані та операції Виду і його обробляти логіку та поведінку. Добре спроектована Модель-виду повинна постачати дані залежні від стану замість візуально-специфічних даних.

Рівні зв'язуються вони між собою наступним чином.

Модель та Модель-вид не мають складних зв'язків. Зазвичай, Модель-виду бере на себе задачу передачі даних з серверу сервісному агенту, який

передасть їх назад Моделі-виду після їх обробки. Після цього Модель-виду може зберегти посилання на Модель і передати його у Вид.

Для з'єднання Моделі-виду і Виду, зазвичай екземпляр Моделі-виду приписується до властивості DataContext (контекст даних) Виду. Це дозволяє властивостям компонентів Виду напряму прив'язуватися до властивостей Моделі-виду, чи моделі, яку вона представляє.

Існує декілька підходів до створення екземплярів (інстанціювання) Моделі-виду і приписування їх до властивості контексту даних Виду, включаючи наступні:

- інстанціювання Моделі-виду безпосередньо в описі Виду;
- інстанціювання Моделі-виду у виділеному коді Виду;
- інстанціювання Моделі-виду окремо і передача її як параметр конструктора Виду;
- використовувати локатор Моделі-виду (поширена утиліта, що входить в склад різноманітних MVVM фреймворків, що здійснює пошук специфічного екземпляру класів).

Забезпечення зв'язків між Видом і Моделлю-виду є лише частиною роботи. Після того як Вид зв'язується з Моделлю-виду, йому потрібно отримувати дані та операції, які представляє Модель-вид. Якщо Модель-вид виставляє дані як публічні властивості, Вид може використовувати біндинги для прив'язування до цих властивостей і отримання даних. Це ефективно забезпечує двосторонній зв'язок, вносячи межу між Видом та Моделлю-виду. Таким чином, Вид може використовувати дані виставлені Моделлю-виду і оновлювати Модель-виду відповідно будь-яким змінам, які користувач вніс в дані через Вид, завдяки такому двонаправленому зв'язуванню.

Операції які передають логіку і поведінку від Модулі-виду до Представлення можуть бути втілені як за допомогою публічних методів, так і з використанням команд.

**Команди** використовуються для інкапсуляції частини логіки, до якої можуть прив'язатися різноманітні компоненти Виду і виконуватись у

відповідь на подію, наприклад натиснення кнопки. Команду можна представити у Модулі-виду як публічну властивість і прив'язати до неї властивість `Command` у Виді. Коли над компонентом буде здійснена відповідна дія, команда буде виконана.

**Методи** це загальне поняття для передачі операцій з класу відповідно до принципів об'єктно-орієнтованого програмування. За допомогою ще однієї парадигми ООП – рефлексії, можливе прив'язування Виду до конкретного методу Модулі-виду, з подальшим викликом цього методу у відповідь на події компонентів Виду.

Серед розробників немає чіткого рішення який із підходів для передачі операцій є кращим – використання команд чи методів. Команди можуть інкапсулювати частину логіки і виділити її з Моделі-виду таким чином, що ця логіка може бути повторно використана багато разів, але команди потребують написання більшого об'єму коду для своєї реалізації, ніж методи. З іншого боку, методи легко реалізувати, але для їх виклику потрібно використовувати спеціальні утиліти, які дозволяють знаходити методи по їх опису у Виді, але для вони потребують обчислювальні ресурсів, що зменшує ефективність роботи ПЗ. До того ж, оголошення методів в описі Виду, збільшує його об'єм та складність.

Також існує інша сторона взаємодії, а саме: як Модель-виду повідомляє Вид про те, що виникла якась подія на яку йому, можливо, потрібно відреагувати. Більшість класів Моделі-виду повинні мати певний набір базових функцій. Їм, зазвичай, потрібно реалізовувати програмний інтерфейс `INotifyPropertyChanged`, мати осмислене ім'я, можливість бути закритими (тобто, бути прибраними з інтерфейсу). Вирішення цієї задачі, як правило, покладається на один, чи декілька, базових класів, від яких новий клас Модулі-виду може успадковувати всю його базову функціональність. Таким чином, утворюється ієрархія спадкування класів Моделі-виду.

Таким же чином можна групувати і компоненти Виду. Елементи, що знаходяться на одному рівні ієрархії можуть легко взаємодіяти між собою,

так як знаходяться в зоні видимості і мають доступ до публічних членів класу один одного. Також, дочірні компоненти мають повний доступ до материнських, що забезпечують висхідну комунікацію. З іншого боку, при такій організації структури, взаємодія елементів з різних рівнів і гілок ієрархії стає складною задачею. Тому, грамотне розподілення компонентів Виду по дереву ієрархії може стати запорукою ефективною розробки та роботи ПЗ.

Таким чином, актуальним є завдання – оцінити ефективність роботи програмної системи при використанні в її розробці різних методів проектування з наступним визначенням методики оптимального (раціонального) використання шаблону проектування Model-View-ViewModel. Дане завдання доцільно формалізувати наступним чином. Нехай шаблон проектування ПЗ формально описаний кортежем

$$\langle I, Z, R, R_z \rangle, \quad (4.1)$$

де  $I$  – ім'я паттерну, за допомогою якого його можна ідентифікувати;  
 $Z$  – задача, що описує в яких випадках потрібно використовувати даний паттерн;

$R$  – вирішення, тобто опис того, яким чином досягається вирішення задачі  $Z$ ;

$R_z$  – результати та наслідки реалізації даного шаблону.

Тоді, доцільно ввести описання множини метрик шаблону

$$M = \{ m_1, m_2, \dots, m_n \}, \quad (4.2)$$

де  $n$  – кількість метрик.

Варте нагадати, що метрика програмного забезпечення – це міра, що дозволяє отримати числове значення деяких властивостей ПЗ та його специфікацій. Таким чином, метрика шаблону – це показник якості ПЗ.

Звідси, отримаємо математичну формалізацію завдання.

Знайти

$$G(V_i R_j) : \begin{cases} m_l \rightarrow \max(\min); l = \overline{1, n} \\ \sigma(V_i R_j) \in \langle I, Z, R, R_z \rangle \end{cases} \quad (4.3)$$

де  $G(V_i R_j)$  – модулі архітектурного шаблону;

$\sigma(V_i R_j) \in \langle I, Z, R, R_z \rangle$  – формалізація обмежень.

В роботі модулем архітектурного шаблону прийнято називати програмну, змінювану частину.

### **Оцінка ефективності реалізації архітектурного шаблону**

Відоме, що якість програмного забезпечення – характеристика програмного забезпечення або ступінь відповідності ПЗ до вимог. При цьому вимоги можуть трактуватись по-різному, що породжує декілька незалежних визначень терміну. Якість ПЗ – набір властивостей продукту (сервісу або програм), що характеризують його здатність задовольнити встановлені або передбачувані потреби замовника. Поняття якості має різні інтерпретації залежно від конкретної програмної системи і вимог до неї.

Найбільш розповсюдженим міжнародним стандартом є ISO/IEC 9126 «Програмування. Якість продукту». Основною його частиною являється Частина 4. «Якість при використанні показників» [123]. Цей стандарт регламентує зовнішні і внутрішні характеристики якості. Перші відображають вимоги до функціонування програмного продукту.

Для кількісного встановлення критеріїв якості, за якими буде здійснюватися перевірка і підтвердження відповідності ПЗ заданим вимогам, визначаються відповідні зовнішні вимірювані властивості (зовнішні атрибути) ПЗ, метрики (наприклад, час виконання окремих компонентів), діапазони зміни значень і моделі їх оцінки. Метрики використовуються на

стадії тестування або функціонування і називаються зовнішніми метриками. Вони являють собою моделі оцінки атрибутів.

Внутрішні характеристики якості і внутрішні атрибути ПЗ використовуються для складання плану досягнення необхідних зовнішніх характеристик якості продукту. Для квантифікації внутрішніх характеристик якості застосовують внутрішні метрики, як інструмент перевірки відповідності проміжних продуктів внутрішнім вимогам до якості, які формулюються на процесах, що передують тестуванню.

Зовнішні і внутрішні характеристики якості відображають властивості самого ПЗ (працюючого або не працюючого), а також погляд замовника і розробника на таке ПЗ. Безпосереднього кінцевого користувача ПЗ цікавить експлуатаційна якість ПЗ – сукупний ефект від досягнення характеристик якості, що вимірюється строком результату, а не властивістю самого ПЗ. Це поняття ширше, ніж будь-яка окрема характеристика (наприклад, зручність використання або надійність).

Стандарт [123] пропонує використовувати для опису внутрішнього та зовнішнього якості ПЗ багаторівневу модель. На верхньому рівні виділено 6 основних характеристик якості ПЗ. Кожна характеристика описується за допомогою кількох вхідних у неї атрибутів. Для кожного атрибута визначається набір метрик, що дозволяють його оцінити. Однак, варто зазначити, що критерії оцінки якості визначені в стандарті [123], не підходять повною мірою для оцінки якості архітектурних шаблонів. Це пов'язано з тим, що паттерни не являють собою програмну систему, а є лише технологією, набором методів за допомогою яких проектується ПЗ. Але, відповідно до вимог шаблону MVVM його реалізація повинна мати конкретну структуру, яка має спільні характеристики з іншими об'єктно-орієнтованими програмами. Тому, можна стверджувати, що є можливість виділити ті характеристики стандарту оцінки якості, які є застосовними до архітектурних шаблонів. Для цього проаналізуємо детальніше характеристики та їх

атрибути, при цьому дамо їх трактування у контексті MVVM, що може дещо відрізнятися від їх трактування в стандарті.

**Функціональність** – найбільш відповідальна характеристика ПЗ. Її важко об'єктивно формалізувати і оцінювати в системному проєкті. Вона зв'язана з тим, які функції і задачі вирішує ПЗ для задоволення потреб користувача, в той час як інші конструктивні характеристики головним чином зв'язані з тим, як і при яких умовах, задані функції можуть виконуватися з необхідною якістю. Субхарактеристики і атрибути функціональності можна характеризувати в основному властивостями, категоріями і якісним описом функцій, для яких зазвичай важко визначити чисельні міри та шкали. Тому вони віднесені в групу номінальних, категорійне-описових характеристик.

В більшості випадків функціональність найбільш залежить від коректності та надійності ПЗ. Значні труднощі при створенні орієнтирів для вибору мір і шкал характеристик якості, проявляються при аналізі коректності (точності), здатності до взаємодії і захищеності ПЗ. Ці субхарактеристики, важко звести до кількісних мір і їх приходить оцінювати по наявності властивостей і ряду типових процедур в ПЗ чи по величині необхідних затрат ресурсів, що достатньо помітно впливають на функціональність.

**Функціональна придатність** – це здатність вирішувати потрібний набір задач. Так як паттерни проєктування описують метод вирішення певної задачі, то можна стверджувати, що функціональна придатність архітектурного шаблону завжди буде максимальною.

**Точність (коректність)** – це здатність шаблону забезпечувати правильні чи прийнятні результати і зовнішні ефекти для користувачів. Еталонами для вибору вимог для коректності при проєктуванні можуть бути: верифікаційні та взаємозв'язані вимоги до функцій системи, компонентів і модулів програми, а також правила їх структурної побудови, організація взаємодій та інтерфейсів. Ці вимоги при розробці повинні бути простежені

зверху вниз до модулів, і використовуватися як еталони при встановленні необхідної коректності відповідних компонентів. Дане поняття включає забезпечення очікуваними даними з необхідною точністю розрахункових значень у відповідності з вимогами технічного завдання та специфікацій. В процесі проектування і розробки модулів та груп програм застосовуються окремі структурні критерії коректності, які включають коректність структури програм, обробки даних і міжмодульних інтерфейсів. Кожен з окремих критеріїв може характеризуватися декількома методами виміру якості і максимальним показником коректності програм: детерміновано, схоластично, чи в реальному часі.

Вибір потрібного рівня коректності ПЗ при системному проектуванні полягає у встановленні ступеню покриття тестами сукупності функцій та маршрутів виконання модулів і всього комплексу програм для подальшого процесу верифікації і тестування. Для визначення цієї величини при розробці ПЗ необхідна організація регулярної реєстрації та накопичення імен і складу функцій та маршрутів, що пройшли тестування, а також контроль долі не протестованих від всієї сукупності. Мірою вибраної коректності може бути відносне число протестованих функцій та маршрутів, яке може вимірюватися у відсотках від загального числа виконуваних. Частково цю величину при відповідній автоматизації та кваліфікації спеціалістів відображає трудомісткість і тривалість тестування, що безпосередньо впливає на функціональність ПЗ.

***Здатність до взаємодії*** – полягає у властивості шаблону та його компонентів взаємодіяти з одним або більшим числом певних компонентів внутрішнього і зовнішнього середовища. При виборі та встановленні при проектуванні здатності програмних та інформаційних компонентів до взаємодії, її можна оцінювати об'ємом технологічних змін в ПЗ, які необхідно виконати при доповненні чи виключенні деяких функцій або компонентів, коли відсутні зміни операційного, апаратного чи користувацького середовища. З цим показником зв'язана коректність та

уніфікація міжмодульних інтерфейсів, що визначаються двома видами зв'язків: по управлінню та по інформації.

Здатність до взаємодії компонентів шаблону між собою і з зовнішнім середовищем визначається ступенем уніфікації їх інтерфейсів та відповідністю стандартам. Ця характеристика полягає в описі властивостей і практично не впливає на якість функціонування поточної версії ПЗ. Її властивості проявляються в основному при модифікації комплексу програм. Степінь уніфікації інтерфейсів може вимірюватися їх відносним числом або об'ємом тексту (наприклад у відсотках від об'єму програми), які підлягають зміні при будь-яких коректуваннях програми. На функціональність, здатність до взаємодії впливає в тих випадках, коли ПЗ повинна піддаватися регулярним адаптаціям чи модифікаціям силами користувачів при зміні зовнішнього середовища.

При реалізації шаблону MVVM здатність до взаємодії доцільно пов'язати з використанням більш ніж одного суперечливих підходів, щодо побудови окремих аспектів паттерну. Так як ця субхарактеристика на пряму впливає на функціональність всієї системи, потрібно детально проводити її оцінку і в разі низьких показників приймати заходи щодо оптимізації системи, оскільки в іншому випадку, при досягненні ПЗ низької гнучкості, знадобиться значна витрата ресурсів для внесення змін в систему.

**Захищеність** – властивості ПЗ захищати програму та інформацію від негативних впливів описуються зазвичай складом та номенклатурою методів і засобів, що використовуються в ПЗ для захисту від зовнішніх та внутрішніх загроз. В критеріях захисту та забезпечення безпеки для конкретних ПЗ, зосереджуються різноманітні характеристики, котрі в ряді випадків важко чи неможливо описати кількісно, і при виборі доводиться оцінювати експертно. Непрямим показником її якості може служити відносна доля обчислювальних ресурсів, які використовуються безпосередньо засобами захисту.

Такі види обчислювальних ресурсів при забезпеченні безпеки використовуються для:

- контролю та корекції спотворення інформації, що надходить від джерела даних;
- оперативного контролю та виявлення дефектів виконання програм і обробки даних при використанні інформаційної системи за прямим призначенням;
- розміщення та забезпечення функціонування використовуваних засобів захисту від всіх видів загроз безпеки ІС;
- генерації тестових наборів або зберігання тестів для контролю працездатності, збереження та цілісності ПЗ при функціонуванні ІС;
- накопичення та зберігання даних про виявлені інциденти, спроби несанкціонованого доступу до інформації, про дефекти, збої та відмови в процесі виконання програми та обробки даних, що впливають на безпеку;
- реалізації процедур аналізу та моніторингу виявлених дефектів і оперативного відновлення обчислювального процесу, програм і даних (реєстру) після виявлення дефектів та відмов функціонування ІС.

Однією із головних задач, які вирішує паттерн MVVM – це забезпечення захисту даних. На додачу до базових механізмів безпеки даних об'єктно-орієнтованої парадигми – інкапсуляції за допомогою модифікаторів доступу, MVVM визначає розділення даних на три принципові групи. Це забезпечує захист споріднених даних від випадкового чи несанкціонованого доступу ззовні. Також в системі взаємодії Моделі-виду та Виду визначені механізми валідації даних та перевірка помилок. Це гарантує врегулювання випадків некоректного вводу даних користувачем чи помилок при передачі даних в Вид. Таким чином, базова захищеність ПЗ на базі MVVM має доволі високий показник і тому, її визначенням можна знехтувати при оцінці функціональності системи.

**Надійність** – це сукупність атрибутів, які визначають здатність ПЗ перетворювати вихідні дані на результати за умов, що залежать від

періоду часу життя (зношування і старіння не враховуються). Зниження надійності ПЗ відбувається через помилки у вимогах, проектуванні й виконанні. Відмови й помилки в програмах з'являються на визначеному проміжку часу.

Надійність ПЗ найбільш повно характеризується стійкістю або здатністю до безвідмовного функціонування і відновлюваністю працездатного стану після збоїв та відмов. В свою чергу стійкість залежить від ступеня покриття тестами функцій і структури програми, від рівня не встановлених дефектів та помилок (завершеності) та від здатності ПЗ реагувати на їх прояви так, щоб це не відобразалося на показниках надійності. Останні, визначаються ефективністю контролю даних, що надходять із зовнішнього середовища та від засобів виявлення аномалій функціонування ПЗ. В реальних умовах по різноманітним причинам, вхідні дані можуть потрапляти в області значень, що не перевірені при розробці та випробуваннях, а також не задані вимогами специфікації і технічного завдання, і викликають збої та відмови. При цьому некоректна програма може функціонувати цілком надійно.

**Завершеність** – це властивість ПЗ не потрапляти в стан відмов внаслідок помилок та дефектів в програмах і даних. Кількість або щільність проявів прихованих дефектів та помилок безпосередньо віддзеркалюється на тривалості нормального функціонування системи між відмовами. Завершеність можна характеризувати напрацюванням (тривалістю) на відказ (при відсутності автоматичного відновлення – рестарту), що вимірюється зазвичай годинами. На цю субхарактеристику впливають тільки відмови, внаслідок проявлених дефектів. Вони можуть бути обумовленими не повним тестовим покриттям при випробуваннях компонентів та ПЗ в цілому, а також недостатньою завершеністю тестування їх функцій.

**Стійкість до відмов** – властивість ПЗ автоматично підтримувати заданий рівень якості функціонування при проявах дефектів та помилок або порушення встановленого інтерфейсу. Для цього в ПЗ повинна вводитися

часова, програмна та інформаційна надлишковість, яка реалізує оперативне виявлення дефектів та помилок функціонування, їх ідентифікацію і автоматичне відновлення (рестарт) нормального функціонування ПЗ. Ефективне, оперативне усунення проявлення дефектів, помилок і некоректної взаємодії з операційним та зовнішнім середовищем визначають субхарактеристику – стійкість комплексів програм. Відносна доля обчислювальних ресурсів, що використовуються безпосередньо для швидкої ліквідації наслідків відмов та оперативного відновлення нормального функціонування ПЗ (рестарт) відображається на підвищенні надійності програм.

Як і у випадку завершеності стійкість до відмов являється зовнішньою характеристикою якості. Тому, її неможливо використовувати для оцінки якості шаблону.

**Здатність до відновлення** – властивість ПЗ у випадку відмови відновлювати необхідний рівень якості функціонування, а також пошкоджені програми та дані. Після відмови, ПЗ іноді буває непрацездатною на протязі деякого часу, тривалість якого визначається її відновлюваністю. Для цього необхідні обчислювальні ресурси і час на виявлення непрацездатного стану, діагностику причин відмови, а також на реалізацію процесів відновлення. Основними показниками процесу відновлення являються його тривалість і ймовірнісні характеристики. Відновлюваність характеризується також повнотою відновлення нормального функціонування програм в процесі ручного чи автоматичного їх перезапуску – рестарту. Перезапуск повинен забезпечувати відновлення нормального функціонування ПЗ, на що необхідні апаратні ресурси і час, які можна характеризувати відносною величиною (відсоток від загальних ресурсів). Повнота та тривалість відновлення після збоїв та відмов визначає надійність ПЗ і її функціональну придатність для використання по прямому призначенню.

**Доступність (готовність)** – властивість ПЗ бути в змозі виконувати необхідну функцію в даний момент часу при заданих умовах використання.

Ззовні, доступність може оцінюватися відносним часом, протягом якого ПЗ знаходиться в працездатному стані, в пропорції до загального часу застосування. Звідси, доступність – комбінація завершеності (від якої залежить частота відмов), стійкості до помилок та відновлюваності, які в сукупності обумовлюють тривалість простою для відновлення після кожної відмови, а також тривалості напрацювання на відмову. Для визначення цієї величини вимірюється відносний час працездатного стану програми між послідовними відмовами. Узагальнення характеристик відмов та відновлення виконується в критерії «коефіцієнт готовності». Цей показник виражає ймовірність мати відновлені програми і дані в працездатному стані в довільний момент часу.

Нижня границя шкали атрибутів надійності відображена значенням, при якому різко зменшується функціональна придатність, і використання даного типу ПЗ стає незручним, небезпечним або нерентабельним. Прикладом таких найгірших, граничних величин для багатьох класів ПЗ можуть бути напрацювання на відмову менш ніж десять годин, коефіцієнт готовності нижче 0,9 та час відновлення більше десяти хвилин. З іншого боку, найкращі значення цих атрибутів практично обмежені тими ресурсами, які можуть бути виділені для їх досягнення при розробці та експлуатації. Обчислювальні та програмні ресурси на безпосереднє забезпечення надійності функціонування ПЗ зазвичай знаходиться в діапазоні від 10% до 90%, причому останні значення відповідають критичним, особливо для високонадійних систем. Навіть для таких критичних програмних засобів рідко напрацювання на відмову перевищує декілька тисяч годин, коефіцієнт готовності не вище 0,999, а час відновлення при відмовах не менше декількох секунд. Перераховані параметри для конкретних проектів можуть вибиратися у вказаних діапазонах, в залежності від призначення і функцій програми, приближаючись до їх верхніх чи нижніх меж, з урахуванням впливу на функціональну придатність і доступних ресурсів на підтримку надійності.

Доцільне зробити висновок, що архітектурний шаблон MVVM не забезпечує надійність ПЗ. Визначне значення для надійності має реалізація обробки даних, а також наявність дефектів в коді. Помилки можуть бути виявлені в будь-якому модулі ПЗ, не залежно від архітектури за якою вона побудована. Тому, забезпечення надійності повністю визначається методами написання коду та обробки даних.

**Ефективність.** В стандарті [123] відображені дві субхарактеристики якості – *часова ефективність* і *використання ресурсів ЕОМ*, які рекомендується описувати, в основному кількісними, атрибутами, що характеризують динаміку функціонування компонентів ПЗ. В цій стандартизованій характеристиці відображається тільки окрема конструктивна ефективність використання ресурсів ЕОМ, яку не слід змішувати з системною ефективністю функціональної придатності ПЗ при застосування в конкретній інформаційній системі.

**Часова ефективність** – властивості ПЗ, що характеризують потрібний час відгуку та обробки завдань, а також продуктивність вирішення задач з урахуванням кількості використовуваних обчислювальних ресурсів в установлених умовах. Ці ресурси можуть включати інші програмні продукти, апаратні засоби, засоби телекомунікації і т.п. Часова ефективність ПЗ визначається тривалістю виконання заданих функцій та очікування результатів в середніх і-або найгірших випадках, з урахуванням пріоритетів задач. Вона залежить від швидкості обробки даних, що безпосередньо впливає на інтервал часу завершення конкретного обчислювального процесу, і від пропускної здатності – продуктивності, тобто від числа завдань, які можна реалізувати на даній ЕОМ в заданий інтервал часу. Ці показники якості тісно зв'язані з дисципліною диспетчеризації та часом реакції (відгуку) ПЗ на завдання при вирішенні різноманітних, функціональних задач. Величина цього часу залежить від тривалості вирішення сукупності задач центральним процесором ЕОМ, від затрат часу на обмін із зовнішньою пам'яттю, на введення та виведення даних та від тривалості очікування в

черзі до початку рішення задачі. Ця субхарактеристика тісно зв'язана з тривалістю обробки типового завдання, а також з інтервалом часу вирішення типових чи функціональних задач, що найбільш часто викликаються даною ПЗ. Пропускна здатність програми на конкретній ЕОМ виражається числом повідомлень чи завдань на вирішення певних задач, що обробляється за одиницю часу, яка залежить від характеристик середовища. Також вона залежить від функціонального вмісту ПЗ та його конструктивної реалізації, тим самим може розглядатися як один із внутрішніх показників якості програми.

**Використання ресурсів** – степінь завантаження доступних обчислювальних ресурсів протягом заданого часу при виконанні функцій ПЗ у встановлених умовах. Ресурсна економічність виражається зайнятістю ресурсів центрального процесору, оперативної, зовнішньої та віртуальної пам'яті, каналів вводу-виводу, терміналів та каналів мереж зв'язку. Ця величина визначається структурою і функціями ПЗ а також архітектурними особливостями та доступними ресурсами ЕОМ. В залежності від конкретних особливостей ПЗ і ЕОМ при виборі атрибутів може домінувати або величина абсолютної зайнятості ресурсів різних видів, або відносна величина використання ресурсів кожного виду при нормальному функціонуванні ПЗ. Ресурсна економія впливає не тільки на вартість вирішення функціональних задач, але й зазвичай, особливо для вбудовуваних систем, визначає принципіальну можливість повноцінного функціонування конкретної ПЗ в умовах реально обмежених обчислювальних ресурсів. Незважаючи на швидкий зріст доступних ресурсів пам'яті та продуктивності ЕОМ, часто переганяють їх технічне збільшення, і задача оцінки ефективного використання обчислювальних ресурсів залишається актуальною.

Якісним аналізом з урахуванням впливу на функціональну придатність можна визначити граничні значення для основних атрибутів конструктивної характеристики – ефективність. Використання обчислювальних ресурсів пам'яті та продуктивності ЕОМ для кожної з функціональних задач може

складати декілька відсотків. При нераціональних потоках завдань на вирішення основних, функціональних задач ПЗ необхідні деякі резерви пам'яті та потужностей ЕОМ, що визначає раціональне значення використовуваних ресурсів до 80-90% від максимальних значень. Атрибут часової ефективності – час відгуку на завдання користувача, безпосередньо залежить від вирішуваних функціональних задач і в загальному випадку може встановлюватися в діапазоні від 0,1 секунди до декількох десятків секунд. Ці значення залежать від динамічних характеристик об'єктів зовнішнього середовища, для яких вирішуються функціональні задачі ПЗ. У відповідності з цими значеннями часу відгуку може бути встановлений діапазон для атрибуту – пропускна здатність по числу типових завдань, що виконуються за одиницю часу. Ця величина в загальному випадку може бути в діапазоні 1-1000 завдань за хвилину.

Використання програмою ресурсів середовища вкрай важливий показник якості, тому при реалізації будь-якого шаблону потрібно звертати на нього увагу, тому можна стверджувати, що ефективність є однією із головних характеристик для оцінки ефективності паттерну.

**Практичність** – властивості ПЗ, які відображають складність її розуміння, вивчення та використання, а також привабливість для користувачів. В число користувачів можуть бути включені оператори, кінцеві та непрямі користувачі, які знаходяться під впливом або залежать від якості функціонування ПЗ. В практичності слід враховувати всі різноманітні характеристики зовнішнього середовища користувачів, на які може впливати ПЗ, включаючи необхідну підготовку до використання та оцінки результатів функціонування програми. Практичність використання – поняття достатньо суб'єктивне і важко формалізується, однак в результаті зазвичай значним чином визначає функціональну придатність та корисність застосування ПЗ. В цю групу показників входять атрибути, що з різних сторін відображають функціональну зрозумілість, зручність освоєння або простоту використання.

Деякі субхарактеристики можна оцінювати економічними показниками – затратами праці та часу спеціалістів на реалізацію відповідних функцій.

Так як концепція архітектурного шаблону покликана, перш за все, спростити роботу розробника ПЗ, і цим самим підвищити його її ефективність, практичність може стати визначною характеристикою оцінки якості паттерну.

**Зрозумілість** – властивість ПЗ, що забезпечує користувачу розуміння, чи являється програма придатною для його цілей, і як її можна використовувати для конкретних задач та умов використання. Зрозумілість залежить від якості документації та суб'єктивних вражень від функцій та характеристик ПЗ. Її можна описати якісно чіткістю функціональної концепції, широтою демонстраційних можливостей, повнотою, комплексністю та наочністю представлення можливих функцій і особливостей їх реалізації в експлуатаційній документації. Вона повинна забезпечуватись коректністю і повнотою опису вхідної та результуючої інформації, а також всіх деталей функцій ПЗ для користувачів. Крім того, ця субхарактеристика повинна відражати розпізнання параметрів, що можуть модифікуватися і адаптивність програм до конкретного середовища та умовам використання користувачами.

**Простота використання** – можливість користувачу зручно і комфортно експлуатувати та керувати ПЗ. Аспекти змінюваності, адаптивності та легкості інсталяції можуть бути передумовами для простоти використання та вибору конкретного ПЗ, а у випадку розробника програми – вибору конкретного шаблону. Характеризується рядом динамічних параметрів: часом введення та відгуком на завдання, тривалістю вирішення типових задач, часом на реєстрацію результатів, які перекриваються з атрибутами субхарактеристики часової ефективності.

Простоту використання інформаційних систем, значною мірою, характеризує коректність і адекватність опису інтерактивних директив управління, об'єм та час введення завдань, і час очікування користувачем

результатів при їх виконанні. На етапі розробки, тобто при використанні паттернів проектування, це може виражатися в часі компіляції та іншої обробки введених розробником коду і даних. Простота використання може узагальнено оцінюватися якісно шкалами з двома-чотирма категоріями. Такий же метод найбільш адекватний для оцінки комфортності експлуатації та простоти управління функціями ПЗ. Однак деякі атрибути цієї субхарактеристики доступні для більш повної кількісної оцінки шляхом вимірювання трудомісткості і тривалості відповідних процесів підготовки та навчання кваліфікованих користувачів (розробників) до повної та ефективної експлуатації ПЗ (шаблону).

**Зручність навчання** – властивість ПЗ, що забезпечує зручне освоєння її застосування достатньо кваліфікованим користувачам. В контексті архітектурних шаблонів проектування ця субхарактеристика тісно зв'язана із зрозумілістю. Вона може трудомісткість та тривалість підготовки користувача (розробника) до повноцінної експлуатації ПЗ (паттерну). Атрибути даної субхарактеристики залежать від можливості попереднього навчання та від вдосконалення знань в процесі експлуатації, від можливості оперативної допомоги та підказки при використанні ПЗ, а також від повноти, доступності і зручності використання посібників та інструкцій по експлуатації. Якість вивчення ПЗ залежить від внутрішніх властивостей та складності програми, а також від суб'єктивних характеристик кваліфікації конкретних користувачів (розробників).

На значення зручності вивчення значно впливають демонстраційні можливості довідкових засобів навчання, якість та об'єм експлуатаційної документації, а також електронних посібників, які можна оцінювати відповідно по числу супроводжуючих сторінок документів або зайнятий посібниками об'єм пам'яті ЕОМ. Зручність навчання можна відражати трудомісткістю та протяжністю вивчення користувачами відповідної кваліфікації, методів та інструкцій застосування ПЗ для повноцінної експлуатації. Ці атрибути може характеризувати трудомісткість від одиниці

до сотень людино-годин і тривалість від одиниць до тисяч годин, необхідних для освоєння кваліфікованого застосування особливо складних програм. Вибір діапазонів доцільних значень цих характеристик можна здійснювати по відгукам користувачів про прецеденти освоєння ПЗ, добре підтримуваних засобів навчання. Звичайно, для створення навчальних посібників необхідні певні затрати праці та часу розробників, які в деякій мірі пропорційні складності функцій ПЗ.

**Привабливість** – суб'єктивна властивість ПЗ подобатися замовникам, покупцям і/або користувачам. Вона зв'язана із зовнішніми атрибутами, наочністю реклами та естетикою оформлення ПЗ, інтерфейсів з користувачами і експлуатаційної документації, що обумовлюють більшу чи меншу її привабливість для придбання і використання користувача споживачами. Так як ця субхарактеристика є зовнішньою вона не має відношення до оцінки якості архітектурного шаблону, і тому її можна виключити з ієрархії характеристик якості паттернів.

**Супроводжуваність (зручність супроводу)** – пристосованість ПЗ до модифікацій та змін модифікацій. Модифікації можуть включати виправлення, вдосконалення або адаптацію ПЗ до змін в зовнішньому середовищі застосування, а також вимогах та функціональних специфікаціях замовника. Простота і трудомісткість модифікацій визначається внутрішніми метриками якості програми, які відбиваються на зовнішній якості і якості використання, а також на складності управління конфігураціями версій ПЗ.

**Здатність до аналізу** – підготованість ПЗ до діагностики її дефектів чи причин відмов, а також до ідентифікації і виділення її компонентів для модифікації та виправлення. Ця субхарактеристика залежить від чіткості архітектури, уніфікації інтерфейсів, повноти і коректності технологічної і експлуатаційної документації на ПЗ. На здатність до аналізу впливає якість засобів контролю і моніторингу змін функціональних характеристик, а також дефектів і корегування програм і даних.

При реалізації архітектурного шаблону слід приділяти увагу можливій придатності до аналізу при виборі того чи іншого підходу, щодо реалізації. Тому що, якщо в результаті вибраного підходу структура проекту може стати сильно складною, це також ускладнює аналіз програми, а відповідно і впровадження в неї виправлень та модифікацій.

**Змінюваність** – пристосування ПЗ до простої реалізації специфікованих змін і до управління конфігурацією. Реалізація модифікацій включає проектування, кодування та документації змін. Для цього потребується певна трудомісткість та час, що зв'язані з виправленням дефектів і/або модернізацією функцій, а також із зміною процесів експлуатації. При виборі атрибутів цієї характеристики слід враховувати вплив структури, інтерфейсів та технічних особливостей ПЗ. Змінюваність залежить не тільки від внутрішніх властивостей ПЗ, але й також від організації та інструментальної оснащеності процесів супроводження і конфігураційного управління, на які орієнтується архітектура, зовнішні та внутрішні інтерфейси програми. Якщо ПЗ повинен модифікувати кінцевий користувач, змінюваність може бути представлена передумовою і частиною простоти використання.

Цілком закономірно, що ПЗ буде змінюватися під час її проектування та розробки. Тому, при виборі підходів реалізації архітектурного шаблону слід приділяти увагу, як цей вибір вплине на змінюваність системи, і приймати на основі цього остаточне рішення.

**Стабільність** – здатність ПЗ до запобігання і мінімізації непередбачені негативні ефекти від змін, можливість локалізувати і обмежувати область впливу змін в програмі та даних. Ця внутрішня субхарактеристика визначається архітектурою ПЗ, уніфікованістю її інтерфейсів, коректністю технологічної документації і може суттєво впливати на функціональність, надійність та адекватність застосування програми при її змінах.

**Зручність тестування** – властивість ПЗ, що забезпечує простоту перевірки якості змін і приймання модифікованих компонентів програм. Ця

субхарактеристика залежить від величини області впливу змін, які необхідно тестувати при модифікаціях програми і даних, від складності тестів для перевірки їх характеристик. Її атрибути залежать від чіткості правил структурної побудови компонентів і всієї програми, від уніфікації міжмодульних та зовнішніх інтерфейсів, від повноти та коректності технологічної документації. Можливість локалізації змін та уніфікація інтерфейсів компонентів з частиною ПЗ, що не корегується, дозволяє знизити складність, трудомісткість і тривалість тестування, спрощує підготовку тестів та аналіз результатів. В цій субхарактеристиці враховується, в основному, технічна і організаційна складові процесу тестування модифікацій і не входить функціональна частина їх підготовки. Узагальнено її можна оцінювати затратами праці та часу на тестування деяких середніх по об'єму та складності модифікацій програми.

Субхарактеристики зручності аналізу та стабільності у складі супроводжуваності якісно характеризуються атрибутами близькими до атрибутів практичності: чіткістю архітектури програми, уніфікацією інтерфейсів, повнотою і коректністю документації. Для цих субхарактеристик може застосовуватися елементарна порядкова шкала.

**Переносимість** – це підготовленість ПЗ до переносу з одного апаратно-операційного середовища в інше. Переносимість програм і даних на різноманітні апаратні та операційні платформи являється важливим показником функціональної придатності для багатьох сучасних ПЗ. Ця властивість може оцінюватися об'ємом, трудомісткістю та тривалістю необхідних доробок компонентів та операцій по адаптації, які слід виконувати для забезпечення повноцінного функціонування ПЗ після перенесення на іншу платформу. Мобільність може здійснюватися, на рівні вхідних текстів програм на мові програмування або на рівні об'єктного коду, що виконується ЕОМ. Вона залежить від структурованості та розширюваності програми і даних, а також від наявності додаткових

ресурсів, необхідних для реалізації переносимості та модифікації компонентів при їх переносі.

**Простота установки-інсталяції** – властивість ПЗ до простого впровадження (інсталяції) в новому апаратному і операційному середовищі замовника або користувача. Якщо ПЗ повинна встановлюватися кінцевим користувачем, легкість встановлення буде передумовою для зручності використання. Так само як адаптованість вона може вимірюватись трудомісткістю і тривалістю процедур встановлення, а також мірою задоволення вимог замовника та користувачів до характеристик і складності інсталяції.

**Відповідність** – здатність ПЗ співіснувати та взаємодіяти з іншими незалежними програмними засобами в спільному обчислювальному середовищі, розділяючи спільні ресурси. Ця субхарактеристика залежить від ступені стандартизації інтерфейсів ПЗ з операційним і апаратним середовищем застосуванням, від сумісності функцій і даних та може оцінюватися експертно.

**Зручність заміни** (зміщуваність) – пристосованість кожного компоненту ПЗ до відносно простого використання замість іншого виділеного та вказаного замінюваного компоненту. Зміщуваність не припускає, що замінюваний компонент ПЗ здатний повністю виконувати функції попереднього компоненту. Велику роль для цієї властивості грає чітка структурованість архітектури і стандартизації зовнішніх та внутрішніх інтерфейсів ПЗ. Ця властивість впливає на трудомісткість і тривалість заміни в основному комплексних ПЗ.

Як відомо, елементи архітектурних шаблонів знаходяться на доволі високому рівні абстракції. Тому, особливості операційного та апаратного середовищ не можуть впливати на загальну структуру ПЗ, яка проектується на базі конкретного архітектурного паттерну. Зважаючи на це можна виключити характеристику мобільності з класифікації оцінки якості архітектурного шаблону.

І останнє, на що потрібно звернути увагу при оцінюванні ПЗ – це **відповідність різноманітним стандартам якості**. Нажаль, так як архітектурний шаблон MVVM не є стандартизованим, то неможливо використовувати дані субхарактеристики при оцінці його якості.

Таким чином, проаналізувавши модель оцінки якості ПЗ та провівши порівняння її характеристик відповідно до концепції архітектурного шаблону, можна сформулювати модель оцінки якості архітектурного шаблону, зображену на рис. 4.1.



Рис. 4.1. Модель оцінки якості архітектурного шаблону

### **Аналіз метрик для оцінки ефективності використання архітектурного шаблону MVVM**

Як вже було зауважено, для оцінки даних атрибутів та характеристик використовують спеціальні метрики. Проаналізуємо відомі визначення метрики та дамо те, що досліджується в роботі.

В математиці існує таке поняття як **метричний простір** — це пара  $(X, d)$ , яка складається з деякої множини  $X$  елементів і відстані  $d$ , визначеної для будь-якої пари елементів цієї множини.

Виходячи з цього, **метрика** – це міра деякого метричного простору.

**Метрика програмного забезпечення** — це міра, що дозволяє отримати числове значення деяких властивостей програмного забезпечення та його специфікацій. Кількісні методи оцінювання добре показали себе в інших сферах науки, а тому багато теоретиків та практиків в галузі інформаційних технологій, спробували перенести цей підхід в розробку програмного забезпечення. В загальному випадку застосування метрик дозволяє визначити складність розробленого проекту, або проекту, що перебуває у розробці, оцінити об'єм робіт, стилістику розроблюваного проекту і зусилля, витрачені кожним розробником для реалізації того чи іншого рішення, однак метрики можуть служити лише рекомендаційними характеристиками.

Набір метрик, що використовуються для оцінки якості програмного забезпечення, складається з:

- кількісні метрики коду (кількість пустих рядків програми, відсоток коментарів, середнє число рядків для функцій, класів або файлів, середнє число рядків для модулів);
- порядок зростання (мається на увазі аналіз алгоритмів, в термінах теорії складності обчислень);
- кількість рядків коду;
- цикломатична складність – метрика програмного забезпечення, розроблена Томасом Мак Кабе. Використовується для оцінки складності програм. Обчислює кількість лінійно незалежних шляхів в алгоритмі роботи програми на основі її вихідних текстів;
- ООП - метрики коду;
- аналіз функціональних точок;
- кількість помилок на рядок коду;
- ступінь покриття коду тестуванням;
- покриття вимог;
- кількість класів та інтерфейсів;

- метрики від Роберта Сесіль Мартіна;
- зв'язність – це міра в якій модуль програми залежить від кожного іншого модуля;
- пов'язаність – це міра того наскільки пов'язаним є код в одному модулі програми (наприклад через спільну семантику);
- час завантаження програми;
- час виконання програми;
- розмір бінарних файлів.

З цього списку можна виділити окремих підклас метрик – ООП-метрики коду. Так як, архітектурний шаблон MVVM застосовується саме для розробки об'єктно-орієнтованих програм, ці метрики найбільш повно оцінюють якість даного паттерну. Тому розглянемо їх детальніше з метою застосувати їх для оцінки підходів до реалізації MVVM-шаблону.

**Глибина дерева спадкування** (DIT – Depth of Inheritance Tree). Глибиною класу в ієрархії спадкування є максимальна довжина від вузла класу до кореня дерева, і вимірюється в предках класу. Чим глибше успадкування класу в ієрархії, тим більшу кількість методів він, можливо, успадковує. І тим більш непередбачуваним є поведінка класу. Для даної метрики відсутнє чітке визначення контексту. Велика частина ланцюжка спадкування може бути розташована всередині використовуваної бібліотеки класів, в самому додатку може бути розташована її менша частина. Слід розрізняти ієрархії спадкування, що виникають в результаті розширення класу і в результаті реалізації інтерфейсу. Слід також розрізняти в ієрархії класи стандартної бібліотеки, класи сторонніх бібліотек і класи власне додатка. Граничним оптимальним значенням цієї метрики рекомендується вважати 5 або 6.

**Кількість нащадків** (Number of Children – NOC) – кількість безпосередніх нащадків класу. Чим більше таких нащадків, тим більшого тестування вимагає даний клас. Оскільки при обчисленні метрики

враховується тільки безпосередні нащадки, то оцінити якість ієрархії успадкування дана метрика не дозволяє.

**Кількість методів** (Number of Methods - NOM) визначених локально в класі, без урахування видимості класу. Враховуються також і перевизначені методи. Дана метрика добре підходить для оцінки складності класу.

**Кількість зважених методів на клас** (Weighted Methods Per Class – WMC) є сума цикломатична складність методів визначених у цьому класі.

**Цикломатична складність** (Cyclomatic Complexity Metric –  $V(G)$ ) обчислюється як максимальне число лінійно незалежних ділянок коду в методі. Ділянка коду лінійна, якщо не містить команд розгалуження. Значення метрики обчислюється таким чином

$$V(G) = e - n + p, \quad (4.4)$$

де  $n$  - кількість вершин,  $e$  - кількість ребер,  $p$  - кількість з'єднаних компонент (вузлів виходу) у графі потоку управління.

Ця метрика є індикатором психологічної складності розуміння програмного коду.

Метрики зчеплення (coupling) призначені вимірювання залежності різних класів один від одного.

**Зчеплення між класами об'єктів** (Coupling Between Object classes - CBO). Два класи зчеплені один з одним, якщо один клас використовує інший клас. Наприклад, метод одного класу викликає метод іншого класу, або використовує атрибути іншого класу. Зчепленням також вважається успадкування одного класу від іншого класу, або поліморфний виклик іншого класу. Значення цієї метрики для класу – кількість класів з яким він пов'язаний. При обчисленні цієї метрики використовуються лише класи безпосередньо пов'язані один з одним. Класи з надмірним зчепленням складніше розуміти, змінювати і коригувати. Метрику CBO слід відрізнити від метрики відцентрового зчеплення (підраховуються тільки вхідні

залежності) і метрики доцентрового зчеплення (підраховуються лише вхідні залежності). Лімітом оптимального значення для одиничного члену ПЗ є  $CVO = 9$ .

**Коефіцієнт зчеплення (Coupling Factor - CF).** Область дії цієї метрики - не конкретний клас, а програма в цілому. Коефіцієнт зчеплення обчислюється як нормалізоване співвідношення між кількістю клієнтських відношень і загальним числом можливих клієнтських відношень. Клієнтське відношення існує, якщо клас посилається на метод або атрибут іншого класу, за винятком випадку, коли клас є нащадком іншого класу. Таким чином, зчеплення через успадкування виключається, але виклики поліморфних методів класів-предків враховуються.

**Зчеплення посилкою повідомлень (Message Passing Coupling - MPC).** Значення метрики – кількість повідомлень, що посилаються, визначених у класі. Виклик власних методів класу при цьому не враховується. Інтерпретація метрики наступна: кількість викликів методів зроблених з класу дозволяє виміряти наскільки реалізація методів даного класу залежить від методів в інших класах. Поліморфізм методів при цьому не враховується.

Метрики **зв'язаності (cohesion)** класів дозволяють оцінити якість проектування, аналізуючи зв'язки всередині класу, а не зв'язки між класами, як це робилося за допомогою метрик зчеплення. Зв'язки всередині класу виникають, зокрема, через зв'язки між методами класу. Зв'язаність методів в класі бажана, оскільки це сприяє інкапсуляції зв'язаних методів в одному класі. Відсутність пов'язаності методів у класі означає, що можливо даний клас слід перепроєктувати, розщепивши його на два і більше підкласів.

Набір метрик **«Відсутність зв'язаності в методах»** [124] (Lack of Cohesion in Methods - LCOM) використовується для обчислення зв'язаності коду.

Метрика **LCOM1** представляє кількість пар методів в класі, які не посилаються на загальний атрибут. Недоліком методу є використання для

обчислення значення метрики тільки власних атрибутів і методів класу. Ця метрика враховує тільки власні методи і атрибути класу.

**LCOM2** – для кожної пари методів в класі, якщо вони мають доступ до різної множини екземплярів змінних, значення змінної  $P$  збільшується на 1, в іншому випадку збільшується значення змінної  $Q$ . Якщо  $P$  більше  $Q$ , то значення метрики

$$LCOM2 = P - Q. \quad (4.5)$$

В іншому випадку значення метрики  $LCOM2 = 0$ . Значення 0 означає, що клас зв'язаний. В іншому випадку клас може бути розщеплений на два і більше класів, оскільки їх екземпляри змінних належать різним множинам.

**LCOM3** – обчислює кількість з'єднаних компонент в графі методів. Методи в графі з'єднані, якщо методи мають доступ до того ж атрибуту.

**LCOM4** – ця метрика обчислюється на основі кількості з'єднаних компонент в графі методів. Методи в графі з'єднані, якщо методи мають доступ до того ж атрибуту. Ця метрика, на відміну від **LCOM3**, приймає до уваги транзитивний виклик графа.

$LCOM4 = 1$  означає зв'язаний клас.

$LCOM4 > 2$  означає, що цей проблемний клас повинен бути розділений на менші класи.

$LCOM4 = 0$  означає, що в класі немає методів.

**LCOM5** обчислюється за формулою

$$LCOM5 = \frac{NOM - \frac{\sum_{m \in M} NOAcc(m)}{NOA}}{NOM - 1}, \quad (4.6)$$

де  $M$  – множина методів класу;

$NOM$  – кількість методів;

$NOA$  – кількість атрибутів;

NOAcc ( $m$ ) – кількість атрибутів класу доступних методу  $m$ .

Значення метрики LCOM5 нормалізовано (змінюється в межах  $[0,1]$ ). Завдяки нормалізованим значенням можливе порівняння різних версій вимірюваних класів.

**Щільність зв'язаності класу** [124] (Tight Class Cohesion - TTC). Значення цієї метрики – нормалізоване співвідношення між кількістю безпосередньо зв'язаних через атрибути методів до загальної кількості з'єднань між методами. Безпосереднє з'єднання між методами існує, якщо обидва методи мають відношення до атрибуту безпосередньо, або через виклик методу.

Максимальне число можливих з'єднань

$$NP = 0,5 * N / (N-1), \quad (4.7)$$

де  $N$  – кількість видимих методів.

$$TCC = NDC / NP, \quad (4.8)$$

де  $NDC$  – кількість безпосередніх з'єднань.

Метрика TCC приймає значення в межах  $[0,1]$ . Для цієї метрики розглядаються тільки видимі методи. Ігноруються конструктори і деструктори, а також методи, що реалізують інтерфейси або дії обробки. Ця метрика вимірює ступінь зв'язаності між видимими методами класу. Більше значення метрики означає більшу зв'язаність. Клас зі значенням метрики меншим 0,5 незв'язаний. Конструктори класів через непрямі з'єднання з атрибутами завищують значення метрики, ускладнюючи її точне вимірювання.

**Втрата зчеплення класом** [124] (Loose Class Cohesion – LCC). Значення цієї метрики – нормалізоване співвідношення між кількістю безпосередньо або опосередковано зв'язаних через атрибути методів до

загальної кількості з'єднань між методами. Непряме з'єднання між методами існує, якщо існує послідовний шлях від одного методу до іншого через послідовність прямих з'єднань між методами. Це визначається використанням транзитивного замикання графу, що визначається метрикою ТСС. Метрика ТСС обчислюється за формулою

$$LCC = (NDC + NIC) / NP, \quad (4.9)$$

де  $NDC$  – кількість безпосередніх з'єднань;

$NIC$  – кількість непрямих з'єднань;

$NP$  – максимальне число можливих з'єднань.

За визначенням  $LCC >$  або  $=$  ТСС.

Метрика ТСС приймає значення в межах  $[0,1]$ . Для цієї метрики розглядаються тільки видимі методи. Ігноруються конструктори і деструктори.

Метрика  $LCC$  вимірює загальну ступінь зв'язаності між видимими методами класу. Значення метрики  $LCC < 0,5$  означає незв'язаний клас.  $LCC = 0,8$  означає «сильно зв'язаний» клас.  $TCC = LCC$  означає, що клас має тільки прямі зв'язки.  $TCC = LCC = 1$  означає, що всі методи класу зв'язані.

Виділивши найбільш вагомi метрики з розглянутих вище, отримуємо наступну формалізацію оптимальної структури програми побудованої відповідно до архітектурного шаблону MVVM

$$G(V_i R_j) : \begin{cases} DIT, NOC, V(G) \rightarrow \min \\ CBO \leq 9, \\ LCOM4 = LCOM5 = TCC = LCC = 1 . \\ \sigma(V_i R_j) \in \langle I, Z, R, R_z \rangle \end{cases} \quad (4.10)$$

## **Методика оптимального використання архітектурного шаблону проектування програмного забезпечення Model-View-Viewmodel**

Аналіз програмних метрик, що використовуються для оцінки якості програмних систем дозволяє виділити з них ті, які доцільне застосувати для оцінки якості реалізації архітектурного шаблону. Саме, за допомогою цих метрик, можливо отримати адекватну оцінку ефективність будь якого підходу до реалізації паттерну і таким чином вибрати з них ті, що забезпечують найбільшу ефективність та якість для додатку, що розробляється.

**1 етап.** В першій групі суперечливих підходів до реалізації архітектурного шаблону MVVM стоїть питання про розміщення певних частин логіки поведінки компонентів Виду в класі Моделі-виду, проти їх розміщення безпосередньо у виділеному коді цього Виду.

Теоретично вся логіка поведінки Виду повинна знаходитися в його моделі, яка буде делегувати цю логіку через операції. Ідеальною, в цьому випадку буде ситуація досягнення програмою «нульового виділеного коду». Але вимоги до програмної системи можуть диктувати такі умови, коли частині коду, що описує деяку логічну операцію необхідний прямий доступ до компонентів, з яких складається Вид, що керується цією логікою. В таких випадках, при розміщенні цього коду в Моделі-виду, доведеться затратити багато ресурсів на організацію доступу коду до Моделі-виду до компонентів Виду. Вирішенням даної проблеми може бути перенесення такої логіки в область виділеного коду, що керує компонентами Виду, і в області видимості якого є прямий доступ до всіх компонентів. Це значно спрощує взаємодію логіки та компонентів Виду, але разом з цим робить цю логіку залежною від конкретного екземпляру Виду.

Для вирішення подібних суперечливих ситуацій доцільне виміряти зв'язаність, яку забезпечуватиме дана частина коду, при розміщенні її в тому чи іншому контексті. Для цього слід використовувати групу метрик: LCOM4,

LCOM5, TCC, LCC. Чим більше їх сумарне значення тим вища зв'язаність забезпечується перенесенням коду у відповідний модуль програми.

Таким чином, у програмі забезпечуються високі показники декількох показників якості: зрозумілість та зручність аналізу, класи, що мають велику зв'язаність своїх методів та атрибутів характеризуються простотою, тому подібний програмний код легко зрозуміти, а отже він також легко піддається аналізу для виявлення дефектів та внесення змін; зручність тестування – коли взаємодія між класами не є надмірною, трудомісткість тестування значно зменшується, так як зменшується кількість прямих та неявних зв'язків між цими класами і потенційних непередбачуваних помилок, що виникають при такій взаємодії. Також це забезпечує інкапсуляцію операцій і даних всередині одного класу, а тестування замкнутої системи є найефективнішим.

**2 етап.** Вибір методу ініціалізації екземплярів Моделі-виду. Для цього існує два основних підходи: оголошення екземпляру в описі Виду або його безпосередня ініціалізація в основному коді програми.

У випадку опису екземпляру всю роботу по ініціалізації беруть на себе спеціальні інструменти розробки, що виконують пошук відповідного класу Моделі виду за його назвою, виділення пам'яті для цільового екземпляру, внесення в нього початкових даних. Подібні операції відбуваються вже на етапі виконання програми, а отже на їх виконання витрачається певна кількість обчислювальних ресурсів ЕОМ. Ще одним недоліком даного підходу є потенційне виникнення відмов на етапі виконання додатку, якщо при описі компонентів Виду, розробником були допущені помилки. Для виявлення подібних помилок потрібно проводити додаткове тестування і перевірку описів Представлень. Також існують механізми для виявлення та виправлення подібних помилок та відмов, що через них виникли, вже під час виконання програми, але подібні операції теж потребують витрати програмних ресурсів. Все це прямим чином впливає на ефективність ПЗ, що проектується.

З іншого боку при ініціалізації екземплярів Моделі-виду безпосередньо в кодї програми, всі необхідні для цього дії розробнику потрібно виконувати власноруч. З одного боку це збільшує затрати праці розробника, але з іншого боку, при такому підході всі помилки допущені в кодї програми будуть виявлені на етапі компіляції, після чого їх доволі легко виправити. Також варто зауважити, що об'єм опису екземпляру Моделі-виду в розмітці Виду значно більший від об'єму коду, необхідного для виконання тих самих дій, так як розробнику або дизайнеру потрібно додати значну кількість службових символів і даних в опис компонентів Виду. Але при даному підході також можливе виникнення відмов на етапі виконання програми. Їх причиною можуть стати втрата невизначеність зв'язків екземплярів та введення некоректних даних. Ймовірність виникнення подібних ситуацій зростає при збільшенні кількості екземплярів Моделей-виду і масштабуванні проекту.

Таким чином обидва підходи потребують однакових затрат ресурсів на підтримку надійності підчас експлуатації програми. Але стосовно етапу розробки, ініціалізація екземплярів Моделі-виду в основному кодї програми дає кращі показники зрозумілості, зручності аналізу і тестування, при умові не надто великої кількості екземплярів Моделі-виду. При великих масштабах проекту, розробнику краще скористатися ініціалізацією екземплярів в описі Виду, що забезпечує автоматизацію управління створеними компонентами.

Зазвичай, кожен окремий клас Моделі-виду має лише один екземпляр і створення дублікатів є небажаною практикою. В такому випадку для всієї програми можна порахувати кількість нащадків (NOC) базового класу Моделі-виду, за допомогою інструментів рефакторингу інтегрованого середовища розробки (IDE). При значенні цього показника в декілька десятків, можна вважати дану програму простою і використовувати ініціалізацію всередині основного коду програми. Інакше, слід вдаватися до опису екземплярів через Вид. Обчислення даної метрики краще за все проводити ще на етапі проектування і робити на їх результатах відповідні

висновки. Це дозволить уникнути затрат праці на перепроєктування в майбутньому.

На основі попереднього вибору повинен здійснюватися вибір методу зв'язування Виду з його моделлю. При ініціалізації екземпляру класу Моделі-виду безпосередньо в коді програми, розробник має прямий доступ до цього екземпляру. Тому, зв'язування доступне через передачу цього об'єкту напряду до відповідного об'єкту Виду. Звичайно, при цьому потрібно мати доступ до цього Виду, а також описати в його виділеному коді певну логіку, що буде виконуватись при передачі їх екземпляру Моделі-виду для зв'язування з ним. В іншому випадку, при описі об'єкту Моделі-виду в самому Виді, за їх зв'язування будуть відповідати ті ж самі інструменти, що виконують ініціалізацію цього об'єкту. Як і у випадку з ініціалізацією, таке зв'язування виконується на етапі виконання програми і тому потребує затрат певної кількості обчислювальних ресурсів.

Тому в розрахунках ефективності роботи програми слід враховувати не тільки затрати ресурсів та часову ефективність ініціалізації об'єктів, але й ефективність зв'язування цих об'єктів з відповідним об'єктом Виду. Також варто зауважити, що інструменти, які виконують автоматичну ініціалізацію та зв'язування об'єктів використовують під час дуже складні алгоритми, що значно зменшує зрозумілість всієї системи та ускладнює аналіз збоїв та відмов, що можуть виникати під час виконання додатку. Отже, можна зробити висновок, що застосовувати описовий метод ініціалізації та зв'язування екземплярів Виду та його моделі слід тільки у випадку великомасштабних проєктів.

**3 етап.** Вибір підходу до передачі операцій від Моделі-виду до Виду. Це можна здійснювати за допомогою команд або методів. Команди можуть інкапсулювати частину логіки і відділити її від Моделі-виду таким чином, щоб цю частину можна було повторно використовувати безліч разів. Але реалізація команд потребує більшу кількість коду на своє описання ніж методи. З іншого боку методи забезпечують більш простий спосіб

виставлення операцій від Моделі-виду до Виду і також дає змогу повторного їх використання, але для реалізації такого підходу потрібно використовувати спеціальні інструменти для пошуку і виконання відповідних методів. Такі інструменти виконують виклик методів з опису Виду, для чого в ньому необхідно прописати відповідну службову інформацію, збільшення об'єму опису Виду підвищує його складність, тобто зменшується зрозумілість та зручність аналізу програми. До того ж, далеко не всі середовища розробки мають вбудовані побідні інструменти. Тому найбільш оптимальним способом виставлення операцій з Моделі-виду можна вважати використання команд. Хоч він і потребує більших затрат праці програмістів, але він забезпечує більшу зрозумілість та ефективність роботи програми в порівнянні з методами.

Іншою стороною взаємодії Виду з його моделлю є сповіщення Виду про виникнення різноманітних подій в Моделі та Моделі-виду, на які потрібно відреагувати.

Першим способом реалізації такого сповіщення є оголошення перехоплювачів подій у виділеному коді Виду та написання відповідного коду, що реалізує логіку обробки перехопленої події. Подібний підхід суперечить концепції «нульового виділеного коду», а також збільшує складність тестування програми, так як виділений код потрібно тестувати окремо і враховувати його можливий вплив на інші модулі програми.

Другим способом є реалізація тригера в середині самого Виду, що буде чекати виникнення події і реагувати на неї безпосередньо. Такий підхід виявляється більш ефективним, так як дозволяє розміщувати логіку поведінки Виду при реакції на подію в спеціально визначеному для цього місці – в його моделі. Таким чином, операції і дані інкапсулюються всередині однієї сутності, виключаючи роздроблення програми на безліч частин. Це, в свою чергу, спрощує тестування програми, так як немає необхідності в перевірці можливих непрямих впливів різних модулів однієї сутності один на

одного. Також значно підвищується зрозумілість програми для розробника, так як, всі основні функціональні частини коду знаходяться в одному класі.

**4 етап.** Забезпечення оптимальної ієрархії спадкування класів. Для її оцінки можна використовувється метрика глибини дерева спадкування – DIT. Високе значення DIT означає також високу ймовірність виникнення помилок, але низьке її значення означає низький показник повторного використання коду, а отже і надлишковість класу. Тому рекомендується розбивати логіку класів між відповідними нащадками, кожен з яких буде відповідати виконанню специфічних для нього задач, але потрібно дотримуватися граничного значення DIT, яке дорівнює 5 або 6. Такий ліміт забезпечує зменшення складності всієї системи, а отже і її зрозумілість та зручність тестування.

Таким чином, сформульовані рекомендації щодо практичної реалізації розроблених в дисертації наукових результатів – моделей та методів прогнозу технічного стану ЗВТ на основі м'яких обчислень у вигляді методики оптимального використання архітектурного шаблону проектування програмного забезпечення Model-View-Viewmodel.

#### **4.2. Перевірка достовірності запропонованих наукових результатів**

Оцінку ефективності прогнозу технічного стану ЗВТ доцільне здійснити на основі методу статистичних випробувань (Монте-Карло). Ідея методу полягає у здійсненні розіграшу – моделювання випадкового явища за допомогою деякої процедури, яка дає випадковий результат. У результаті такого розіграшу отримується одна реалізація випадкового явища. Виконавши такий розіграш велику кількість разів, можна отримати статистичний матеріал, який, у подальшому обробляється методами математичної статистики [17,24].

Випадковим фактором, який підлягає урахуванню при статистичному моделюванні прогнозу технічного стану ЗВТ під час кожної реалізації

випадкового числа окремих зовнішніх та зовнішніх факторів є результат прогнозу.

Основним елементом, із сукупності яких складається модель статистичних випробувань, є одна випадкова реалізація явища, яке моделюється. Якщо множина зовнішніх та внутрьшніх факторів, які впливають на технічний стан ЗВТ складається з  $N$  елементів, то необхідно буде одночасно кинути  $N$  одиничних жеребів, кожен з яких визначатиме факт існування елемента у працездатному стані на протязі життєвого циклу системи. Для запровадження такого жереба з використанням елементів комп'ютерної алгебри вводиться оператор генерації випадкового числа  $\text{Rand} [\zeta]$ , який при кожному запуску видає випадкову величину, розподілену з постійною щільністю на проміжку  $[0,1]$ . Тоді визначення елементів вектору  $X$  при реалізації системи може бути здійснено на основі

$$x_i = \begin{cases} 1, & \text{якщо } \text{Rand} [\zeta]_i < R_i, \\ 0, & \text{якщо } \text{Rand} [\zeta]_i \geq R_i, \end{cases} \quad (i = \overline{1, N}). \quad (4.11)$$

У (4.11) у якості  $R_i$  та використовуються деяке межове значення параметра контролю.

*Оцінка достовірності статистичного моделювання.* Метод Монте-Карло ґрунтується на граничних теоремах теорії ймовірностей, які стверджують, що при великій кількості дослідів  $N$  частота події наближається до її ймовірності, а середнє арифметичне спостережених значень випадкової величини – до її математичного очікування. У таких умовах виникає питання щодо оцінки достовірності характеристик випадкового явища отриманих даним методом, а також визначення необхідного числа дослідів (реалізацій) для того, щоб із заданою ймовірністю  $P^D$  можна було очікувати, що частота  $Q^*$  події, яка досліджується відхилиться від її ймовірності  $Q$  менш ніж на деяку величину  $\varepsilon$ .

У [12] показано, що необхідну кількість дослідів  $N^N$  можна визначити на основі співвідношення

$$N^N = \frac{Q(1-Q)}{\varepsilon^2} \left[ \Phi^{-1} \left( \frac{1}{2} P^D \right) \right]^2, \quad (4.12)$$

де  $Q$  – ймовірність досліджуваної події;  
 $\varepsilon$  – задане відхилення ймовірності;  
 $\Phi^{-1}$  – функція, обернена функції Лапласа;  
 $P^D$  – довірча ймовірність.

Так, наприклад, якщо встановити:  $Q = 0,05$ ,  $\varepsilon = 0,02$ ,  $P^D = 0,95$  то необхідна кількість реалізацій, визначених за (4.12), складе  $N^N = 456$ .

Основна мета заходів системи технічної експлуатації формує логіку оцінки ефективності розроблених в дисертації моделей та методів прогнозу по впливу на таку властивість як надійність. А саме, так як ці моделі та методи є підсистемою для забезпечення безвідмовної роботи в цілому ЗВТ або комплектуючих підсистем та елементів, то і оцінювати їх ефективність доцільно за тим впливом, який вони надають на безвідмовність. Дослідження на модельному прикладі показує, що застосування запропонованих моделей і методів дозволяє підвищити достовірність прогнозу технічного стану підсистем ЗВТ до 8-15 %. При цьому, за результатами математичного моделювання, можливе підвищення ймовірності виконання безвідмовної роботи до 10-12 % за рахунок впровадження на практиці.

#### **Висновки до розділу 4**

1. Аналіз стандартів оцінки якості програмних систем визначив основні риси моделі оцінки якості програмної реалізації моделей та методів прогнозу технічного стану ЗВТ з використанням паттерну MVVM. На основі побудованої моделі оцінки якості були обрана система програмних метрик,

за допомогою яких можна виміряти ефективність використання того чи іншого підходу до реалізації архітектурного шаблону, що досліджується в даній роботі.

2. Встановлено, що при виборі місця розміщення фрагментів логіки Виду між його моделлю та виділеним кодом, слід вибирати такий контекст, який має найбільший показник зв'язаності з даним фрагментом логіки.

3. Визначено, що при невеликих масштабах проекту більш ефективним підходом до ініціалізації екземплярів Моделі-виду та зв'язування їх з відповідним Представленням являється виконання цих дій безпосередньо в основному коді програми, але при збільшенні числа таких екземплярів більш ефективним виявляється метод їх оголошення в описі Виду.

4. Встановлено, що найбільш універсальним та ефективним підходом до передачі логічних операцій від Моделі-виду до Виду є використання команд, в той час як оптимальним методом отримання Видом подій, що відбулися в Моделі-виду або Моделі, є створення спеціального триггеру.

5. Визначено, що для забезпечення оптимальної ієрархії спадкування класів компонентів MVVM-паттерну слід дотримуватися граничного значення глибини дерева спадкування рівним 5-6.

6. Доцільне представити рекомендації щодо практичної реалізації розроблених в дисертації наукових результатів – моделей та методів прогнозу технічного стану ЗВТ на основі м'яких обчислень у вигляді методики оптимального використання архітектурного шаблону проектування програмного забезпечення Model-View-Viewmodel.

7. Аналіз показав, що оцінку ефективності запропонованих в роботі моделей та методів прогнозу технічного стану доцільне виконувати по впливу на складову надійності – безвідмовність. Це обумовлене тим що моделі та методи прогнозу є підсистемою для забезпечення безвідмовної роботи в цілому ЗВТ або комплектуючих підсистем та елементів, тому оцінювати їх ефективність доцільно за тим впливом, який вони надають на безвідмовність.

8. Дослідження на модельному прикладі показує, що застосування запропонованих моделей і методів дозволяє підвищити достовірність прогнозу технічного стану підсистем ЗВТ до 8-15 %. При цьому, за результатами математичного моделювання, можливе підвищення ймовірності виконання безвідмовної роботи до 10-12 % за рахунок впровадження на практиці.

## ВИСНОВКИ

В результаті дисертаційних досліджень, виконаних автором, вирішено нове актуальне наукове завдання удосконалення існуючих та розробки нових моделей та методів прогнозування технічного стану засобів водного транспорту на основі м'яких обчислень. Дане наукове завдання має важливе значення для теорії і практики підвищення ефективності контролю технічного стану транспортної техніки, встановлення закономірностей змінювання параметрів технічного стану у процесі експлуатації, впровадження методів і засобів діагностування та прогнозування технічного стану засобів транспорту, що забезпечують високу ефективність їх використання та надійність роботи. Відсутність аналогічних рішень в нашій країні і за кордоном робить результати досліджень пріоритетними.

На підставі проведених досліджень зроблені наступні висновки:

1. Доведено, що на теперішній час активне виконується державне завдання інтенсивного економічного та соціального розвитку країни та її участі в міжнародній інтеграції. Значна роль в цьому відводиться транспортній галузі, а саме, морським та річним перевезенням. Водний транспорт, як інфраструктурна галузь, має розвиватися швидкими темпами. У 2020-2022 роках, очікується збільшення обсягу перевезення вантажів морськими та річними судами до 2500 млн. тон; переробка вантажів у державних торговельних портах також збільшиться та становитиме близько 240 млн. тон; обсяги пасажирських перевезень становитимуть більш 10900 млн. пасажирів.

2. Дослідження, яке виконане в дисертації показує, що рівень безаварійного судноводіння, показники якості та ефективності перевезень пасажирів та вантажів продовжують покращуватися. Особлива увага приділяється саме заходам забезпечення заданого рівня надійності технічних засобів. Значна роль в цьому напрямку – рішення завдання прогнозування технічного стану засобів водного транспорту.

3. Аналіз основних напрямків підвищення ефективності прогнозу технічного стану засобів водного транспорту визначив те, що сучасним інноваційним напрямком покращення математичного та програмного забезпечення є м'які обчислення. Питанням прогнозування технічного стану засобів водного транспорту присвячено багато наукових та практичних робіт, але це не знижує актуальність досліджень в даному напрямку.

4. Основною тенденцією розвитку систем контролю та прогнозування технічного стану як суднового обладнання, так і ЗВТ в цілому вважається перехід від екстенсивного розвитку систем централізованого контролю до якісної зміни їх функціональних можливостей за рахунок застосування спеціальних діагностичних методик, алгоритмів та апаратури, адаптованих до специфічних особливостей об'єктів діагностування, тобто зроблене висновок про необхідність впровадження в практику експлуатації суден принципово нового класу інформаційних систем – інтелектуальних та інтелектуалізованих систем.

5. Доведена актуальність теми дисертації, яка визначається невідповідністю існуючих методів та способів прогнозування реальним умовам функціонування. Аналіз існуючих науково-обґрунтованих підходів підвищення ефективності складних технічних систем, до яких повною мірою відноситься й інтелектуальна транспортна система, дозволив зробити висновок про формування за останні роки нового пріоритетного підходу, пов'язаного із реалізацією принципів теорії штучного інтелекту – м'яких обчислень. Таким чином, загальне наукове завдання щодо розвитку моделей, методів, методик та алгоритмів прогнозу технічного стану ЗВТ на основі м'яких обчислень є важливим та актуальним для науки та практики.

6. Аналіз світового досвіду впровадження інтелектуальних технологій на водному транспорті свідчить о можливості значного підвищення ефективності, якості та достовірності прогнозу технічного стану ЗВТ. Таким чином, незважаючи на стрімкий розвиток теорії штучного інтелекту та м'яких обчислень взагалі, актуальним є наукове завдання якому присвячена дана дисертаційна робота.

7. Найбільш ефективною є стратегія удосконалення концептуальної моделі прогнозування технічного стану засобів водного транспорту, в алгоритмічному сенсі, яка поєднує OLAP-технологію інтелектуального аналізу даних; алгоритм нечіткої субтрактивної кластеризації для визначення кількості значень лінгвістичних змінних (рангу терм-множини); метод нечітких с-середніх (Fuzzy C-Means) для побудови функцій приналежностей, а також гібридні мережі, так звані адаптивні нейроні нечіткі мережі ANFIS для самоналаштування нечіткого висновку на вхідні умови. Аналіз наукових робіт в даній галузі дозволяє зробити висновок про високу ефективність саме такого комплексного застосування добре апробованих підходів.

8. Доведено те, що при удосконаленні методу формування функції приналежності в моделі прогнозування технічного стану ЗВТ доцільне використовувати нечітку кластеризацію, а саме: метод субтрактивної кластеризації для визначення кількості кластерів, яка інтерпретуються як ранг базової терм-множини лінгвістичної змінної та методу нечітких с-середніх для розрахунку значень функції приналежності; це дозволяє значно покращити адекватність процедури фазифікації та точність прогнозу.

9. Моделювання прогнозу технічного стану ЗВТ на основі використаних моделей нечіткої кластеризації свідчить про те, що розробка є функціональною, а математичне та програмне забезпечення задовольняє усім потребам користувача. Моделі є адекватними, а дане програмне забезпечення є досить надійним. Інтерфейс програми зручний та ергономічний. Завдяки використанню сучасних алгоритмів для розрахунків та відображення результату програмне забезпечення досить ефективне. Але програмне забезпечення є цілком не портативним, тому що воно розроблялось тільки для роботи на основі пакету MATLAB.

10. Аналіз показав, що оцінку ефективності розроблених в дисертації моделей та методів прогнозу доцільне виконати по впливу на таку властивість як надійність. А саме, так як ці моделі та методи є підсистемою для забезпечення безвідмовної роботи в цілому ЗВТ або комплектуючих

підсистем та елементів, то і оцінювати їх ефективність доцільно за тим впливом, який вони надають на безвідмовність.

11. Дослідження на модельному прикладі показує, що застосування запропонованих моделей і методів дозволяє підвищити достовірність прогнозу технічного стану підсистем ЗВТ до 8-15 %. При цьому, за результатами математичного моделювання, можливе підвищення ймовірності виконання безвідмовної роботи до 10-12 % за рахунок впровадження на практиці.

12. Таким чином, мета дослідження щодо підвищення достовірності прогнозу технічного стану засобів водного транспорту за рахунок впровадження запропонованих моделей та методів її досягнута і всі часткові завдання вирішені повністю. Наукові результати досліджень є внеском в розвиток наукових і методологічних основ підвищення ефективності прогнозування технічного стану транспортної техніки, встановлення закономірностей змінювання параметрів технічного стану у процесі експлуатації, впровадження методів і засобів діагностування та прогнозування технічного стану засобів транспорту, що забезпечують високу ефективність їх використання та надійність роботи, а також внеском в теорію автоматизованих систем контролю та прогнозування технічного стану даних засобів.

13. Основні результати дисертаційних досліджень можуть бути використані дослідно-конструкторськими організаціями та підприємствами при розробці або удосконаленні програмних засобів автоматизованих систем прогнозування технічного стану засобів водного транспорту.

14. Напрямами подальших досліджень у зазначеній галузі можуть бути ті, які стосуються створення та модифікація ерготичних інтелектуальних систем для автоматизації прогнозування технічного стану засобів водного транспорту.

**СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ**

1. Абрамов О.В., Розенбаум, А.Н. Прогнозирование состояния технических систем. Москва: Наука, 1990. 126 с.
2. Абрамов О.В. Контроль и прогнозирование технического состояния систем ответственного назначения. *Надежность и качество сложных систем*. 2018. № 4 (24). С. 108–115.
3. Акіменко В.В., Загородній Ю.В. Лабораторний практикум з основ проектування баз знань: Київ : КНУ, 2010. 140 с.
4. Аналіз стану безпеки руху, польотів, судноплавства та аварійності на транспорті в Україні за I півріччя 2017 р. URL: [http://dsbt.gov.ua/sites/default/files/imce/Bezpeka\\_DTP/analiz\\_2017/analiz\\_avariynosti\\_i\\_pivrichchya\\_2017.pdf](http://dsbt.gov.ua/sites/default/files/imce/Bezpeka_DTP/analiz_2017/analiz_avariynosti_i_pivrichchya_2017.pdf) (дата звернення 12.06.2019).
5. Анучин О. Н. Емельянцева Т.Н. Интегрированные навигационные системы ориентации и навигации для морских подвижных объектов. Санкт-Петербург: Государственный Научный Центр Российской Федерации. 1999. 357 с. URL: <https://www.twirpx.com/file/126419/> (дата звернення 12.12.2019).
6. Баранов Г. Л., Носовський А. М., Тихонов І. В. Функціональна стійкість навігаційного обслуговування безпеки судноплавства на внутрішніх водних шляхах: монографія. Київ: КДАВТ, 2012. 149 с.
7. Баранов Г.Л., Тихонов І.В. Концепція побудови функціонально стійкого навігаційного обслуговування об'єктів водного транспорту в зонах підвищеного ризику плавання. *Системи управління, навігації та зв'язку*. 2009. Вип.2(10). С. 17-21.
8. Баранов Г.Л., Тихонов І.В. Ефективність інтелектуалізації інтегрованих систем навігації і управління рухомими транспортними засобами. *Системи управління, навігації та зв'язку*. 2010. Вип. 1 С. 13-20.

9. Беллман Р. Заде Л. Принятие решений в расплывчатых условиях: сборник переводов. Москва: Мир, 1976. С. 172-215.
10. Белов П. Г. Запороженко Ю. Ф. Сущность и методы прогнозирования техногенного риска. *Вісник КМУЦА*. 1999. №1. С. 260–264.
11. Бень А.П. Методы оценки опасности траектории движения судов в системах поддержки принятия решений. *Вестник ХНТУ*: сб. науч. трудов Херсонского национального технического университета. 2009. Вып. 1 (34). С. 429-433.
12. Бенькович Е. С. Колесов Ю. Б., Сениченков Ю. Б. Практическое моделирование динамических систем: учебное пособие. Петербург, 2002. 464 с.
13. Боглаев Ю. П. Вычислительная математика и программирование: учебное издание. Москва: Высшая Школа, 1990. 546 с.
14. Богом'я В. І., Коломієць О. М. Методи підвищення ефективності процесу експлуатації судових комплексів. *Новітні технології*. 2017. Вип. 1(3). С. 42–48.
15. Вагущенко Л. Л. Цымбал Н. Н. Системы автоматического управления движением судна: учебное пособие. Одесса: Латстар, 2002. 310 с.
16. Вагущенко Л. Л., Цымбал Н. Н. Системы автоматического управления движением судна: учебное пособие. 3-е издание перераб.и доп. Одесса: Феникс, 2007. 376 с.
17. Вентцель Е.С. Исследование операций. Москва: Советское радио, 1972. 552 с.
18. Вычужанин В.В., Рудниченко Н.Д. Технические риски сложных комплексов функционально взаимосвязанных структурных компонентов судовых энергетических установок. *Вісник Одеського національного морського університету*. 2014. № 2. С. 68–77.
19. Гаврилова Т. А., Хорошевский В. Ф. Базы знаний интеллектуальных систем: учебник. Питер, 2000. 384 с.

20. Гаранов М. Ю., Шепета Ю. Н., Лебедев А. Е. Технология интегрирования оборудования рулевой рубки скоростных судов. *Морские информационно-управляющие системы*. 2014, № 1 (4). С. 112 – 115.
21. Гаскаров Д. В. Истомин Е. П., Кутузов О. И. Сетевые модели распределенных автоматизированных систем. Санкт-Петербург: Энергоатомиздат, 1998. 353 с.
22. Герасимов Б. М. Оксіюк О. Г., Шворов С. О. Проектування та застосування експертно-навчальних систем: монографія. Київ: Європейський університет, 2008. 263 с.
23. Герасимов Б. М. Тарасов В. А., Токарев И. В. Человеко-машинные системы принятия решений с элементами искусственного интеллекта. Київ: Наукова думка, 1993. 184 с.
24. Горбань І. І. Теорія ймовірностей і математична статистика для наукових працівників та інженерів . Київ : НАНУ. ППММС. 2003. 244 с.
25. Горошко К. О. Оцінка сучасного стану та перспективні шляхи розвитку внутрішнього водного транспорту України: збірник наукових праць ДЕГУТ. Сер. «Економіка і управління». Вип 26. Київ: ДЕГУТ, 2013. С. 169 – 173.
26. Груверман А. Технология создания мостиковых систем с применением интеграторов информации от разнородных датчиков. Алгоритмы распределенной обработки и регистрации полученных данных. *Управление и диагностика технических средств для судов различной класса автоматизации ЗАО «Транзас»*. URL: <https://docplayer.ru/31427735-Tehnologiya-sozdaniya-mostikovyih-sistem-s-primeneniem-integratorov-informacii-ot-raznorodnyh-datchikov.html> (дата звернення: 19.02.2019).

27. Довідник кваліфікаційних характеристик професій працівників. «Водний транспорт». Розділ «Морський транспорт». Міністерство транспорту України. 2002. Вип.67. 128 с.
28. ДСТУ 2389-94. Технічне діагностування та контроль технічного стану. Терміни та визначення. URL: <https://www.twirpx.com/file/2414014/> (дата звернення: 03.07.2019).
29. ДСТУ 2860-94. Надійність техніки. Терміни та визначення. URL: [https://dnaop.com/html/2273/doc\\_2860-94](https://dnaop.com/html/2273/doc_2860-94) (дата звернення: 03.07.2019).
30. Дюбуа Д., Прад А. Теория возможностей. Приложения к представлению знаний в информатике. Москва, 1990. 286 с.
31. Заде Л. А. Роль мягких вычислений и нечеткой логики в понимании, конструировании и развитии информационных интеллектуальных систем *Новости искусственного интеллекта*. Перевод с англ. И.З. Батыршина. 2001. № 2–3. С. 7–11.
32. Зайченко Ю. П. Нечеткие модели и методы в интеллектуальных системах: учебное пособие для студентов высших учебных заведений. Киев: Слово, 2008. 344 с.
33. Зайченко Ю.П. Основи проектування інтелектуальних систем. Киев: Слово, 2004. 352 с.
34. Закон України «Про транспорт» від 28.12.2015 р. №233/94-ВР. URL: <http://zakon2.rada.gov.ua/laws/show/232/94-вр> (дата звернення 12.12.2019).
35. Захаров И. Г. Проблемы комплексирования и интеграции боевых систем и технических средств при создании кораблей нового поколения. *Морская радиоэлектроника*. 2006. №3(17).
36. Інструкція з проведення технічного нагляду за суднами рибної промисловості України, що не підлягають нагляду класифікаційного товариства. URL: <https://zakon.rada.gov.ua/laws/show/z0749-17> (дата звернення 04.07.2019).

37. Интеллектуальные системы поддержки принятия решений в нештатных ситуациях с использованием информации о состоянии природной среды / Геловани В. А. Башлыков А. А., Бритков В. Б., Вязилов Б. Д. Москва: Эдиториал УРСС, 2001. 304 с.
38. Інтелектуальні системи підтримки прийняття рішень / Герасимов Б. М., Локазюк В. М., Оксикюк А. Г., Поморова О. В. Київ: Європейський університет, 2007. 335 с.
39. Качинский А. Б. Засады системного аналізу безпеки складних систем. Киев: ДП «НВЦ» Євроатлантикінформ, 2006. 336 с.
40. Клир Дж. Системология. Автоматизация решения системных задач. Москва: Радио и связь, 1990. 544 с.
41. Колесников А. В. Кириков И. А. Методология и технология решения сложных задач методами функциональных гибридных интеллектуальных систем. Москва: ИПИ РАН, 2007. 387 с.
42. Коломієць О. М., Тимощук О. М., Данік О. В. Выбор показателей надежности с учетом интенсивности эксплуатации судна. *Proceedings of Azerbaijan State Marine Academy*. 2017. №2. С. 90–95.
43. Кормен Т., Лейзерсон Ч., Ривест Р. Алгоритмы: построение и анализ. Москва: МЦМНО, 1999. 960 с.
44. Кофман А. Введение в теорию нечетких множеств. Москва: Мир, 1982. 432с.
45. Кристофидес Н. Теория графов. Алгоритмический подход. Москва: Мир, 1978. 402 с.
46. Критерій оптимальності процесу технічного обслуговування суднових комплексів/ Тимощук О.М., Коломієць О. М., Дакі О.А., Трофименко І.В. *Наука і техніка Повітряних сил Збройних сил України*. 2017.№4(29). С.132–136.
47. Круглов В. В., Борисов В. В. Искусственные нейронные сети. Теория и практика. Москва, 2001. 382 с.

48. Круглов В. В., Дли М. И., Голунов Р. Ю. Нечеткая логика и искусственные нейронные сети. Москва: Издательство физ.-мат. лит., 2001. 224 с.
49. Крыжановский Г. А., Купин В. В., Плясовских А. П. Теория транспортных систем: учебное пособие / под ред. Г. А. Крыжановского. С. Петербург, 2008. 208 с.
50. Кудрицька Н. В. Транспортнодорожній комплекс України: сучасний стан, проблеми та шляхи розвитку: монографія. Киев: НТУ, 2010. 338 с.
51. Кузнецов О. П., Адельсон-Вельский Г. М. Дискретная математика для инженера. Москва, 1988. 480 с.
52. Кунцевич В. М. Управление в условиях неопределенности: гарантированные результаты в задачах управления и идентификации. Киев: Наукова думка, 2006. 264с.
53. Леоненков А. В. Нечеткое моделирование в среде MATLAB и fuzzyTECH. Санкт- Петербург: БХВ-Петербург, 2005. 736 с.
54. Лорьер Ж.-Л. Системы искусственного интеллекта. Москва: Мир, 1991. 568 с.
55. Майстренко О., Гурченков О. Стан виробничого потенціалу суднобудування та перспективи його розвитку URL: [http://www.nbuu.gov.ua/portal/Soc\\_Gum/Ecan/2011\\_9\\_2/pdf](http://www.nbuu.gov.ua/portal/Soc_Gum/Ecan/2011_9_2/pdf) (дата звернення 12.12.2019).
56. Максимей И. В. Имитационное моделирование на ЭВМ. Москва: Радио и связь, 1988. 232 с.
57. Маринов М. Л., Клименко В. Д. Учет человеческого фактора в аварийных ситуациях на море. Эксплуатация морского транспорта. 2008. № 2. С. 25-29.
58. Митюшкин Ю. И., Мокин Б. И., Ротштейн А.П. Soft Computing: Идентификация закономерностей нечеткими базами знаний. Вінниця: УНІВЕРСУМ-Вінниця, 2002. 145 с.

59. Моделювання та оптимізація інформаційних систем в судноводінні: підручник / Кравченко Ю.В. та ін., за ред. О.М. Тимощук. Київ :ДУІТ, 2019, 312 с.
60. Мусорін О. О. Методи підвищення ефективності засобів й систем навігаційного обслуговування вантажних суден необмеженого району судноплавства: дис... к-та техніч. наук : 05.22.13/ ДП «УНДНЦ ПССЯ». Київ, 2017. 187 с.
61. Навігаційне забезпечення управління рухом суден : навчальний посібник / Богом'я В.І. та ін. Київ :ДВВП «Компас», 2012, 336 с.
62. Національної доповіді про стан техногенної природної безпеки в Україні у 2013 році. URL:[http://www.mns.gov.ua/files/prognoz/report/2013/2\\_5.pdf](http://www.mns.gov.ua/files/prognoz/report/2013/2_5.pdf) (дата звернення 12.12 2018).
63. Небылов А.В. Гарантирование точности управления. Москва: *Наука. Физматлит*, 1998. 304 с.
64. Нечаев Ю. И., Сизов В. Г. Принятие решений при управлении судном в экстремальных ситуациях на основе современной теории катастроф / Судовождение: сб. научн. трудов ОНМА. Одесса: «ИздатИнформ», 2010. Вып. 20. С. 130–142.
65. Нечеткие множества в моделях управления и искусственного интеллекта / Аверкин А. Н., и др.; Москва: Наука, 1986. 312 с.
66. Основи технічної експлуатації автоматизованої системи управління судном: підручник для студентів вищих навчальних закладів. / Богом'я В.І. та ін. ; за ред. О. М. Тимощук. Київ. ДУІТ. 2018. 305 с.
67. Особливості системного підходу до вирішення наукових завдань експлуатації суднового обладнання./ Богом'я В.І. та ін. ; за ред. О. М. Тимощук. Київ, ДУІТ, 2018, 305 с.
68. Паспорт спеціальності 05.22.20 - Експлуатація та ремонт засобів транспорту. URL: <http://inmad.vntu.edu.ua/upload/aspi/052220.pdf> (дата звернення: 09.12.2019).

69. Письменна К. С. Стан і тенденції розвитку суднобудівної промисловості в Україні: господарсько-правовий аспект URL: <http://vuzlib.com/content/view/2516/27/> (дата звернення 12.04.2019).
70. Прогнозирование и оценка технического состояния корпуса судна по замерам остаточных толщин /Баева Л.С. та ін. *Международный журнал прикладных и фундаментальных исследований*. 2016. № 7-1. С. 7-10. URL: <http://www.applied-research.ru/ru/article/view?id=9746> (дата звернення: 30.06.2019).
71. Проект Закону України «Про внутрішній водний транспорт» від 09.07.2018 № 2475а-д URL: [http://w1.c1.rada.gov.ua/pls/zweb2/webproc4\\_1?pf3511=64397](http://w1.c1.rada.gov.ua/pls/zweb2/webproc4_1?pf3511=64397) (дата звернення 12.12.2018).
72. Публічний звіт Голови Державної служби України з безпеки на транспорті М. Ноняка за 2017 рік. Державна служба України з безпеки на транспорті : веб сайт. URL: <http://dsbt.gov.ua/storinka/publicnyu-zvit-golovy-derzhavnoyi-sluzhby-ukrayiny-z-bezpeky-na-transporti-muha-la-0> (дата звернення: 09.12.2018).
73. Публічний звіт В.о. Голови Морської адміністрації за 2019 рік URL: <https://marad.gov.ua/storage/app/sites/1/gromadska-rada/Zvit/Publichnyi%20Zvit.pdf> (дата звернення: 21.03.2020).
74. Пятаков Э. Н. Требование к процедуре формирования второго уровня иерархической системы управления взаимодействием судов. Судовождение: Сб. научн. трудов. / ОНМА, Одесса: *ИздамИнформ*, Вып.13. 2007. С. 145 – 148.
75. Равин А.А. Диагностическое обеспечение судового энергетического оборудования. PRoATOM: веб-сайт. URL: <http://proatom.ru/modules.php?name=News&file=print&sid=4403> (дата звернення: 09.12.2018).
76. Равин А.А. Диагностическое обеспечение судового энергетического оборудования: проблемы и решения : дис.... д-ра техн. наук : 05.08.05.

- URL: <http://docplayer.ru/44925881-Ravin-aleksandr-aleksandrovich-diagnosticheskoe-obespechenie-sudovogo-energeticheskogo-oborudovaniya-problemy-i-resheniya.html> (дата звернення: 30.11.2018).
77. Разработка метода оценки и прогнозирования технического состояния судовых сложных систем / В. В. Вычужанин та ін. Восточно-Европейский журнал передовых технологий. 2016. №6/9 (84). URL: <http://journals.uran.ua/eejet/article/download/85605/87615> (дата звернення: 06.01.2020).
  78. Розвиток транспорту з метою відновлення і зростання української економіки: наукова доповідь / за ред. О.І. Никифороук. Київ: Ін-т екон. та прогнозув. НАН України. 2018. 200 с.
  79. Ротштейн А.П. Интеллектуальные технологии идентификации: нечеткая логика, генетические алгоритмы, нейронные сети. Винница: УНИВЕРСУМ-Винница, 1999. 320 с.
  80. Рутковская Д., Пилиньский М., Рутковский Л. Нейронные сети, генетические алгоритмы и нечеткие системы. Москва: Горячая линия. Телеком, 2006. 452 с.
  81. Садловська І.П. Аналіз сучасних вантажопотоків в Україні та напрями їх розвитку / І.П. Садловська // Вісник Дніпропетровського національного університету залізничного транспорту імені академіка В. Лазаряна. Дніпропетровськ: *Вид-во Дніпропетр. нац. ун-ту залізн. трансп. ім. акад. В. Лазаряна*, 2012. Вип. 42. С. 242 – 250.
  82. Седова Н. А., Седова Н. А. Теоретические аспекты нейросетевого управления курсом судна. Транспортное дело России. Москва: *Морские вести России*, 2006. Спецвыпуск № 7.С. 54– 57.
  83. Сетлак Г. Интеллектуальные системы поддержки принятия решений. Киев: Логос, 2004. 251 с.
  84. Сидоренко В. Ф. Кораблекрушения на море. Ленинград: Издательство Ленинградского университета, 1990. 296 с.

85. Структурное моделирование та символні перетворення для управління рухом транспортних засобів/ Баранов Г. Л. та ін. Київ, 2014. 310 с.
86. Тимощук О. М., Коломієць О. М., Дакі О. А. Обґрунтування застосування сигналів з нормованим спектром для контролю технічного стану радіонавігаційних приладів засобів водного транспорту. *Новітні технології*. 2018. Вип. 2(6). С. 39–45.
87. Тимощук О. М., Коломієць О. М., Данік О. В. Вибір критерію оптимальності системи відновлення суднових комплексів. *Економіка та держава*. 2017. № 4. С. 102–104.
88. Управління технічною експлуатацією флоту : конспект лекцій. URL: [http://www.kma.ks.ua/ua/images/2\\_library/methodical/sud\\_energ/department/avtomatyka/utef/u1.pdf](http://www.kma.ks.ua/ua/images/2_library/methodical/sud_energ/department/avtomatyka/utef/u1.pdf) (дата звернення: 09.12.2018).
89. Хайкин С. Нейронные сети: полный курс. Москва: Вильямс, 2006. 1104 с.
90. Шапиро Д. И. Принятие решений в системах организационного управления. Использование расплывчатых категорий. Москва: Энергоатомиздат, 1983. 185 с.
91. Шевченко А. П., Герасимов С. В., Тимочко О. І., Тимощук О. М., Трішин В. В. Оптимальний алгоритм обробки навігаційної інформації у системах управління засобами водного транспорту. *Наукоємні технології*. 2018, №4(40). С.450-457.
92. Шевченко А.П., Воробьёв Е.С., Мазур А.М., Коломиец О.М., Демьяненко С.К. Разработка методов сжатия сообщений о воздушных объектах и управления дискретностью их выдачи от источников радиолокационной информации. *Новітні технології*. 2018. Вип.3(7). С.217-230.
93. Шевченко А.П., Трофименко І.В., Мазур А.М. Підвищення точності вимірювань в суднових радіолокаційних системах з врахуванням

- тропосферного впливу в умовах неоднорідності морського середовища. *Новітні технології*. 2018, Вип.2(6). С.61-68.
94. Шевченко А.П., Михайлова Т.І, Бойко С.О. Спосіб прогнозування відмов агрегатів суднових комплексів за даними експлуатаційних спостережень. *Новітні технології*. 2019. Вип. 1(8). С.52-58.
95. Шевченко А.П., Будолак С.Ю., Ткаченко В.В., Гуменніков Р.В. Метод структурного синтезу системи управління засобів водного транспорту. *Наукоємні технології*. 2019. №1(41). С.101-108.
96. Шевченко А.П., Дробина В.В., Ковтун О. В., Корнєв І.О., Котигора В.А. Особливості створення автоматизованої системи управління судном. *Новітні технології*. 2019. Вип. 2(9). С. 125-130.
97. Шевченко А.П., Пліта Л.Л., Дакі О.А. Особливості організації процесу експлуатації засобів річкового та морського транспорту. *Новітні технології*. 2019. Вип. 3(10). С. 6-12.
98. Шевченко А.П., Штрибець В.В., Трофіменко А.О. Розроблення фільтрових методів спектрального аналізу випадкових сигналів для контролю технічного стану двигунів засобів водного транспорту. *Slovak international scientific journal*. 2019. Vol.1. No.34. pp.30-38.
99. Шевченко А.П., Богом'я В.І. Обґрунтування моделі функціонування автоматизованої системи управління засобів водного транспорту. Міжнародна науково-практична конференція «Водний транспорт: сучасний стан та перспективи розвитку»: тези доповідей, 16-17 травня 2019 року. Київ : ДУІТ, 2019. С.379-380.
100. Шевченко А.П., Коломієць О.М., Бойко С.О. Методи автоматизації контролю технічного стану засобів водного транспорту у різноманітних умовах експлуатації. Всеукраїнська інтернет-конференція студентів, аспірантів та молодих вчених я «Технічні науки в Україні: сучасні тенденції розвитку»: тези доповідей, 20-21 листопада 2019 року. Київ : ДУІТ, 2019. С.121-123.

101. Штовба С. Д. Проектирование нечетких систем средствами MATLAB. Москва: Горячая линия: Телеком, 2007. 228 с.
102. Шуляков В. М. Аналіз використання методу субтрактивної кластеризації при створенні нечітких регуляторів електрогідравлічних слідкуючих приводів автомобілів. *Вісник НТУ «ХПІ». Серія: Нові рішення в сучасних технологіях.* 2013. № 4 (978). С. 69-73.
103. Ярушкина Н. Г. Нечеткие нейронные сети. *Новости искусственного интеллекта.* 2001. № 3. С. 47-51.
104. Bootstrap. URL: <http://twbs.docs.org.ua/> (дата звернення 08.01.2019)
105. Buckley J.J., Hayashi Y. Fuzzy neural networks: a survey *Fuzzy Sets and Systems.* 1994. Vol. 66. P.p. 1-13.
106. De Marco T. *Structured Analysis and System Specification* N.Y.: Yourdon Press, 1988. 236p.
107. M. Schneider, A. Kandel, G. Langholz, G. Chew. *Fuzzy Expert System Tools / John Willey & Sons,* 1996. 198 p.
108. Goldberg D.E. *Genetic algorithms in search, optimization and machine learning.* Addison Wesley, 1989. 196 p.
109. Holland J.H. *Adaptation in natural and artificial systems. An introductory analysis with application to biology, control, and artificial intelligence.* London: Bradford book edition, 1994. 211 p.
110. Hollstein R.B. *Artificial genetic adaptation in computer control systems.* PhD Thesis: University of Michigan, 1971. 213 p.
111. Jang J.-S. R. ANFIS: Adaptive-Neuro-based Fuzzy Inference System. *IEEE Transactions on Systems, Man, and Cybernetics: Systems.* 1993. Vol. 23. no. 3. pp. 665–685.
112. Lisowski J. *Game control methods in navigator decision support system / The Archives of Transport.* 2005. No 3-4, Vol. XVII. P. 133 –147.
113. Recommendation of the European Parliament and of the Council of 23 April 2008 on the establishment of the European Qualifications Framework for lifelong learning. Text with EEA relevance. 2008. C 111/01.

114. Reform in the inland water transport: China's experience URL: <https://www.unescap.org/our-work/transport> (дата звернення 12.12.2018).
115. Sustainable development of inland waterway transport in China (2009). Theme I of a World Bank Project: Comprehensive Transport System Analysis in China URL: <http://siteresources.worldbank.org/EXTPRAL/Resources/china.pdf> (дата звернення 12.12.2018).
116. Takagi T., Sugeno M Fuzzy identification of systems and its application to modeling and control. IEEE Trans. Systems, Man, and Cybernetics. 1985. Vol. 15. P. 116–132.
117. The work plans of the 11 European Coordinators for the TENT have been finalised, establishing the basis for action until 2030. / Carlo Secchi, Karla Peijs, Laurens Jan Brinkhorst and others. Brussel, Belgium: European Commission, Directorate General for Mobility and Transport, May 2015 URL: [http://ec.europa.eu/transport/themes/infrastructure/news/20150528\\_coordinatorworkplans\\_en.htm](http://ec.europa.eu/transport/themes/infrastructure/news/20150528_coordinatorworkplans_en.htm) (дата звернення 12.12.2018).
118. Von Altrock C. Fuzzy Logic & Neuro-Fuzzy Applications Explained / New Jersey: Prentice Hall PTR. 1995. 350 p.
119. Yager, R. and D. Filev, "Generation of Fuzzy Rules by Mountain Clustering," *Journal of Intelligent & Fuzzy Systems*, 1994, Vol. 2, No. 3, pp. 209-219.
120. Zadeh L.A. A computational approach to fuzzy quantifiers in natural languages. *Computer and Mathematics*. 1983. Т № 9, p. 149-184.
121. Zadeh L. A. Fuzzy Logic, Neural Networks, and Soft Computing. *Communications of the ACM*, March 1994, Vol. 37 No. 3, pp. 77-84.
122. Glenn E. Krasner and Stephen T. Pope. A cookbook for using the model-view controller user interface paradigm in Smalltalk-80. *Journal of Object-Oriented Programming*, 1(3), 1988. pp. 26-49.
123. ISO/IEC 9126. Information Technology. – Software Quality Characteristics and metrics. 1997.

124. Lionel C. Briand, John W. Daly, and Jurgen K. Wust. A Unified Framework for Coupling Measurement in Object-Oriented Systems. *IEEE Transactions on Software Engineering*, 25(1), , 1999. pp.91–121.

## Лістинг А.1. Нечіткий висновок (алгоритм Мамдані)

```

[System]
Name='Tex_stan'
Type='mamdani'
Version=2.0
NumInputs=2
NumOutputs=1
NumRules=5
AndMethod='min'
OrMethod='max'
ImpMethod='min'
AggMethod='max'
DefuzzMethod='centroid'

[Input1]
Name='P±PsPePsPIP°'
Range=[0 3]
NumMFs=2
MF1='PSPsCᄁPjP°': 'smf', [1 1.5]
MF2='PSPᄁC%Pμ__PSPsCᄁPjPᄁ': 'zmf', [1 1.5]

[Input2]
Name='PrPSPᄁC%Pμ'
Range=[0 3]
NumMFs=2
MF1='PSPsCᄁPjP°': 'smf', [1 2]
MF2='PSPᄁC%Pμ__PSPsCᄁPjPᄁ': 'zmf', [1 2]

[Output1]
Name='PŸPŸ'
Range=[0 110]
NumMFs=3
MF1='P·P°P±PsCᄁPsPSP°': 'trimf', [-1330 48.5 60.0054171180932]
MF2='PsP±PjPμP¶PμPSPμ': 'trimf', [60.2437703141928 79.8 90]
MF3='PìCᄁPᄁPrP°C, PSPμ': 'trimf', [90 100 13800]

[Rules]
1 1, 3 (1) : 1
1 2, 2 (1) : 1
1 1, 3 (1) : 1
2 2, 1 (1) : 2
2 2, 1 (1) : 1

```

## Вхідні данні (текстовий файл fcmdata1.dat)

1.1895919e-01	7.1971108e-01	2.4784176e-01	9.1691429e-01	7.5294041e-01	3.3203456e-01
6.7929641e-01	2.1389837e-01	4.7643180e-01	5.7842135e-01	6.6952064e-01	5.0487466e-01
7.3096535e-01	4.7903993e-01	3.8931417e-01	5.2762005e-01	6.3342991e-01	4.2841739e-01
5.3461635e-02	6.7931485e-01	2.0325033e-01	7.1023354e-01	2.2700772e-01	8.4878311e-01
5.8677271e-01	5.2219568e-01	9.4748678e-01	3.3117965e-01	6.9983444e-01	2.6863618e-01
8.1964891e-02	7.1405761e-01	1.3118853e-01	7.5624968e-01	5.2612328e-01	4.7793654e-01
6.5391896e-01	1.6631787e-01	8.8564837e-01	2.8843555e-01	3.2966639e-01	7.5485767e-01
7.0119059e-01	4.2222400e-01	9.2173630e-02	7.3633707e-01	4.8532518e-01	4.0283149e-01
9.1032083e-01	3.1883553e-01	3.6533903e-01	5.4023723e-01	8.6022570e-01	3.8888944e-01
7.3608188e-01	4.5760685e-01	2.5305736e-01	7.6713302e-01	5.5683578e-01	2.2796971e-01
6.3263857e-01	2.4380738e-01	7.8315317e-01	5.0865692e-01	7.3899661e-01	4.8685151e-01
7.2266040e-01	3.2066780e-01	3.4952414e-01	7.3388084e-01	5.2854827e-01	3.0300425e-01
7.5335583e-01	4.6373323e-01	2.1524838e-01	8.6596086e-01	8.1073887e-01	5.9246226e-01
7.2685883e-02	8.7749311e-01	6.7959237e-01	2.0414434e-01	5.8811912e-01	5.1315049e-01
2.7270997e-01	6.4868126e-01	2.5012559e-01	8.2884477e-01	5.1808359e-01	5.2021714e-01
7.6649478e-01	3.9238853e-01	8.6085984e-01	3.9399870e-01	3.7022628e-01	7.2335304e-01
3.5926498e-01	6.4307502e-01	8.1756148e-01	4.9034177e-01	4.7589622e-01	8.9663054e-01
9.0465309e-01	1.8695199e-01	7.5584353e-01	1.7416348e-01	7.8263208e-02	7.6459728e-01
4.9397668e-01	5.4814877e-01	8.2469739e-01	3.6823673e-01	3.6974201e-01	5.8650577e-01
2.6614451e-01	7.3637975e-01	1.0343393e-01	7.9543708e-01	6.7178415e-01	2.1699766e-01
7.3749075e-02	7.9043769e-01	5.7671664e-01	3.0030901e-01	6.7623692e-01	7.9747635e-02
5.2974739e-01	4.0064069e-01	8.7656572e-01	2.9353873e-01	5.1393637e-01	3.1850888e-01
4.6444582e-01	5.6809172e-01	4.4003866e-01	5.0536909e-01	7.2860836e-01	1.7882267e-01
7.7020455e-01	1.6379344e-01	8.6926374e-01	2.8842873e-01	7.2076782e-01	4.7258700e-01
6.2954342e-01	2.3056262e-01	8.8603112e-01	1.3881256e-01	3.2156009e-01	7.4857788e-01
7.3622451e-01	6.5969527e-02	4.6332273e-01	5.5548080e-01	4.6043417e-01	3.4848546e-01
8.8857221e-01	2.3088594e-01	7.1342232e-01	4.7622826e-01	6.6135550e-01	6.4064059e-02
5.1327370e-01	3.1833422e-01	6.6767907e-01	2.4682623e-01	6.0563970e-01	8.4123407e-01
5.9111358e-01	2.4322982e-01	6.8204912e-01	4.0841180e-01	6.7009840e-01	4.9318005e-01
5.3730398e-01	3.8338957e-01	3.1573241e-01	6.8839481e-01	5.2280771e-01	2.9752862e-01
4.6791737e-01	6.2853425e-01	4.6753178e-01	5.3192527e-01	2.6661332e-01	7.0293582e-01
2.8721237e-01	7.7519880e-01	3.1917759e-01	8.8998057e-01	2.4673315e-01	6.6149041e-01
1.7832770e-01	7.6630850e-01	6.8249423e-01	4.8122673e-01	8.1710133e-01	5.1471396e-01
8.0240573e-01	2.5589577e-01	8.3641988e-01	1.4312388e-01	5.6074891e-01	8.8552542e-01
4.9848012e-01	3.3800649e-01	7.0892061e-01	4.8309189e-01	7.0782630e-01	1.3539050e-01
5.5458385e-01	5.1581954e-01	8.2870795e-01	3.2547527e-01	4.3663845e-01	5.0815926e-01
8.9073748e-01	3.7895436e-01	2.1354680e-01	6.5412826e-01	7.5171009e-01	1.9508763e-01
6.2484929e-01	8.5975014e-02	3.8985359e-01	6.1988540e-01	9.0330118e-01	3.4870540e-01
7.1470997e-01	5.0341800e-01	7.7686582e-01	3.0887354e-01	5.8478720e-01	8.0508049e-02
2.3991080e-01	6.7094390e-01	7.8386520e-01	2.3765611e-01	6.2686137e-01	4.7778535e-01
6.8134621e-01	6.0274597e-02	2.8215589e-01	9.2267568e-01	6.5905333e-01	8.3837822e-01
1.4753300e-01	8.7615123e-01	8.1972609e-01	5.6874402e-01	4.3866129e-01	9.3050593e-01
5.8718662e-01	4.7366216e-01	6.0101011e-01	2.7616354e-01		
5.9010861e-01	4.8229278e-01	8.2835472e-01	5.0918299e-01		
5.5614614e-01	1.5842396e-01	1.5773118e-01	8.3858058e-01		
4.0876669e-01	6.7277244e-01	2.3359919e-01	7.2674889e-01		
5.6489868e-01	4.2322414e-01	6.3471744e-01	3.3295071e-01		
4.8851455e-01	2.4175423e-01	7.9476981e-01	4.5126372e-01		
6.5125374e-01	5.4492761e-01	6.9624281e-01	3.8926233e-01		

## Лістинг А.2. Нечітка кластеризація (алгоритм FCM)

```

>> load fcmdatal.dat
plot(fcmdatal(:, 1), fcmdatal(:,2), 'o', 'color', 'k')
[center, U, obj_fcm] = fcm(fcmdatal, 2)
maxU = max(U);
index1 = find(U(1, :) == maxU);
index2 = find(U(2, :) == maxU);
line(fcmdatal(index1, 1), fcmdatal(index1, 2), 'linestyle', 'none',
'marker', 'o', 'color', 'g');
line(fcmdatal(index2, 1), fcmdatal(index2, 2), 'linestyle', 'none',
'marker', 'x', 'color', 'r');
hold on
plot(center(1,1), center(1,2), 'ko', 'MarkerSize', 12, 'LineWidth', 2)
plot(center(2,1), center(2,2), 'kx', 'MarkerSize', 12, 'LineWidth', 2)
plot(obj_fcm)
Iteration count = 1, obj. fcn = 8.740506
Iteration count = 2, obj. fcn = 7.113987
Iteration count = 3, obj. fcn = 5.988443
Iteration count = 4, obj. fcn = 4.474270
Iteration count = 5, obj. fcn = 4.081747
Iteration count = 6, obj. fcn = 4.044424
Iteration count = 7, obj. fcn = 4.039581
Iteration count = 8, obj. fcn = 4.038732
Iteration count = 9, obj. fcn = 4.038576
Iteration count = 10, obj. fcn = 4.038547
Iteration count = 11, obj. fcn = 4.038542

center =

    0.2847    0.7315
    0.7007    0.3375

U =
Columns 1 through 12
    0.9461    0.0358    0.0739    0.9051    0.2587    0.9268    0.0646
0.0260    0.0731    0.0533    0.0360    0.0021
    0.0539    0.9642    0.9261    0.0949    0.7413    0.0732    0.9354
0.9740    0.9269    0.9467    0.9640    0.9979

Columns 13 through 24
    0.0604    0.9119    0.9756    0.0207    0.9401    0.0862    0.5296
0.9989    0.9258    0.1639    0.6489    0.0590
    0.9396    0.0881    0.0244    0.9793    0.0599    0.9138    0.4704
0.0011    0.0742    0.8361    0.3511    0.9410

Columns 25 through 36
    0.0427    0.1039    0.0705    0.1374    0.0592    0.1348    0.7588
0.9947    0.9733    0.0332    0.1694    0.3082
    0.9573    0.8961    0.9295    0.8626    0.9408    0.8652    0.2412
0.0053    0.0267    0.9668    0.8306    0.6918

Columns 37 through 48
    0.0715    0.1147    0.1048    0.9828    0.1127    0.9375    0.1660
0.1761    0.1164    0.9130    0.1295    0.1615
    0.9285    0.8853    0.8952    0.0172    0.8873    0.0625    0.8340
0.8239    0.8836    0.0870    0.8705    0.8385

Columns 49 through 60
    0.2119    0.9380    0.6430    0.7172    0.9820    0.0923    0.9538
0.0616    0.9345    0.7810    0.9941    0.1080
    0.7881    0.0620    0.3570    0.2828    0.0180    0.9077    0.0462
0.9384    0.0655    0.2190    0.0059    0.8920

Columns 61 through 72

```

	0.9852	0.9574	0.0403	0.9765	0.0608	0.0977	0.0528
0.0371	0.9388	0.0582	0.0571	0.5610			
	0.0148	0.0426	0.9597	0.0235	0.9392	0.9023	0.9472
0.9629	0.0612	0.9418	0.9429	0.4390			

Columns 73 through 84

	0.0542	0.0938	0.6230	0.0724	0.0238	0.0201	0.9897
0.5573	0.9449	0.0868	0.0795	0.0809			
	0.9458	0.9062	0.3770	0.9276	0.9762	0.9799	0.0103
0.4427	0.0551	0.9132	0.9205	0.9191			

Columns 85 through 96

	0.0346	0.9683	0.8824	0.0155	0.0331	0.9340	0.1779
0.0427	0.1172	0.9519	0.9929	0.0153			
	0.9654	0.0317	0.1176	0.9845	0.9669	0.0660	0.8221
0.9573	0.8828	0.0481	0.0071	0.9847			

Columns 97 through 108

	0.0605	0.0094	0.0072	0.1270	0.0566	0.9660	0.0121
0.2907	0.9918	0.2547	0.0589	0.0907			
	0.9395	0.9906	0.9928	0.8730	0.9434	0.0340	0.9879
0.7093	0.0082	0.7453	0.9411	0.9093			

Columns 109 through 120

	0.0820	0.1126	0.2067	0.2376	0.4025	0.9722	0.8505
0.9288	0.8586	0.0357	0.1039	0.1364			
	0.9180	0.8874	0.7933	0.7624	0.5975	0.0278	0.1495
0.0712	0.1414	0.9643	0.8961	0.8636			

Columns 121 through 132

	0.0491	0.0677	0.9948	0.2458	0.1150	0.6956	0.1093
0.1195	0.9965	0.9800	0.1198	0.7620			
	0.9509	0.9323	0.0052	0.7542	0.8850	0.3044	0.8907
0.8805	0.0035	0.0200	0.8802	0.2380			

Columns 133 through 140

	0.0711	0.5754	0.0433	0.0722	0.1339	0.1217	0.6251
0.8691							
	0.9289	0.4246	0.9567	0.9278	0.8661	0.8783	0.3749
0.1309							

obj\_fcm =

8.7405  
7.1140  
5.9884  
4.4743  
4.0817  
4.0444  
4.0396  
4.0387  
4.0386  
4.0385  
4.0385

## Лістинг А.3. ANFIS (прогнозування ТС корінних та шатунних підшипників)

```

[System]
Name='ANFIS1'
Type='sugeno'
Version=2.0
NumInputs=8
NumOutputs=1
NumRules=6
AndMethod='prod'
OrMethod='probor'
ImpMethod='prod'
AggMethod='max'
DefuzzMethod='wtaver'
  [Input1]
Name='in1'
Range=[9 9]
NumMFs=6
MF1='in1cluster1':'gausmf',[0.000353553390593078 9]
MF2='in1cluster2':'gausmf',[0.000353553390593078 9]
MF3='in1cluster3':'gausmf',[0.000353553390593078 9]
MF4='in1cluster4':'gausmf',[0.000353553390593078 9]
MF5='in1cluster5':'gausmf',[0.000353553390593078 9]
MF6='in1cluster6':'gausmf',[0.000353553390593078 9]
  [Input2]
Name='in2'
Range=[4 44]
NumMFs=6
MF1='in2cluster1':'gausmf',[7.07106781657203 43.00000000248988]
MF2='in2cluster2':'gausmf',[7.07106781328177 4.00000000026354]
MF3='in2cluster3':'gausmf',[7.07204133452204 41.0022889007352]
MF4='in2cluster4':'gausmf',[7.07106781093849 41.9999999973819]
MF5='in2cluster5':'gausmf',[7.07106800949145 41.9999999580435]
MF6='in2cluster6':'gausmf',[7.07134211062111 43.9993537648333]
  [Input3]
Name='in3'
Range=[9 9]
NumMFs=6
MF1='in3cluster1':'gausmf',[0.000353553390593078 9]
MF2='in3cluster2':'gausmf',[0.000353553390593078 9]
MF3='in3cluster3':'gausmf',[0.000353553390593078 9]
MF4='in3cluster4':'gausmf',[0.000353553390593078 9]
MF5='in3cluster5':'gausmf',[0.000353553390593078 9]
MF6='in3cluster6':'gausmf',[0.000353553390593078 9]
  [Input4]
Name='in4'
Range=[4 45]
NumMFs=6
MF1='in4cluster1':'gausmf',[7.24784468036982 43.0000000039032]
MF2='in4cluster2':'gausmf',[7.24784450718489 36.0000000000033]
MF3='in4cluster3':'gausmf',[7.25278834894597 38.0050987885963]
MF4='in4cluster4':'gausmf',[7.24784434423104 3.99999997277231]
MF5='in4cluster5':'gausmf',[7.24784448804463 40.9999999691726]
MF6='in4cluster6':'gausmf',[7.24923223020753 44.998564104396]
  [Input5]
Name='in5'
Range=[9 9]
NumMFs=6
MF1='in5cluster1':'gausmf',[0.000353553390593078 9]
MF2='in5cluster2':'gausmf',[0.000353553390593078 9]
MF3='in5cluster3':'gausmf',[0.000353553390593078 9]
MF4='in5cluster4':'gausmf',[0.000353553390593078 9]
MF5='in5cluster5':'gausmf',[0.000353553390593078 9]
MF6='in5cluster6':'gausmf',[0.000353553390593078 9]

```

```

[Input6]
Name='in6'
Range=[3 39]
NumMFs=6
MF1='in6cluster1': 'gaussmf', [6.36396276877668 37.9999997395618]
MF2='in6cluster2': 'gaussmf', [6.36396103076486 28.0000000000686]
MF3='in6cluster3': 'gaussmf', [6.36410944953793 3.00094183839951]
MF4='in6cluster4': 'gaussmf', [6.36396101841394 31.9999999912708]
MF5='in6cluster5': 'gaussmf', [6.36396100834939 35.9999999565735]
MF6='in6cluster6': 'gaussmf', [6.36400368859446 3.99973421386556]
[Input7]
Name='in7'
Range=[9 9]
NumMFs=6
MF1='in7cluster1': 'gaussmf', [0.000353553390593078 9]
MF2='in7cluster2': 'gaussmf', [0.000353553390593078 9]
MF3='in7cluster3': 'gaussmf', [0.000353553390593078 9]
MF4='in7cluster4': 'gaussmf', [0.000353553390593078 9]
MF5='in7cluster5': 'gaussmf', [0.000353553390593078 9]
MF6='in7cluster6': 'gaussmf', [0.000353553390593078 9]
[Input8]
Name='in8'
Range=[2 38]
NumMFs=6
MF1='in8cluster1': 'gaussmf', [6.36396179186055 35.0000000800307]
MF2='in8cluster2': 'gaussmf', [6.36396103068038 2.00000000000857]
MF3='in8cluster3': 'gaussmf', [6.38986787001366 25.0125160332718]
MF4='in8cluster4': 'gaussmf', [6.36396099967046 26.9999999841633]
MF5='in8cluster5': 'gaussmf', [6.36396304416203 3.00000061586578]
MF6='in8cluster6': 'gaussmf', [6.37106295193534 37.9965309810442]
[Output1]
Name='out1'
Range=[16000 26000]
NumMFs=6
MF1='out1cluster1': 'linear', [182.837749410364 -520.583096121013
182.837748456675 331.639469710727 182.837732690823 230.062713269318
182.837745501464 479.421062093068 20.3153012363645]
MF2='out1cluster2': 'linear', [59.3814756174988 26.3918014033636
59.381429803886 237.525663745671 59.381460358717 184.742370308825
59.381474663839 13.1958724158206 6.59793748265177]
MF3='out1cluster3': 'linear', [36.1687408179693 165.709513223291
36.1687256560615 154.907940952184 36.1687256411461 12.3645465712573
36.168742807282 104.546304324338 4.01875440487204]
MF4='out1cluster4': 'linear', [41.991008502692 195.95715383956 41.9909932439098
18.6523288104508 41.9909932439031 149.300056979884 41.9909894292049
125.97218242589 4.66565961362901]
MF5='out1cluster5': 'linear', [35.4758503516991 165.52985690203 35.475811661303
161.593934869371 35.4758269187048 141.885029166416 35.4758393164659
11.1656773512269 3.94176429351656]
MF6='out1cluster6': 'linear', [41.829671964855 203.714132773926
41.8296714880162 207.314755205447 41.8296710112078 18.329432674852
41.8296483614457 173.209345152071 4.64774668641003]

[Rules]
1 1 1 1 1 1 1 1, 1 (1) : 1
2 2 2 2 2 2 2 2, 2 (1) : 1
3 3 3 3 3 3 3 3, 3 (1) : 1
4 4 4 4 4 4 4 4, 4 (1) : 1
5 5 5 5 5 5 5 5, 5 (1) : 1
6 6 6 6 6 6 6 6, 6 (1) : 1

```

## Акти впровадження

ЗАТВЕРДЖУЮ

Ректор Державного університету  
інфраструктури та технологій

д.т.н., професор

В.В. ПАНІН

«03» 2020 року

АКТ № 12/а

впровадження результатів дисертаційних досліджень  
на тему: «Моделі та методи прогнозування технічного стану засобів  
водного транспорту на основі м'яких обчислень»

Шевченка Антона Петровича

Комісія у складі: голови – завідувача кафедри експлуатації засобів транспорту на внутрішніх водних шляхах Войченко Т.О. та членів: Богом'я В.І., Ганношиної І.М. встановила, що результати наукових досліджень автора, а саме:

удосконалений автором метод прогнозування технічного стану засобів водного транспорту на основі нечіткого висновку при застосуванні гібридної мережі, який на відміну від існуючих базується на моделі адаптивної нейронної нечіткої мережевої системи ANFIS (Adaptive Neuro-Fuzzy Inference System) та надає можливість одночасно використовувати переваги нечіткої логіки та нейронних мереж для підвищення достовірності прогнозування технічного стану;

використовувались в навчальному процесі при підготовці освітньо-професійних програм бакалаврів і магістрів спеціальності 271 «Річковий та морський транспорт» спеціалізації «Судноводіння» при створенні курсу лекцій з дисциплін «Автоматизовані комплекси судноводіння», «Сучасні автоматизовані системи управління рухом судна», «Інноваційні технології в судноводінні».

Також, результати дисертаційних досліджень Шевченка А.П. впроваджені у науково-дослідної роботі: «Розробка комплексного показника якості пасажирських круїзних суден змішаного плавання в системі безпересадкових круїзних перевезень між портами Дніпра, Чорного моря та Дунаю» (номер держреєстрації 0116U03946), яка виконувалася у Київській державній академії водного транспорту, в якій автор приймав участь як виконавець.

Акт не є підставою для фінансових розрахунків.

Голова:

Т.О. Войченко

Члени:

В.І. Богом'я

І.М. Ганношина

**ЗАТВЕРДЖУЮ**

В.о. начальника ДП «Укрводшлях»

Шевурдін І. В.

«09» 06 2020 року

**АКТ № 3/03/2020**

впровадження результатів дисертаційних досліджень Шевченка А. П.

Комісія у складі: голови – заступника начальника підприємства з безпеки судноплавства, канд. техн. наук Дороніна В. В. та членів: начальника служби безпеки судноплавства Безносенка Д. П., провідного інженера служби безпеки судноплавства Чернокапської С. М. встановила, що результати наукових досліджень автора, а саме:

концептуальна модель прогнозування технічного стану засобів водного транспорту, яка базується на OLAP-технології з інтелектуальним аналізом даних на м'яких обчисленнях, а саме, комплексному використанні нечіткої кластеризації на основі методів субтрактивної кластеризації (для визначення кількості значень лінгвістичних змінних), а також адаптивної нейронної нечіткої мережі ANFIS включено в програму навчання в Центрі підготовки судноводіїв та використовується при виконанні практичної тренажерної підготовки судноводіїв.

Практичне значення одержаних результатів полягає в тому, що їх реалізація доцільна в математичному та програмному забезпеченні, як складової технології автоматизації процесу прогнозування технічного стану судна в інтегрованої місткової системі.

Застосування запропонованої моделі дозволяє підвищити достовірність прогнозу технічного стану в деяких випадках до 20-25 %.

Програмна реалізація моделі значно покращує показники оперативності та достовірності прогнозу за рахунок зменшення кількості операцій та використання нових принципів обробки апріорної інформації та самонавчання.

Акт не є підставою для фінансових розрахунків.

Голова комісії:

В. В. Доронін

Члени комісії:

Д. П. Безносенко

С. М. Чернокапська

**СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА**

*Наукові праці, в яких опубліковані основні наукові результати дисертації:*

1. Шевченко А. П., Герасимов С. В., Тимочко О. І., Тимошук О. М., Трішин В. В. Оптимальний алгоритм обробки навігаційної інформації у системах управління засобами водного транспорту. Наукоємні технології. 2018, №4(40). С.450-457.

2. Шевченко А.П., Воробьёв Е.С., Мазур А.М., Коломиец О.М., Демьяненко С.К. Разработка методов сжатия сообщений о воздушных объектах и управления дискретностью их выдачи от источников радиолокационной информации. *Новітні технології*. 2018. Вип.3(7). С.217-230.

3. Шевченко А.П., Трофименко І.В., Мазур А.М. Підвищення точності вимірювань в суднових радіолокаційних системах з врахуванням тропосферного впливу в умовах неоднорідності морського середовища. *Новітні технології*. 2018, Вип.2(6). С.61-68.

4. Шевченко А.П., Михайлова Т.І, Бойко С.О. Спосіб прогнозування відмов агрегатів суднових комплексів за даними експлуатаційних спостережень. *Новітні технології*. 2019. Вип. 1(8). С.52-58.

5. Шевченко А.П., Будолак С.Ю., Ткаченко В.В., Гуменніков Р.В. Метод структурного синтезу системи управління засобів водного транспорту. *Наукоємні технології*. 2019. №1(41). С.101-108.

6. Шевченко А.П., Дробина В.В., Ковтун О. В., Корнев І.О., Котигора В.А. Особливості створення автоматизованої системи управління судном. *Новітні технології*. 2019. Вип. 2(9). С. 125-130.

7. Шевченко А.П., Пліта Л.Л., Дакі О.А. Особливості організації процесу експлуатації засобів річкового та морського транспорту. *Новітні технології*. 2019. Вип. 3(10). С. 6-12.

8. Шевченко А.П., Штрибець В.В., Трофіменко А.О. Розроблення фільтрових методів спектрального аналізу випадкових сигналів для контролю технічного стану двигунів засобів водного транспорту. Slovak international scientific journal. 2019. Vol.1. No.34. pp.30-38.

9. Шевченко А.П., Пліта Л.Л. Аналіз методів прогнозування технічного стану засобів водного транспорту. Водний транспорт. 2020. Вип.1(29). С.23–30.

*Наукові праці, які засвідчують апробацію матеріалів дисертації:*

10. Шевченко А.П., Богом'я В.І. Обґрунтування моделі функціонування автоматизованої системи управління засобів водного транспорту. Міжнародна науково-практична конференція «Водний транспорт: сучасний стан та перспективи розвитку»: тези доповідей, 16-17 травня 2019 року. Київ : ДУІТ, 2019. С.379-380.

11. Шевченко А.П., Коломієць О.М., Бойко С.О. Методи автоматизації контролю технічного стану засобів водного транспорту у різноманітних умовах експлуатації. Всеукраїнська інтернет-конференція студентів, аспірантів та молодих вчених «Технічні науки в Україні: сучасні тенденції розвитку»: тези доповідей, 20-21 листопада 2019 року. Київ :ДУІТ, 2019. С.121-123.

12. Шевченко А.П., Богом'я В.І. Моделі та методи прогнозування технічного стану засобів водного транспорту на основі використання м'яких обчислень. II Міжнародна науково-практична морська конференція кафедри СЕУ і ТЕ Навчально-наукового інституту Морського флоту Одеського національного морського університету. Одеса– 2020. С.174–178.