

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ОДЕСЬКА ПОЛІТЕХНІКА»
МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ОДЕСЬКА ПОЛІТЕХНІКА»
МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

*Кваліфікаційна наукова
праця на правах рукопису*

КРИКУН ВАЛЕНТИН АНДРІЙОВИЧ

УДК 004.942: 004.052

ДИСЕРТАЦІЯ

**МЕТОД ТА ПРОГРАМНІ ЗАСОБИ ІНТЕРПРЕТАЦІЇ МОДЕЛЕЙ
МАШИННОГО НАВЧАННЯ НЕЛІНІЙНИХ ДИНАМІЧНИХ ОБ'ЄКТІВ**

Спеціальність: 121 Інженерія програмного забезпечення

Галузь знань: 12 Інформаційні технології

Подається на здобуття наукового ступеня доктора філософії

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

_____ В. А. Крикун

Науковий керівник – Фомін О. О. доктор технічних наук, професор

Одеса – 2024

АНОТАЦІЯ

Крикун В. А. Метод та програмні засоби інтерпретації моделей машинного навчання нелінійних динамічних об'єктів. – Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття наукового ступеня доктора філософії за спеціальністю 121 Інженерія програмного забезпечення. – Національний університет «Одеська політехніка» Міністерства освіти і науки України, Одеса, 2024.

Роботу присвячено вирішенню актуальної науково-практичної задачі, яка полягає у створенні методу інтерпретації моделей машинного навчання нелінійних динамічних об'єктів з неперервними характеристиками та його застосуванню у вигляді програмно-алгоритмічних засобів ідентифікації як складової частини інтелектуальних систем.

Актуальність теми дослідження визначається наявним протиріччям між сталим ускладненням моделей машинного навчання при ідентифікації нелінійних динамічних об'єктів типу «чорна скриня», з одного боку, та зростанням труднощів інтерпретації цих моделей людиною — з іншого боку. Встановлене протиріччя долається з застосуванням методу побудови сурогатних моделей у вигляді непараметричних динамічних моделей на основі інтегро-степеневих поліномів Вольтерри, що дозволяє спростити структуру сурогатних моделей та, завдяки урахуванню нелінійних та динамічних властивостей моделей машинного навчання, підвищити точність ідентифікації сурогатних моделей.

Метою роботи є підвищення точності сурогатних моделей шляхом розвитку методу інтерпретації моделей машинного навчання нелінійних динамічних об'єктів за допомогою непараметричних динамічних моделей на основі інтегро-степеневих поліномів Вольтерри та реалізації запропонованого методу у вигляді програмно-алгоритмічних засобів ідентифікації неперервних об'єктів у складі інтелектуальних систем.

Для досягнення вказаної мети дослідження в роботі поставлено і розв'язано такі **задачі**:

– виконано аналіз існуючих методів інтерпретації моделей машинного навчання неперервних нелінійних динамічних об'єктів, визначено базові проблеми інтерпретації нейромережових моделей; обґрунтовано вибір напрямку досліджень в області побудови сурогатних моделей у вигляді інтегральних непараметричних динамічних моделей на основі інтегро-степеневих поліномів Вольтерри для аналітичного опису нелінійних динамічних об'єктів;

– запропоновано використання непараметричних динамічних моделей на основі інтегро-степеневих поліномів Вольтерри в якості сурогатних моделей для інтерпретації нейронних мереж з часовими затримками; встановлено зв'язок в аналітичному вигляді між нейронними мережами з часовими затримками та непараметричними динамічними моделями на основі інтегро-степеневих поліномів Вольтерри для підвищення точності сурогатних моделей неперервних нелінійних динамічних об'єктів;

– розвинуто метод інтерпретації нейронних мереж з часовими затримками шляхом побудови сурогатних моделей у вигляді непараметричних динамічних моделей на основі інтегро-степеневих поліномів Вольтерри для опису нелінійних динамічних об'єктів із забезпеченням прийнятної точності моделювання;

– розроблено інформаційну технологію інтерпретації моделей машинного навчання нелінійних динамічних об'єктів, що ґрунтується на використанні нейронної мережі з часовими затримками для ідентифікації нелінійної динамічної моделі об'єкта за даними експерименту «вхід–вихід» та побудови сурогатної моделі у вигляді непараметричних динамічних моделей на основі інтегро-степеневих поліномів Вольтерри із забезпеченням прийнятної точності моделювання для інтерпретації нейронної мережі;

– розроблено інструментальні засоби комп'ютерного моделювання для побудови нейромережових моделей нелінійних динамічних об'єктів та сурогатних моделей для їх інтерпретації в системі ідентифікації неперервних об'єктів;

застосовано розроблений метод та інструментальні засоби для розв'язання прикладних задач ідентифікації неперервних об'єктів.

Наукова новизна отриманих результатів, які виносяться на захист, полягає в наступному.

– *Вперше запропоновано* сурогатну модель у вигляді непараметричної динамічної моделі на основі інтегро-степеневих поліномів Вольтерри для інтерпретації нейронних мереж з часовими затримками, побудованих за результатами експерименту «вхід–вихід», яка, на відміну від відомих моделей, дозволяє одночасно спростити структуру сурогатних моделей та підвищити точність їх ідентифікації, завдяки урахуванню нелінійних та динамічних властивостей моделей машинного навчання.

– *Отримав подальший розвиток* метод оцінки ядер Вольтерри за допомогою нейронних мереж з часовими затримками шляхом встановлення аналітичного зв'язку між нейронною мережею з активацією у вигляді лінійного випрямляча та непараметричною динамічною моделлю на основі інтегро-степеневих поліномів Вольтерри, що дозволяє підвищити точність ідентифікації нелінійних динамічних об'єктів у вигляді поліномів Вольтерри.

– *Вперше запропоновано* метод інтерпретації моделей машинного навчання у вигляді нейронних мереж з часовими затримками, збудованими за даними експерименту «вхід–вихід» шляхом побудови сурогатної моделі на основі непараметричних динамічних моделей у вигляді інтегро-степеневих поліномів Вольтерри за допомогою встановленого аналітичного зв'язку між нейронною мережею та ядрами Вольтерри. Це дозволяє підвищити точність та зменшити розмірність сурогатних моделей при ідентифікації нелінійних динамічних об'єктів.

– *Удосконалено* математичну модель оцінки якості програмного забезпечення інтерпретації моделей машинного навчання, яка відрізняється від існуючих одночасним врахуванням метрик функціональності та складності, що дозволяє кількісно виразити здатність до інтерпретації для формального

порівняння сурогатних моделей, збудованих різними методами та автоматизації відбору моделей машинного навчання та сурогатних моделей.

Практичне значення одержаних результатів полягає у розробці інструментальних програмно-алгоритмічних засобів інформаційної технології інтерпретації моделей машинного навчання нелінійних динамічних об'єктів, які реалізують обчислювальні алгоритми ідентифікації нелінійної динамічної моделі об'єкта за даними експерименту «вхід–вихід» за допомогою нейронної мережі з часовими затримками та побудови сурогатної моделі у вигляді непараметричних динамічних моделей на основі інтегро-степеневих поліномів Вольтерри із забезпеченням прийнятної точності моделі машинного навчання, а також у впровадженні створених інструментальних засобів у навчальний процес та наукові дослідження.

На основі створених ІЗ розроблено ефективні хмарні сервіси для вирішення ресурсномістких задач моделювання, що забезпечує високу обчислювальну потужність та швидкість побудови моделей в задачах ідентифікації неперервних об'єктів.

Розроблені ІЗ та хмарні сервіси впроваджено в ТОВ «Дискрет» (м. Одеса), де вони використовуються для ідентифікації моделей тензометричного зважування транспортних засобів у русі та їх інтерпретації.

Результати, отримані в дисертаційній роботі, використано у навчальному процесі Національного університету «Одеська політехніка» при викладанні дисциплін «Інтелектуальні системи» та «Математичне моделювання нелінійних динамічних процесів» на кафедрі комп'ютеризованих систем та програмних технологій; а також застосовано при розробці тем магістерських кваліфікаційних робіт.

Об'єкт досліджень – процес інтерпретації моделей машинного навчання неперервних нелінійних динамічних об'єктів.

Предмет досліджень – моделі та методи інтерпретації нейромережевих моделей нелінійних динамічних об'єктів, орієнтовані на впровадження в системах ідентифікації неперервних об'єктів.

Ключові слова: інтерпретація моделей машинного навчання, нейронні мережі з часовими затримками, нелінійні динамічні об'єкти, непараметричні динамічні моделі, поліноми Вольтерри, сурогатні моделі, ідентифікація.

Список публікацій здобувача

Наукові праці, в яких опубліковано основні наукові результати дисертації.

1. Krykun V. Improving the accuracy of the neural network models interpretation of nonlinear dynamic objects. *Математичне та комп'ютерне моделювання. Серія: Технічні науки*. 2023. Вип. 24. С. 45–55. DOI: 10.32626/2308-5916.2023-24.45-55. *Видання включено до переліку наукових фахових видань України, категорія «Б».*

<http://mcm-tech.kpnu.edu.ua/issue/view/17449>

2. Mathematical models of software quality assurance for interpretation of dynamic neural networks / O. Fomin, V. Krykun, A. Orlov et al. *Вчені записки ТНУ імені В.І. Вернадського. Серія: Технічні науки*. 2023. Том 34 (73), № 5. С. 250–256. DOI: 10.32782/2663-5941/2023.5/39. *Видання включено до переліку наукових фахових видань України, категорія «Б».*

<http://tech.vernadskyjournals.in.ua/34-73-5>

3. Models of dynamic objects with significant nonlinearity based on time-delay neural networks / O. Fomin, V. Speranskyu, V. Krykun et al. *Вісник черкаського державного технологічного університету. Технічні науки*. 2023. № 3. С. 97–112. DOI: 10.24025/2306-4412.3.2023.288284. *Видання включено до переліку наукових фахових видань України, категорія «Б».*

<http://vtn.chdtu.edu.ua/issue/view/16885>

4. Fomin O.O., Krykun, V.A. Assessment of the Quality of Neural Network Models Based on a Multifactorial Information Criterion. *Вісник сучасних інформаційних технологій*. 2024. Том 7, № 1. С. 13–23. DOI: 10.15276/hait.07.2024.1. *Видання включено до переліку наукових фахових видань України, категорія «Б».*

<https://hait.od.ua/index.php/journal/article/view/201>

Наукові праці, які засвідчують апробацію матеріалів дисертації.

5. Interpretation of Dynamic Models Based on Neural Networks in the Form of Integral-Power Series / O. Fomin et al. ; in: Arsenyeva, O., Romanova, T., Sukhonos, M., Tsegelnyk, Y. (eds). *Smart Technologies in Urban Engineering. STUE 2022. Lecture Notes in Networks and Systems*. Springer, Cham, 2022. Vol. 536. P. 258–265. DOI: 10.1007/978-3-031-20141-7_24. Видання проіндексовано у базах даних *Web of Science Core Collection* та *Scopus*.

<https://link.springer.com/book/10.1007/978-3-031-20141-7?page=2>

6. Interpretation Method for Dynamic States Neural Network Models / S. Polozhaenko et al. *IEEE 3rd International Conference on System Analysis & Intelligent Computing (SAIC)*. Kyiv, Ukraine, 2022. P. 1–5. DOI: 10.1109/SAIC57818.2022.9923008. Видання проіндексовано у базі даних *Scopus*.

<https://ieeexplore.ieee.org/xpl/conhome/9922952/proceeding?isnumber=9922911&pageNumber=2>

7. Крикун В. А., Фомін О. О. Інтерпретація динамічних моделей у вигляді нейронних мереж з часовими затримками. *Матеріали Дванадцятій Міжнародної наукової конференції студентів та молодих учених «Сучасні інформаційні технології – 2022»*. Одеса, Україна, 2022. С. 140-141.

<http://dspace.opu.ua/jspui/handle/123456789/12912>

8. Use of dynamic neural networks for modeling nonlinear objects with significant nonlinearity / S. Polozhaenko et al. *Збірник тез IV Міжнародної науково-практичної Інтернет-конференції "Математика та інформатика в науці й освіті: виклики сучасності"*. Вінниця, Україна, 2023. С. 121-124.

<https://press.vntu.edu.ua/index.php/vntu/catalog/book/791>

9. Крикун В. А., Фомін О. О. Нелінійне моделювання об'єктів на основі динамічних нейронних мереж. *Матеріали Тринадцятій Міжнародної наукової конференції студентів та молодих учених «Сучасні інформаційні технології – 2023»*. Одеса, Україна, 2023. С. 153-155.

<http://dspace.opu.ua/jspui/handle/123456789/14172>

10. Use of Dynamic Neural Networks for Modeling Nonlinear Objects with Significant Nonlinearity / O. Fomin et al. *18th Conference on Computer Science and Intelligence Systems*. Warsaw, Poland, 2023. P. 97–102. DOI: 10.15439/2023F3874.

<https://annals-csis.org/proceedings/2023/communication.html>

11. Modeling of the agricultural crops development using satellite imagery / O. Fomin et al. *Біоінтенсивні та SMART-технології у тваринництві: матеріали II Міжнародної науково-практичної конференції науково-педагогічних працівників та молодих науковців*. Одеса, Україна, 2023. С. 10–14.

<https://osau.edu.ua/wp-content/uploads/2023/11/TEZY-2023-II-Mizhn-konf-NNIBtaA-2906.pdf>

12. Крикун В. А. Математична модель оцінки якості програмного забезпечення інтерпретації моделей машинного навчання. *Комп'ютеризовані системи та програмні технології*. 2023. № 1. С. 7–11.

https://ccs.od.ua/wp-content/uploads/journal/cspt_1_2023.pdf

ABSTRACT

V. Krykun. Method and software tools for interpreting machine learning models of nonlinear dynamic objects. – Qualifying scientific work on the rights of a manuscript.

Thesis for the degree of Doctor of Philosophy in specialty 121 Software Engineering. – Odessa Polytechnic National University, Ministry of Education and Science of Ukraine, Odessa, 2024.

The dissertation is devoted to solving an urgent scientific and practical problem, which is to create a method for interpreting machine learning models of nonlinear dynamic objects with continuous characteristics and its application in the form of software and algorithmic identification tools as an integral part of intelligent systems.

The relevance of the research topic is determined by the existing contradiction between the growing complexity of machine learning models for identifying nonlinear dynamic objects of the "black box" type, on the one hand, and the growing difficulty of interpreting these models by humans, on the other hand. This contradiction is overcome

by applying the method of building surrogate models in the form of nonparametric dynamic models based on integral-power Volterra polynomials, which allows simplifying the structure of surrogate models and, due to the consideration of nonlinear and dynamic properties of machine learning models, increasing the accuracy of surrogate model identification.

The purpose of the study is to improve the accuracy of surrogate models by developing a method for interpreting machine learning models of nonlinear dynamic objects using nonparametric dynamic models based on integral-power Volterra polynomials and implementing the proposed method in the form of software and algorithmic tools for identifying continuous objects as part of intelligent systems.

To achieve the research purpose, the following tasks were set and solved:

- the analysis of existing methods for interpreting machine learning models of continuous nonlinear dynamic objects is carried out, the basic problems of interpreting neural network models are identified; the choice of research direction in the field of building surrogate models in the form of integral nonparametric dynamic models based on integral-power Volterra polynomials for the analytical description of nonlinear dynamic objects is substantiated;

- the use of nonparametric dynamic models based on Volterra integral polynomials as surrogate models for the interpretation of neural networks with time delays was proposed; an analytical connection between neural networks with time delays and nonparametric dynamic models based on Volterra integral polynomials was established to improve the accuracy of surrogate models of continuous nonlinear dynamic objects;

- a method for interpreting neural networks with time delays was developed by constructing surrogate models in the form of nonparametric dynamic models based on Volterra integral-power polynomials to describe nonlinear dynamic objects with acceptable modeling accuracy;

- an information technology for interpreting machine learning models of nonlinear dynamic objects based on the use of a neural network with time delays to identify a nonlinear dynamic model of an object based on input-output experiment data

and to build surrogate models in the form of nonparametric dynamic models based on integral-power Volterra polynomials with acceptable modeling accuracy for interpreting the neural network was developed;

– computer modeling tools for constructing neural network models of nonlinear dynamic objects and surrogate models for their interpretation in continuous object identification systems was developed; the developed method and tools to solve applied problems of continuous object identification was applied.

The scientific novelty of the results presented for defense is as follows.

– For the first time, a surrogate model in the form of a nonparametric dynamic model based on integral-power Volterra polynomials is proposed for interpreting time delay neural networks built on the results of an input-output experiment, which, unlike the known models, allows simultaneously simplifying the structure of surrogate models and increasing the accuracy of their identification by taking into account the nonlinear and dynamic properties of machine learning models.

– The method of estimating Volterra kernels using time delay neural networks was further developed by establishing an analytical connection between a neural network with activation in the form of a linear rectifier and a nonparametric dynamic model based on integral-power Volterra polynomials, which allows to increase the accuracy of identification of nonlinear dynamic objects in the form of Volterra polynomials.

– For the first time, a method for interpreting machine learning models in the form of time delay neural networks built from input-output experiment data is proposed by constructing a surrogate model based on nonparametric dynamic models in the form of integral-power Volterra polynomials using the established analytical relationship between the neural network and Volterra kernels. This makes it possible to increase the accuracy and reduce the dimensionality of surrogate models when identifying nonlinear dynamic objects.

– A mathematical model for assessing the quality of machine learning model interpretation software has been improved, which differs from the existing ones by simultaneously taking into account the functionality and complexity metrics, which

allows quantifying the interpretation ability for formal comparison of surrogate models built by different methods and automating the selection of machine learning models and surrogate models.

The practical significance of the obtained results lies in the development of instrumental software and algorithmic means of information technology for interpreting machine learning models of nonlinear dynamic objects, which implement computational algorithms for identifying a nonlinear dynamic model of an object based on the data of an input-output experiment using a time delay neural network and building surrogate models in the form of nonparametric dynamic models based on integral-power Volterra polynomials with acceptable accuracy.

On the basis of the created tools, effective cloud services for solving identification tasks have been developed. These tools ensure the speed of building of continuous objects models due to high computing power.

The developed software and cloud services have been implemented in Diskret LLC (Odesa), where they are used to identify models of strain gauge weighing of vehicles in motion and their interpretation.

The results obtained in this thesis are used in the educational process of the Odessa Polytechnic National University in teaching the disciplines "Intelligent Systems" and "Mathematical Modeling of Nonlinear Dynamic Processes" at the Department of Computerized Systems and Software Technologies; and are also used in the development of topics for master's qualification works.

The object of research is the process of interpreting machine learning models of continuous nonlinear dynamic objects.

The subject of research is models and methods for interpreting neural network models of nonlinear dynamic objects, focused on implementation in continuous object identification systems.

Keywords: interpretation of machine learning models, time delay neural networks, nonlinear dynamic objects, nonparametric dynamic models, Volterra polynomials, surrogate models, identification.

List of publications of the applicant

Scientific works in which the main scientific results of the dissertation are published.

1. Krykun V. Improving the accuracy of the neural network models interpretation of nonlinear dynamic objects. *Mathematical and computer modelling. Series: Technical sciences*. 2023. Issue 24. P. 45–55. DOI: 10.32626/2308-5916.2023-24.45-55. *The issue is included in the list of scientific professional publications of Ukraine, category "B"*.

<http://mcm-tech.kpnu.edu.ua/issue/view/17449>

2. Mathematical models of software quality assurance for interpretation of dynamic neural networks / O. Fomin, V. Krykun, A. Orlov et al. *Scientific notes of Taurida National V.I. Vernadsky University. Series: Technical Sciences*. 2023. Vol. 34 (73), № 5. P. 250–256. DOI: 10.32782/2663-5941/2023.5/39. *The issue is included in the list of scientific professional publications of Ukraine, category "B"*.

<https://tech.vernadskyjournals.in.ua/34-73-5>

3. Models of dynamic objects with significant nonlinearity based on time-delay neural networks / O. Fomin, V. Speransky, V. Krykun et al. *Bulletin of Cherkasy State Technological University. Technical Sciences*. 2023. № 3. P. 97–112. DOI: 10.24025/2306-4412.3.2023.288284. *The issue is included in the list of scientific professional publications of Ukraine, category "B"*.

<http://vtn.chdtu.edu.ua/issue/view/16885>

4. Fomin O.O., Krykun, V.A. Assessment of the Quality of Neural Network Models Based on a Multifactorial Information Criterion. *Herald of Advanced Information Technology*. 2024. Vol. 7, № 1. P. 13–23. DOI: 10.15276/hait.07.2024.1. *The issue is included in the list of scientific professional publications of Ukraine, category "B"*.

<https://hait.od.ua/index.php/journal/article/view/201>

Scientific works that confirm the approbation of the dissertation materials.

5. Interpretation of Dynamic Models Based on Neural Networks in the Form of Integral-Power Series / O. Fomin et al. ; in: Arsenyeva, O., Romanova, T., Sukhonos, M., Tsegelnyk, Y. (eds). *Smart Technologies in Urban Engineering. STUE 2022. Lecture*

Notes in Networks and Systems. Springer, Cham, 2022. Vol. 536. P. 258–265. DOI: 10.1007/978-3-031-20141-7_24. *The issue is indexed in the Web of Science Core Collection and Scopus databases.*

<https://link.springer.com/book/10.1007/978-3-031-20141-7?page=2>

6. Interpretation Method for Dynamic States Neural Network Models / S. Polozhaenko et al. *IEEE 3rd International Conference on System Analysis & Intelligent Computing (SAIC)*. Kyiv, Ukraine, 2022. P. 1–5. DOI: 10.1109/SAIC57818.2022.9923008. *The issue is indexed in the Scopus database.*

<https://ieeexplore.ieee.org/xpl/conhome/9922952/proceeding?isnumber=9922911&pageNumber=2>

7. Krykun V., Fomin O. Interpretation of dynamic models in the form of neural networks with time delays. *Proceedings of the Twelfth International Scientific Conference of Students and Young Scientists «Modern Information Technology – 2022»*. Odessa, Ukraine, 2022. P. 140-141.

<http://dspace.opu.ua/jspui/handle/123456789/12912>

8. Use of dynamic neural networks for modeling nonlinear objects with significant nonlinearity / S. Polozhaenko et al. *Proceedings of the IV International Scientific and Practical Internet Conference "Mathematics and Informatics in Science and Education: Challenges of Modernity"*. Vinnytsia, Ukraine, 2023. P. 121-124.

<https://press.vntu.edu.ua/index.php/vntu/catalog/book/791>

9. Krykun V., Fomin O. Nonlinear modeling of objects based on dynamic neural networks. *Proceedings of the Thirteenth International Scientific Conference of Students and Young Scientists «Modern Information Technology – 2023»*. Odessa, Ukraine, 2023. P. 153-155.

<http://dspace.opu.ua/jspui/handle/123456789/14172>

10. Use of Dynamic Neural Networks for Modeling Nonlinear Objects with Significant Nonlinearity / O. Fomin et al. *18th Conference on Computer Science and Intelligence Systems*. Warsaw, Poland, 2023. P. 97–102. DOI: 10.15439/2023F3874.

<https://annals-csis.org/proceedings/2023/communication.html>

11. Modeling of the agricultural crops development using satellite imagery / O. Fomin et al. *Biointensive and SMART technologies in animal husbandry*: Proceedings of the II International Scientific and Practical Conference of Academic Staff and Young Scientists. Odesa, Ukraine, 2023. P. 10–14.

<https://osau.edu.ua/wp-content/uploads/2023/11/TEZY-2023-II-Mizhn-konf-NNIBtaA-2906.pdf>

12. Krykun V. A mathematical model for assessing the quality of software for interpreting machine learning models. *Computerized systems and software technologies*. 2023. № 1. P. 7–11.

https://ccs.od.ua/wp-content/uploads/journal/cspt_1_2023.pdf

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ	18
ВСТУП	19
РОЗДІЛ 1 АНАЛІЗ ПРОБЛЕМИ ІНТЕРПРЕТАЦІЇ МОДЕЛЕЙ МАШИННОГО НАВЧАННЯ В СИСТЕМАХ ІДЕНТИФІКАЦІЇ НЕПЕРЕРВНИХ ДИНАМІЧНИХ ОБ'ЄКТІВ	25
1.1. Аналіз застосування математичних моделей інтерпретації в складі інтелектуальних систем	26
1.2. Класифікація моделей неперервних динамічних об'єктів у вигляді нейронних мереж	35
1.2.1. Динамічний нейропросторовий мепінг	38
1.2.2. Динамічні нейронні мережі вінірівського типу	39
1.2.3. Нейронні мережі з часовими затримками	40
1.3. Методи інтерпретації моделей машинного навчання нелінійних динамічних об'єктів	43
1.3.1. Класифікація методів інтерпретації моделей машинного навчання	43
1.3.2. Візуалізація моделей	45
1.3.3. Числові оцінки моделей	46
1.3.4. Аналітичні моделі	47
1.4. Непараметричні динамічні моделі на основі інтегро-степеневих рядів Вольтерри	48
1.5. Методи оцінювання інтерпретованості моделей машинного навчання	52
1.6. Постановка задач дослідження	55
1.7. Висновки за розділом	56
РОЗДІЛ 2 МАТЕМАТИЧНІ МОДЕЛІ ІНТЕРПРЕТАЦІЇ ДИНАМІЧНИХ НЕЙРОННИХ МЕРЕЖ НА ОСНОВІ РЯДУ ВОЛЬТЕРРИ	59
2.1. Ідентифікація моделей нелінійних динамічних об'єктів у вигляді інтегро-степеневих поліномів Вольтерри в часовій області	60

2.2. Моделювання нелінійних динамічних об'єктів за допомогою нейронних мереж з часовими затримками	66
2.3. Інформаційний зв'язок між моделями Вольтерри та нейронними мережами з часовими затримками	71
2.4. Дослідження сурогатних моделей у вигляді поліномів Вольтерри	74
2.4.1. Імітаційна модель тестового об'єкта	75
2.4.2. Модель об'єкта на основі нейронної мережі з часовою затримкою	76
2.4.3. Побудова інтерпретаційних моделей	79
2.5. Висновки за розділом	84
РОЗДІЛ 3 МЕТОД ТА ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ІНТЕРПРЕТАЦІЇ МОДЕЛЕЙ МАШИННОГО НАВЧАННЯ	85
3.1. Метод інтерпретації моделей машинного навчання нелінійних динамічних об'єктів	85
3.2. Інформаційна технологія інтерпретації моделей машинного навчання нелінійних динамічних об'єктів	88
3.3. Дослідження моделей машинного навчання динамічних об'єктів з гладкими та значними нелінійностями	94
3.3.1. Дослідження масштабованості TDNN моделей до різних вхідних сигналів	98
3.3.2. Дослідження інтерполяційних та екстраполяційних властивостей TDNN моделей	101
3.4. Висновки за розділом	105
РОЗДІЛ 4. ЗАСОБИ КОМП'ЮТЕРНОГО МОДЕЛЮВАННЯ ТА ПРАКТИКА ЇХ ЗАСТОСУВАННЯ ПРИ РОЗВ'ЯЗУВАННІ ЗАДАЧ ІНТЕРПРЕТАЦІЇ МОДЕЛЕЙ МАШИННОГО НАВЧАННЯ	107
4.1. Проектування інструментальних програмних засобів інтерпретації моделей машинного навчання	107
4.1.1. Проектування інструментальних програмних засобів інтелектуальної системи	108
4.1.2. Вимоги до інструментальних програмних засобів комп'ютерного моделювання	111

4.2. Модель якості програмного забезпечення інтерпретації моделей машинного навчання	118
4.2.1. Визначення компонентів мета-моделі якості програмного забезпечення та метрик для їх кількісної оцінки	119
4.2.2. Побудова багатофакторної моделі якості моделей машинного навчання	122
4.3. Розробка інструментальних програмних засобів інтерпретації моделей машинного навчання	126
4.3.1. Модульно-орієнтована архітектура інструментальних засобів	126
4.3.2. Технології розробки програмного забезпечення розгортання моделей	130
4.4. Організація структур даних та моделей	136
4.4.1. Формати і типи даних	136
4.4.2. Серіалізація моделей	138
4.5. Інтерфейс системи інтерпретації моделей машинного навчання	139
4.6. Висновки за розділом	146
ЗАГАЛЬНІ ВИСНОВКИ	148
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	151
ДОДАТОК А. Програма генерації датасету для імітаційної моделі тестового нелінійного динамічного об'єкта	170
ДОДАТОК Б. Реалізація засобів комп'ютерного моделювання для розв'язування задач інтерпретації моделей машинного навчання	172
ДОДАТОК В. Список публікацій здобувача	177
ДОДАТОК Г. Відомості про апробацію результатів дисертації	180
ДОДАТОК Д. Документи про впровадження результатів дисертації	182

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

- AIC – інформаційний критерій Акаїке (Akaike Information Criterion)
- BIC – байєсівський інформаційний критерій (Bayesian Information Criterion)
- DNN – динамічні нейронні мережі (Dynamic Neural Network)
- LIME – Local Interpretable Model-agnostic Explanations
- MAE – середнє абсолютне відхилення (Mean Absolute Error)
- MAPE – нормована абсолютна помилка (Mean Absolute Percentage Error)
- MSE – середньоквадратична помилка (Mean Squared Error)
- MSPE – нормована середньоквадратична помилка (Mean Squared Percentage Error)
- R^2 – коефіцієнт детермінації
- ReLU – функція активації у вигляді випрямляючої лінійної ланки (Rectified Linear Unit)
- SHAP – SHapley Additive exPlanations
- TDNN – нейронна мережа з часовими затримками (Time Delay Neural Network)
- UI – інтерфейс користувача (User Interface)
- UX – досвід користувача (User Experience)
- БД – база даних
- ІС – інформаційна система
- ІТ – інформаційна технологія
- ІЗ – інструментальні засоби
- ММ – математична модель
- НДР – науково-дослідницька робота
- НМ – нейронна мережа
- ОД – об'єкт дослідження
- ПЗ – програмне забезпечення
- ПС – парціальна складова
- ПФ – перехідна функція

ВСТУП

Актуальність теми. Сучасні процеси розробки, виробництва та експлуатації у більшості галузей характеризуються зростаючою роллю моделювання. При цьому великий інтерес викликають прикладні завдання моделювання складних технічних та біологічних об'єктів, що належать до класу нелінійних динамічних об'єктів з безперервними характеристиками та невідомою структурою. Такі об'єкти прийнято розглядати як «чорну скриню».

При моделюванні об'єктів типу «чорний ящик» велике поширення набуває нейромережвий підхід, оскільки традиційні детерміновані методи не підходять для моделювання таких об'єктів. Однак, недоліком нейромережвих моделей є складність інтерпретації їхньою людиною. Моделі у вигляді нейронних мереж (НМ) не дають аналітичного вираження залежності між входом та виходом об'єкта. Для аналізу властивостей складних систем зручно використовувати саме аналітичну модель, яка дозволяє робити наочні та однозначні висновки про функціонування системи. Тому широке поширення нейромережвих моделей суттєво стримується в таких критичних галузях, як медицина, фінанси, транспорт, де є підвищені вимоги до безпеки обґрунтування довіри моделі.

В останні роки помітно зростає інтерес до досліджень у напрямі інтерпретації роботи чорних ящиків на основі НМ, щоб підвищити довіру до цих моделей, проаналізувати структуру та закони функціонування об'єктів дослідження (ОД).

Таким чином, актуальність теми дослідження визначається наявним протиріччям між сталим ускладненням моделей машинного навчання при ідентифікації нелінійних динамічних об'єктів типу «чорна скриня», з одного боку, та зростанням труднощів інтерпретації цих моделей людиною — з іншого боку. Встановлене протиріччя може бути подолано з застосуванням методу побудови сурогатних моделей на основі непараметричних динамічних моделей у вигляді інтегро-степеневих поліномів Вольтерри, що дозволяє спростити структуру сурогатних моделей та, завдяки урахуванню нелінійних та динамічних

властивостей моделей машинного навчання, підвищити точність ідентифікації сурогатних моделей.

Зв'язок роботи з науковими програмами, планами, темами. Дисертація виконувалася у відповідності до переліку пріоритетних тематичних напрямків наукових досліджень і науково-технічних розробок на період до 2023 року (Постанова Кабінету Міністрів України від 7.09.2011 р. №942 із змінами), згідно пп. 1.2.1.1, 1.2.1.2, 1.2.1.4 і 1.2.4.6 «Основних наукових напрямів та найважливіших проблем фундаментальних досліджень у галузі природничих, технічних, суспільних і гуманітарних наук Національної академії наук України на 2019–2023 роки» (постанова Президії НАН України від 30.01.2019 р. № 30), та у межах держбюджетної науково-дослідної роботи «Методи та програмні засоби інтерпретації моделей машинного навчання непараметричних динамічних об'єктів», №ДР 0122U002161 (Національного університету «Одеська політехніка», 2021 — 2025 р.р.).

Метою роботи є підвищення точності сурогатних моделей шляхом розвитку методу інтерпретації моделей машинного навчання нелінійних динамічних об'єктів за допомогою непараметричних динамічних моделей на основі інтегро-степеневих поліномів Вольтерри та реалізації запропонованого методу у вигляді програмно-алгоритмічних засобів ідентифікації неперервних об'єктів у складі інтелектуальних систем.

Для досягнення вказаної мети дослідження в роботі поставлено і розв'язано такі **завдання**:

– виконано аналіз існуючих методів інтерпретації моделей машинного навчання неперервних нелінійних динамічних об'єктів, визначено базові проблеми інтерпретації нейромережових моделей; обґрунтовано вибір напрямку досліджень в області побудови сурогатних моделей у вигляді інтегральних непараметричних динамічних моделей на основі інтегро-степеневих поліномів Вольтерри для аналітичного опису нелінійних динамічних об'єктів;

– запропоновано використання непараметричних динамічних моделей на основі інтегро-степеневих поліномів Вольтерри в якості сурогатних моделей для

інтерпретації НМ з часовими затримками; встановлено зв'язок в аналітичному вигляді між нейронними мережами з часовими затримками та непараметричними динамічними моделями на основі інтегро-степеневих поліномів Вольтерри для підвищення точності сурогатних моделей неперервних нелінійних динамічних об'єктів;

– розвинуто метод інтерпретації НМ з часовими затримками шляхом побудови сурогатних моделей у вигляді непараметричних динамічних моделей на основі інтегро-степеневих поліномів Вольтерри для опису нелінійних динамічних об'єктів із забезпеченням прийнятної точності моделювання;

– розроблено інформаційну технологію інтерпретації моделей машинного навчання нелінійних динамічних об'єктів, що ґрунтується на використанні НМ з часовими затримками для ідентифікації нелінійної динамічної моделі об'єкта за даними експерименту «вхід–вихід» та побудови сурогатної моделі у вигляді непараметричних динамічних моделей на основі інтегро-степеневих поліномів Вольтерри із забезпеченням прийнятної точності моделювання для інтерпретації НМ;

– розроблено інструментальні засоби комп'ютерного моделювання для побудови нейромережових моделей нелінійних динамічних об'єктів та сурогатних моделей для їх інтерпретації в системі ідентифікації неперервних об'єктів; застосовано розроблений метод та інструментальні засоби для розв'язання прикладних задач ідентифікації неперервних об'єктів.

Об'єкт досліджень – процес інтерпретації моделей машинного навчання неперервних нелінійних динамічних об'єктів.

Предмет досліджень – моделі та методи інтерпретації нейромережових моделей нелінійних динамічних об'єктів, орієнтовані на впровадження в системах ідентифікації неперервних об'єктів.

Методи дослідження. В теоретичних дослідженнях використано положення теорії машинного навчання для побудови моделей ОД; теорії непараметричної ідентифікації на основі рядів Вольтарри для побудови сурогатних моделей. Експериментальні дослідження базуються на методах організації комп'ютерних засобів моделювання та обчислювального

експерименту для чисельного розв'язування тестових задач та підтвердження одержаних теоретичних результатів; методах програмної інженерії для розробки інструментального забезпечення.

Наукова новизна отриманих результатів, які виносяться на захист, полягає в наступному.

– *Вперше запропоновано* сурогатну модель у вигляді непараметричної динамічної моделі на основі інтегро-степеневих поліномів Вольтерри для інтерпретації НМ з часовими затримками, побудованих за результатами експерименту «вхід–вихід», яка, на відміну від відомих моделей, дозволяє одночасно спростити структуру сурогатних моделей та підвищити точність їх ідентифікації, завдяки урахуванню нелінійних та динамічних властивостей моделей машинного навчання.

– *Отримав подальший розвиток* метод оцінки ядер Вольтерри за допомогою НМ з часовими затримками шляхом встановлення аналітичного зв'язку між НМ з активацією у вигляді лінійного випрямляча та непараметричною динамічною моделлю на основі інтегро-степеневих поліномів Вольтерри, що дозволяє підвищити точність ідентифікації нелінійних динамічних об'єктів у вигляді поліномів Вольтерри.

– *Вперше запропоновано* метод інтерпретації моделей машинного навчання у вигляді НМ з часовими затримками, збудованими за даними експерименту «вхід–вихід» шляхом побудови сурогатної моделі на основі непараметричних динамічних моделей у вигляді інтегро-степеневих поліномів Вольтерри за допомогою встановленого аналітичного зв'язку між НМ та ядрами Вольтерри. Це дозволяє підвищити точність та зменшити розмірність сурогатних моделей при ідентифікації нелінійних динамічних об'єктів.

– *Удосконалено* математичну модель оцінки якості програмного забезпечення інтерпретації моделей машинного навчання, яка відрізняється від існуючих одночасним врахуванням метрик функціональності та складності, що дозволяє кількісно виразити здатність до інтерпретації для формального порівняння сурогатних моделей, збудованих різними методами та автоматизації

відбору моделей машинного навчання та сурогатних моделей.

Практичне значення одержаних результатів полягає у розробці інструментальних програмно-алгоритмічних засобів інформаційної технології інтерпретації моделей машинного навчання нелінійних динамічних об'єктів, які реалізують обчислювальні алгоритми ідентифікації нелінійної динамічної моделі об'єкта за даними експерименту «вхід–вихід» за допомогою нейронної мережі з часовими затримками та побудови сурогатної моделі у вигляді непараметричних динамічних моделей на основі інтегро-степеневих поліномів Вольтерри із забезпеченням прийнятної точності моделі машинного навчання, а також у впровадженні створених інструментальних засобів у навчальний процес та наукові дослідження.

На основі створених ІЗ розроблено ефективні хмарні сервіси для вирішення ресурсномістких задач моделювання, що забезпечує високу обчислювальну потужність та швидкість побудови моделей в задачах ідентифікації неперервних об'єктів.

Розроблені ІЗ та хмарні сервіси впроваджено в ТОВ «Дискрет» (м. Одеса), де вони використовуються для ідентифікації моделей тензометричного зважування транспортних засобів у русі та їх інтерпретації.

Результати, отримані в дисертаційній роботі, використано у навчальному процесі Національного університету «Одеська політехніка» при викладанні дисциплін «Інтелектуальні системи» та «Математичне моделювання нелінійних динамічних процесів» на кафедрі комп'ютеризованих систем та програмних технологій; а також застосовано при розробці тем магістерських кваліфікаційних робіт.

Особистий внесок здобувача. Всі наукові положення та висновки, які викладено в дисертації та винесено на захист, одержано здобувачем самостійно. Роботи [1, 12] здобувачем виконано особисто. В наукових працях, написаних у співавторстві, здобувачу належать: сурогатні моделі на основі інтегро-степеневих поліномів Вольтерри для інтерпретації нейронних мереж з часовими затримками [5, 6]; встановлення зв'язку між нейронними мережами з часовими затримками та

непараметричними динамічними моделями на основі інтегро-степеневих поліномів Вольтерри [2]; розвиток методу оцінки ядер Вольтерри за допомогою нейронних мереж з часовими затримками [5, 6, 7]; метод інтерпретації моделей машинного навчання у вигляді нейронних мереж з часовими затримками на основі інтегро-степеневих поліномів Вольтерри [3, 8, 10]; математична модель оцінки якості програмного забезпечення [2, 4]; програмно-алгоритмічні засоби ідентифікації неперервних об'єктів в системах високопродуктивних обчислень [9, 11].

Апробація результатів дисертації. Основні положення та результати дисертаційної роботи доповідалися та обговорювалися на XII та XIII Міжнародних наукових конференціях студентів та молодих вчених «Сучасні інформаційні технології — 2022 (MIT2022)» і «Сучасні інформаційні технології — 2023 (MIT-2023)» (Одеса, 2022, 2023); IV Міжнародній науково-практичній Інтернет-конференції "Математика та інформатика в науці й освіті: виклики сучасності" (Вінниця, 2023); IEEE 3rd International Conference on System Analysis & Intelligent Computing (Kiyv, 2022); 18th Conference on Computer Science and Intelligence Systems. (Warsaw, Poland, 2023); II Міжнародній науково-практичній конференції «Біоінтенсивні та SMART-технології у тваринництві» (Одеса, 2023); Молодіжному науковому семінарі кафедри комп'ютеризованих систем і програмних технологій для аспірантів, магістрів і студентів «Інтелектуальні інформаційні технології в системах контролю та управління» (Одеса, 2023).

Публікації. Основні наукові результати дисертації викладено у 12 публікаціях, у тому числі: 4 статті – в наукових фахових виданнях України категорії “Б”, 7 публікації у працях і матеріалах міжнародних наукових конференцій (матеріали двох з яких проіндексовано у наукометричних базах Scopus та Web of Science Core Collection), 2 статті надруковано без співавторів.

Структура та обсяг роботи. Дисертація має загальний обсяг 185 сторінок і складається з вступу, чотирьох розділів, висновків, списку використаних джерел з 177 найменувань, 5 додатків, 7 рисунків та 5 таблиць на окремих сторінках. Основний текст дисертації викладено на 169 сторінках.

РОЗДІЛ 1

АНАЛІЗ ПРОБЛЕМИ ІНТЕРПРЕТАЦІЇ МОДЕЛЕЙ МАШИННОГО НАВЧАННЯ В СИСТЕМАХ ІДЕНТИФІКАЦІЇ НЕПЕРЕРВНИХ ДИНАМІЧНИХ ОБ'ЄКТІВ

Протягом довгого часу сфера штучного інтелекту була зосередженою, здебільшого, на розвитку теоретичних досліджень. В останньому десятилітті ця ситуація кардинальним чином змінилася завдяки суттєвому стрибку обчислювальної потужності комп'ютерної техніки у поєднанні з розвитком алгоритмів обробки інформації та навчання і доступом до великих обсягів даних.

Результатом цього став істотний прогрес у машинному навчанні, що призвів до підвищення точності моделювання і, як наслідок, стрімкого розповсюдження штучного інтелекту в різних сферах діяльності [11, 25, 34]. При цьому, точність моделей машинного навчання з їхньою здатністю автоматично виявляти, вивчати та отримувати корисні знання з великих обсягів даних досягається за рахунок збільшення складності самої моделі [59, 121]. В результаті, знижується можливість пояснення принципів роботи такої моделі і, як наслідок, отримані результати стають досить складними для розуміння людиною, тобто, погіршується інтерпретованість моделі [27, 97].

Визначення 1. Інтерпретованість – це ступінь, до якого людина може зрозуміти причину рішення [59]. При цьому, розрізняють глобальну та локальну інтерпретованість. Глобальна інтерпретованість означає пояснення p , чому модель f передбачила \hat{Y} для всіх випадків в X , на основі визначення умовної взаємодії між незалежними та залежними змінними $\sigma(\hat{Y}, X)$. Локальна інтерпретованість пояснює, чому модель f передбачила \hat{y} для випадку x , $x \in X$, на основі умовної взаємодії $\sigma(\hat{y}, x)$ між незалежною та залежними змінними.

Визначення 2. Пояснення p для прогнозу $\hat{y}=f(x)$ – це об'єкт, отриманий з моделі f за допомогою деякої функції $p = \sigma(f, x)$, $p \in P$, де P – область інтерпретації людиною.

Інтерпретованість є важливою властивістю моделей машинного навчання. Вона полегшує процеси контролю та діагностування об'єктів та процесів, пояснюючи, з яких причин отримано те чи інше рішення, що сприяє поліпшенню моделей та зростанню довіри до їх використання [5, 31, 48, 97].

Так, ускладнення моделей машинного навчання обмежує їх використання з огляду на складності інтерпретації людиною у критично важливих сферах діяльності. Це викликає протиріччя між складністю ОД та інтерпретованістю його моделі. Потреба практики в ефективних моделях, які б легко сприймалися людиною, призвела до розвитку напряму інтерпретованого штучного інтелекту, що сфокусував свою увагу на розумінні та інтерпретації функціонування інтелектуальних систем [52, 97].

Для вирішення означеного протиріччя необхідно виконати детальний аналіз проблеми інтерпретації моделей машинного навчання в процесах ідентифікації неперервних об'єктів в складі інтелектуальних систем, виконати систематизацію моделей машинного навчання неперервних динамічних об'єктів у вигляді НМ, розглянути методи інтерпретації моделей машинного навчання нелінійних динамічних об'єктів. Для порівняння, до огляду бажано додати непараметричні динамічні моделі на основі інтегро-степеневих рядів Вольтерри, як ефективний апарат опису нелінійних динамічних об'єктів.

На основі наведеного аналізу треба визначити проблематику інтерпретації математичних моделей машинного навчання нелінійних динамічних об'єктів в складі інтелектуальних систем, виокремити задачі інтерпретації моделей машинного навчання з забезпеченням необхідної точності ідентифікації.

1.1. Аналіз застосування математичних моделей інтерпретації в складі інтелектуальних систем

Розвиток науки і технологій дозволяє забезпечити якісне зростання характеристик сучасних пристроїв і процесів у різних галузях діяльності. З іншого

боку, цей безперервний процес веде до постійного ускладнення ОД, посилення вимог до їх функціонування [25, 72].

Для успішної взаємодії з подібними об'єктами (розв'язання задач контролю, управління, діагностування) насамперед необхідно забезпечити їхнє адекватне математичне забезпечення та ефективні засоби моделювання. Це досягається насамперед за рахунок збільшення складності моделей [59, 121].

Ускладнення моделі, як правило, погіршує її інтерпретованість з кількох причин [15, 97]:

- *зростання розмірності*: збільшення кількості параметрів моделі призводить до ускладнення її інтерпретування, особливо, при наявності залежних параметрів;

- *нелінійність*: зростання порядку нелінійності моделі ускладнює встановлення закономірностей, що діють між вхідними та вихідними змінними та визначення ознак, що впливають на поведінку моделі;

- *обчислювальні навантаження*: збільшення обчислювальних ресурсів для навчання і застосування обмежує доступність і використання моделей, особливо в задачах з обмеженими ресурсами.

У ряді галузей, де є підвищені вимоги до адекватності моделі, а саме, де модель може істотно впливати на життя людей, наприклад: медицина, фінанси, транспорт, кібербезпека – інтерпретованість є не тільки бажаною властивістю моделі, але й невід'ємною вимогою до моделей машинного навчання, закріпленою юридично. Прикладом цього є європейський регламент GDPR [44], що вимагає право пояснення рішення, прийнятого моделлю.

Більший науковий та практичний інтерес викликають задачі моделювання складних об'єктів з невідомими законами функціонування та невідомою структурою, коли застосування простих інтерпретованих моделей не призводить до задовільного результату. Такі об'єкти прийнято розглядати як «чорну скриню» [5, 113, 121].

Визначення 3. Нехай f є моделлю «чорної скриньки», внутрішній принцип роботи та параметри θ якої або невідомі, або відомі, але не можуть бути

інтерпретовані людиною. Модель f є інтерпретованою, якщо її параметри $\hat{\theta}$ відомі, і існує достатньо знань про те, як і чому модель генерує певні прогнози [59].

У зв'язку з цим великого значення набувають задачі побудови адекватних інтерпретуючих моделей машинного навчання для об'єктів типу «чорна скринька» [5, 59, 97, 111, 113], що дозволяє покращити адекватність моделей та сприяє зростанню довіри до їх використання.

В якості прикладів об'єктів типу «чорна скриня» можна розглядати нелінійні динамічні об'єкти з невідомими законами функціонування та невідомою структурою [111]. Завдяки нелінійним динамічним характеристикам ОД можуть функціонувати в більш складних режимах, які не можуть бути реалізовані за допомогою лінійних характеристик. Для таких об'єктів характерна деяка апіорна невизначеність: брак даних про об'єкти, присутність перешкод та збурень зовнішнього середовища [168]. Тому, традиційні детерміновані методи не підходять для моделювання таких об'єктів [112].

При моделюванні об'єктів типу «чорна скриня» велике поширення отримувє нейромережевий підхід [56, 73, 113]. Популярність НМ отримали завдяки тому, що процес побудови моделі потребує тільки вимірювання даних на вході і виході «чорної скрині» і не вимагає жодних припущень про структуру об'єкта і внутрішні закони його функціонування. Тому, останнім часом використання НМ для опису нелінійних динамічних об'єктів, зокрема, з безперервними характеристиками значно розширилося.

Однак через високу нелінійність і складні взаємодії великої кількості параметрів моделі НМ не відображають у явному вигляді структуру і внутрішні закони функціонування об'єкта [51, 87, 133]. Тому, НМ сприймаються як моделі «чорної скрині». Як наслідок, серйозним недоліком нейромережевих моделей є те, що прогнози, зроблені такою складною моделлю, не можна простежити до вхідних даних та зрозуміти чому вихідні дані перетворюються певним чином [59, 97].

Таким чином, складні взаємодії великої кількості параметрів НМ нелегко простежити, у той час як їх розплутування може дати уявлення про процеси, що

відображені в об'єкті, та параметри, на яких засновані рішення моделі. Як результат, моделі у вигляді НМ не дають наочного аналітичного вираження залежності між множиною вхідних $\mathbf{X}=(x_1, x_2, \dots, x_v)$ і вихідних $\mathbf{Y}=(y_1, y_2, \dots, y_\mu)$ сигналів об'єкта. При цьому, для аналізу властивостей динамічного об'єктів типу «чорна скринька» зручно використовувати саме аналітичну модель, яка дозволяє робити наочні та однозначні висновки про структуру об'єкта та його функціонування у вигляді оператора перетворення $A_\tau(t)$ [174]:

$$\mathbf{Y}(t)=A_\tau(t)[\mathbf{X}(\tau)], \quad (1.1)$$

де $t_0 \leq \tau \leq t \leq T$, t_0 , T – час почату та закінчення спостереження відповідно.

В результаті, при моделюванні об'єктів типу «чорна скриня», доводиться мати справу з моделями, принципи роботи яких не очевидні, а ознаки найчастіше не мають фізичного сенсу, що ускладнює інтерпретацію моделей людиною. Як наслідок, широке поширення нейромережевих моделей суттєво стримується в таких критичних областях, як медицина, фінанси, транспорт, де присутні підвищені вимоги до безпеки результатів моделювання і довіри до моделі.

З іншого боку, ситуація, що склалася в останні роки стимулює інтерес до досліджень у напрямку інтерпретації роботи «чорних скринь» на основі НМ, щоб підвищити довіру до цих моделей, проаналізувати структуру і закони функціонування ОД [59, 97, 113]. Тому, актуальним завданням лишається розвиток напряму інтерпретації нейромережевих моделей нелінійних динамічних об'єктів.

Процес інтерпретації моделей типу «чорна скриня» тісно пов'язані з процесами ідентифікації та машинного навчання. Узагальнено, зв'язок між цими процесами наведений на рис. 1.1.

На етапі ідентифікації визначається математична модель $A_\tau(t)$, яка найкращим чином відповідає даним спостережень сигналів $x(t) \in \mathbf{X}$ та $y(t) \in \mathbf{Y}$ (експеримент «вхід/вихід»).

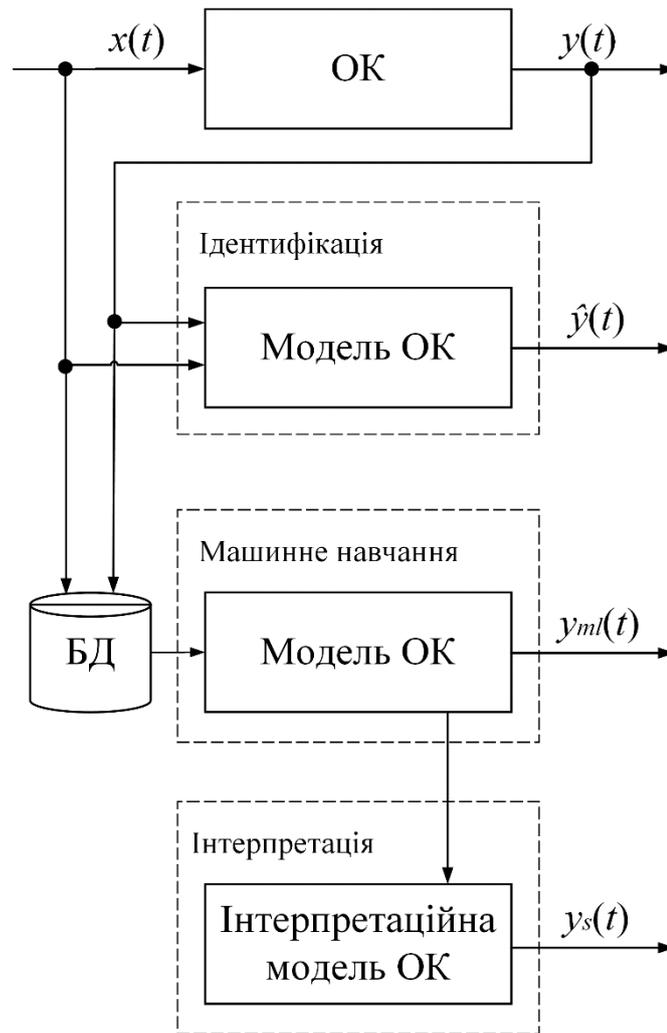


Рисунок 1.1 – Зв'язок процесу інтерпретації моделей з процесами ідентифікації та машинного навчання.

Головні переваги ідентифікації математичних моделей при правильному виборі класу моделей і підборі її параметрів полягають у наступному:

- висока точність моделі;
- низька обчислювальна складність методів моделювання;
- масштабованість моделі на різні завдання і дані.

Недоліки ідентифікації математичних моделей полягають у наступному:

- низька точність при помилковому виборі класу моделей та підборі її параметрів;
- залежність від якості даних та дії завад.

Якщо математична модель не відповідає даним експерименту «вхід/вихід» (не задовольняє вимогам точності при зіставленні з результатами спостережень) внаслідок невдалого визначення класу моделі або оцінки її параметрів, то складність моделі підвищують.

Поширеним підходом побудови ускладнених моделей є методи машинного навчання. Цей підхід базується на формуванні бази даних (БД) зі спостережень вхідних та вихідних сигналів та складається з кількох етапів [13, 83, 139].

Етап 1. Підготовки даних: очищення даних від викидів і помилок, нормалізація і шкалювання ознак, поділ даних на тренувальні, валідаційні та тестові набори.

Етап 2. Вибір моделі: лінійні моделі, дерева рішень, НМ або інше.

Етап 3. Навчання моделі: визначення параметрів моделі, які найкраще відповідають вхідним даним.

Етап 4. Інтерпретація моделі: пояснення, як окремі ознаки моделі впливають на цільову змінну.

Головні переваги моделей машинного навчання полягають у наступному [13, 102]:

- висока точність за рахунок зростання порядку нелінійності моделі;
- здатність до навчання на великих обсягах даних;
- адаптивність до нових даних і умов, що змінюються в часі;

Недоліки моделей машинного навчання полягають у наступному [102, 139]:

- вимоги до мінімальних обсягів даних;
- висока обчислювальна складність методів моделювання;
- схильність до перенавчання, що погіршує точність моделі;
- складність інтерпретації моделей.

В критично важливих областях, як згадувалося раніше, завершальним етапом машинного навчання є процес інтерпретації отриманої моделі об'єкта. Результатом інтерпретації є спрощена (сурогатна) модель, що відбиває параметри об'єкту з заданою точністю та є зрозумілою для людини (зацікавлених сторін та кінцевих користувачів).

Головні переваги сурогатних моделей полягають у наступному [5, 15, 59]:

- прозорість моделі і довіра до неї, оскільки користувачі можуть бачити і розуміти, як ухвалюються рішення;
- невисока обчислювальна складність;
- спрощення ухвалення рішень шляхом надання інформації про те, які ознаки впливають на вихід об'єкта;
- сприяння покращенню взаємодії людини і машини.

До недоліків сурогатних моделей відносяться [15, 87]:

- зменшення точності за рахунок обмеження складності порівняно з моделями машинного навчання;
- обмеження на типи задач, де складні залежності важко інтерпретувати.

Не всі моделі машинного навчання потребують інтерпретації. Так, наприклад, регресійні моделі, моделі у вигляді неглибоких дерев рішень та моделі k-найближчих сусідів (у просторі інформативних ознак) відносяться до простих моделей і легко сприймаються людиною, тобто. є інтерпретованими. Але ці моделі не задовольняють сучасним вимогам за точністю та надійністю при роботі зі складними об'єктами реального світу. На відміну від них, НМ, градієнтний бустинг – це потужні моделі, принцип роботи яких важко зрозуміти, тому ці моделі потребують додаткової інтерпретації [31, 87].

В табл. 1.1. наведено огляд поширених моделей машинного навчання з позначенням області їх застосування, основних характеристик та здатності до інтерпретації результатів навчання.

Моделі машинного навчання, наведені в табл. 1.1 зручно розділити на 2 частини: статичні та динамічні. Інтерес в даному дослідженні представляють саме динамічні моделі у вигляді НМ, які потребують більш детальної уваги.

Таблиця 1.1 – Поширені моделі машинного навчання

№ з/п	Модель	Тип завдання	Застосування	Цільова функція	Нелінійність	Динаміка	Інтерпретованість
1	2	3	4	5	6	7	8
1	Лінійна регресія	Регресія	Прогнозування числових значень на основі лінійної залежності ознак	$y = a_0 + \sum_{i=0}^n a_i x_i$ n – розмірність простору	Ні	Ні	Висока
2	Логістична регресія	Класифікація	Оцінка ймовірності приналежності об'єкта до одного з класів	$P(y=1 x) = f(z),$ $z = a_0 + \sum_{i=0}^n a_i x_i$ n – розмірність простору	Ні	Ні	Висока
3	Дерева рішень	Класифікація, регресія	Створення структури рішень як дерева для класифікації чи регресії	Правила формату <i>if-else</i> на основі питань про ознаки	Так	Ні	Висока
4	Гradientний бустинг	Класифікація, регресія	Комбінація множини слабких моделей поліпшення продуктивності	Сума виважених прогнозів базових моделей.	Так	Ні	Середня
5	Метод опорних векторів	Класифікація, регресія	Пошук оптимальної гіперплощини для поділу класів чи прогнозу числових значень.	Гіперплощина у просторі ознак.	Ні	Ні	Середня

1	2	3	4	5	6	7	8
6	Метод k-найближчих сусідів	Класифікація, регресія, кластеризація	Оцінка класу об'єкта чи числового значення з урахуванням його подібності певному класу	Середні значення ознак у кожному класі	Так	Ні	Середня
7	Асоціативний аналіз	Пошук асоціацій	Виявлення асоціативних правил у даних	Правила формату <i>if-else</i>	Ні	Ні	Середня
8	Гаусівські процеси	Імовірнісна класифікація, регресія	Прогнозування значень цільової змінної	$y=f(x)$	Так	Ні	Висока
9	Нейронні мережі прямого розповсюдження сигналу	Класифікація, регресія	Вирішення різних завдань машинного навчання за допомогою мережі нейронів	НМ, що апроксимує нелінійні функції високих порядків	Так	Ні	Низька
10	Динамічні нейронні мережі	Регресія	Обробка послідовних даних: часові ряди, тексти, звукові сигнали тощо	НМ з прямим розповсюдженням сигналу	Так	Так	Низька

Динамічні нейронні мережі (Dynamic Neural Network, DNN) – це клас нейронних мереж, які можуть відтворювати динамічну поведінку об'єкта після завершення процедури навчання. Вони можуть бути використані в множині задач моделювання, комп'ютерного зору, обробки природної мови, прогнозування часових рядів, управління роботами та іншими пристроями [3, 32, 101, 161, 167]. ДНМ можуть бути реалізовані за допомогою різних архітектур.

Аналіз даних наведених в табл. 1.1 свідчить про те, що сучасним, універсальним та потужним апаратом моделювання динамічних об'єктів будь-якої складності є саме моделі у вигляді DNN.

1.2. Класифікація моделей неперервних динамічних об'єктів у вигляді нейронних мереж

Моделі динамічних об'єктів являють собою вирази, які описують зміну стану об'єкта в часі (для моделей безперервних ОД – в безперервному часі). Ці моделі широко використовуються в інженерії, економіці, біології та інших галузях для аналізу та прогнозування динамічних процесів [95, 153, 173]. Нижче розглядаються деякі поширені типи моделей безперервних динамічних об'єктів, розділені за принципом наявної апріорної інформації про об'єкт. З цього погляду, оператор динамічного об'єкта $A_{\tau}(t)$ може бути записаний у однієї з двох форм [29, 168]:

- параметричній, коли оператор $A_{\tau}(t)$ має вигляд певного математичного виразу і має фіксовану кількість параметрів; використовуються, коли наявна апріорна інформація про структуру ОД та форму залежності його входів та виходів.

- непараметричній, коли оператор $A_{\tau}(t)$ не має заданої форми математичного виразу і може мати більш гнучку структуру, адаптуватися до даних і мати нескінченну кількість параметрів; використовуються, коли відсутня повністю або частково апріорна інформація про структуру ОД та форму залежності його входів та виходів.

Так, параметричні моделі зазвичай менш гнучкі, оскільки форма моделі фіксована. Це може бути перевагою, якщо дані відповідають передбачуваній моделі, але недоліком, якщо модель занадто проста або недостатньо гнучка для адекватного опису даних. Параметричні моделі є інтерпретованими, оскільки задають у явному вигляді залежність між входами та виходами ОД [15, 98, 106].

Непараметричні моделі більш гнучкі та здатні краще підлаштовуватися під дані, особливо якщо форма залежності складна або невідома. Вони можуть виявляти нелінійні та складні взаємозв'язки в даних та точніше їх описувати. Непараметричні моделі, зазвичай, менш інтерпретовані, оскільки їхні параметри можуть бути абстрактними і не мати прямої семантики [48, 65, 161].

За наявності апріорної інформації про ОД непараметричні моделі можна розглядати як [122, 123, 168]:

– «сіра скриня», коли частково доступна інформації про структуру ОД або закони його функціонування;

– «чорна скриня», коли повністю відсутня інформація про структуру ОД та закони його функціонування, об'єкт можна вивчати тільки по його входах і виходах.

Розглянуту класифікацію моделей нелінійних динамічних об'єктів представлено на рис. 1.2.

Як згадувалося раніше, важливе значення для науки та практики мають задачі ідентифікації складних об'єктів, для яких властива відсутність будь-якої інформації про їх закони функціонування та структуру [74, 153, 154]. У таких випадках використання параметричних моделей та моделей «сірої скрині», як правило, не призводять до рішень, які задовольняють вимоги до точності моделювання.

У зв'язку з цим, велика увага приділяється методам побудови адекватних моделей машинного навчання [81, 100, 115, 150, 155, 156], які здатні задовольнити сучасні вимоги до точності моделей, отриманих за результатами ідентифікації. Для ідентифікації цих об'єктів найдоцільніше застосовувати непараметричні динамічні моделі. Вони дають змогу одночасно враховувати нелінійні та динамічні властивості об'єкта, володіють високим ступенем гнучкості і можуть точніше апроксимувати складні залежності в даних [17, 35, 45, 107, 135, 150, 161, 163, 175].

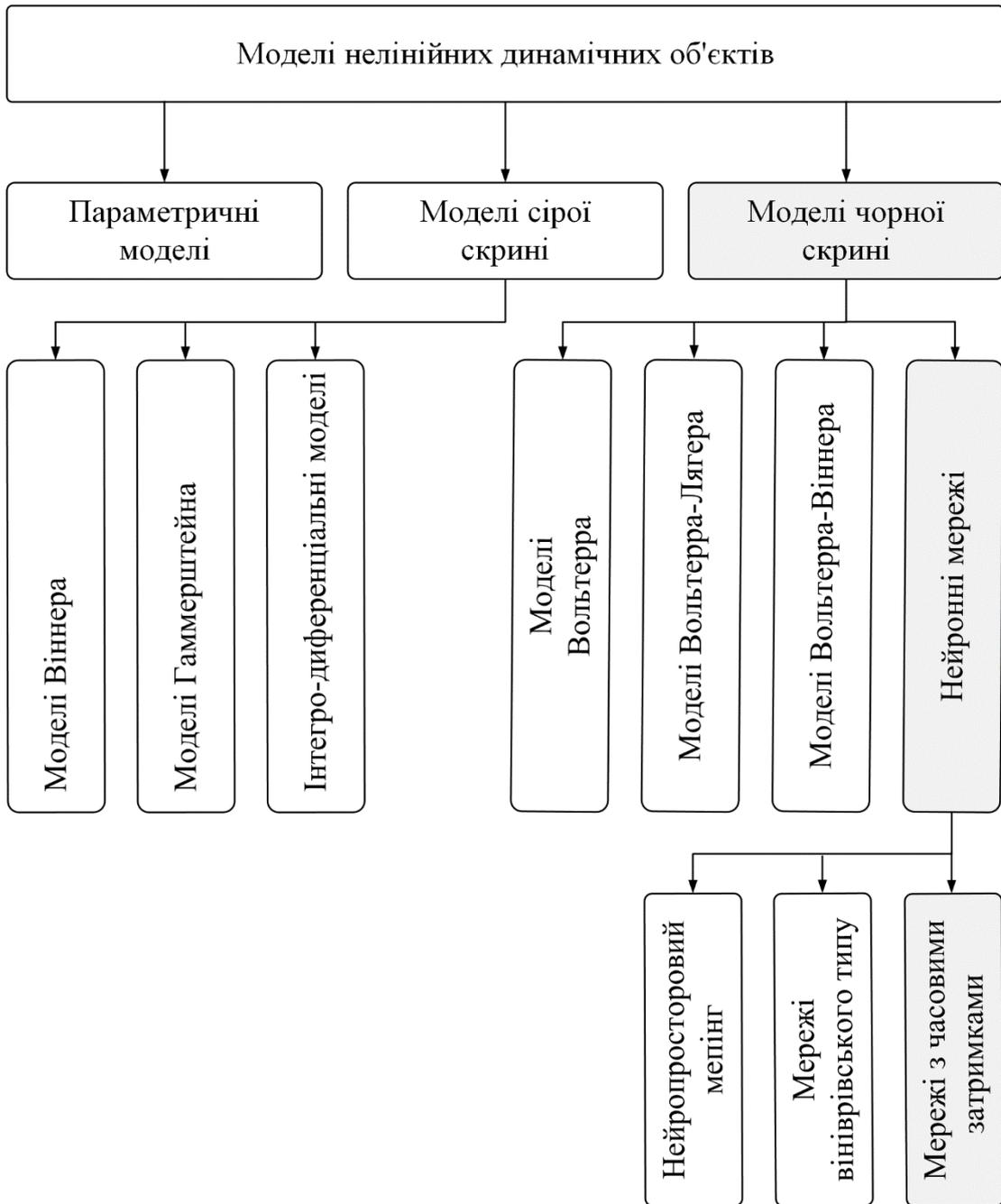


Рисунок 1.2 – Класифікація моделей нелінійних динамічних об'єктів

До найпоширеніших непараметричних моделей «чорної скрині», згідно класифікації, наведеної на рис. 1.2, відносяться аналітичні моделі Вольтерри [18, 30, 33, 64], Вольтерри-Лягера [45], Вольтерри-Віннера [67, 122] та імітаційні моделі у вигляді нейронних мереж [3, 104, 118].

Моделі Вольтерри являють собою потужний апарат математичного моделювання ОД у вигляді інтегро-степеневих рядів, що здатний одночасно

враховувати як нелінійні, так і динамічні характеристики об'єкта. Завдяки цьому, моделі здатні забезпечити високу точність ідентифікації. Однак, існують певні обмеження у використанні моделей Вольтерри: умова на обмеження амплітуди тестових сигналів на вході ОД та ускладнені алгоритми ідентифікації моделей, що потребують значних зусиль і ресурсів, особливо, при побудові моделей вищих порядків [108, 109, 112, 148].

Моделі Вольтерри-Лягер являють собою розширення моделей Вольтерри та дозволяють знизити кількість коефіцієнтів моделі, завдяки застосуванню ортогональних функцій Лягера.

Моделі Вольтерра-Вінера також є розширенням моделей Вольтерри, що дозволяють будувати менш складні вирази для розкладання реакції об'єкту $y(t)$, що ортогональні вхідним сигналом ОД $x(t)$.

Загалом, моделі Вольтерри-Лягера і Вольтерри-Віннера надають більш складні інструменти для ідентифікації ОД. Їхнє використання може вимагати глибшого розуміння математичних обмежень застосування задля забезпечення збіжності інтегро-степеневих рядів.

Цих недоліків позбавлені підходи до ідентифікації, засновані на принципі імітаційного моделювання на основі НМ. Сьогодні відомо кілька поширених методів моделювання нелінійних динамічних об'єктів за допомогою НМ: динамічний нейпросторовий мепінг (Dynamic Neuro-SM) [2, 10], динамічні нейронні мережі вінерівського типу (Wiener-type DNN) [4, 7, 8] та нейронні мережі з часовими затримками (TDNN) [6, 84, 92, 132, 134].

1.2.1. Динамічний нейпросторовий мепінг. Метод статичного нейпросторового відображення [2, 10] засновано на ідеї просторового відображення заданої наближеної (грубої) моделі на бажану (точну) модель за допомогою НМ таких як багатосаровий персептрон. Цей метод ефективно працює при нелінійному моделювання об'єктів але не здатний працювати з динамічними моделями.

Моделі типу Dynamic Neuro-SM – це удосконалені в порівнянні зі статичними моделями нейропросторового мепінгу. Ці моделі створюються шляхом поєднання грубої моделі та DNN. Метою Dynamic Neuro-SM є відображення заданої грубої моделі об'єкта на точну модель шляхом автоматичного зіставлення та зміни існуючої грубої моделі в точну модель шляхом процесу машинного навчання за допомогою DNN, наприклад Wiener-type DNN або TDNN [10]:

$$y(t_n) = S(\mathbf{x}(t_n), \mathbf{w}), \quad (1.2)$$

де S – DNN, що використовується для поточного відображення, \mathbf{w} – вагові коефіцієнти ДНМ, $\mathbf{x}(t_n) = [x(t_n), x(t_{n-1}), \dots, x(t_{n-M+1})]$, $t_n = n\Delta t$, $n = 1, 2, \dots, M$ – загальна кількість кроків затримки, Δt – параметр часової затримки.

З (1.2) випливає, що сигнал на виході грубої моделі залежить не тільки від поточного сигналу точної моделі, але й від історії сигналу, що робить запропоноване відображення динамічним. Оскільки в нейронній мережі задіяні запізнілі сигнали, метод навчання для Dynamic Neuro-SM не може бути застосований безпосередньо. Для навчання цієї DNN необхідно встановити зв'язок динамічного відображення з динамічними характеристиками ОД. Тому, Dynamic Neuro-SM забезпечують підвищення точності порівняно зі статичними моделями нейропросторового мепінгу, проте передбачають наявність деякої апріорної інформації про закони функціонування ОД.

1.2.2. Динамічні нейронні мережі вінерівського типу. Wiener-type DNN базується на принципі побудови нелінійної динамічної системи Віннера, яка складається з послідовно з'єднаних спрощеної лінійної динамічної моделі та нелінійної статичної моделі [4, 7, 8]. Так, динамічні характеристики пов'язані з лінійною частиною; нелінійні характеристики пов'язані з статичною частиною.

Аналітичний запис моделі Wiener-type DNN можна представити у вигляді системи рівнянь [4]:

$$\begin{cases} u''(t) = \mathbf{A}u(t) + \mathbf{B}x(t) \\ y(t) = S(x(t), u(t), \mathbf{w}) \end{cases}, \quad (1.3)$$

де \mathbf{A} та \mathbf{B} – матриці, елементи яких є дійсними коефіцієнтами лінійних динамічних рівнянь, S – невідома статична нелінійна функція, виражена нейронною мережею.

При цьому, лінійні динамічні рівняння використовуються для вираження динамічної частини моделі, статичні нелінійні рівняння використовуються для вираження нелінійної частини моделі.

Оскільки статична нелінійна функція між вихідним сигналом, вхідним сигналом і змінними стану невідома, для її представлення використовується НМ. Це відрізняє Wiener-type DNN від існуючих нейромережових моделей, такими як Dynamic Neuro-SM та TDNN, де нелінійні і динамічні характеристики представлені в єдиній нейронній мережі.

Структурну схему моделі Wiener-type DNN представлено на рис. 1.3. На цьому рисунку сигнал $x(t)$ подається на лінійний динамічний блок і статичний нелінійний блок. Вхідний сигнал $x(t)$ і змінні стану $u(t)$ є входами статичної нелінійної функції.

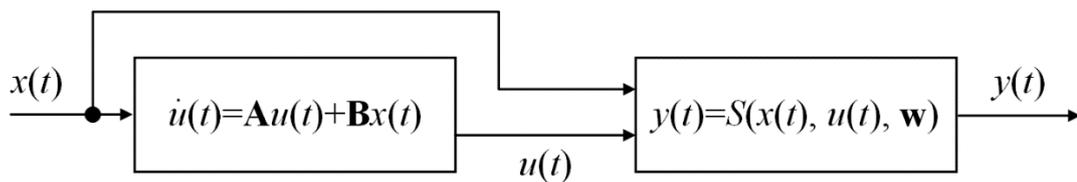


Рисунок 1.3 – Структурна схема моделі Wiener-type DNN

Така структура може суттєво підвищити надійність динамічної нейромережової моделі та покращити збіжність під час нелінійного моделювання. Проте Wiener-type DNN має складну (гібридну) структуру, що висуває додаткові вимоги до алгоритмів навчання мережі та звужує сферу застосування моделі, а також вимагає наявності апріорної інформації про лінійну динамічну частину моделі.

1.2.3. Нейронні мережі з часовими затримками. TDNN – є однією зі структур DNN. Це клас динамічних мереж, які можуть опрацьовувати послідовності даних з урахуванням часових залежностей між ними, такі як часові

ряди, а також моделювати динамічні системи [6, 84, 134]. Вони можуть використовуватися для прогнозування часових рядів, розпізнавання мови, обробки відео та аудіо, генерації тексту, моделювання динамічних систем, а також для управління технічними об'єктами.

Серед означених варіантів нелінійних динамічних нейромережевих моделей TDNN є найбільш загальною структурою, що складаються з кількох шарів з прямим поширенням сигналу. Такі моделі здатні ефективно відтворювати динамічну поведінку ОД з нелінійністю високого порядку [6, 134]. TDNN здатні навчатися на основі даних експерименту «вхід-вихід» нелінійних динамічних об'єктів і мають відмінні властивості збіжності [134, 144], що є перевагами порівняно зі згаданими методами Dynamic Neuro-SM та Wiener-type DNN.

Завдяки своїй простоті та універсальності при моделюванні нелінійних динамічних об'єктів TDNN набули найбільшого поширення. У літературі досить повно представлені дослідження моделей TDNN для динамічних об'єктів із слабкою нелінійністю [68, 134]. Слід зазначити низку публікацій, присвячених інтерпретації TDNN, зокрема встановленню інформаційного зв'язку цих моделей з моделями Вольтерра [84, 120, 130, 136]. Так, моделі TDNN є ефективним інструментом для моделювання нелінійних динамічних об'єктів з неперервними характеристиками.

Існує багато структур нейронних мереж TDNN, що відрізняються кількістю прихованих шарів, функціями активації та топологією. Для спрощення опису моделі на основі TDNN в подальшому розглядається найчастіше використовувана структура TDNN, що складається з трьох шарів: вхідного, прихованого та вихідного [84, 136]. В означеній структурі вхідний шар TDNN включає M нейронів, прихований шар включає K нейронів, вихідний шар включає 1 нейрон.

На рис. 1.4 наведено архітектуру TDNN у вигляді тришарової мережі з прямим поширенням сигналу з M входами, прихованим шаром з K нейронами та одним вихідним нейроном [174].

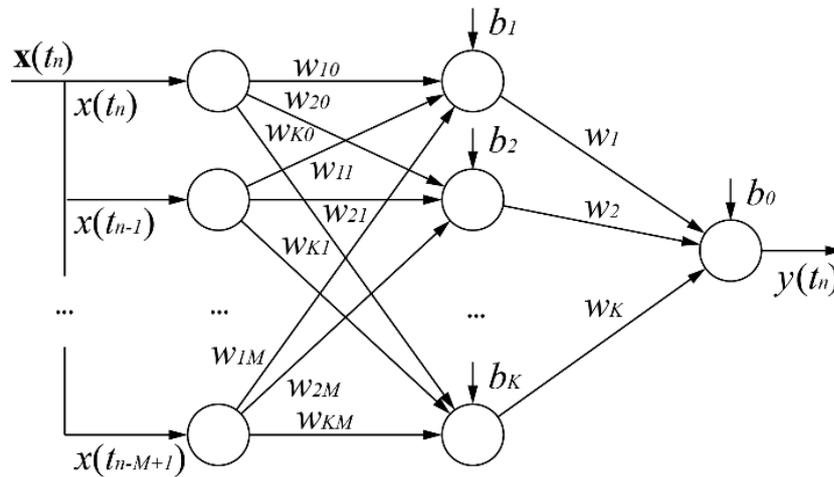


Рисунок 1.4 – Архітектура TDNN у вигляді тришарової мережі з прямим поширенням сигналу

Використання цієї структури дає менш складний у порівнянні з іншими структурами вираз для виходу НМ. Сигнал $y(t_n)$ на вихідному шарі в момент часу t_n залежить від значень вхідного сигналу $\mathbf{x}(t_n)$ та визначаються за виразом [130]:

$$y(t_n) = b_0 + S_0 \sum_{i=1}^K w_i S_i \left(b_i + \sum_{j=1}^M w_{i,j} x(t_{n-j}) \right), \quad (1.4)$$

де b_0, b_i – зсуви нейронів вихідного та прихованого шарів відповідно; S_0, S_i – функції активації нейронів вихідного та прихованого шарів відповідно; $w_i, w_{i,j}$ – вагові коефіцієнти нейронів вихідного та прихованого шарів відповідно.

Часові затримки у мережі реалізуються за допомогою послідовної подачі вхідних даних, зсунутих відносно попередніх на одне значення. Така модель може бути навченою динамічній поведінці з урахуванням нелінійних характеристик на даних експерименту «вхід-вихід».

В більш складних НМ часові затримки реалізуються у декількох прихованих шарах, але це призводить до істотного ускладнення мережі [93] і на практиці не отримало значного поширення.

1.3. Методи інтерпретації моделей машинного навчання нелінійних динамічних об'єктів

1.3.1. Класифікація методів інтерпретації моделей машинного навчання. В напрямку інтерпретації моделей машинного навчання сьогодні зосереджено найбільші зусилля вчених та інженерів-практиків. Всесвітньо відомі ІТ-компанії також бачать в цьому напрямі великий потенціал та створюють власні інструменти для інтерпретації моделей машинного навчання:

- aix360.readthedocs.io від компанії IBM,
- aws.amazon.com/sagemaker від компанії Amazon,
- captum.ai від компанії Facebook,
- explainable.ai від компанії Google,
- interpret.ml від компанії Microsoft.

Аналіз наукових досягнень та наведених сервісів дозволяє встановити існування декількох підходів до процесу інтерпретації моделей [27, 97]. Для визначення ефективних методів інтерпретації моделей машинного навчання нелінійних динамічних ОД доцільно прийняти до уваги всі аспекти існуючих підходів та всебічно обґрунтувати подальший вибір напрямку досліджень. Для цього на рис. 1.5 представлено узагальнену класифікацію методів інтерпретації моделей машинного навчання за різними аспектами розподілу. Наведена класифікація базується на меті, в залежності від якої створюються методи інтерпретації моделей.

Згідно з представленою класифікацією, вирізняють чотири види мети створення методів інтерпретації:

- сприяння покращенню достовірності моделі,
- пояснення складних моделей,
- аналізу чутливості прогнозів моделі,
- створення сурогатних моделей.

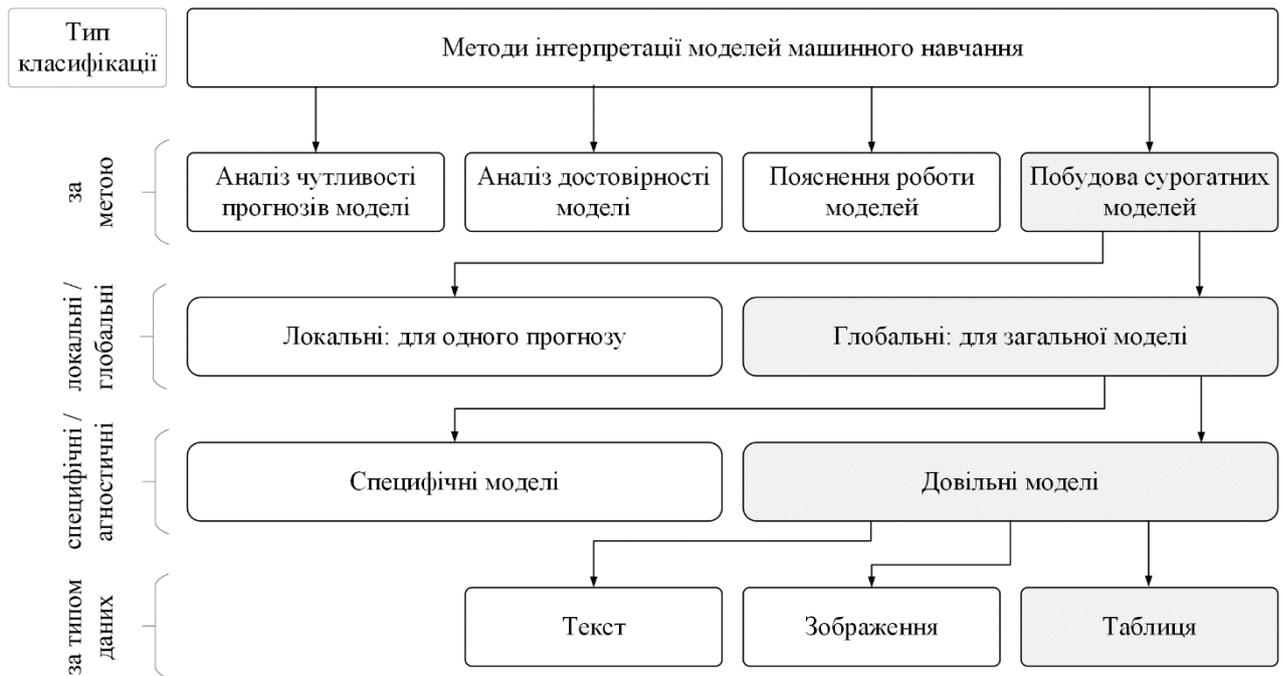


Рисунок 1.5 – Узагальнена класифікація методів інтерпретації моделей машинного навчання

Для розв'язання встановленого у Вступі протиріччя між складністю об'єкта та інтерпретованістю його моделі необхідно вирішувати саме задачу створення нових адекватних сурогатних моделей.

Визначення 4. Нехай x – екземпляр вектору незалежних змінних навчального датасету \mathbf{X} ($x \in \mathbf{X}$), y – вектор залежних змінних (міток датасету), $y \in \mathbf{Y}$, f – модель «чорної скрині», $f(x) = \hat{y} \approx y$. Модель f_S є сурогатною моделлю для f , якщо $f_S(x) - f(x) \leq E$, де E – деяка міра близькості двох функцій.

Розподіл методів інтерпретації за цим напрямом прийнято поділяти за наступними аспектами [27, 97]:

- локальні (метод пояснює рішення лише для певного випадку) та глобальні (метод пояснює загальну модель);
- специфічні (застосування обмежується лише певним сімейством моделей) та агностичні (застосовуються до довільних моделей);
- за типами даних, до яких методи застосовуються: табличні (числові, категоріальні), зображення, текстові.

Наведена класифікація дозволяє визначити напрям дослідження як побудову глобальних агностичних моделей інтерпретації за даними, представленими у табличному вигляді. Поширені на практиці методи інтерпретації моделей машинного навчання, що вирішують задачі у визначеному напрямі, можна розділити на такі категорії [5, 27, 48, 65, 142]:

- візуалізація (2D, 3D графіки, графи тощо),
- побудова текстових моделей у вигляді антецедентів (наприклад, «фактор A і фактор B призвели до передбачення C »),
- розрахунок числових оцінок (міри важливості ознак, коефіцієнти, ваги),
- аналітичні моделі з безпосередньою інтерпретованою залежністю виходу об'єкта від його входу.

Більшим попитом з боку практики сьогодні користуються методи інтерпретації нейронних мереж на основі візуалізації та оцінки важливості ознак [65, 142]. Для інтерпретації нелінійних динамічних моделей машинного навчання як самостійні мало придатними є методи побудови текстових моделей. Проте, найбільший науковий інтерес можуть представляти аналітичні моделі. Для ознайомлення з означеними напрямками, що становлять інтерес для даного дослідження та визначення їх переваг і недоліків далі розглядається кожний з наведених напрямів більш детально.

1.3.2. Візуалізація моделей. Широко відомі методи інтерпретації нейронних мереж, що ґрунтуються на візуалізації процесів прийняття рішень, для обробки нечислових даних: зображень та відео, звуку та мови, тексту [65, 142]. Методи забезпечують глибоке розуміння даних, дозволяють вирішувати задачі перевірки та пояснення даних, підвищення довіри до моделей. Даний підхід до інтерпретації дозволяє візуалізувати процес навчання нейронної мережі та результати навчання у наступних формах:

- скалярні функції у розрізі часу:

$$c(t) = f(y(t), \hat{y}(t)), \quad (1.5)$$

де $\hat{y}(t)$ – оцінка вихідного сигналу ОД. В якості $c(t)$ найчастіше використовують функції втрат (*loss function*), вартості (*cost function*) або точності (*accuracy*) нейронної мережі;

– зображення, сформовані на даних у 2D просторі $c(a, b)=f((a, b))$. В якості c можуть бути використані вагові коефіцієнти мережі w_{ij} ;

– граф моделі $c(b_i, w_{ij}) = \{b_i, w_{ij}\}$;

– гістограма зміни розподілу даних у шарах у часі $c(a)=\{a_i\}$, $i=1, n$, n – розмірність даних.

За графічними інтерпретаційними моделями можна встановити якісні показники значущості і пов'язаності властивостей тестового об'єкта. Але ці моделі не надають інформацію, як саме певні властивості впливають на передбачення моделі.

Переваги: наочне відображення вихідних даних у просторі ознак, оцінки якості процесу машинного навчання моделі.

Недоліки: відсутність числових оцінок взаємозв'язку ознак та їх важливості, залежності у явному вигляді виходу об'єкта від його входу.

1.3.3. Числові оцінки моделей. При інтерпретації нейронних мереж часто використовуються методи, що ґрунтуються на оцінці важливості ознак для пояснення індивідуальних передбачень моделі. Популярним методом оцінки важливості ознак вважається SHAP (SHapley Additive exPlanations) [65, 79].

Даний підхід до інтерпретації дозволяє оцінити процес навчання нейронної мережі та результати навчання у наступних формах:

– розрахунок оцінок значущості і пов'язаності ознак;

– візуалізація оцінок значущості і пов'язаності ознак;

– текстове пояснення (моделі у вигляді антецедентів: «при зменшенні/збільшенні ознаки x_i , зростає ймовірність зменшення/збільшення» передбачення $y(x)$).

Переваги: оцінка важливості ознак, визначення ознак, що впливають на роботу моделі.

Недоліки: відсутність функціональної залежності результату від виявлених ознак, залежності у явному вигляді виходу об'єкта від його входу.

1.3.4. Аналітичні моделі. Аналітичні вирази використовуються для інтерпретації моделей значно рідше. Інтерпретація моделей у вигляді аналітичних виразів дозволяє забезпечити математичну достовірність: прозоро показати відсутність прихованої поведінки чи логіки, які впливають на властивості моделі [57, 97]. Особливої уваги серед моделей машинного навчання потребують нейромережеві моделі. Актуальність задачі інтерпретації нейронних мереж за допомогою аналітичних виразів зростає у зв'язку з тим, що нейронні мережі, здатні нести достовірну інформацію про структуру та функції ОД, використовуються все частіше для моделювання складних об'єктів та процесів навколишнього світу (технічні та біологічні ОД, засоби виробництва, управління та автоматизації). Незважаючи на це, нейромережеві моделі таких ОД досліджені недостатньо, а методи інтерпретації цих моделей набагато менше.

Переваги: відображення у явному вигляді залежності виходу об'єкта від його входу.

Недоліки: алгоритмічна та обчислювальна складність побудови моделей, здебільшого, з використанням процедури ідентифікації, складність забезпечення адекватності моделі, чутливість до початкових умов та діючих похибок.

Математичне забезпечення для побудови інтерпретаційних аналітичних виразів «вхід/вихід» для нейромережевих моделей розроблено недостатньо повно. Методи інтерпретації нейронних мереж, представлені в літературі, як правило, зводяться до апроксимації моделі в локальній області більш простими у обчислювальному плані сурогатними моделями, зокрема, лінійними [9, 79, 97] або поліноміальними моделями [130, 136]. При цьому, інтерпретуючі моделі, як правило, мають вигляд лінійних динамічних або нелінійних статичних залежностей та не відображають всіх властивостей ОД. Популярними аналітичними методами інтерпретації моделей машинного навчання є LIME

(Local Interpretable Model-agnostic Explanations) [65] і лінійна регресія [79], які будують сурогатні моделі, локально наближаючи вихідну модель до лінійної.

Проблема інтерпретації нейромережових моделей у вигляді аналітичних виразів може бути вирішена, якщо в якості інтерпретаційних моделей замість лінійних або поліноміальних використовувати нелінійні динамічні моделі на основі інтегро-степеневих рядів Вольтерри [58]. Основними перевагами зазначених моделей є одночасне урахування нелінійних та динамічних властивостей ОД, що забезпечує адекватність моделі та підвищення точності інтерпретації нейромережових моделей нелінійних динамічних об'єктів.

1.4. Непараметричні динамічні моделі на основі інтегро-степеневих рядів Вольтерри

Непараметричні моделі здатні описувати складні системи за відсутністю будь-якої апіорної інформації про них. Ці моделі часто використовуються для опису об'єктів з неперервними характеристиками та нелійними і динамічними характеристиками, що функціонально виражають залежність між вхідними $x(t)$ і вихідними $y(t)$ сигналами ОД [69, 150, 154, 151, 157, 162]. Функціональна залежність може бути представлена у вигляді інтегро-степеневого ряду Вольтерри [16, 18, 64, 112, 164], що використовуються для математичного опису динамічних систем з врахуванням нелінійних ефектів ОД, які не піддаються аналітичному опису параметричними моделями. Ці моделі ґрунтуються на поданні функцій у вигляді рядів, що містять у собі як степеневі, так і інтегральні члени. Такий підхід дає змогу врахувати як миттєвий вплив, так і накопичені ефекти в системі. Їх застосування охоплює різноманітні галузі досліджень і має великий потенціал для вирішення практичних завдань моделювання у різних сферах людської діяльності.

Згідно теорії рядів Вольтерри відношення «вхід-вихід» для нелінійного динамічного об'єкта типу «чорна скриня» з багатьма входами і виходами у неперервному часі на основі інтегро-степеневого ряду має наступний вигляд [18, 109]:

$$\begin{aligned}
y_j(t) = & \sum_{n=1}^{\infty} \sum_{i_1=1}^{\eta} \dots \sum_{i_n=1}^{\eta} \int_0^t \dots \int_0^t v_{i_1 i_2 \dots i_n}^j(\tau_1, \tau_2, \dots, \tau_n) \prod_{k=1}^n x_{i_k}(t - \tau_k) d\tau_k = \\
& \sum_{i_1=1}^{\eta} \int_0^t v_{i_1}^j(\tau) x_{i_1}(t - \tau) d\tau + \\
& + \sum_{i_1=1}^{\eta} \sum_{i_2=1}^{\eta} \int_0^t \int_0^t v_{i_1 i_2}^j(\tau_1, \tau_2) x_{i_1}(t - \tau_1) x_{i_2}(t - \tau_2) d\tau_1 d\tau_2 + \\
& + \sum_{i_1=1}^{\eta} \sum_{i_2=1}^{\eta} \sum_{i_3=1}^{\eta} \int_0^t \int_0^t \int_0^t v_{i_1 i_2 i_3}^j(\tau_1, \tau_2, \tau_3) x_{i_1}(t - \tau_1) x_{i_2}(t - \tau_2) x_{i_3}(t - \tau_3) d\tau_1 d\tau_2 d\tau_3 + \dots,
\end{aligned} \tag{1.6}$$

де $y_j(t)$ – сигнал на j -му виході об'єкта в момент часу t при нульових початкових умовах, $j=1, \dots, \mu$, μ – кількість виходів об'єкту; η – кількість входів об'єкту, $v_{i_1 i_2 \dots i_n}^j(\tau_1, \tau_2, \dots, \tau_n)$ – багатовимірні вагові функції (ядра Вольтерри) n -го порядку по i_1, \dots, i_n входах та j -му виходу; $x_1(t), \dots, x_{\eta}(t)$ – сигнали на вході об'єкта.

В окремому випадку, для об'єкту з одним входом $\eta=1$ і одним виходом $\mu=1$ модель набуває вигляду [109, 174]:

$$\begin{aligned}
y(t) = & \sum_{n=0}^{\infty} \int_0^t \dots \int_0^t w_n(t, \tau_1, \dots, \tau_n) \prod_{i=1}^n x(\tau_i) d\tau_i = \\
& \int_0^t w_1(\tau) x(t - \tau) d\tau + \\
& + \int_0^t \int_0^t w_2(\tau_1, \tau_2) x(t - \tau_1) x(t - \tau_2) d\tau_1 d\tau_2 + \\
& + \int_0^t \int_0^t \int_0^t w_3(\tau_1, \tau_2, \tau_3) x(t - \tau_1) x(t - \tau_2) x(t - \tau_3) d\tau_1 d\tau_2 d\tau_3 + \dots
\end{aligned} \tag{1.7}$$

де $x(t)$ та $y(t)$ – вхідний та вихідний сигнали об'єкту; $w_n(\tau_1, \dots, \tau_n)$ – багатовимірні вагові функції n -го порядку ($n=1, 2, 3, \dots$).

Дискретна форма ряду Вольтерри для об'єкта з одним входом $\eta=1$ і одним виходом $\mu=1$ набуває вигляду [109, 174]:

$$\begin{aligned}
y(n) &= \sum_{n=0}^{\infty} \sum_{k_1, \dots, k_n=0}^n v_n(k_1, \dots, k_n) \prod_{i=1}^M x(n - k_i) = \\
y(n) &= v_0 + \sum_{k=0}^M v_1(k)x(n - k) + \\
&+ \sum_{k_1=0}^M \sum_{k_2=0}^M v_2(k_1, k_2)x(n - k_1)x(n - k_2) + \\
&\sum_{k_1=0}^M \sum_{k_2=0}^M \sum_{k_3=0}^M v_3(k_1, k_2, k_3)x(n - k_1)x(n - k_2)x(n - k_3).
\end{aligned} \tag{1.8}$$

де $x(n)$ та $y(n)$ – вхідний та вихідний дискретні сигнали об'єкту; $v_n(k_1, \dots, k_n)$ – дискретні багатовимірні вагові функції n -го порядку ($n=1,2,3,\dots$), n – тривалість інтервалу спостереження; $[0, M]$ – інтервал запам'ятовування ОД.

На практиці модель у вигляді інтегро-степеневого ряду обмежують кількома першими членами.

Моделі у вигляді інтегро-степеневого ряду на основі багатовимірних вагових функцій (1.6) – (1.8) є потужним інструментом подання нелінійних та динамічних характеристик об'єкту, тому, можуть забезпечити достатню точність моделювання. Це особливо корисно у випадках, коли невідомі характеристики ОД, тобто коли об'єкт моделювання відноситься до класу «чорної скрині».

Узагальнюючи вище наведене, далі перелічені принципові переваги інтегро-степеневих моделей Вольтерри при моделюванні нелінійної динаміки:

- здатність одночасного враховувати нелінійних і динамічних характеристики системи, що дає змогу повніше й точніше описати поведінку ОД;
- функціональна залежність у явному вигляді між вхідними та вихідними сигналами, що полегшує аналіз і розуміння впливу вхідних параметрів на вихідні сигнали об'єкта, що є важливим у контексті управління та прогнозування його поведінки;
- інваріантність до виду вхідного сигналу: модель зберігає свою застосовність незалежно від характеру вхідних збурень;

– здатність описувати широкий спектр об'єктів, включно неперервні та імпульсні, із зосередженими та розподіленими параметрами, стаціонарні та нестаціонарні.

Але повсюдному практичному застосуванню інтегро-степеневих моделей Вольтерри заважають такі обмеження:

– обчислювальна складність під час побудови моделей вищих порядків, вимагає великих обчислювальних ресурсів і експертних знань.

– обмеження в точності та інтерпретованості, особливо за високих порядків моделей.

– обмеження видів нелінійності ОД: моделі Вольтерри не придатні для опису деяких видів нелінійних об'єктів зі складними нелінійними характеристиками (кусково-неперервними, з розривами, гістерезисом тощо).

– обмеження на амплітуди вхідних тестових сигналів,

– необхідність великого обсягу експериментальних даних.

– чутливість до шумів, що може призвести до неточних результатів.

Перелічені недоліки часто заважають використанню моделей Вольтерри для безпосереднього моделювання ОД, тому для первинної ідентифікації об'єктів використовуються інші методи, наприклад, нейронні мережі. Важливо зазначити, що ці недоліки не роблять моделі Вольтерри менш корисними, а лише наголошують на важливості їх застосування в областях їх ефективного використання, що визначається переліченими вище перевагами та недоліками.

Одною з областей ефективного використання моделей Вольтерри може бути інтерпретація нелінійних динамічних моделей машинного навчання, якщо в якості сурогатних моделей використовувати інтегро-степеневі ряди. Це дозволить забезпечити адекватність моделі та підвищення точності інтерпретації динамічних нейронних мереж за рахунок одночасного урахування нелінійних та динамічних властивостей ОД.

1.5. Методи оцінювання інтерпретованості моделей машинного навчання

В роботах [79, 97] розглядаються наступні підходи до оцінювання інтерпретованості моделей машинного навчання:

1. *Оцінювання орієнтоване на застосування* спрямоване на визначення впливу результатів процесу інтерпретації на експерта, кінцевого користувача інтелектуальної системи для конкретного застосування. Наприклад, оцінювання зміни помилок у прогнозах інтерпретованої моделі. За результатами оцінювання можна відповісти на питання яку інформацію і яким чином модель надає кінцевим користувачам. Часто це оцінювання проводиться за участю людини (експертів і аудиторії, що тестує модель).

2. *Оцінювання орієнтоване на людину* спрямоване на визначення впливу результатів процесу інтерпретації на будь-яку людину, користувача інтелектуальної системи для перевірки якості збудованої інтерпретації в більш загальному контексті. Наприклад, оцінювання наскільки добре схоплені інтерпретаційною моделлю загальні характеристики ОД [47, 131, 149].

3. *Оцінювання на основі функціональних залежностей* не вимагає виконання експериментів за участю людини, а використовує формальні поняття інтерпретованості для визначення якості методу інтерпретації. Для оцінки інтерпретації можна порівнювати інтерпретаційні моделі з моделями чорної скриньки, щоб визначити, яку точність забезпечують методи інтерпретації.

Останній тип оцінювання зручніше за інші для подальшого ранжування якості різних моделей інтерпретованості. Оцінювання якості на основі функціональних залежностей також зручно, коли експерименти за участю людини не можуть бути проведені, наприклад, через існуючу небезпеку, етику, фінансові витрати.

На практиці оцінювання якості інтерпретаційних моделей на основі функціональних залежностей виконується шляхом порівняння та чисельної оцінки виходу $y(t)$ ОД або його моделі $\hat{y}(t)$, $y_{ml}(t)$ (наприклад, у вигляді нейронної мережі) та інтерпретаційної моделі $y_s(t)$, отриманих при подачі однакових

сигналів $x(t)$ на їх входи.

Порівняння безперервних моделей, що використовуються у машинному навчанні, може проводитися різними методами залежно від конкретних цілей і характеристик даних. Нижче наведено кілька методів порівняння безперервних моделей, які широко використовуються в машинному навчанні [50, 146, 147].

Середньоквадратична помилка (Mean Squared Error, mse) вимірює середньоквадратичне відхилення між значеннями виходів початкової моделі $y(t)$ і сурогатної моделі $y_S(t)$:

$$\varepsilon_{mse} = \frac{1}{k} \sum_{t=1}^k (y_t - y_{St})^2 \quad (1.9)$$

де k – кількість відліків моделей y і y_S на часовому інтервалі стеження $[0, T]$.

Модель із меншим значенням mse вважається більш точною.

В окремих випадках використовується критерій нормованої середньоквадратичної помилки (Mean Squared Percentage Error, $mspe$):

$$\varepsilon_{mspe} = \sum_{t=1}^k (y_t - y_{St})^2 / \sum_{t=1}^k y_t^2 \quad (1.10)$$

Цей критерій можна розглядати як зважену версію mse , де вага обернено пропорційна квадрату спостережуваного значення. Таким чином, при зростанні спостережуваних значень помилка зменшується.

Середнє абсолютне відхилення (Mean Absolute Error, mae) вимірює середнє абсолютне відхилення між значеннями початкової моделі $y(t)$ і сурогатної моделі $y_S(t)$:

$$\varepsilon_{mae} = \frac{1}{k} \sum_{t=1}^k |y_t - y_{St}| \quad (1.11)$$

Метод є менш чутливим до викидів, ніж mse .

Аналогічно критерію $mspe$ можна ввести критерій нормованої абсолютної помилки (Mean Absolute Percentage Error, $mape$):

$$\varepsilon_{mape} = \sum_{t=1}^k |y_t - y_{St}| / \sum_{t=1}^k |y_t| \quad (1.12)$$

Коефіцієнт детермінації R^2 вказує, наскільки добре модель відповідає даним.

$$R^2 = 1 - \frac{\sum_{t=1}^k (y_t - y_{st})^2}{\sum_{t=1}^k (\bar{y}_t - y_{st})^2} \quad (1.13)$$

Значення R^2 близьке до 1 означає хорошу відповідність, а близьке до 0 - погану.

Перелічені вище критерії якості моделей важко інтерпретувати та за їх значенням визначити, що потрібно зробити, щоб поліпшити модель. У зв'язку з цим для оцінювання якості моделей представляє інтерес також використання показників, що визначають, наскільки дана модель працює краще, ніж певна модель у вигляді константи (коефіцієнти при вхідних змінних дорівнюють нулю). Такі критерії можуть застосовуватися з метою порівняння моделей та не мають смислового змісту самих критеріїв, чим вони і відрізняються від метрик, заснованих на помилках, розглянутих вище. Нижче перелічені основні критерії порівняння моделей [1].

Інформаційний критерій Акаїке (Akaike Information Criterion, *aic*) враховує баланс між якістю моделі та її складністю. Модель із меншим значенням *aic* вважається кращою.

Байєсівський інформаційний критерій (Bayesian Information Criterion, *bic*) як і *aic* враховує якість моделі та її складність. Він штрафує моделі з великим числом параметрів, тому простіші моделі можуть мати перевагу.

Порівняння кривих навчання для різних моделей дає можливість оцінити їхню продуктивність на різних обсягах даних. Це особливо корисно за наявності обмеженої кількості даних.

Тести гіпотез використовуються для порівняння параметрів моделей, наприклад, порівняння коефіцієнтів регресії або порівняння середніх значень.

Візуалізація даних і результатів моделей також може допомогти в наочному порівнянні різних моделей.

Вибір конкретного методу порівняння залежить від цілей дослідження. Крім того, часто використовується комбінація декількох методів для повнішого

розуміння продуктивності моделей. В даній роботі для оцінювання інтерпретованості моделей машинного навчання використовуються методи *mse* для встановлення точності моделі і *cross-validation* для оцінки її перенавчання. Для порівняння моделей між собою при автоматичному виборі найкращого рішення в Розділі 4 використовуються *aic*.

1.6. Постановка задач дослідження

Формальна постановка задачі інтерпретації моделей машинного навчання нелінійних динамічних об'єктів може бути сформульована наступним чином. Припустимо F – нелінійна динамічна модель машинного навчання, яка з заданою точністю відображає множину вхідних даних $\mathbf{x}(t) \in \mathbf{X}$, де $\mathbf{x}(t) = [x(t_n), x(t_{n-1}), \dots]$, $t_n = n\Delta t$, $n = 1, 2, \dots$) у множину вихідних даних $y(t) \in \mathbf{Y}$: $F: \mathbf{X} \rightarrow \mathbf{Y}$. При цьому модель F не відображає в явному вигляді структуру та закони функціонування ОД.

Завдання інтерпретації моделі машинного навчання F полягає в побудові сурогатної моделі $F' = f_s(F, \mathbf{X}, t)$ яка відображає множину вхідних даних у множину вихідних даних $\mathbf{X} \rightarrow \mathbf{Y}$, з заданою точністю E . До моделі F' пред'являється вимога відображення у явному вигляді (в аналітичній формі) внутрішніх механізмів і процесів, що відбуваються всередині ОД шляхом встановлення інформаційного зв'язку між вхідними $\mathbf{x}(t) \in \mathbf{X}$ та вихідними сигналами $y(t) \in \mathbf{Y}$ з метою визначення конкретних аспектів або характеристик моделі F , які потребують пояснення для людини.

Задача дослідження полягає у побудові сурогатної моделі F' в аналітичній формі у вигляді непараметричних динамічних моделей на основі інтегро-степеневих рядів Вольтерри, що адекватно відображають нелінійні та динамічні характеристики ОД.

1.7. Висновки за розділом

В даному розділі висвітлено практичні аспекти застосування математичних моделей інтерпретації в інтелектуальних системах. На основі аналізу предметної області розглянуто причини появи протиріччя між складністю ОД та інтерпретованістю його моделі; здійснено огляд та пошук ефективних напрямів подолання означеної проблеми.

Аналіз розглянутих джерел дозволяє наголосити, що на сьогоднішній день задача інтерпретації моделей машинного навчання нелінійних динамічних об'єктів з неперервними характеристиками ще не набула розв'язку. Також, існує брак алгоритмічних та програмних засобів інтерпретації нейронних мереж у вигляді аналітичних виразів. Залишаються певні перспективи підвищення точності сурогатних моделей нелінійних динамічних об'єктів з неперервними характеристиками.

За результатами аналітичного огляду літературних джерел зроблено наступні висновки.

1. Проведено систематизацію поширених моделей машинного навчання з позначенням області їх застосування, основних характеристик та здатності до інтерпретації результатів навчання. Окремо проведено класифікацію моделей нелінійних динамічних об'єктів, як таких, що викликають найбільший інтерес з боку практики. В результаті аналітичного огляду поточного стану проблеми інтерпретації нелінійних динамічних моделей машинного навчання виокремлено актуальний напрям створення прозорих моделей об'єктів типу «чорна скриня» у вигляді аналітичних виразів.

2. Встановлено поширені методи побудови нейромережових моделей нелінійних динамічних об'єктів, розглянуто їх переваги та недоліки та окреслено особливості їх використання. На основі порівняльного аналізу обґрунтовано подальше використання нейронних мереж з часовими затримками для моделювання нелінійних динамічних об'єктів типу «чорна скриня». Показано, що

застосування нейронних мереж у критично важливих сферах діяльності людини обмежено дією ряду ускладнень, пов'язаних з проблемами їх інтерпретації.

3. Розглянуто поширені підходи до інтерпретації нейронних мереж. В результаті аналітичного огляду розглянутих підходів визначено сфери ефективного застосування аналітичних сурогатних моделей для інтерпретації складних об'єктів типу «чорна скриня». Застосування таких моделей забезпечує підвищенні вимоги до безпеки результатів моделювання: переконаності у відсутності прихованої поведінки чи логіки, які впливають на властивості моделі. Визначено напрям дослідження як побудова глобальних агностичних моделей інтерпретації за даними, представленими у табличному вигляді.

На підставі наведеного встановлено, що вирішення протиріччя, наведеного у вступі, може бути досягнуто при використанні сурогатних моделей – апроксимації точних але складних і повільних нейромережових моделей за допомогою значно простіших адекватних моделей, що можуть бути інтерпретовані людиною. Запропоновано та обґрунтовано підхід до інтерпретації нейромережових моделей складних об'єктів з нелінійними і динамічними характеристиками типу «чорна скриня» шляхом застосування в якості аналітичних виразів залежності «вхід/вихід» сурогатних моделей у вигляді інтегро-ступеневих рядів Вольтерри. Це дозволить забезпечити підвищення точності інтерпретації складних ОД типу «чорна скриня» у порівнянні з нелінійними статичними або лінійними динамічними сурогатними моделями.

Наведені загальні проблеми інтерпретації нейромережових моделей нелінійних динамічних об'єктів в інтелектуальних системах дозволяють сформулювати постановку задачі дослідження, як розвиток підходу до застосування сурогатних моделей у вигляді інтегральних непараметричних динамічних моделей Вольтерри для апроксимації нейронних мереж з часовими затримками, що забезпечує підвищення точності інтерпретації моделей при створенні програмного забезпечення спеціалізованих інтелектуальних систем.

Вирішення встановленої задачі інтерпретації нейромережових моделей на основі апарата інтегро-ступеневих рядів Вольтерри ускладнюється наступними

проблемами.

1. Відсутність розвинутого математичного апарату побудови сурогатних моделей інтерпретації динамічних нейронних мереж на основі рядів Вольтерри. Існуючі спроби побудови інформаційного зв'язку між нейронними мережами та рядами Вольтерри обмежуються окремими випадками використання перцептронів з активаційними функціями у вигляді гіперболічного тангенсу або поліномів низьких порядків.

2. Брак ефективних методів побудови та оцінки точності сурогатних моделей на основі рядів Вольтерри.

3. Відсутність засобів комп'ютерного моделювання для побудови сурогатних моделей на основі рядів Вольтерри та інтерпретації моделей машинного навчання нелінійних динамічних об'єктів.

Вирішення проблеми, наведеної у п.1, полягає у розробці математичних моделей інтерпретації динамічних нейронних мереж на основі ряду Вольтерри, отриманих за даними експерименту «вхід–вихід».

Вирішення проблеми, наведеної у п.2, полягає у розробці методу та інформаційної технології інтерпретації моделей машинного навчання, що забезпечує належну точність сурогатної моделі на основі експериментальних даних.

Вирішення проблеми, наведеної у п.3, полягає у розробці засобів комп'ютерного моделювання та дослідження області їх ефективного застосування при розв'язуванні задач інтерпретації моделей машинного навчання нелінійних динамічних об'єктів.

Вирішенню наведених проблем та створенню ефективного методу та програмних засобів інтерпретації моделей машинного навчання нелінійних динамічних об'єктів присвячено наступні розділи даного дослідження.

РОЗДІЛ 2

МАТЕМАТИЧНІ МОДЕЛІ ІНТЕРПРЕТАЦІЇ ДИНАМІЧНИХ НЕЙРОННИХ
МЕРЕЖ НА ОСНОВІ РЯДУ ВОЛЬТЕРРИ

На основі аналізу літературних джерел та предметної області, виконаного в Розділі 1, встановлені можливості підвищення точності сурогатних моделей нелінійних динамічних об'єктів з неперервними характеристиками. Запропоновано та обґрунтовано підхід до інтерпретації нейромережевих моделей об'єктів з нелійними і динамічними характеристиками типу «чорна скриня» на основі використання в якості сурогатних моделей аналітичних виразів у вигляді інтегро-степеневих рядів Вольтерри.

Інтегральні непараметричні динамічні моделі Вольтерри мають низку переваг, в тому числі, забезпечують високу точність ідентифікації нелінійних динамічних моделей та добре інтерпретуються людиною завдяки здатності представляти залежність між вхідними $x(t)$ та вихідними сигналами $y(t)$ у явній формі. Однак, їх застосування стримується наявністю низки обмежень. Так, моделі Вольтерри більшою мірою підходять для моделювання об'єктів зі слабкою гладкою однозначною нелінійністю. Наявність неоднозначних або кусково-неперервних нелінійних характеристик робить застосування інтегро-степеневих рядів Вольтерри мало ефективним. Крім того, наявні обмеження на амплітуди вхідних тестових сигналів також перешкоджають використанню таких моделей під час ідентифікації ОД.

Тому, як самостійний апарат, моделі Вольтера мають суттєві обмеження при використанні в практичних застосуваннях. Але вони можуть досить ефективно використовуватись як сурогатні моделі для нейронних мереж, до яких не ставляться жорсткі обмеження. З іншого боку, брак методів побудови сурогатних моделей машинного навчання нелінійних динамічних об'єктів обумовлює необхідність розробки нових моделей інтерпретації нейронних мереж. Тому, вбачається перспективним напрямом розробка та дослідження сурогатних

моделей у вигляді інтегро-степеневих рядів Вольтерри для вирішення проблеми інтерпретації моделей машинного навчання нелінійних динамічних об'єктів.

В даному розділі поставлено задачу розробки сурогатних моделей у вигляді інтегро-степеневих рядів Вольтерри на основі моделей машинного навчання у вигляді нейронних мереж з часовими затримками. Для виконання цієї задачі попутно розглядаються питання вдосконалення алгоритмів детермінованої ідентифікації за даними експерименту «вхід-вихід» нелінійних динамічних об'єктів на основі моделей у вигляді інтегро-степеневих рядів Вольтерри, а також, дослідження області ефективного застосування запропонованих сурогатних моделей. Для дослідження точності запропонованих сурогатних моделей, та моделей збудованих за допомогою алгоритмів детермінованої ідентифікації організовано та проведено низку експериментів з тестовими імітаційними моделями нелінійних динамічних об'єктів.

2.1. Ідентифікація моделей нелінійних динамічних об'єктів у вигляді інтегро-степеневих поліномів Вольтерри в часовій області

У ряді випадків, в умовах дотримання обмежень застосування інтегро-степеневих рядів і за допустимості використання спеціальних тестових сигналів на вході ОД, модель Вольтерри, що може бути інтерпретованою людиною, може бути побудована безпосередньо за результатами експерименту «вхід/вихід» без побудови нейронної мережі. Використання апарату інтегро-степеневих рядів Вольтери потребує багато експериментального матеріалу та значних обчислювальних потужностей [170, 172]. Це обумовлює пошук нових методів ідентифікації нелінійних динамічних об'єктів, які не мають означених незручностей при їх використанні. Тому, актуальними залишаються задачі побудови моделей нелінійних динамічних об'єктів з застосуванням апарату рядів Вольтерри у спрощеному вигляді.

В літературі існує багато підходів до визначення ядер Вольтерри [21, 30, 38]. Так, в роботах [110, 112, 164, 165] наведено спрощені вирази для визначення ядра

Вольтерри першого порядку та головних перетинів ядер Вольтерри вищих порядків з використанням тестових ступінчастих вхідних сигналів у часовій області та описано відповідний експеримент. Перевагами цього підходу є зменшення обчислювальної складності експерименту «вхід-вихід» та простота створення тестових сигналів $x(t)$. Однак похибка оцінки моделі від некомпенсованих членів ряду Вольтерри обмежує застосування цього метода в умовах реального експерименту.

Для подолання вказаного обмеження здійснюється спроба подальшого розвитку цього підходу з використанням поліномів Вольтерри у часовій області, що дозволяє зменшити похибку оцінки моделі. Відношення «вхід-вихід» для нелінійного динамічного об'єкта типу «чорна скриня» з багатьма входами ν і багатьма виходами μ у неперервному часі на основі інтегро-степеневого поліному Вольтери має наступний вигляд [18, 30]:

$$y_j(t) = \sum_{n=1}^N \sum_{i_1=1}^{\eta} \dots \sum_{i_n=1}^{\eta} \int_0^t \dots \int_0^t v_{i_1 i_2 \dots i_n}^j(\tau_1, \tau_2, \dots, \tau_n) \prod_{k=1}^n x_{i_k}(t - \tau_k) d\tau_k, \quad (2.1)$$

де N – порядок поліному Вольтери.

В окремому випадку для об'єкта з одним входом $\eta=1$ і одним виходом $\mu=1$ модель набуває вигляду:

$$y(t) = \sum_{n=0}^N \int_0^t \dots \int_0^t v_n(t, \tau_1, \dots, \tau_n) \prod_{i=1}^n x(\tau_i) d\tau_i. \quad (2.2)$$

Згідно [110, 112, 161], ідентифікація моделі нелінійного динамічного об'єкта у вигляді поліномів Вольтерри в часовій області полягає у використанні спеціальних вхідних сигналів $x(t)$ і вимірюванні вихідних сигналів $y(t)$. За даними експерименту «вхід/вихід» необхідно розробити алгоритм, за яким можуть бути оцінені ядра Вольтерри $v_n(\tau_1, \dots, \tau_n)$.

Ідентифікація моделі нелінійного динамічного ОД з одним входом і одним виходом у вигляді поліному Вольтерри 2-го порядку виконується наступним чином. Якщо на вхід ОД по черзі подаються сигнали $x_1(t)$ та $x_2(t)$, то згідно (2.2) можна записати систему рівнянь:

$$\begin{cases} y_1(t) = \int_0^t v_1(\tau)x_1(t-\tau)d\tau + \int_0^t \int_0^t v_2(\tau,\tau)x_1(t-\tau)x_1(t-\tau)d\tau d\tau \\ y_2(t) = \int_0^t v_1(\tau)x_2(t-\tau)d\tau + \int_0^t \int_0^t v_2(\tau,\tau)x_2(t-\tau)x_2(t-\tau)d\tau d\tau \end{cases} . \quad (2.3)$$

Розглянемо випадок, коли вхідний тестовий сигнал $x(t)$ являє собою δ -функцію (функція Дірака) $\delta(t)$. Якщо в якості вхідних сигналів використовувати пропорційні δ -функцію $x_1(t)=a\delta(t)$ та $x_2(t)=2a\delta(t)$, де a – масштабуючий коефіцієнт, то систему рівнянь (2.3) можна записати у вигляді:

$$\begin{cases} y_1(t) = \int_0^t v_1(\tau)a\delta(t-\tau)d\tau + \int_0^t \int_0^t v_2(\tau,\tau)a\delta(t-\tau)a\delta(t-\tau)d\tau d\tau \\ y_2(t) = \int_0^t v_1(\tau)2a\delta(t-\tau)d\tau + \int_0^t \int_0^t v_2(\tau,\tau)2a\delta(t-\tau)2a\delta(t-\tau)d\tau d\tau \end{cases} . \quad (2.4)$$

Зважаючи на те, що

$$\int_0^t \delta(t-\tau)d\tau = 1, \quad (2.5)$$

отримуємо систему лінійних алгебраїчних рівнянь:

$$\begin{cases} y_1(t) = av_1(t) + a^2v_2(t,t) \\ y_2(t) = 2av_1(t) + 4a^2v_2(t,t) \end{cases} . \quad (2.6)$$

Рішенням цієї системи рівнянь є вирази для ядра Вольтерри першого порядку $v_1(t)$ та діагонального перетину ядра Вольтерри другого $v_2(t,t)$ порядку:

$$\begin{cases} v_1(t) = (4y_1(t) - y_2(t))/2a \\ v_2(t,t) = (y_2(t) - 2y_1(t))/2a^2 \end{cases} . \quad (2.7)$$

Модель нелінійного динамічного ОД з одним входом і одним виходом у вигляді поліному Вольтерри 2-го порядку має вигляд:

$$y(t) = (4y_1(t) - y_2(t))/2a x(t) + (y_2(t) - 2y_1(t))/2a^2 x(t) \quad (2.8)$$

Ідентифікація моделі нелінійного динамічного ОД з одним входом і одним виходом у вигляді поліному Вольтерри 3-го порядку виконується наступним чином. Якщо на вхід ОД по черзі подаються сигнали $x_1(t)$, $x_2(t)$ та $x_3(t)$, то згідно (2.2) можна записати систему рівнянь:

$$\left\{ \begin{array}{l} y_1(t) = \int_0^t v_1(\tau)x_1(t-\tau)d\tau + \int_0^t \int_0^t v_2(\tau,\tau)x_1(t-\tau)x_1(t-\tau)d\tau d\tau + \\ + \int_0^t \int_0^t \int_0^t v_3(\tau,\tau,\tau)x_1(t-\tau)x_1(t-\tau)x_1(t-\tau)d\tau d\tau d\tau \\ y_2(t) = \int_0^t v_1(\tau)x_2(t-\tau)d\tau + \int_0^t \int_0^t v_2(\tau,\tau)x_2(t-\tau)x_2(t-\tau)d\tau d\tau + \\ + \int_0^t \int_0^t \int_0^t v_3(\tau,\tau,\tau)x_2(t-\tau)x_2(t-\tau)x_2(t-\tau)d\tau d\tau d\tau \\ y_3(t) = \int_0^t v_1(\tau)x_3(t-\tau)d\tau + \int_0^t \int_0^t v_2(\tau,\tau)x_3(t-\tau)x_3(t-\tau)d\tau d\tau + \\ + \int_0^t \int_0^t \int_0^t v_3(\tau,\tau,\tau)x_3(t-\tau)x_3(t-\tau)x_3(t-\tau)d\tau d\tau d\tau \end{array} \right. \quad (2.9)$$

Подаючи на вхід ОД почергово тестові сигнали $x_1(t)=a\delta(t)$, $x_2(t)=2a\delta(t)$ та $x_3(t)=3a\delta(t)$, отримуємо систему рівнянь:

$$\left\{ \begin{array}{l} y_1(t) = \int_0^t v_1(\tau)a\delta(t-\tau)d\tau + \int_0^t \int_0^t v_2(\tau,\tau)a\delta(t-\tau)a\delta(t-\tau)d\tau d\tau + \\ \int_0^t \int_0^t \int_0^t v_3(\tau,\tau,\tau)a\delta(t-\tau)a\delta(t-\tau)a\delta(t-\tau)d\tau d\tau d\tau \\ y_2(t) = \int_0^t v_1(\tau)2a\delta(t-\tau)d\tau + \int_0^t \int_0^t v_2(\tau,\tau)2a\delta(t-\tau)2a\delta(t-\tau)d\tau d\tau + \\ \int_0^t \int_0^t \int_0^t v_3(\tau,\tau)2a\delta(t-\tau)2a\delta(t-\tau)2a\delta(t-\tau)d\tau d\tau d\tau \\ y_3(t) = \int_0^t v_1(\tau)3a\delta(t-\tau)d\tau + \int_0^t \int_0^t v_2(\tau,\tau)3a\delta(t-\tau)3a\delta(t-\tau)d\tau d\tau + \\ \int_0^t \int_0^t \int_0^t v_3(\tau,\tau)3a\delta(t-\tau)3a\delta(t-\tau)3a\delta(t-\tau)d\tau d\tau d\tau \end{array} \right. \quad (2.10)$$

Виконуючи спрощення, аналогічні наведеним вище, отримуємо систему лінійних алгебраїчних рівнянь:

$$\left\{ \begin{array}{l} y_1(t) = av_1(t) + a^2v_2(t,t) + a^3v_3(t,t,t) \\ y_2(t) = 2av_1(t) + 4a^2v_2(t,t) + 8a^3v_3(t,t,t) \\ y_3(t) = 3av_1(t) + 9a^2v_2(t,t) + 27a^3v_3(t,t,t) \end{array} \right. \quad (2.11)$$

Рішенням системи рівнянь (2.11) є вирази для ядра Вольтерри першого порядку w_1 та діагональних перетинів ядер Вольтерри другого w_2 та третього w_3 порядків:

$$\begin{cases} v_1(t) = (3y_1(t) - 3/2 y_2(t) + 1/3 y_3(t))/a \\ v_2(t,t) = (-5y_1(t) + 4y_2(t) + y_3(t))/2a^2 \\ v_3(t,t,t) = (3y_1(t) - 3y_2(t) + y_3(t))/6a^3 \end{cases} \quad (2.12)$$

Модель нелінійного динамічного ОД з одним входом і одним виходом у вигляді поліному Вольтерри 3-го порядку має вигляд:

$$y(t) = (3y_1(t) - 3/2 y_2(t) + 1/3 y_3(t))/a x(t) + (-5y_1(t) + 4y_2(t) + y_3(t))/2a^2 x(t) + (3y_1(t) - 3y_2(t) + y_3(t))/6a^3 x(t) \quad (2.13)$$

Для зменшення кількості експериментів з отримання даних про вхідні та вихідні сигнали та значного спрощення алгоритму обробки експериментальних даних опис ОД буде здійснено у вигляді перетинів багатовимірних моделей. Вираз для піддіагональних перетинів ядер Вольтерри n -го порядку $\hat{h}_{v_1 \dots v_n}^\mu(t - \tau_1, \dots, t - \tau_n)$ з багатьма входами v і багатьма виходами μ у неперервному часі на основі інтегро-степеневого поліному Вольтери має наступний вигляд [174]:

$$h_{v_1 \dots v_n}^\mu(t - \tau_1, \dots, t - \tau_n) = \int_0^\infty \dots \int_0^\infty v_{v_1 \dots v_n}^\mu(t - \tau_1 - \lambda_1, \dots, t - \tau_n - \lambda_n) d\lambda_1 \dots, d\lambda_n. \quad (2.14)$$

В окремому випадку для об'єкту з одним входом $v=1$ і одним виходом $\mu=1$ модель набуває вигляду:

$$h_n(t - \tau_1, \dots, t - \tau_n) = \int_0^\infty \dots \int_0^\infty v_n(t - \tau_1 - \lambda_1, \dots, t - \tau_n - \lambda_n) d\lambda_1 \dots, d\lambda_n. \quad (2.15)$$

При фіксованих значеннях τ_1 і τ_2 перехідна функція $h_2(t - \tau_1, t - \tau_2)$ являє собою вираз від змінної t та визначає перетин моделі $h_2(t_1, t_2)$ площиною, яка проходить під кутом в 45° до осей t_1 і t_2 та зсунутої по осі t_1 на дистанцію $\tau_0 = \tau_2 - \tau_1$. По різним перетинам $h_2(\tau_1, \tau_2)$ можна визначити повну модель $h_2(\tau_1, \tau_2)$, якщо змінювати зсув τ_0 у діапазоні $[0, \infty)$.

Обчислювальний алгоритм методу експериментального визначення моделей нелінійних динамічних об'єктів з одним входом $v=1$ і одним виходом $\mu=1$ у вигляді поліномів Вольтерри в часовій області набуває наступного вигляду.

Крок 1. На вхід об'єкта по чергово подається тестові імпульсні сигнали $x(t)$ амплітуд $a, 2a, 3a, \dots, Na$, де N – порядок моделі.

Крок 2. На виході реєструються відповідні відгуки об'єкту $y_i(t)$, $i=1, 2, \dots, N$.

Крок 3. Обчислюють ядра Вольтерри порядків $1, 2, 3, \dots, N$.

Крок 4. Будується модель ОД N -го порядку у вигляді поліномів Вольтерри.

Наведений алгоритм може бути компактно записаний за допомогою псевдокоду:

Algorithm: *surrogate_nn_model*

1: **Input:** $M, K, \varepsilon, E, N, \mathbf{D}$

2: **Output:** \mathbf{V}_n

3: *normalize* $\leftarrow \mathbf{D}$

4: $\mathbf{D}_{\text{train}}, \mathbf{D}_{\text{test}} \leftarrow \mathbf{D}$

5: *random* $\leftarrow \mathbf{B}[K], \mathbf{W}[M, K]$

6: *epoch* $\leftarrow 0$

7: **while** *epoch* $< E$ or $e_s(t) > e$ **do**

8: *epoch* \leftarrow *epoch* + 1

9: **for** $i = 1, \dots, N$ **do**

10: *evaluations* \leftarrow *training procedure*($\mathbf{B}, \mathbf{W}, \mathbf{D}_{\text{train}}, e$)

11: **end for**

12: *loss* \leftarrow *mse*($\mathbf{B}, \mathbf{W}, \mathbf{D}_{\text{test}}$)

13: **end while**

14: $\mathbf{V} \leftarrow$ *bwf*(\mathbf{B}, \mathbf{W})

Таким чином, в роботі використовується модифікований алгоритм методу непараметричної ідентифікації нелінійних динамічних об'єктів на основі аналізу відгуків об'єкта на тестові входні впливи у вигляді імпульсних або східчастих сигналів з використанням поліномів Вольтерри в якості моделі ОД [150, 161, 164]. Перевагою алгоритму методу у порівнянні з [161, 164] є спрощенні вирази для обчислення ядер Вольтерри та підвищення точності моделі на певному інтервалі амплітуд тестових сигналів.

2.2. Моделювання нелінійних динамічних об'єктів за допомогою нейронних мереж з часовими затримками

Використання TDNN в інформаційних системах забезпечує потужні можливості ідентифікації динамічних об'єктів з нелінійними характеристиками [6, 134, 139, 130, 136], в тому числі з нелінійностями високих порядків. Ці властивості виявляються вкрай важливими для побудови імітаційних моделей, які здатні повторювати поведінку складних об'єктів при дії різних вхідних сигналів. Крім того, означені нейронні мережі здатні забезпечити унікальні можливості ідентифікації моделей в умовах, коли використання інших методів є досить важким, наприклад, в умовах наявності складних нелінійностей у вигляді розривів, зламів, невизначеності.

Для ефективного вирішення таких задач необхідні спеціальним образом організовані вихідні дані, архітектурне рішення нейронної мережі та алгоритми навчання мережі.

Вихідні дані. Для відображення динамічних характеристик ОД вихідний сигнал мережі $y(t_n)$ в момент часу t_n має залежати не тільки від вхідного сигналу $x(t_n)$ в даний момент часу, але також від вхідних сигналів, що діяли в попередні моменти часу $t_{n-1}, t_{n-2}, \dots, t_0, t_n = n\Delta t, n=1, 2, \dots, M$, де M – довжина часової затримки (пам'яті системи).

В цьому випадку нейронна мережа з часовими затримками повинна отримати спеціальним чином підготовані вхідні дані:

$$\mathbf{x}(t_n) = [x(t_n), x(t_{n-1}), \dots, x(t_{n-M+1})], \quad (2.16)$$

або у матричній формі:

$$\begin{array}{c} x_n, x_{n-1}, \dots, x_{n-M+1} \\ x_{n-1}, x_{n-2}, \dots, x_{n-M} \\ \dots \\ x_M, x_{M-1}, \dots, x_1 \end{array} \quad (2.17)$$

Архітектура нейронної мережі. Незважаючи на певну множину архітектурних рішень TDNN, на практиці здебільшого застосовується топологія у

вигляді тришарового персептрона з прямим розповсюдженням сигналу [130, 136, 164]. Тришарова НМ – це добре відома структура TDNN, яка може бути навченою динамічній поведінці з урахуванням нелінійних характеристик на даних експерименту «вхід-вихід». Шари, що присутні у мережі: вхідний (приймає вхідні дані), вихідний (формує остаточний результат) і прихований (обробляє дані від вхідного шару).

Вхідний шар в означеній структурі TDNN включає M нейронів, де M – довжина пам'яті моделі об'єкта. Кількість нейронів M обирається таким чином, щоб найкраще відображати динамічні властивості об'єкта. Питання вибору значення M детально розглядається в Розділі 3.

Зазвичай, у вхідному шарі функції активації не використовуються, оскільки їх завдання – передати вхідні дані в наступний шар без спотворення.

Кількість нейронів у вихідному шарі мережі в задачі моделювання дорівнює кількості виходів ОД. Для випадку ОД з одним виходом вихідний шар містить один нейрон. Вихідні дані в цьому випадку мають вигляд:

$$y(t_n)=[y_n, y_{n-1}, \dots, y_{n-M+1}], t_n=n\Delta t, n=1, 2, \dots, M+1 \quad (2.18)$$

Вибір активаційної функції для вихідного шару залежить від типу завдання, що вирішується. В якості активаційної функції вихідного нейрону в задачі регресії, яка вирішується при моделюванні, доцільно використовувати лінійну функцію, що дозволяє генерувати вихідний сигнал у вигляді лінійної комбінації його входів. В цьому випадку, вихідний сигнал тришарової мережі має вигляд (1.18).

Прихований шар включає K нейронів з нелінійною функцією активації. Кількість нейронів K обирається таким чином, щоб найкраще відображати нелінійні властивості об'єкта. Питання вибору значення K детально розглядається в Розділі 3.

Кожен нейрон прихованого шару здійснює нелінійне перетворення даних, що подаються на його входи, за допомогою обраної активаційної функції. Для прихованих шарів поширеним рішенням є такі активаційні функції:

– сигмоїдна функція перетворює вхідний сигнал згідно з виразом $S_i = \max(0, x)$, що робить її корисною для моделей, де ми повинні передбачити ймовірність.

– гіперболічний тангенс (\tanh) перетворює вхідний сигнал згідно з виразом $S_i = \max(0, x)$.

Рідше можуть застосовуватися такі активаційні функції як поліноми, гаусівські, синусоїдальні, тощо. Застосування таких функцій пояснюється кількома перевагами: нелінійні функції досить ефективно наближають інші нелінійні функції; вони неперервні та мають гладкі неперервні похідні.

Але у розглянутих функцій є істотні недоліки [20, 114, 117]:

– при збільшенні значення зваженої суми штучного нейрона $S_i(x)$ значення активаційної функції $f(S)$ змінюються повільніше і градієнти активаційної функції зменшуються до дуже малих значень; це призводить до істотного уповільнення процесу навчання або, навіть, до його зупинки;

– активаційна функція $f(S)$ визначена на всьому діапазоні значень зваженої суми штучного нейрона $S_i(x)$; це призводить до активації всіх нейронів для визначення виходу мережі, незалежно від рівня та знаку входів і, як наслідок, до збільшення обчислювального навантаження.

Для подолання означених недоліків та, як наслідок, прискорення процесу навчання НМ часто на практиці застосовується функція активації у вигляді випрямляючої лінійної ланки (Rectified Linear Unit) – ReLU. Найбільший ефект від її використання припадає на згорткові і глибокі НМ з великою кількістю нейронів. Функція ReLU описується виразом [117]:

$$S_i(x) = \max(0, x). \quad (2.19)$$

Популярність використання цієї функції в якості активаційної пояснюється її властивостями [114, 117]:

– ненасиченість градієнта: на відміну від сигмовидних функцій, де градієнти можуть "зникнути" за дуже великих або дуже малих значень, градієнт ReLU залишається постійним (дорівнює 1) для всіх позитивних входів; це прискорює збіжність стохастичного градієнтного спуску;

– простота обчислення: ReLU можна реалізувати за допомогою простих

арифметичних операцій; це робить її ефективнішою з погляду обчислень порівняно з іншими функціями активації, які потребують складніших операцій, таких як експоненціальні;

– сприяє розрідженості активацій: ReLU обнуляє всі негативні входи, що призводить до викреслення частини нейронів з обчислювального процесу – розрідженості активацій, тобто, в будь-який момент часу, тільки невелика кількість нейронів активна; це призводить до істотного зменшення обчислювального навантаження;

– моделювання нелінійності: ReLU допомагає моделювати нелінійності в даних, що є важливим аспектом у більшості реальних завдань.

Важкою проблемою при використанні ReLU є труднощі в процесі навчання, пов'язані з вимогою гладкості та диференційованості активаційної функції. Це обмежує використання ReLU в алгоритмах навчання НМ, побудованих на принципі зворотного розповсюдження помилки. Знаною практикою вирішення цієї проблеми є заміна функції ReLU інтерполяційним поліномом [117], що максимально наближений до активації на певному інтервалі $[-q, q]$:

$$S_i = \sum_{p=0}^N a_p x^p, \quad (2.20)$$

де p – порядок полінома, $p=0,1,2,\dots$

Для зменшення обчислювальної складності та підвищення швидкості навчання TDNN в якості активаційної функції прихованого шару запропоновано використовувати апроксимаційний поліном випрямляючої лінійної ланки ReLU.

Точність наближення поліноміальної функції (2.20) до активаційної функції значно впливає на процес навчання та якість роботи НМ. Тому, в роботі проведено допоміжне дослідження для побудови поліноміальної апроксимаційної функції (2.19) на множині $\{x_i, \max(0, x_i)\}$, де x_i – випадкові значення з нормальним розподілом на певному інтервалі $[-q, q]$. Для пошуку коефіцієнтів апроксимаційного поліному використовується метод регресії на основі найменших квадратів [147, 152].

Для обчислення поліноміальної апроксимації ReLU на стандартному нормальному розподілі, використовується функція поліноміальної апроксимації *Polynomial.fit* з пакету *Python numpy*. Використання даної функції для вхідних даних $\{x_i, \max(0, x_i)\}$ та різних порядків p функції апроксимації дає набори коефіцієнтів $a_i, i=1, \dots, p$. Результати апроксимації функції ReLU за допомогою поліномів різних порядків з стандартним нормальним розподілом значень 99,73% на інтервалі $[-3, 3]$ наведені в табл. 2.1.

Таблиця 2.1 – Апроксимація випрямляючої лінійної ланки за допомогою поліномів різних порядків

Ступінь поліному	Поліном S_i	Точність апроксимації
1	$0.85714286 + 1.5 x$	1.857
2	$0,1992 + 0,5002 x + 0,1997 x^2$	0,279
3	$0,1995 + 0,5002 x + 0,1994 x^2 - 0,0164 x^3$	0,161
4	$0.1298 + 1.500 \cdot x + 2.5909 \cdot x^2 - 0.0001 x^3 - 1.2272 \cdot x^4$	0.038

Алгоритми навчання мережі. Навчання TDNN з активаційною функцією ReLU у прихованому шарі для моделювання нелінійної динаміки ОД зазвичай здійснюється за допомогою принципу зворотнього поширення помилки. Цей принцип включає в себе два етапи: пряме поширення помилки, під час якого робиться передбачення відповіді моделі та зворотне поширення помилки, коли помилка між фактичною відповіддю і передбаченою мінімізується.

В роботі TDNN модель нелінійного динамічного ОД будується на основі алгоритму Левенберга-Марквардта [46, 100, 126, 129]. Цей метод є ефективним алгоритмом навчання нейронних мереж, який може розглядатися як комбінація методу Ньютона і методу градієнтного спуску. Він забезпечує високу точність навчання нейронної мережі, хоча вимагає складних обчислень, які можуть займати багато часу. У деяких дослідженнях пропонуються підходи до його оптимізації та паралелізації для збільшення продуктивності [100, 124].

В результаті навчання за алгоритмом Левенберга-Марквардта можна збудувати неперервну TDNN модель з відтворенням нелінійних та динамічних характеристик. Але модель у вигляді НМ залишається не зручною для вивчення та інтерпретації її поведінки. Тому, далі здійснюється пошук способу побудови інтерпретованих моделей нелінійних динамічних ОД, що поєднують переваги обох моделей TDNN та поліномів Вольтерри. Цей напрям дослідження є перспективним та багатообіцяючим, якщо спиратися на висновки Розділу 1, згідно якому, треба збудувати сурогатну модель на основі встановлення інформаційного зв'язку між нелінійними динамічними моделями Вольтерри та моделями TDNN.

2.3. Інформаційний зв'язок між моделями Вольтерри та нейронними мережами з часовими затримками

Аналітичні моделі намагаються точно виразити виміряну поведінку об'єкта, формулюючи рівняння, яке представляє вимірюваний параметр, який може бути функцією декількох незалежних змінних. Проте, поведінкові моделі на основі рядів Вольтерри зберігають свою ефективність тільки для слабких нелінійностей і вимагають важких зусиль по ідентифікації ядер.

Зростаючий інтерес до застосування обох підходів до моделювання нелінійних динамічних систем мотивує порівняльне дослідження для встановлення зв'язку між обома моделями та удосконалення моделювання при їх комбінованому використанні для підвищення швидкості та точності моделювання.

В публікаціях існують спроби оцінки ядер Вольтерри на основі навчених тришарових НМ з сигмоїдальною функцією активації [130, 136]. Недоліки сигмоїдальної та похідних від неї активаційних функцій, що відзначалися вище, не сприяли поширенню цього підходу в практичних задачах. Тому в даному дослідженні виправляється цей недолік і здійснюється спроба оцінки ядер Вольтерри на основі навченої тришарової TDNN з функцією активації у вигляді ReLU.

Модель TDNN у вигляді залежності вихідного сигналу $y(t_n)$ в момент часу t_n від значень вхідного сигналу $x(t_n)$ в загальному вигляді визначаються за виразом

(1.4). При використанні активаційної функції ReLU модель TDNN (1.4) приймає вигляд:

$$y(t_n) = b_0 + S_0 \sum_{i=1}^K w_i S_i \left(b_i + \sum_{j=1}^M w_{i,j} x(t_{n-j}) \right), \quad (2.21)$$

Якщо апроксимувати активаційну функцію ReLU (2.19) поліномом (2.20) на певному інтервалі $[-q, q]$, а вихідну активаційну функцію прийняти $S_0=x$ для спрощення математичних викладок, модель TDNN (2.21) приймає вигляд:

$$y(n) = b_0 + S_0 \sum_{i=0}^K r_i^2 \sum_{h=0}^H a_h \left(b_i + \sum_{j=0}^M r_{i,j}^1 x(n-j) \right)^h \quad (2.22)$$

Функціональний ряд у вигляді моделі Вольтерри є узагальненням степеневого ряду на функціональний простір (1.6). В даному виразі перший член ряду – це лінійна модель у вигляді інтеграла згортки. Члени ряду вищих порядків враховують динамічні та нелінійні характеристики вищих порядків.

Аналіз моделі Вольтерри (1.6) та TDNN (2.22) дозволяє зробити висновок про ізоморфізм обох виразів, тобто складовим однієї моделі можна зіставити складові іншої моделі. Тому, для побудови сурогатної моделі для TDNN у вигляді моделі Вольтерри достатньо встановити інформаційний зв'язок між виразами (1.6) та (2.22).

Отже, оцінки ядер Вольтарри можна виразити через значення зсуву нейронів b_0, b_j та вагові коефіцієнти $w_i, w_{i,j}$ ($i=1, \dots, K, j=1, \dots, M$) вихідного та прихованого шарів НМ відповідно, а також, коефіцієнтів a_p поліному, що апроксимує функцію активації в прихованому шарі.

Так, задаючи ступінь апроксимуючого поліному H , можна отримати інформаційний зв'язок моделей у вигляді TDNN та функціональним поліномом у аналітичному вигляді.

При $p=0$:

$$y(n) = b_0 + a_0 \sum_{i=0}^K w_i \cdot \quad (2.23)$$

$$v_0 = b_0 + a_0 \sum_{i=0}^K w_i \cdot \quad (2.24)$$

При $p=1$:

$$y(n) = b_0 + \sum_{i=0}^K w_i (a_0 + a_1 b_i) + a_1 \sum_{i=0}^K w_i \sum_{j=0}^M w_{i,j} x(n-j) \cdot \quad (2.25)$$

$$v_0 = b_0 + \sum_{i=0}^K w_i (a_0 + a_1 b_i); \quad v_1 = a_1 \sum_{i=0}^K w_i \sum_{j=0}^M w_{i,j} \cdot \quad (2.26)$$

При $p=2$:

$$y(n) = b_0 + \sum_{i=0}^K w_i (a_0 + a_1 b_i + a_2 b_i^2) + \\ + \sum_{i=0}^K w_i (a_1 + 2a_2 b_i) \sum_{j=0}^M w_{i,j} x(n-j) + \cdot \quad (2.27)$$

$$+ a_2 \sum_{i=0}^K w_i \sum_{j=0}^M w_{i,j}^2 x^2(n-j)$$

$$v_0 = b_0 + \sum_{i=0}^K w_i (a_0 + a_1 b_i + a_2 b_i^2);$$

$$v_1 = \sum_{i=0}^K w_i (a_1 + 2a_2 b_i) \sum_{j=0}^M w_{i,j}; \quad \cdot \quad (2.28)$$

$$v_2 = a_2 \sum_{i=0}^K w_i \sum_{j=0}^M w_{i,j}^2$$

При $p=3$:

$$y(n) = b_0 + \sum_{i=0}^K w_i (a_0 + a_1 b_i + a_2 b_i^2 + a_3 b_i^3) + \\ + \sum_{i=0}^K w_i (a_1 + 2a_2 b_i + 3a_3 b_i^2) \sum_{j=0}^M w_{i,j} x(n-j) + \cdot \quad (2.29)$$

$$+ \sum_{i=0}^K w_i (1 + 3a_3 b_i) \sum_{j=0}^M w_{i,j}^2 x^2(n-j) +$$

$$+ a_3 \sum_{i=0}^K w_i \sum_{j=0}^M w_{i,j}^3 x^3(n-j)$$

$$\begin{aligned}
v_0 &= b_0 + \sum_{i=0}^K w_i (a_0 + a_1 b_i + a_2 b_i^2 + a_3 b_i^3); \\
v_1 &= \sum_{i=0}^K w_i (a_1 + 2a_2 b_i + 3a_3 b_i^2) \sum_{j=0}^M w_{i,j}; \\
v_2 &= \sum_{i=0}^K w_i (1 + 3a_3 b_i) \sum_{j=0}^M w_{i,j}^2; \\
v_3 &= a_3 \sum_{i=0}^K w_i \sum_{j=0}^M w_{i,j}^3.
\end{aligned} \tag{2.30}$$

Вирази (2.24), (2.26), (2.28), (2.30) є оцінками ядер Вольтерри порядків $p=0,1,2,3$ відповідно, отриманими за допомогою TDNN. Аналогічно, можна отримати вирази для оцінок ядер Вольтерри більших порядків, але на практиці моделі у вигляді поліномів Вольтерри вищих порядків використовуються досить рідко.

Дослідження області ефективного використання запропонованих моделей оцінки ЯВ за допомогою TDNN, проведені в наступному параграфі, доводять переваги запропонованих моделей: високу точність при моделюванні динамічних об'єктів зі значними нелінійностями.

2.4. Дослідження сурогатних моделей у вигляді поліномів Вольтерри

Сурогатні моделі у вигляді поліномів Вольтерри, запропоновано використовувати для ідентифікації динамічних об'єктів з нелінійними динамічними характеристиками. Ці моделі можуть бути особливо корисні у випадках, коли складність системи робить аналітичне моделювання трудомістким або навіть неможливим. Натомість, сурогатні моделі у вигляді поліномів Вольтерри можуть бути побудовані на основі моделей машинного навчання, що робить їх більш гнучкими та застосовними до різних прикладних завдань. Інтеграція методів детермінованої ідентифікації та машинного навчання надає можливість створення більш точних та стійких сурогатних моделей.

Для дослідження ефективності запропонованих сурогатних моделей підготовано та проведено обчислювальний експеримент з тестовим нелінійним динамічним ОД.

2.4.1. Імітаційна модель тестового об'єкта. Тестовий ОД являє собою структуру з динамічним блоком першого порядку та нелінійним блоком $f(y)$ у зворотному зв'язку [39, 108, 174] (рис. 2.1).

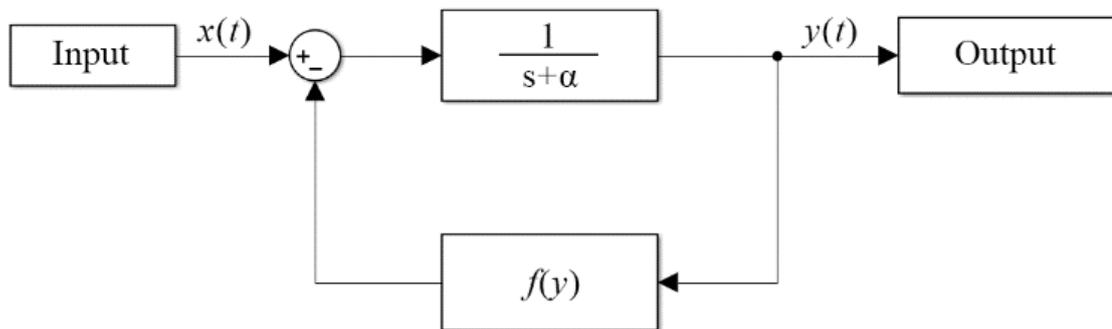


Рис. 2.1 – Імітаційна модель тестового нелінійного динамічного об'єкта

В якості нелінійної функції $f(y)$ в блоці зворотного зв'язку розглядається поліноміальна функція $f(y)=y^2$. При проведенні експерименту прийнято параметр імітаційної моделі: $\alpha=2.64$.

Імітаційна модель досліджується за допомогою тестових сигналів з різною амплітудою: імпульсних, ступінчастих, лінійних і гармонійних. Навчальний набір даних формується з результатів експерименту «вхід-вихід» – множини векторів $\{x(t), y(t)\}$ для кожного типу вхідного сигналу.

На рис. 2.2 наведено перехідні характеристики $y_{03}(t)$, $y_{06}(t)$, $y_{09}(t)$ тестового об'єкта при дії на вході ступінчастого сигналу $x(t)=a\Theta(t)$, $a=0.3, 0.6, 0.9$. Рисунок демонструє нелінійні та динамічні властивості тестового об'єкта.

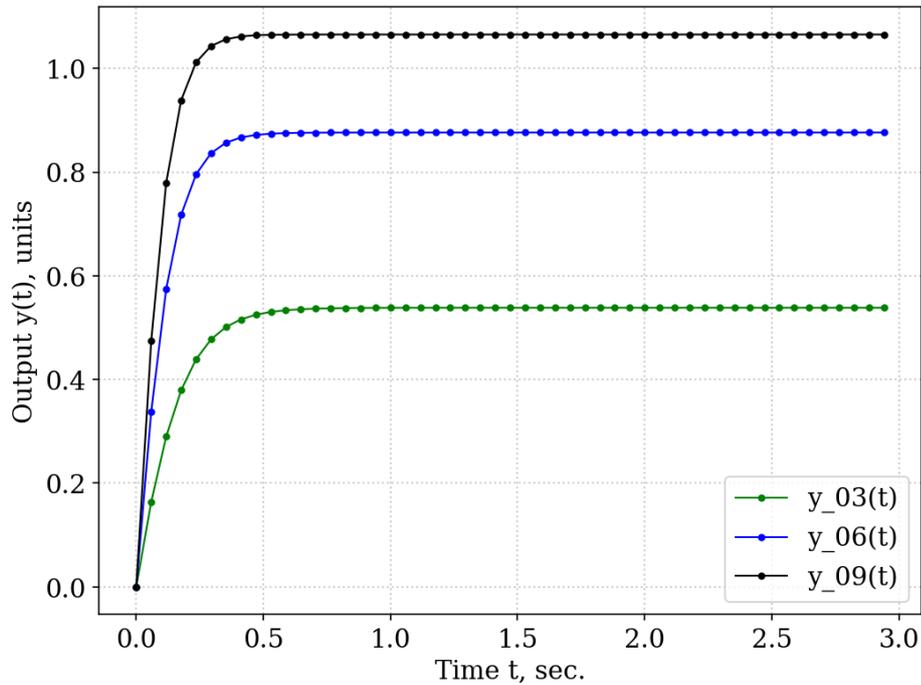


Рис. 2.2 – Перехідні характеристики $y_{03}(t)$, $y_{06}(t)$, $y_{09}(t)$ тестового об’єкта при дії на вході ступінчастого сигналу $x(t)=a\Theta(t)$, $a=0.3, 0.6, 0.9$.

2.4.2. Модель об’єкта на основі нейронної мережі з часовою затримкою.

Для побудови TDNN моделі треба визначити наступні параметри M , K , w_i , w_{ij} та b_0 , b_i . При цьому, M та K визначають архітектуру НМ, а w_i , w_{ij} та b_0 , b_i визначаються в результаті навчання моделі.

Для визначення найкращих значень M та K у заданій структурі TDNN збудовано ряд НМ з різною кількістю нейронів у вхідному і прихованому шарах. Побудова множини НМ здійснюється засобами програмної бібліотеки *Keras*. Це відкрита бібліотека для машинного навчання, написана мовою програмування *Python*. Вона надає високорівневий інтерфейс для побудови та навчання НМ. Детальний опис інструментальних засобів моделювання наведено в Розділі 4.

Результат експерименту у вигляді залежності втрат від кількості нейронів у вхідному і прихованому шарах представлено на рис. 2.3. В якості функції втрат використовується середньоквадратична помилка *mse* з бібліотеки *Keras*.

Результат експерименту у вигляді залежності часу навчання (епохи) від кількості нейронів у вхідному і прихованому шарах представлено на рис. 2.4.

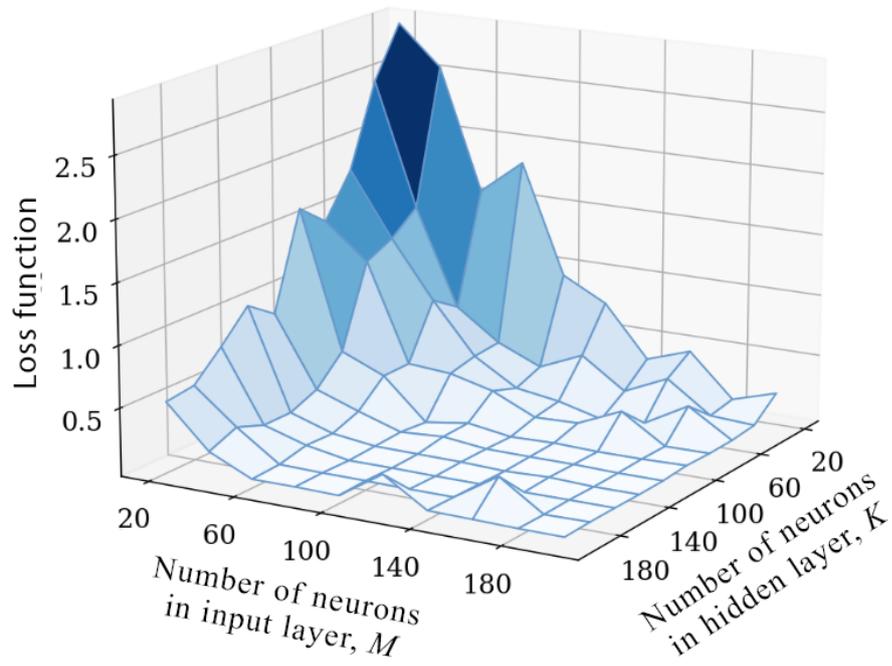


Рис. 2.3 – Залежність функції втрат від кількості нейронів у вхідному і прихованому шарах

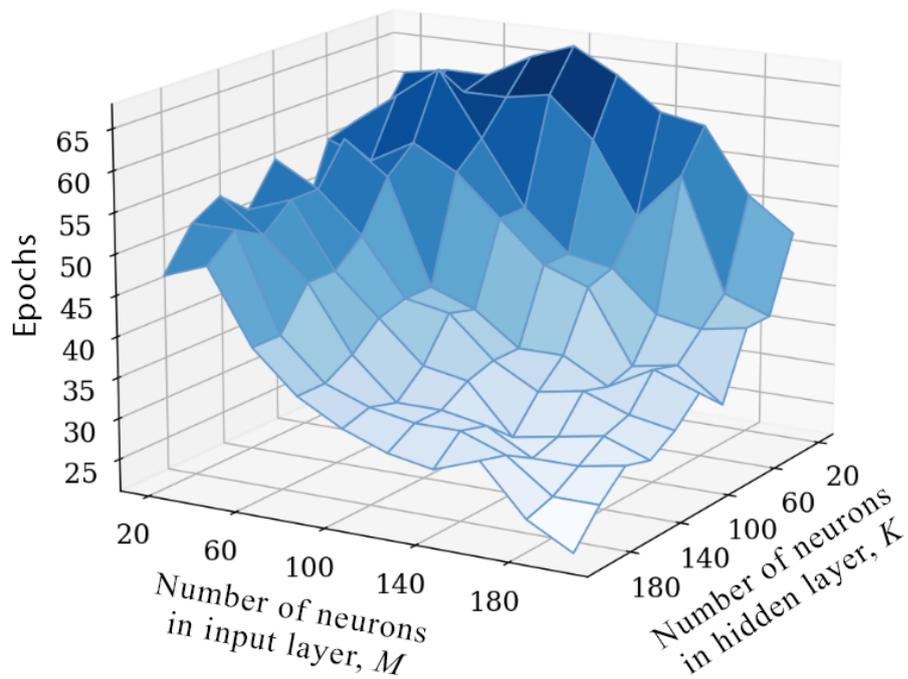


Рис. 2.4 – Залежність часу навчання від кількості нейронів у вхідному і прихованому шарах

В результаті співставлення графіків 2.3 та 2.4 обрано значення кількості нейронів у вхідному та прихованому шарах TDNN $M=15$ та $K=50$ відповідно, що забезпечують встановленим умовами експерименту рівень втрат ($loss < 1.0$) при прийнятному часі навчання ($epochs < 40$).

На рис. 2.5 наведено наочне зображення структури нейромережевої моделі у вигляді графа, отримане за допомогою функції *plot_model* бібліотеки *Keras*. Ця візуалізація може бути корисна при налагодженні та розумінні архітектури НМ, особливо у разі використання складних моделей з багатьма шарами.

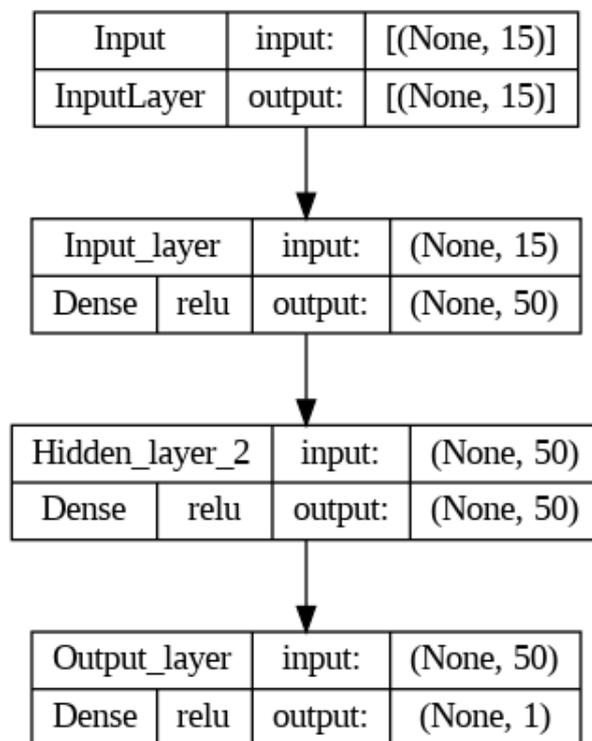


Рис 2.5 – Структурна схема TDNN у вигляді тришарової мережі з прямим поширенням сигналу

За результатами навчання TDNN знайдено значення параметрів w_i , w_{ij} та b_0 , b_i , що визначають модель (2.21). Однак, в результаті отримано модель, параметри w_i , w_{ij} та b_0 , b_i якої не мають фізичного сенсу, тому її роботу важко інтерпретувати.

2.4.3. Побудова інтерпретаційних моделей. Інтерпретація збудованої моделі здійснюється за допомогою розглянутих вище підходів.

Графічна інтерпретація. Даний підхід до інтерпретації дозволяє візуалізувати процес навчання нейронної мережі та результати навчання у наступних формах.

– Скалярні функції втрат (*loss function*) та точності (*accuracy*) нейронної мережі наведені на рис. 2.6;

– Гістограма зміни розподілу у часі параметрів w_i , b_0 у вхідному (рис. 2.7) та $w_{i,j}$, b_i у прихованому (рис. 2.8) шарах.

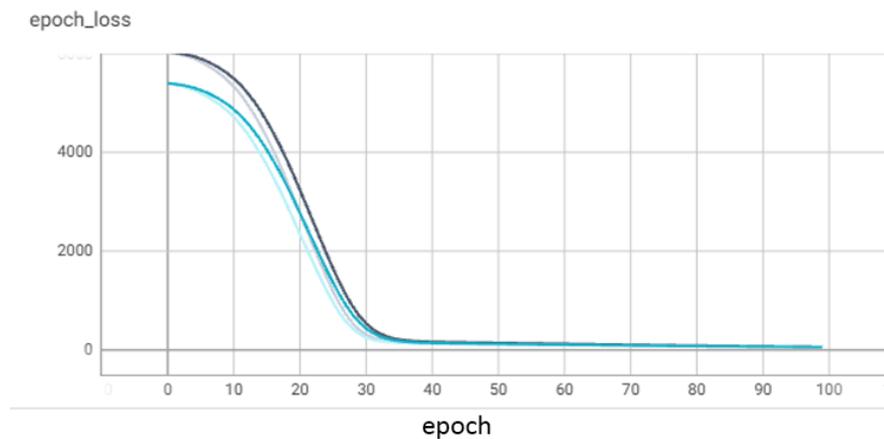


Рис. 2.6 – Скалярні функції втрат (*loss function*) та точності (*accuracy*) НМ

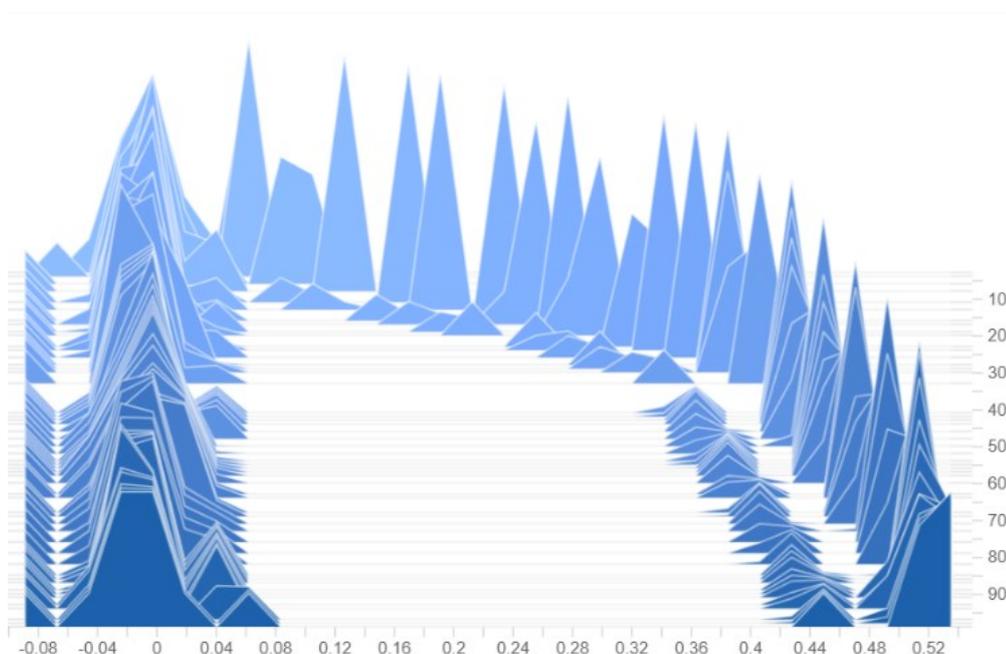


Рис. 2.7 – Гістограма розподілу даних у часі у вхідному шарі

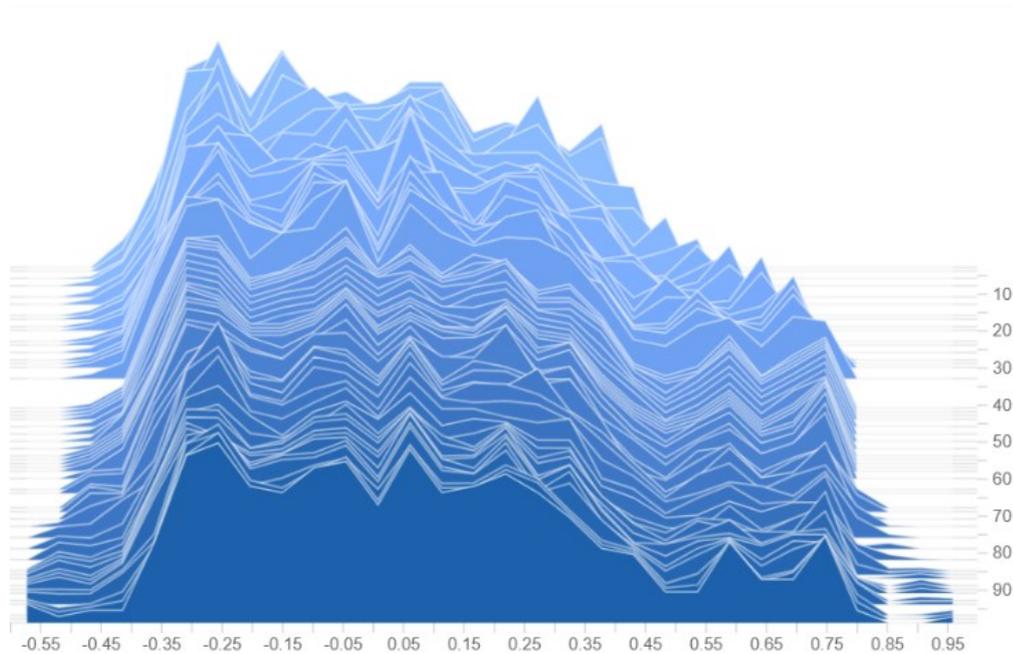


Рис. 2.8 – Гістограма розподілу даних у часі у прихованому шарі

Методи графічної інтерпретації дозволяють наочно встановити якісні показники значущості і пов'язаності ознак тестового об'єкта. Але ці моделі не дозволяють отримати кількісну інформацію щодо впливу вхідних даних на вихід моделі.

Числові оцінки моделі. Даний підхід до інтерпретації дозволяє оцінити процес навчання НМ та результати навчання у наступних формах:

- розрахунок оцінок значущості і пов'язаності ознак;
- візуалізація оцінок значущості і пов'язаності ознак;
- текстове пояснення (моделі у вигляді антецедентів: «при зменшенні/збільшенні ознаки x_i , зростає ймовірність зменшення/збільшення» передбачення $y(x)$).

В якості демонстрації даного підходу вирішується задача інтерпретації тестової TDNN моделі методом SHAP [65, 79]. У відповідності до методу будується графік важливості використовуваних у моделі ознак (рис. 2.9).

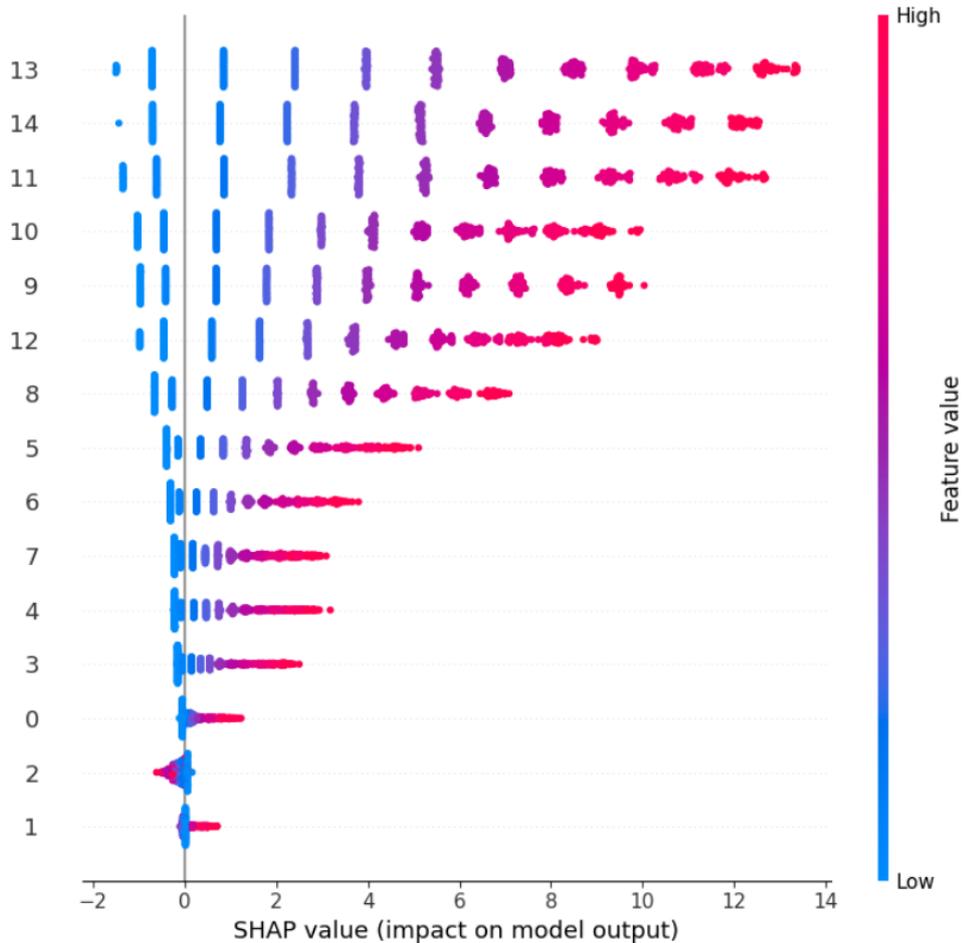


Рис. 2.9 – Графік важливості ознак тестової моделі за методом SHAP

З графіка, наведеного на рис. 2.9 можна зробити наступні висновки:

- практично всі ознаки взаємопов’язані;
- ознаки x_i з меншими порядковими номерами (розташовані ближче до початку вектору $x(t)$) мають більшу значущість (вплив на вихідне значення $y(t)$).

Аналітична модель.

Для інтерпретації нейромережевої моделі тестового об’єкту у вигляді аналітичного виразу інтегро-степеневого ряду використовується залежність (2.26). За допомогою цієї залежності визначено ядро Вольтерри першого $w_1(\tau_1)$ та діагональний перетин ядра Вольтерри другого $w_2(\tau_1, \tau_2)$ порядків. З використанням визначених ядер Вольтерри за виразом (2.25) будується модель інтерпретації нейронної мережі в аналітичному вигляді.

$$\begin{aligned}
y(n) = & b_0 + \sum_{i=0}^K w_i (a_0 + a_1 b_i + a_2 b_i^2) + \\
& + \sum_{i=0}^K w_i (a_1 + 2a_2 b_i) \sum_{j=0}^M w_{i,j} x(n-j) + \cdot \\
& + a_2 \sum_{i=0}^K w_i \sum_{j=0}^M w_{i,j}^2 x^2(n-j)
\end{aligned} \tag{2.31}$$

$$\begin{aligned}
v_0 = & b_0 + \sum_{i=0}^K w_i (a_0 + a_1 b_i + a_2 b_i^2); \\
v_1 = & \sum_{i=0}^K w_i (a_1 + 2a_2 b_i) \sum_{j=0}^M w_{i,j}; \quad \cdot \\
v_2 = & a_2 \sum_{i=0}^K w_i \sum_{j=0}^M w_{i,j}^2
\end{aligned} \tag{2.32}$$

На рис. 2.10 наведено ядро Вольтерри першого порядку $w_1(\tau)$ та діагональний перетин ядра Вольтерри другого порядку $w_2(\tau, \tau)$, отримані на основі навченої TDNN моделі за виразами (2.32).

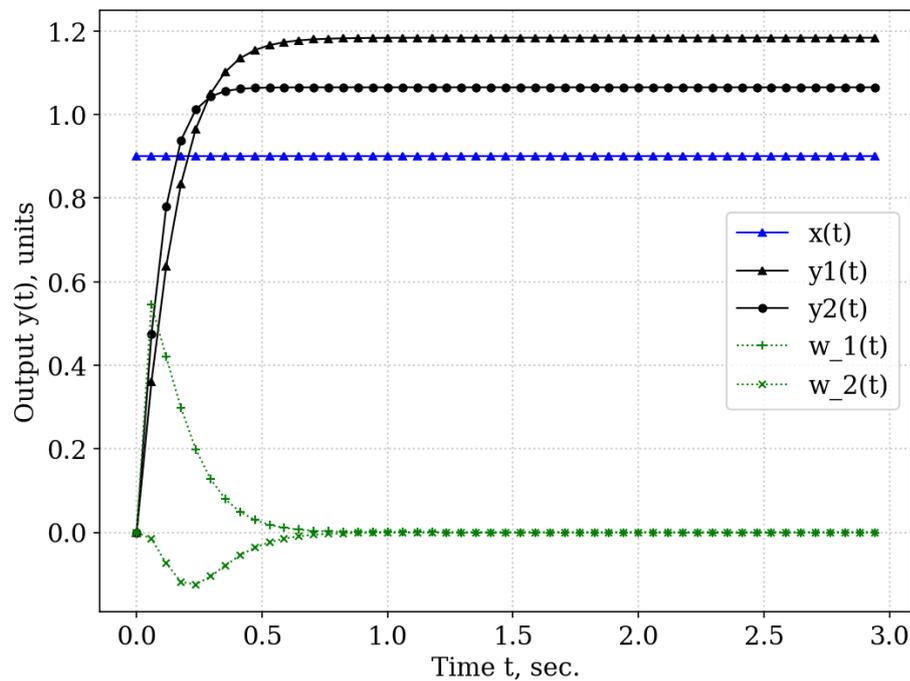


Рис. 2.10. Ядра Вольтерри першого порядку $w_1(\tau)$, діагональний перетин багатовимірної вагової функції другого порядку $w_2(\tau, \tau)$ і сурогатні моделі першого $y_1(t)$ та другого $y_2(t)$ порядку визначені для тестового ОД

Модель тестового ОД $y_v(t)$, збудована на основі виразу (2.32) наведена на рис. 2.11. Отримана модель порівнюється з імітаційною моделлю $y(t)$, нейромережевою моделлю $y_m(t)$ та лінійною сурогатною моделлю $y_l(t)$, збудовану методом LIME, отриманих на тих же даних.

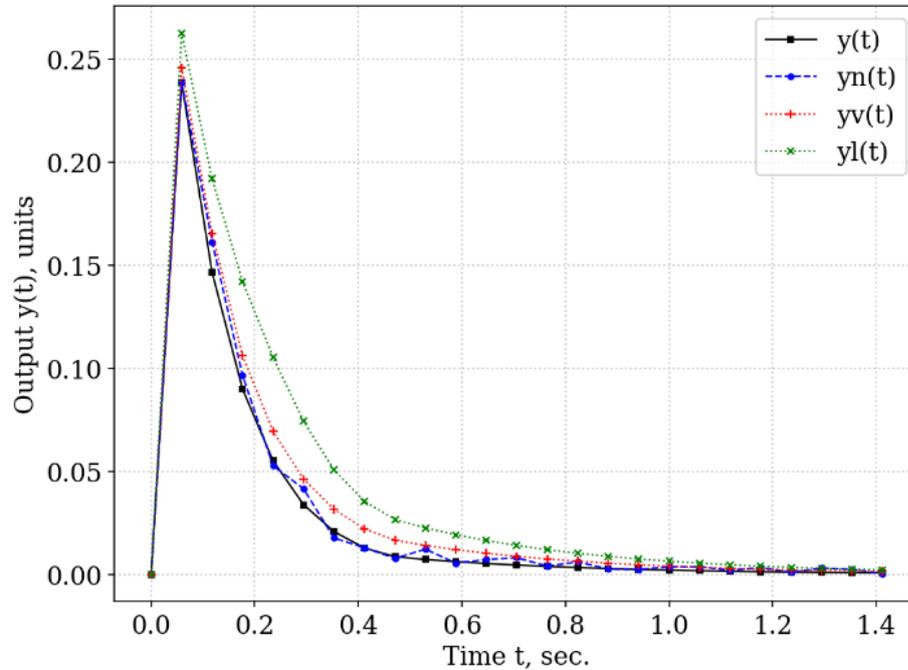


Рис. 2.11. Порівняння сурогатної моделі у вигляді функціонального поліному $y_v(t)$ з імітаційною $y(t)$, нейромережевою $y_l(t)$ та лінійною сурогатною $y_l(t)$ моделями.

Експеримент демонструє, що інтерпретаційна модель, збудована за виразом (2.32) демонструє точність на 10-12% вище за лінійну модель при використанні тестових вхідних сигналів з амплітудами $a \in [0.6, 0.9]$. Експеримент демонструє переваги в точності сурогатної моделі у вигляді функціонального ряду $y_v(t)$ над лінійною сурогатною $y_l(t)$.

Отримана інтерпретаційна модель прозоро демонструє структуру тестового об'єкта та функціональний зв'язок ознак, які впливають на його поведінку.

2.5. Висновки за розділом

В розділі розглянуто підхід до ідентифікації нелінійних динамічних ОД у вигляді моделей Вольтерри за даними експерименту «вхід-вихід» з використанням тестових сигналів у вигляді імпульсних функцій.

Виявлено недоліки підходу до ідентифікації ОД у вигляді моделей Вольтерри, а саме: обчислювальні обмеження при оцінці ядер Вольтерри високих порядків, що обумовило необхідність пошуку нових ефективних способів ідентифікації динамічних ОД з нелінійністю вищих порядків. В якості такого підходу розглядаються штучні НМ з часовими затримками.

Обґрунтовано структуру штучної нейронної мережі з часовими затримками. Для зменшення обчислювальної складності та підвищення швидкості навчання НМ з часовими затримками запропоновано використання в нейронах прихованого шару функції активації у вигляді випрямляючої лінійної ланки.

Для підвищення інтерпретованості нейромережевої структури встановлено аналітичну залежність між ваговими коефіцієнтами TDNN та моделями Вольтерри. На основі встановленого інформаційного зв'язку розроблено метод оцінки ЯВ вищих порядків за допомогою TDNN з використанням функції активації у вигляді лінійної ректифікації. На основі оцінок ЯВ вищих порядків запропоновано сурогатні моделі для інтерпретації нелінійної динаміки нейромережевих моделей.

Перевагою розробленого методу у порівнянні з існуючими є спрощений алгоритм навчання НМ з часовими затримками, що дозволяє підвищити оперативність навчання НМ з часовими затримками та точність ідентифікації інформаційних моделей у вигляді РВ шляхом оцінки ЯВ вищих порядків за допомогою НМ з часовими затримками.

З метою верифікації запропонованих моделей організовано та проведено експерименту з ідентифікації тестового нелінійного динамічного об'єкту. Результати експерименту свідчать про високу точність і ефективність побудови сурогатних моделей на основі поліномів Вольтерри.

Матеріали розділу опубліковані в роботах [57, 58, 70, 89, 159, 160].

РОЗДІЛ 3

МЕТОД ТА ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ІНТЕРПРЕТАЦІЇ МОДЕЛЕЙ
МАШИННОГО НАВЧАННЯ

Для практичного вирішення задачі побудови ефективних інтерпретованих моделей TDNN не достатньо розробки одного лише метода. Використання нейромережових моделей нелінійних динамічних об'єктів, застосування різних методів машинного навчання, побудова сурогатних моделей різних порядків, а також, формування навчальних та тестових датасетів породжує задачу організації процесів багаторазового перетворення інформації для опрацювання, передавання та зберігання даних в рамках єдиної інформаційної технології (ІТ) [19, 23, 42, 167, 171, 174, 176].

ІТ побудови оптимальної сурогатної моделі за критерієм точності моделювання (де якість моделі оцінюється в сукупності з моделлю машинного навчання, побудованої для даного ОД) являє собою досить масштабну задачу, що зводиться до послідовного спрямованого підбору структури нейромережової моделі та перебору моделей Вольтерри різних порядків з метою побудови сурогатної моделі, яка забезпечує достатню точність ідентифікації.

Даний розділ присвячено розвитку методу інтерпретації нейромережових моделей нелінійних динамічних об'єктів з неперервними характеристиками. На основі цього методу розробляється ІТ побудови сурогатної моделі для нейромережової моделі нелінійних інерційних об'єктів.

3.1. Метод інтерпретації моделей машинного навчання нелінійних динамічних об'єктів

Суттєва практична цінність відтвореного зв'язку моделей на основі TDNN та функціональних поліномів Вольтерри полягає в подальшому розвитку методу оцінки ядер Вольтерри безпосередньо за параметрами TDNN [57, 58, 89]. Цей метод є корисним в задачах забезпечення якості програмного забезпечення

інтелектуальних систем шляхом використання більш адекватних та точних сурогатних моделей у вигляді функціональних поліномів Вольтерри.

Метод інтерпретації моделей машинного навчання нелінійних динамічних об'єктів, що розробляється в роботі, полягає у використанні TDNN з активацією у вигляді випрямляючої лінійної ланки у прихованому шарі. На основі відтвореного зв'язку моделей на основі TDNN та ядер Вольтерри для отриманої нейромережевої моделі будується сурогатна модель у вигляді функціональних поліномів Вольтерри. Загалом, розроблений метод може бути представлений у вигляді низки окремих етапів, що при послідовній реалізації дозволяють отримати сурогатну модель, яка інтерпретує поведінку нейронної мережі.

На *етапі 1* виконується підготовка вихідних даних у вигляді вхідних та вихідних сигналів, отриманих за допомогою експерименту «вхід/вихід» для навчання TDNN: навчальний набір даних $x(t_n)$ формується з вхідних сигналів згідно виразу (2.17), цільовий набір даних $y(t_n)$ формується з вихідних сигналів у вигляді вектору дискретних значень – відліків сигналу $y(t_n) = [y_n, y_{n-1}, \dots, y_{n-M-1}]$ в відповідні моменти часу $t_n, t_{n-1}, \dots, t_{n-M-1}, t_n = n\Delta t, n=1, 2, \dots$

На *етапі 2* для ідентифікації моделі ОД за допомогою сформованих даних використовується тришарова TDNN з активацією у вигляді випрямляючої лінійної ланки у прихованому шарі.

На *етапі 3* на основі отриманих TDNN моделей визначаються ядра Вольтерри $w_n(t-\tau_1, \dots, t-\tau_n)$ n -го порядку за допомогою встановлених вирази зв'язку цих моделей (2.23)-(2.30).

На *етапі 4* за визначеними ядрами Вольтерри $w_n(t-\tau_1, \dots, t-\tau_n)$ будується сурогатна модель n -го порядку у вигляді поліномів Вольтерри (2.1) та оцінюється точність побудованої моделі одним з критеріїв (1.9)-(1.13).

Для реалізації методу інтерпретації моделей машинного навчання нелінійних динамічних об'єктів, наведеного вище, розроблено обчислювальний алгоритм, який представляється наступною послідовністю кроків.

Крок 1. Встановити розмір пам'яті моделі M , кількість нейронів у прихованому шарі K та точність ϵ моделювання, порядок сурогатної моделі $k=1$.

Крок 2. Задати множину $\mathbf{D}(t)=\{\mathbf{x}(t_n), y(t_n)\}$ – навчальну вибірку, $n=1, \dots, N$, де N – кількість вимірювань за експериментом «вхід-вихід»; визначити операцію попередньої нормалізації даних за виразом:

$$\mathbf{D}'(t)=[(\mathbf{D}(t)-\text{mean}(\mathbf{D}(t)))/(\max(\mathbf{D}(t))-\min(\mathbf{D}(t)))]. \quad (3.1)$$

Крок 3. Прийняти номер ітерації навчання $k=1$, ініціалізувати за допомогою випадкової величини на інтервалі $(0, 1)$ вектори зсувів $\mathbf{B}=[b_0, b_j]$ та матриці вагових коефіцієнтів $\mathbf{W}=[w_i, w_{i,j}]$ ($i=1, \dots, K, j=1, \dots, M$).

Крок 4. Визначити на основі методу зворотнього розповсюдження помилки з застосуванням алгоритму Левенберга-Марквардта зсуви b_0, b_j та вагові коефіцієнти $w_i, w_{i,j}$ НМ.

Крок 5. Перевірити умови завершення навчання. Якщо $e_k(t) \leq e$, то перейти до *Кроку 6*, якщо $T \leq k$ то збільшити розмір пам'яті моделі M , кількість нейронів у прихованому шарі K та перейти до *Кроку 3*, інакше, прийняти $k=k+1$ і перейти до *Кроку 4*.

Крок 6. Визначити ядра Вольтерри порядків $0, \dots, p$ за допомогою виразів (2.24)- (2.30).

Крок 7. Збудувати сурогатну модель ОД за виразом (2.23).

Крок 8. Визначити точність сурогатної моделі e_s . Перевірити умови завершення навчання. Якщо $e_s \leq e$, то завершити роботу алгоритму, інакше, прийняти $p=p+1$ і перейти до *Кроку 6*.

Наведений алгоритм може бути компактно записаний за допомогою наступного псевдокоду:

Algorithm: surrogate_nn_model

- 1: **Input:** $M, K, e, T, k, N, \mathbf{D}$
- 2: **Output:** \mathbf{V}_n
- 3: $normalize \leftarrow \mathbf{D}$
- 4: $\mathbf{D}_{train}, \mathbf{D}_{test} \leftarrow \mathbf{D}$
- 5: $random \leftarrow \mathbf{B}[K], \mathbf{W}[M, K]$
- 6: $k \leftarrow 0$
- 7: **while** $e_k(t) > e$ or $k < T$ **do**
- 8: $k \leftarrow k + 1$
- 9: **for** $i = 1, \dots, N$ **do**

```

10:           evaluations ← training procedure(B, W, Dtrain, e)
11:       end for
12:       loss ← mse(B, W, Dtest)
13:   end while
14:   Vn ← calcVK(B, W)

```

3.2. Інформаційна технологія інтерпретації моделей машинного навчання нелінійних динамічних об'єктів

Реалізація методу інтерпретації моделей машинного навчання нелінійних динамічних об'єктів окрім виконання означених етапів потребує низки операцій з підготовки та проміжної обробки даних на кожному з етапів. Деякі алгоритми обробки даних, наприклад, визначення оптимальних за швидкістю навчання параметрів структури нейронної мережі, носять характер окремих самостійних задач [88, 166]. Тому для вирішення задачі інтерпретації моделей машинного навчання на базі наведеного методу розроблено базову ІТ побудови сурогатних моделей TDNN-мереж.

Розроблена ІТ вирішує цілий спектр завдань, включаючи сукупність методів, моделей для збору, передачі, оброблення та поширення даних з метою ефективної організації процесу інтерпретації моделей машинного навчання нелінійних динамічних об'єктів.

Для розробки ІТ, що забезпечує підвищення продуктивності та ефективності використання інформації та ресурсів, обрано V-модель розробки інформаційних систем (ІС), спрямовану на спрощення розуміння та вирішення задач, пов'язаних із розробленням систем [12, 61, 86, 91, 116, 174].

V-модель заснована на об'єднанні фази розробки з кожною відповідною стадією тестування. Так, завдання розробки йдуть згори донизу зі зростанням їх деталізації по лівій стороні літери «V», а завдання тестування – вгору по правій стороні літери «V». В цій структурі етапи розробки відповідають етапам тестування, розташованим на одній горизонталі.

Обрана модель розробки надає проекту додаткову прозорість, завдяки чому

дозволяє підвищити якісні показники програмного забезпечення ІС, що розробляється, шляхом встановлення та уніфікації цілей на проміжних етапах та опису відповідних їм рішень і відповідальних осіб.

Згідно прийнятій V-моделі завдання розробки ІТ інтерпретації моделей машинного навчання нелінійних динамічних об'єктів включають наступні етапи:

- аналіз вимог.
- проектування системи.
- архітектурний дизайн.
- розробка модулів.

Тестування ІТ інтерпретації моделей машинного навчання включає наступні етапи:

- модульне тестування.
- інтеграційне тестування.
- системне тестування.
- користувацьке приймальне тестування.

Обрану V-модель розробки з вказаними етапами розробки та тестування ІС наведено на рис. 3.1.

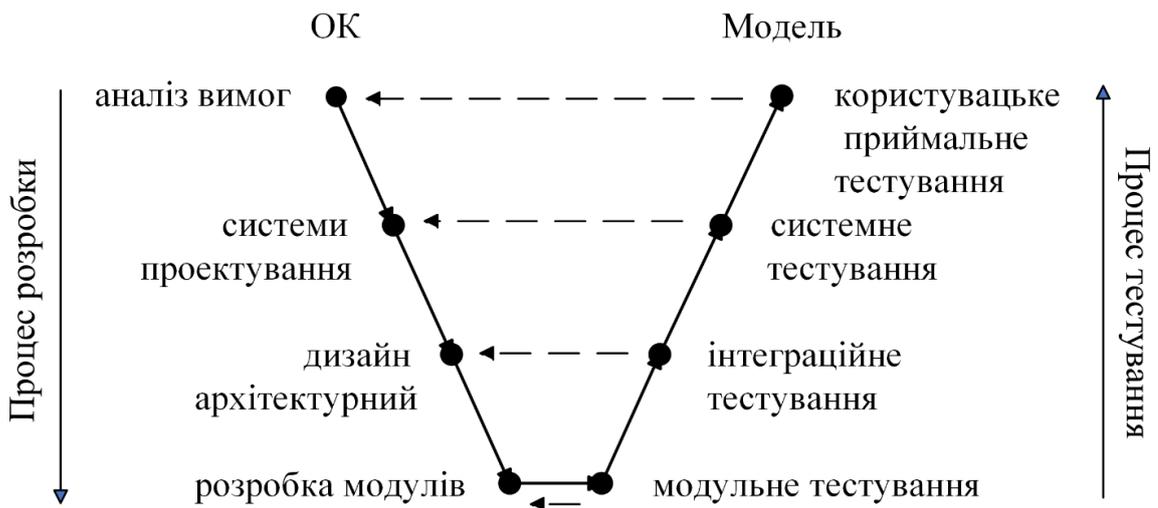


Рисунок 3.1 – V-модель розробки ІС

В табл. 3.1. наведено перелік етапів розробленої ІТ із зазначенням мети, вхідних та вихідних даних та моделі для кожного етапу.

Таблиця 3.1 – IT інтерпретації моделей машинного навчання нелінійних динамічних об'єктів

№	Етап		Опис
1	2		3
1.	Підготовка навчального датасету	1.1. Формування датасету	<p><i>Мета:</i> формування дискретного навчального датасету у вигляді, придатному для використання TDNN</p> <p><i>Вхід:</i> вхідні $x(t)$ та вихідні $y(t)$ сигнали</p> <p><i>Модель:</i> матриця дискретних відліків сигналів $x(t)$ із зсувом у часі (2.15) та відліків сигналів $y(t)$</p> <p><i>Вихід:</i> навчальний датасет $\{x(t_n), y(t_n)\}$</p>
		1.2. Попередня обробка датасету	<p><i>Мета:</i> стабілізація дисперсії відліків сигналів $x(t)$, $y(t)$</p> <p><i>Вхід:</i> навчальний датасет $\{x(t_n), y(t_n)\}$</p> <p><i>Модель:</i> нормалізація даних за виразом (3.1)</p> <p><i>Вихід:</i> навчальний датасет $\{x'(t_n), y'(t_n)\}$ з розподілом ознак близьким до нормального</p>
2.	Ініціалізація структури TDNN		<p><i>Мета:</i> визначити кількість нейронів у вхідному M та скритому K шарах</p> <p><i>Вхід:</i> навчальний датасет $x'(t_n)$</p> <p><i>Модель:</i> оцінка функції втрат(1.5)</p> <p><i>Вихід:</i> повнозв'язна тришарова структура НМ з M нейронами у вхідному та K нейронами у скритому шарах</p>
3.	Ідентифікація моделі ОД		<p><i>Мета:</i> отримати модель ОД у вигляді TDNN</p> <p><i>Вхід:</i> навчальний датасет $\{x'(t_n), y'(t_n)\}$</p> <p><i>Модель:</i> процедура навчання $trainingProcedure(\mathbf{B}, \mathbf{W}, \mathbf{D}_{train}, e)$</p> <p><i>Вихід:</i> навчена нейронна мережа (2.20)</p>

1	2	3
4.	Оцінка ядер Вольтерри	<p><i>Мета:</i> отримати вирази для ядер Вольтерри</p> <p><i>Вхід:</i> вектор зсувів \mathbf{V}, матриця вагових коефіцієнтів \mathbf{W} навченої TDNN</p> <p><i>Модель:</i> вирази (2.22), (2.24), (2.26), (2.28),</p> <p><i>Вихід:</i> ядра Вольтерри $w_n(t-\tau_1, \dots, t-\tau_n)$, $n=1, 2, N$ – порядок моделі</p>
5.	Інтерпретація моделі TDNN	<p>5.1. Побудова сурогатної моделі ОД</p> <p><i>Мета:</i> отримати модель у вигляді полінома Вольтерри</p> <p><i>Вхід:</i> ядра Вольтерри $w_n(t-\tau_1, \dots, t-\tau_n)$</p> <p><i>Модель:</i> вирази (2.21), (2.23), (2.25), (2.27)</p> <p><i>Вихід:</i> модель у вигляді полінома Вольтерри</p>
		<p>5.2. Оцінка якості сурогатної моделі ОД</p> <p><i>Мета:</i> оцінка точності сурогатної моделі $e_s(t)$</p> <p><i>Вхід:</i> середньоквадратична помилка (1.9)</p> <p><i>Вихід:</i> значення точності сурогатної моделі $e_s(t)$</p>

Перший етап полягає у побудові навчального датасету $\{\mathbf{x}(t_n), \mathbf{y}(t_n)\}$ на основі множини сигналів на вході $\mathbf{x}(t)$ та виході $\mathbf{y}(t)$ ОД. Для поліпшення збіжності алгоритмів машинного навчання та запобігання домінуванню ознак датасету отримані дані піддаються додатковій обробці – нормалізації згідно (3.1).

З метою побудови нейромережевої моделі ОД на другому етапі інформаційної технології виконується ініціалізація 3-шарової структури TDNN: визначається кількість нейронів у вхідному M та скритому K шарах. Для оптимізації параметрів M та K використовується алгоритм підбору параметрів, наведений у Розділі 2.

На третьому етапі шляхом навчання нейронної мережі будується TDNN модель нелінійного динамічного об'єкта. Процес навчання здійснюється за даними навчального датасета $\{\mathbf{x}'(t_n), \mathbf{y}'(t_n)\}$. При цьому, встановлюються показники точності моделі у вигляді значення функції втрат e та тривалості процедури навчання у вигляді максимальної кількості епох навчання.

Для інтерпретації отриманої нейромережевої моделі згідно розробленому методу на четвертому етапі будується сурогатна модель у вигляді поліному Вольтерри: за допомогою виразів (2.23), (2.25), (2.27), (2.29) визначаються ядра Вольтерри $w_n(t-\tau_1, \dots, t-\tau_n)$ різних порядків $n=1, 2, N$.

На основі отриманих ядер Вольтерри за виразами (2.24), (2.26), (2.28), (2.30) будується сурогатна модель найменшого порядку, що забезпечує встановлену точність моделювання $e_S(t)$, визначену за критерієм середньоквадратичної помилки (1.9). Якщо точність сурогатної моделі не задовольняє існуючим вимогам до інформаційної системи, управління передається на попередній етап, де здійснюється побудова ядер Вольтерри порядку $n+1$.

Блок-схема ІТ інтерпретації моделей машинного навчання нелінійних динамічних об'єктів наведено на рис. 3.2.

Дослідження методу інтерпретації моделей машинного навчання нелінійних динамічних об'єктів проведено в Розділі 4 при моделювання тестового ОД. Отримані результати доводять переваги методу: підвищення точності сурогатної моделі у порівнянні з лінійною моделлю. Цей метод є добре масштабованим на різні порядки сурогатних моделей та має невисоку обчислювальну складність.

Таким чином, використання методу інтерпретації моделей машинного навчання нелінійних динамічних об'єктів дозволяє підвищити точність ідентифікації, а ІТ, розроблена на його основі являє собою ефективне рішення для створення інструментальних засобів побудови інтерпретаційних моделей.

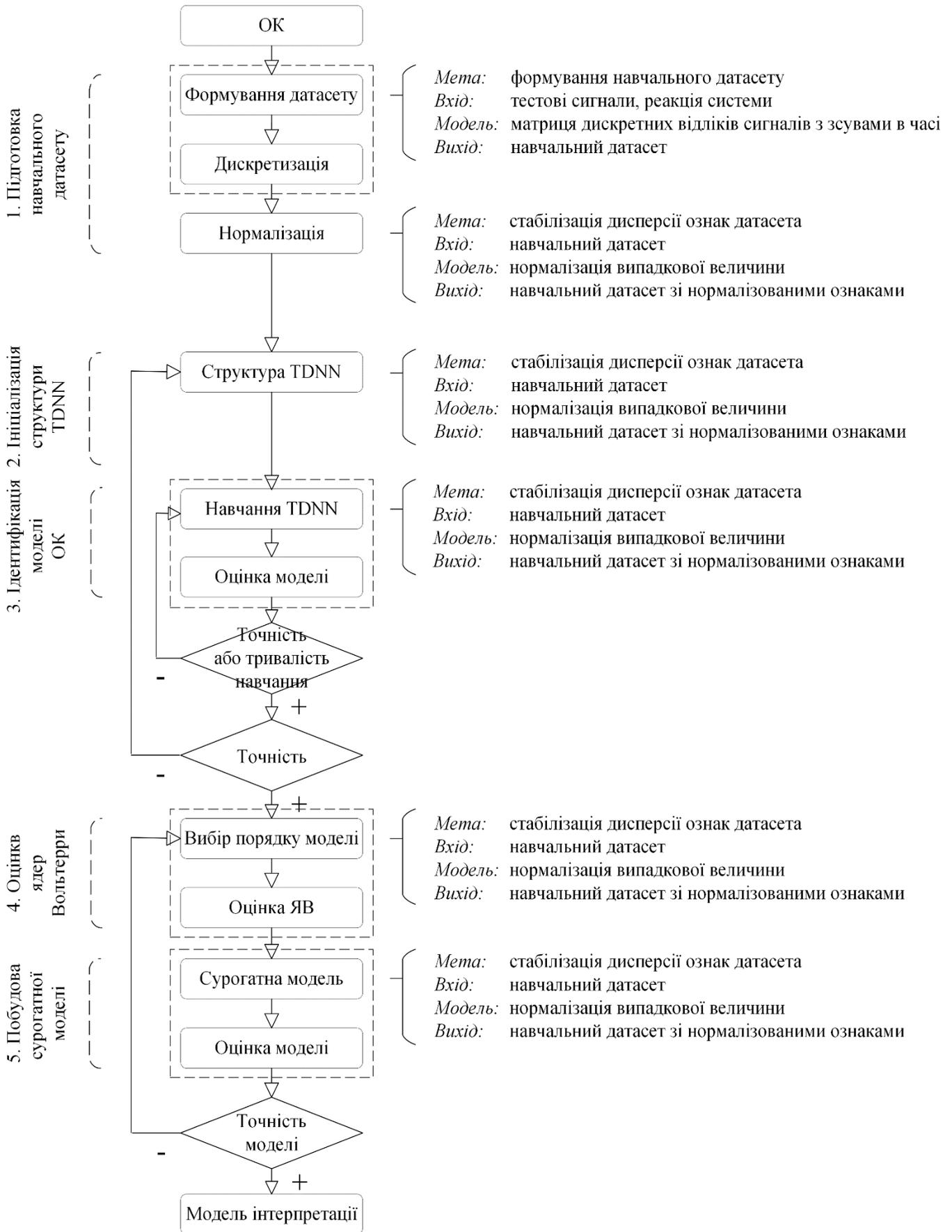


Рисунок 3.2 – IT інтерпретації моделей машинного навчання нелінійних динамічних об'єктів

3.3. Дослідження моделей машинного навчання динамічних об'єктів з гладкими та значними нелінійностями

Визначення галузі ефективного використання розробленого методу інтерпретації моделей машинного навчання є ключовим етапом у процесі його застосування. Це сприяє покращенню якості моделювання, точності результатів та ефективності використання методу у вирішенні практичних завдань.

Вибір моделі ОД завжди залежить від ряду факторів: конкретного завдання, обсягу даних, доступності ресурсів тощо. У деяких випадках використання нейронних мереж, зокрема, TDNN моделей може забезпечити кращу продуктивність, тоді як в інших випадках класичні моделі можуть бути кращими через їхню простоту та інтерпретацію.

Взагалі, класичні підходи до моделювання мають більш жорсткі умови для використання ніж нейронні мережі. Так, в Розділі 1 наведено низку обмежень нелінійних динамічних моделей у вигляді інтегро-степеневих рядів Вольтарри, які істотно звужують область ефективного застосування цих моделей.

Через наявність цих обмежень використовуються нейромережеві моделі, які сприймаються як більш гнучкі та менш обмежені порівняно з класичними моделями по наступним аспектам:

- моделювання складних залежностей: нейронні мережі можуть ефективно моделювати складні та нелінійні залежності між входами та виходами. На відміну від класичних моделей, вони здатні узагальнювати складніші функції, що дозволяє їм краще впоратися зі складними структурами даних;

- адаптація до змін у даних: нейронні мережі здатні адаптуватися до змін даних і навчатися на новій інформації. Це робить їх більш гнучкими та придатними для завдань, де структура даних може змінюватися з часом;

- розв'язання широкого спектра завдань: нейронні мережі можуть застосовуватися до різних типів завдань, таких як класифікація, регресія, сегментація, створення контенту і навіть гри. Ця універсальність робить їх більш

придатними для широкого спектру проблем, ніж деякі спеціалізовані класичні моделі.

Для визначення недоліків та переваг розробленого методу інтерпретації моделей машинного навчання нелінійних динамічних об'єктів, а також встановлення області ефективного використання запропонованих сурогатних моделей проведено низку експериментальних досліджень, направлених на порівняння точності запропонованих моделей з існуючими моделями у вигляді поліномів Вольтерри для ОД з різними характеристиками та при наявності неповної апріорної інформації про об'єкт.

Для дослідження точності моделювання динамічних об'єктів з гладкими та значними нелінійностями за допомогою TDNN моделей та визначення сфери їх ефективного застосування організовано та проведено два експерименти:

1. Дослідження масштабованості TDNN моделей до різних вхідних сигналів $x(t)$.
2. Дослідження екстраполяційних властивостей TDNN моделей.

Результати обох експериментів порівнюються з результатами моделювання за допомогою методу детермінованої ідентифікації, а саме, з моделями у вигляді інтегро-степеневих поліномів Вольтерри $y_v(t)$.

В якості тестового ОД прийнято структуру з динамічним блоком першого порядку та нелінійним блоком $f(y)$ у зворотному зв'язку, що використовується в Розділі 2 (рис. 2.1).

Функція $f(y)$ в блоці зворотного зв'язку ОД має нелінійний характер. В даному дослідженні розглядаються 2 випадки з різними видами нелінійності.

1. Випадок гладкої нелінійності поліноміального типу:

$$f_1(y) = \beta y^2, \quad (3.2)$$

де β – параметр (константа).

2. Випадок значної нелінійності у вигляді насичення:

$$f_2(y) = \begin{cases} s, y > p \\ k \cdot y, |y| \leq p \\ -s, y < -p \end{cases}, \quad (3.3)$$

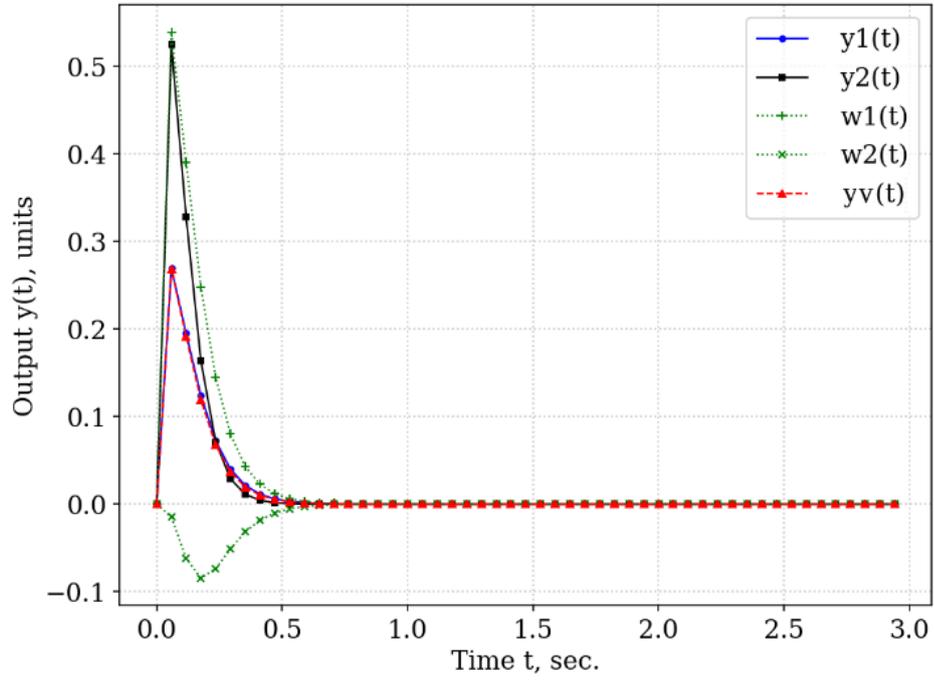
де s – рівень насичення, p – крапка початку насичення, $k=s/p$ – коефіцієнт підсилення.

При виконанні експерименту «вхід/вихід» з різними типами нелінійності прийнято наступні параметри імітаційної моделі: $\alpha=2.64$; для випадку гладкої нелінійності $\beta=1.45$; для випадку значної нелінійності $s=0.7, p=0.7, k=1$.

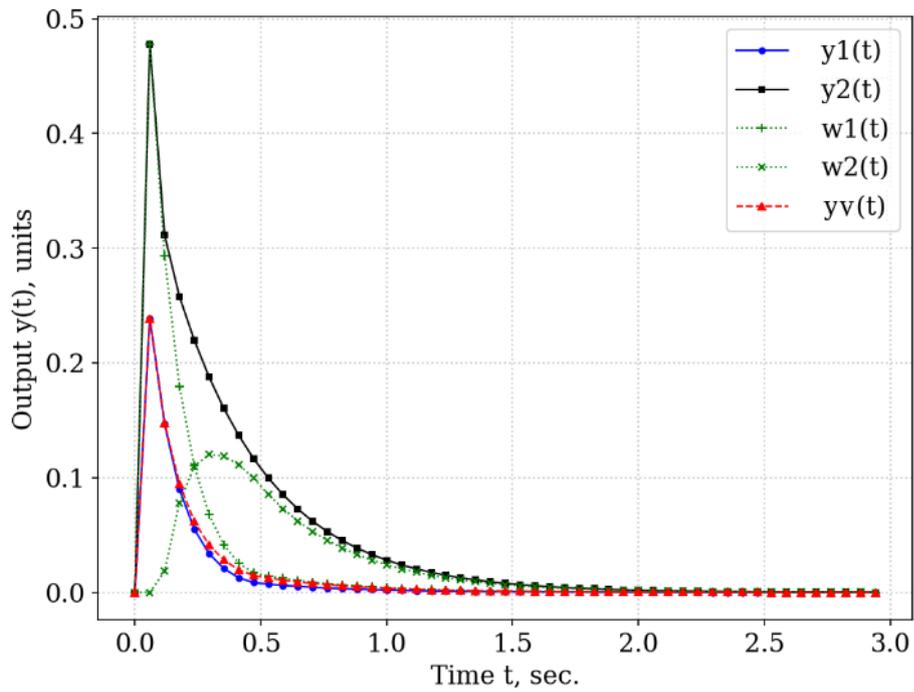
З метою дослідження точності TDNN моделей тестового об'єкта з різними видами нелінійності $f_i(y)$ ($i=1,2$) формуються навчальний $\mathbf{x}_i^{train}(t_n)$ та тестовий $\mathbf{x}_i^{test}(t_n)$ набори даних на основі експерименту «вхід/вихід». Для формування цих даних на вхід імітаційної моделі подаються тестові сигнали $x(t)$ у вигляді імпульсних, ступінчастих, лінійних і гармонічних функцій з різними амплітудами a .

Навчальний набір даних формується з результатів експерименту «вхід-вихід» – множини векторів $\{x(t), y(t)\}$ для кожного типу вхідного сигналу.

На основі отриманого датасету спочатку будується модель $y_v(t)$ у вигляді інтегро-степеневих поліномів Вольтерри. Модель будуються компенсаційним методом, розглянутим у параграфі 2.1. На рис. 3.3 зображено моделі $y_v(t)$, отримані для тестових динамічних об'єктів з гладкою $f_1(y)$ (рис. 3.3а) та значною $f_2(y)$ (рис. 3.3б) нелінійностями. На рисунках також наведені ядра Вольтерри 1-го порядку $w_1(t)$ і діагональні перетини ядер Вольтерри 2-го порядку $w_2(t,t)$, а також відгуки об'єкта $y_1(t)$ і $y_2(t)$ на вхідні сигнали $x_1(t) = a\delta(t)$ і $x_2(t) = 2a\delta(t)$ ($a = 0.5$), що використовуються для ідентифікації ядер Вольтерри.



а



б

Рисунок 3.3 – Моделі $y_v(t)$ у вигляді інтегро-степеневих поліномів Вольтерри, отримані для тестових динамічних об'єктів: а – з гладкою $f_1(y)$ нелінійністю, б – зі значною $f_2(y)$ нелінійністю

3.3.1. Дослідження масштабованості TDNN моделей до різних вхідних сигналів

Навчальний набір даних $\mathbf{x}_i^{train}(t_n)$ сформовано на основі імпульсних сигналів $x(t) = a\delta(t)$ різної амплітуди ($a \in (0, 1]$) на вході об'єкта та відгуків $y(t)$ на його виході.

Тестовий набір даних $\mathbf{x}_i^{test}(t_n)$ включає ступінчасті $x(t)=a\Theta(t)$, лінійний $x(t)=at$ і гармонічні $x(t)=a\sin(t)$ сигнали різної амплітуди ($a \in (0, 1]$) на вході об'єкта та відгуків $y(t)$ на його виході.

Експеримент виконується для об'єктів з нелінійностями у блоці зворотного зв'язку у вигляді поліноміальної функції $f_1(y)$, а також функції з насиченням $f_2(y)$.

Для побудови нейромережевої моделі використовується тришарова TDNN, структура якої визначена в параграфі 2.4.2: кількість нейронів у вхідному та прихованому шарах $M=15$ та $K=50$ відповідно, що забезпечують встановлений умовами експерименту рівень втрат ($loss < 1.0$) при прийнятному часі навчання ($epochs < 40$). Отриману TDNN використано для дослідження точності моделей динамічних об'єктів з гладкими та значними нелінійностями.

TDNN модель будується на даних навчального набору $\mathbf{x}_i^{train}(t_n)$. Дослідження масштабованості отриманої TDNN моделі до різних вхідних сигналів виконується на даних тестового набору $\mathbf{x}_i^{test}(t_n)$.

Вихід TDNN моделі $y_{nl}(t)$ порівнюється з виходом імітаційної моделі $y(t)$ та результатом ідентифікації $y_v(t)$ у вигляді інтегро-степеневого поліному Вольтерри [5, 23].

Експеримент 1. Досліджується точність моделювання з використанням TDNN та інтегро-степеневих моделей під час дії вхідних сигналів $x(t)=a\delta(t)$ різної амплітуди ($a \in (0, 1]$). На рис. 3.4 наведено порівняння вихідних сигналів $y_{nl}(t)$, $y_v(t)$ та $y(t)$, отриманих в результаті дії сигналу $x(t)=a\delta(t)$ ($a=0.65$) на входах моделі TDNN, інтегро-степеневого ряду та імітаційної моделі нелінійного динамічного об'єкта відповідно.

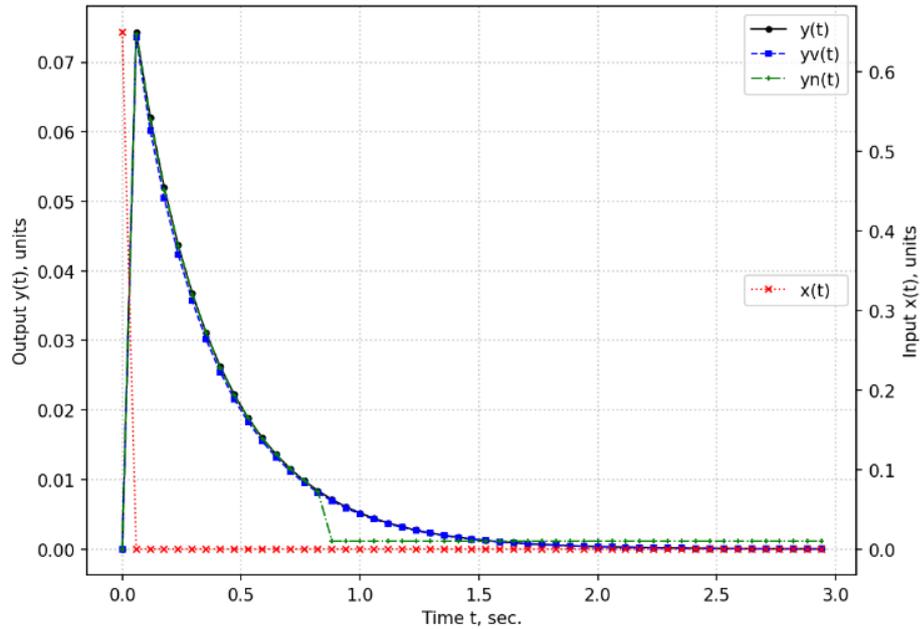


Рисунок 3.4 – Порівняння вихідних сигналів $y_{ml}(t)$, $y_v(t)$ та $y(t)$, отриманих в результаті дії сигналу $x(t)=a\delta(t)$ ($a=0.65$) на входах моделі TDNN, інтегро-степеневого ряду та імітаційної моделі нелінійного динамічного об'єкта відповідно

Даний експеримент демонструє порівнянню точність моделювання з використанням TDNN та інтегро-степеневих моделей під час дії вхідних сигналів $x(t)=a\delta(t)$.

Експеримент 2. Досліджується точність моделювання з використанням TDNN та інтегро-степеневих моделей під час дії вхідних сигналів $x(t)=a\Theta(t)$, $x(t)=at$ та $x(t)=asin(t)$ різної амплітуди ($a \in (0, 1]$). На рис. 3.5–3.7 наведено порівняння вихідних сигналів $y_{ml}(t)$, $y_v(t)$ та $y(t)$, отриманих в результаті дії сигналів $x(t)=a\Theta(t)$ (рис. 3.5), $x(t)=at$ (рис. 3.6) та $x(t)=asin(t)$ (рис. 3.7), $a=0.65$, на входах моделі TDNN, інтегро-степеневого поліному та імітаційної моделі нелінійного динамічного об'єкта відповідно.

Експеримент демонструє, що модель TDNN значно поступається за точністю інтегро-степеневій моделі Вольтерри під час дії вхідних сигналів $x(t)=a\Theta(t)$, $x(t)=at$ та $x(t)=asin(t)$, які не були включені до навчального набору даних $\mathbf{x}_l^{train}(t_n)$.

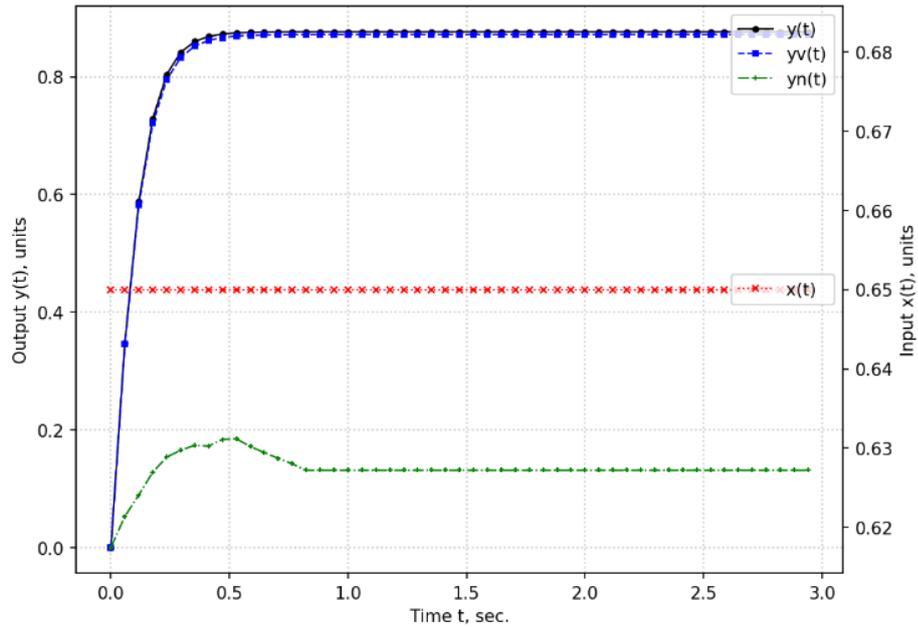


Рисунок 3.5 – Порівняння вихідних сигналів $y_{nn}(t)$, $y_v(t)$ та $y(t)$, отриманих в результаті дії сигналів $x(t)=a\Theta(t)$ на входах моделі TDNN, інтегро-степеневого поліному та імітаційної моделі нелінійного динамічного об'єкта, $a=0.65$

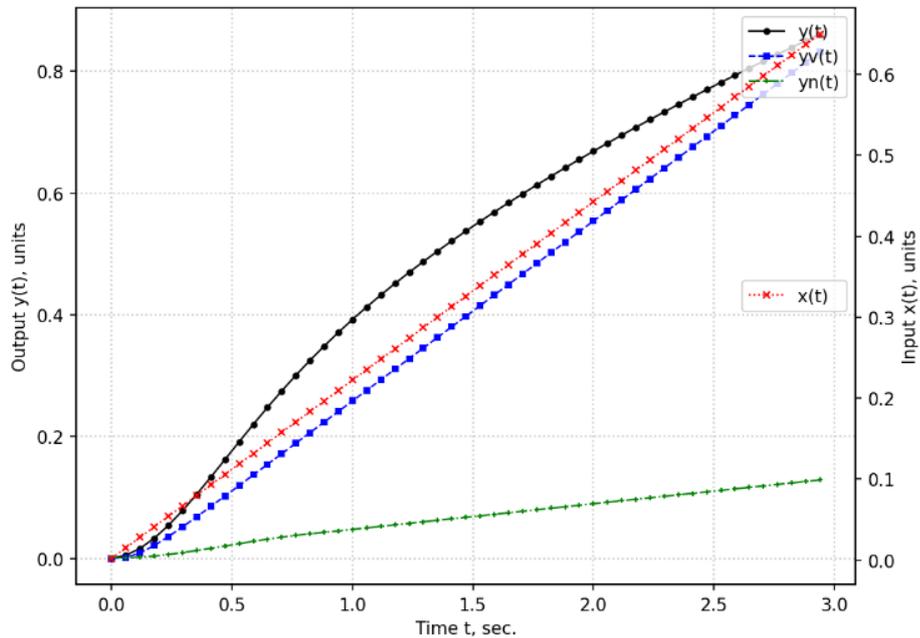


Рисунок 3.6 – Порівняння вихідних сигналів $y_{nn}(t)$, $y_v(t)$ та $y(t)$, отриманих в результаті дії сигналів $x(t)=at$ на входах моделі TDNN, інтегро-степеневого поліному та імітаційної моделі нелінійного динамічного об'єкта відповідно, $a=0.65$

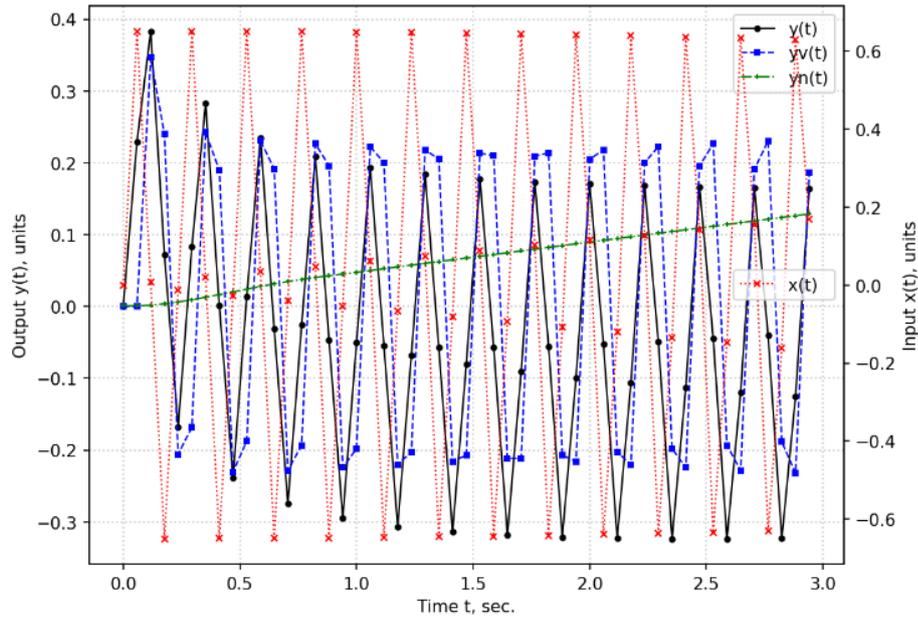


Рисунок 3.7 – Порівняння вихідних сигналів $y_{nn}(t)$, $y_v(t)$ та $y(t)$, отриманих в результаті дії сигналів $x(t)=asin(t)$ на входах моделі TDNN, інтегро-степеневого поліному та імітаційної моделі нелінійного динамічного об'єкта відповідно, $a=0.65$

З експерименту випливає, що моделі TDNN не є інваріантними до форми вхідного сигналу. Модель TDNN може адекватно відображати властивості динамічного об'єкта в разі навчання на достатньому обсязі даних. Навчальний набір даних $x_i^{train}(t_n)$ повинен формуватися на основі вхідних сигналів різної амплітуди того ж типу, що й у тестовому наборі даних $x_i^{test}(t_n)$. Це є недоліком нейромережевих моделей у порівнянні з моделями, що базуються на інтегро-степеневих рядах на основі багатовимірних вагових функцій [5, 23-25].

3.3.2. Дослідження інтерполяційних та екстраполяційних властивостей TDNN моделей

Навчальний набір даних $x_i^{train}(t_n)$ сформований на основі імпульсних $x(t) = a\delta(t)$, ступінчастих $x(t)=a\Theta(t)$, лінійних $x(t)=at$ та гармонічних $x(t)=asin(t)$ сигналів різної амплітуди ($a \in (0, 1]$) на вході об'єкта та відгуків $y(t)$ на його виході.

Тестовий набір даних $\mathbf{x}_i^{test}(t_n)$ включає ті ж сигнали $x(t)$ з різної амплітуди ($a \in (1, 2]$) на вході об'єкта та відгуків $y(t)$ на його виході.

Експеримент виконується для об'єктів з нелінійностями у блоці зворотного зв'язку у вигляді поліноміальної функції $f_1(y)$, а також функції з насиченням $f_2(y)$.

TDNN модель будується на даних навчального набору $\mathbf{x}_i^{train}(t_n)$. Дослідження екстраполяційних властивостей отриманої TDNN моделі виконується на даних тестового набору $\mathbf{x}_i^{test}(t_n)$. Вихід TDNN моделі $y_{ml}(t)$ порівнюється з виходом імітаційної моделі $y(t)$ та результатом ідентифікації $y_v(t)$ у вигляді інтегро-степеневого ряду на основі багатовимірних вагових функцій [5, 23].

Експеримент 3. Досліджується точність моделювання з використанням TDNN та інтегро-степеневих моделей під час дії імпульсних, ступінчастих, лінійних та гармонічних вхідних сигналів різної амплітуди ($a \in [0.1, 0.2, \dots, 1.0]$). На рис. 3.8 наведено порівняння вихідних сигналів $y_{ml}(t)$, $y_v(t)$ та $y(t)$, отриманих в результаті дії сигналу $x(t)=a\Theta(t)$ ($a=0.7$) на входах моделі TDNN, інтегро-степеневого ряду та імітаційної моделі нелінійного динамічного об'єкта відповідно.

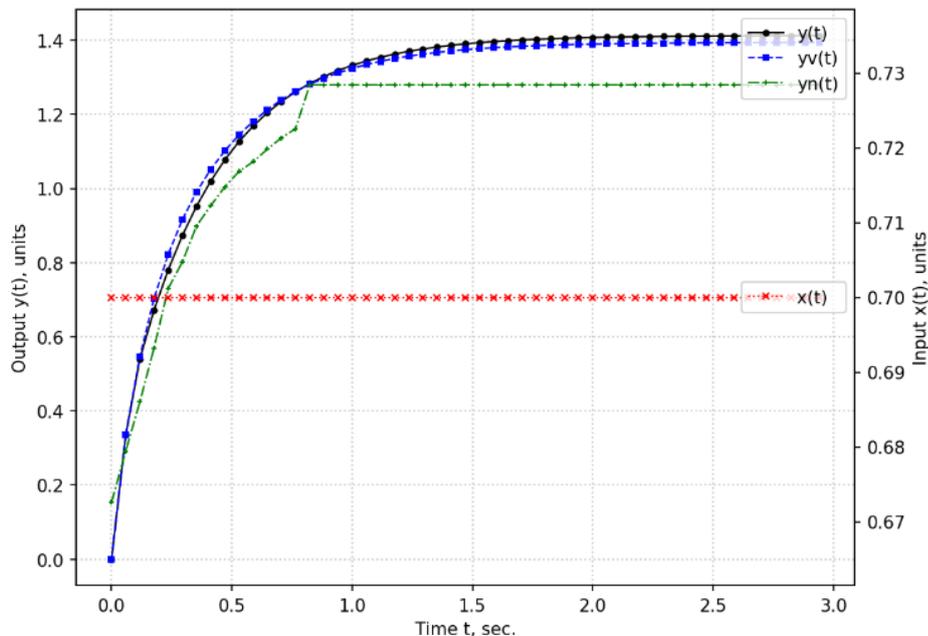


Рисунок 3.8 – Порівняння вихідних сигналів $y_{ml}(t)$, $y_v(t)$ та $y(t)$, отриманих в результаті дії сигналу $x(t)=a\Theta(t)$ ($a=0.7$) на входах моделі TDNN, інтегро-степеневого ряду та імітаційної моделі нелінійного динамічного об'єкта відповідно

За результатами експерименту отримано значення середньої квадратичної похибки для моделей TDNN у порівнянні з моделями у вигляді інтегро-степеневих поліномів Вольтерри для нелінійних функції $f_1(y)$ та $f_2(y)$, що наведено в табл. 3.2.

Таблиця 3.2 – Середня квадратична похибка моделювання тестового об'єкту
($a=0.7$)

Тип нелінійності	Вхідний сигнал $x(t)$							
	імпульсний $a\delta(t)$		ступінчастий $a\Theta(t)$		лінійний at		гармонічний $asin(t)$	
	$y_v(t)$	$y_{ml}(t)$	$y_v(t)$	$y_{ml}(t)$	$y_v(t)$	$y_{ml}(t)$	$y_v(t)$	$y_{ml}(t)$
Поліноміальна	0,074	0,080	0.082	0.087	0.088	0.096	0.103	0.127
З насиченням	0,152	0.121	0.167	0.133	0.209	0.139	0.264	0.184

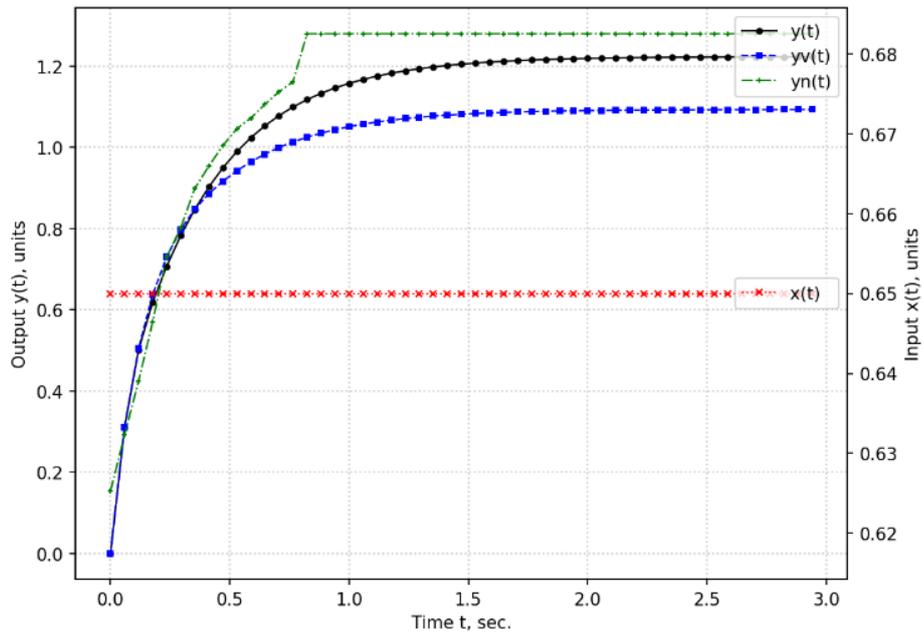
Експеримент 3 демонструє порівнянну точність моделювання з використанням TDNN та інтегро-степеневих моделей тестового об'єкту з гладкою нелінійністю $f_1(y)$. Але при ідентифікації тестового об'єкту з суттєвою нелінійністю $f_2(y)$ TDNN моделі істотно перевищують інтегро-степеневі моделі за точністю.

Експеримент 4. Досліджуються інтерполяційні та екстраполяційні властивості моделей з використанням TDNN та інтегро-степеневих моделей під час дії імпульсних, ступінчастих, лінійних та гармонічних вхідних сигналів різної амплітуди:

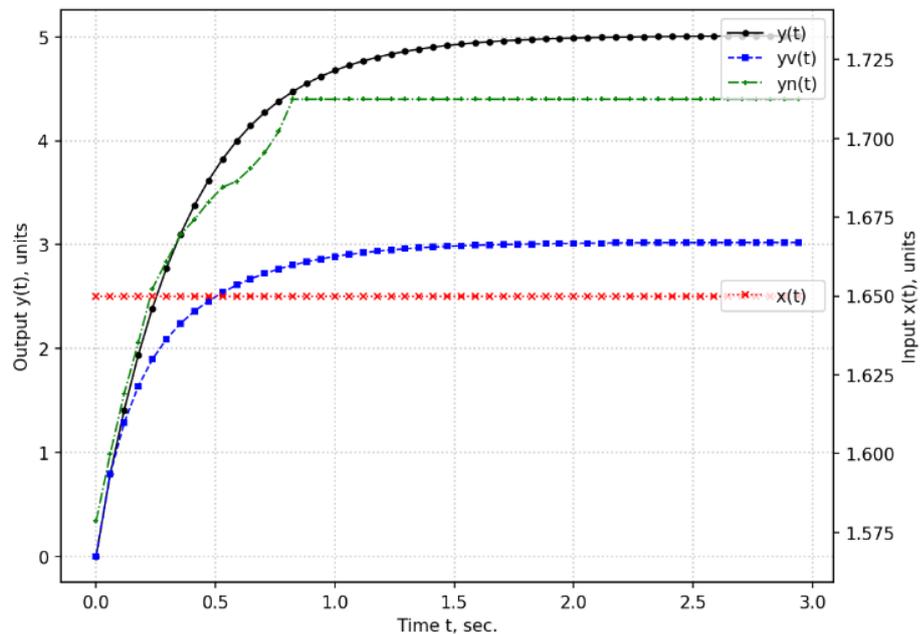
- для випадку дослідження інтерполяційних властивостей моделей $a \in [0.05, 0.15, \dots, 0.95]$;
- для випадку дослідження екстраполяційних властивостей моделей $a \in [1.05, 1.15, \dots, 1.95]$.

На рис. 3.9 наведено порівняння вихідних сигналів $y_{ml}(t)$, $y_v(t)$ та $y(t)$, отриманих в результаті дії сигналу $x(t)=a\Theta(t)$, на входах моделі TDNN, інтегро-степеневого ряду та імітаційної моделі нелінійного динамічного об'єкта

відповідно, для задачі інтерполяції при $a=0.65$ (рис. 3.9а) та задачі екстраполяції при $a=1.65$ (рис. 3.9б).



а



б

Рисунок 3.9 – Порівняння вихідних сигналів $y_{nn}(t)$, $y_v(t)$ та $y(t)$, отриманих в результаті дії сигналу $x(t)=a\Theta(t)$ на входах моделі TDNN, інтегро-степеневого ряду та імітаційної моделі нелінійного динамічного об'єкта відповідно: а – задача інтерполяції ($a=0.65$), б – задача екстраполяції ($a=1.65$)

Експеримент демонструє порівнянну точність моделювання у випадку дослідження інтерполяційних властивостей моделей TDNN та інтегро-степеневих моделей.

Екстраполяційні властивості моделей TDNN погіршувалися зі збільшенням амплітуд вхідних сигналів за межі діапазону амплітуд вхідних сигналів навчальної вибірки $x_i^{train}(t_n)$. Так, використовуючи вхідний сигнал $x(t)=1.65\Theta(t)$, точність TDNN моделі знижується на 25%.

При цьому, точність моделі у вигляді інтегро-степеневого ряду на основі багатовимірних вагових функцій у випадку використання значно нелінійної функції $f_2(y)$ поступається TDNN моделі на 30%.

Таким чином, областю ефективного застосування TDNN моделей є ідентифікація об'єктів зі значно нелінійними характеристиками.

3.4. Висновки за розділом

В даному розділі запропоновано метод інтерпретації моделей машинного навчання у вигляді нейронних мереж з часовими затримками, збудованими за даними експерименту «вхід–вихід». В якості сурогатної моделі, яка має спрощений вигляд та придатна для подальшої інтерпретації використовуються непараметричні динамічні моделі у вигляді інтегро-степеневих поліномів Вольтерри. Побудова сурогатної моделі здійснюється за допомогою встановленого аналітичного зв'язку між тришаровою повнозв'язною нейронною мережею та ядрами Вольтерри.

Перевагою розробленого методу у порівнянні з методами побудови лінійних сурогатних моделей є підвищення точності та зменшити розмірності сурогатних моделей при ідентифікації нелінійних динамічних об'єктів за рахунок найбільш повного урахування діагностичної інформації первинних моделей.

Для реалізації методу інтерпретації моделей машинного навчання нелінійних динамічних об'єктів, наведеного вище, розроблено обчислювальний алгоритм та наведено його реалізацію у вигляді псевдокоду. Створений алгоритм

дає можливість будувати нелінійні сурогатні моделі, що мають переваги в точності перед лінійними сурогатними моделями.

При реалізації розробленого методу необхідно вирішити низку допоміжних операцій з підготовки та проміжної обробки даних на кожному з етапів. Для вирішення задачі інтерпретації моделей машинного навчання на базі наведеного методу, з урахуванням вирішення допоміжних задач розроблено базову ІТ побудови сурогатних моделей TDNN-мереж.

Для визначення області ефективного використання розробленого методу інтерпретації моделей машинного навчання організовано та проведено імітаційні експерименти з використанням даних тестових динамічних об'єктів з гладкими нелінійностями та насиченням.

Експерименти полягають у дослідженні масштабованості TDNN моделей до різних вхідних сигналів та дослідженні екстраполяційних властивостей TDNN моделей.

Отримані результати моделювання свідчать що TDNN моделі не є інваріантними по відношенню до сигналу, який надходить на вхід об'єкта, що є недоліком цих моделей. Але TDNN моделі можуть адекватно відображати властивості нелінійних динамічних об'єктів в разі навчання на достатньому обсязі даних, що формується на основі вхідних сигналів того ж типу, що й у тестовому наборі даних.

Екстраполяційні властивості моделей TDNN погіршуються зі збільшенням амплітуд вхідних сигналів, що виходять за межі діапазону амплітуд вхідних сигналів навчальної вибірки.

При ідентифікації об'єктів з істотно нелінійними властивостями, наприклад, з функцією насиченості, моделі TDNN мають точність на 10-25% вищу, ніж інтегро-ступеневі ряди Вольтерри.

Таким чином, сферою ефективного застосування TDNN є ідентифікація об'єктів з суттєво нелінійними властивостями.

Матеріали розділу відображені в публікаціях [57, 58, 96, 140, 141, 159].

РОЗДІЛ 4

ЗАСОБИ КОМП'ЮТЕРНОГО МОДЕЛЮВАННЯ ТА ПРАКТИКА ЇХ
ЗАСТОСУВАННЯ ПРИ РОЗВ'ЯЗУВАННІ ЗАДАЧ ІНТЕРПРЕТАЦІЇ МОДЕЛЕЙ
МАШИННОГО НАВЧАННЯ

Застосування розробленого методу моделювання стикається з обмеженням, викликаним відсутністю ефективних інструментальних засобів для реалізації у вигляді програмного забезпечення інформаційних систем. При цьому, принципове значення у сфері ідентифікації та моделювання нелінійних динамічних об'єктів із безперервними характеристиками має висока обчислювальна складність [14, 22, 28, 40, 167]. Це завдання порушує важливе актуальне питання – розробку спеціалізованих інструментів високопродуктивного комп'ютерного моделювання, орієнтованих на створення моделей для нелінійних динамічних об'єктів. Ці інструменти мають бути здатними вирішувати складні завдання моделювання, що потребують значних ресурсів.

У цьому розділі розглядаються питання проектування та розробки ефективних інструментальних засобів автоматизованих систем моделювання нелінійних динамічних об'єктів з використанням передових інформаційних технологій, які мають успішно вирішувати ресурсомісткі завдання моделювання.

4.1. Розробка інструментальних програмних засобів інтерпретації моделей машинного навчання

Програмне забезпечення комп'ютерних систем зручно розділяти на три основні класи [12, 62, 80, 128]:

– *системне програмне забезпечення* містить операційні системи та інструменти, які призначені для керування апаратними засобами комп'ютерної системи з метою надання ресурсів для виконання додатків.

– *інструментальне програмне забезпечення* включає в себе додатки, що застосовуються в процесі розроблення та підтримки програмного забезпечення.

– *прикладне програмне забезпечення* включає додатки, призначені для виконання кінцевими користувачам конкретних прикладних завдань.

Свою чергою, *прикладне програмне забезпечення* прийнято поділяти на *професійне програмне забезпечення* та *програмне забезпечення спеціального призначення* [12, 139]. Різниця між цими видами програмного забезпечення полягає в їхньому призначенні, функціональності та сфері застосування.

Професійне програмне забезпечення призначене для використання в загальних професійних або бізнес-сценаріях.

Функціональність: широкий спектр можливостей, які можуть бути адаптовані для розв'язання прикладних завдань у різних галузях або професійних сферах.

Приклади: офісне, графічне програмне забезпечення, CRM та ERP системи.

Програмне забезпечення спеціального призначення призначене для розв'язання конкретних завдань або виконує певні функції в обмеженій галузі.

Функціональність: набір інструментів, зосереджений на розв'язанні певних, як правило, вузькоспеціалізованих прикладних завдань.

Приклади: медичні, фінансові інформаційні системи, програмне забезпечення для автоматизації виробничих процесів, системи управління телекомунікаціями.

Для формування вимог до якості програмних засобів [53, 63, 138, 143], що розробляється, вибору метрик їх оцінювання [54, 76, 85] та організації ефективного процесу проектування і розробки інструментальних засобів [55, 82, 99] необхідно визначити клас, до якого відносяться програмні засоби інтерпретації моделей машинного навчання нелінійних динамічних об'єктів. Саме ці питання розглядаються в даному параграфі.

4.1.1. Проектування інструментальних програмних засобів інтелектуальної системи. Для визначення місця інструментальних засобів інтерпретації моделей машинного навчання в складі програмного застосування

інтелектуальної системи, розглянемо основні складові модулі типової системи (рис. 4.1).

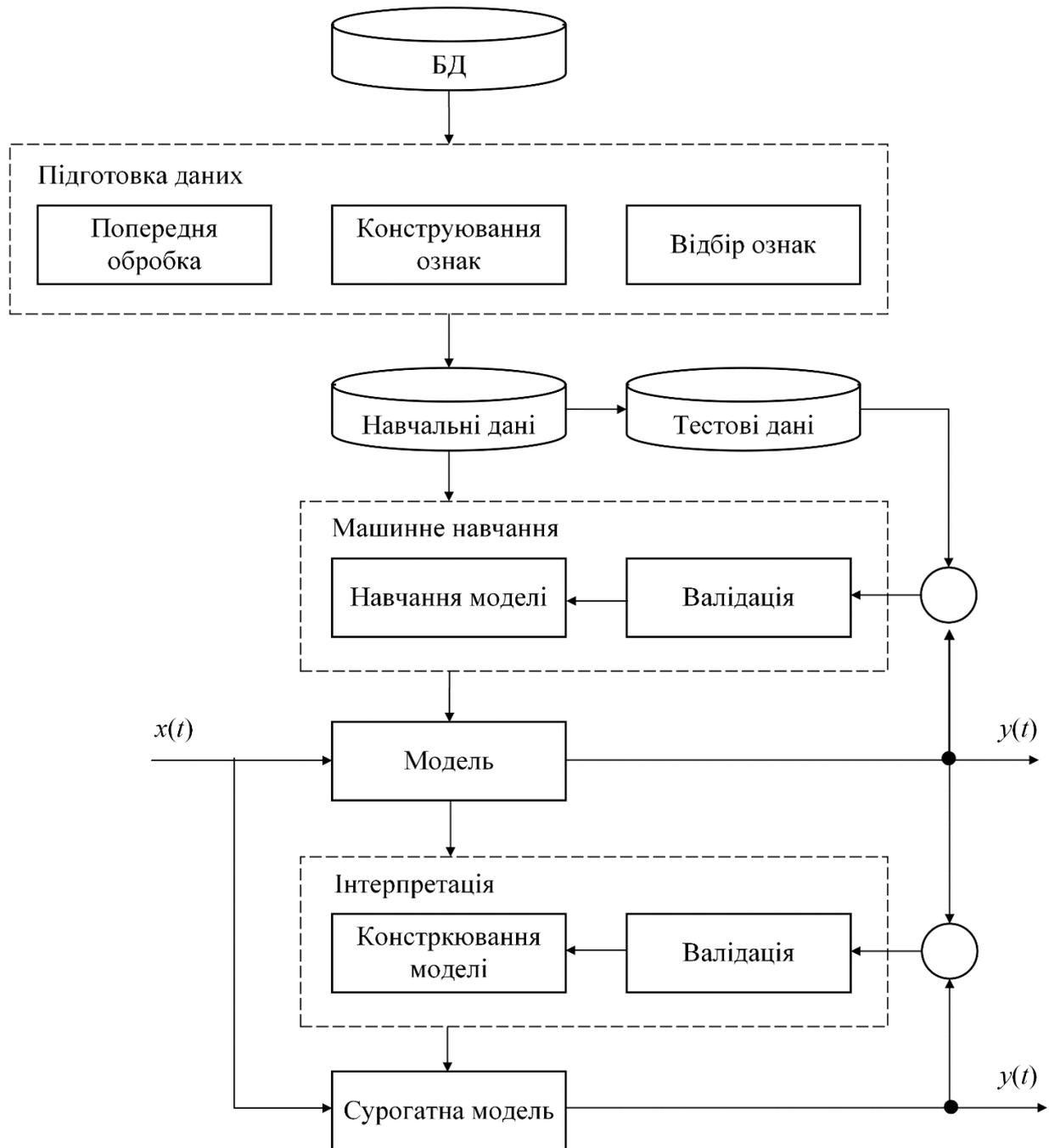


Рисунок 4.1 – Структура інтелектуальної системи з модулями ідентифікації та інтерпретації

Типова інтелектуальна система складається з наступних модулів [34, 39-41, 174]:

- *база знань* (база даних) зберігає дані, інформацію і знання, необхідні системі для розв'язання прикладних завдань. В якості знань можуть виступати факти, відносини між ними і правила, які інтелектуальна система використовує у своїй роботі. Одним із видів знань, отриманих на основі даних, що зберігаються, є моделі;

- *механізм виведення рішень* (блок машинного навчання) займається пошуком розв'язання прикладних задач на основі інформації, що зберігається в базі знань. Механізм виведення використовує наявні в розпорядженні моделі, методи й алгоритми для прийняття рішень;

- *інтелектуальний інтерфейс* слугує для інтерактивної взаємодії користувачів із програмним забезпеченням: засоби організації користувачем уведення інформації в систему та виведення результатів розв'язання задачі.

Окремі складові структури інтелектуальної системи можуть бути, за необхідності, деталізовані за рахунок включення допоміжних модулів. Так, база знань може бути доповнена модулем підготовки даних з такими функціями як попередня обробка даних, конструювання та вибір ознак [26, 90, 169].

Блок механізму виведення рішень може бути доповнений модулями ідентифікації (у випадку вирішення задачі модельної діагностики [37, 43, 176, 177]) та інтерпретації моделей. Модулі ідентифікації та інтерпретації моделей машинного навчання інтелектуальних систем відіграють центральну роль у процесі розробки та функціонування інтелектуальних систем. Вони надають засоби для автоматичного створення математичної моделі задачі, що дає змогу користувачам розуміти роботу системи.

Модуль ідентифікації є ключовим компонентом у процесі розробки, підтримки та оптимізації інтелектуальних систем [125, 135, 157]. Цей модуль відповідає за продукування знань у вигляді моделей даних і процесів на основі інформації, отриманої з бази даних. Моделі можуть бути використані для передбачення поведінки системи або для визначення оптимальних рішень.

Модуль інтерпретації відповідає за пояснення результатів, отриманих від механізму виведення рішень [27, 97]. Він може перетворювати ці результати у

формат, який легко зрозуміти користувачу. У структурі інтелектуальної системи, що наведена на рис. 4.1, цей модуль виконує ключові функції, пов'язані з комп'ютерним моделюванням. Він приймає як вхідні дані моделі машинного навчання, піддає їх деяким перетворенням та видає сурогатну модель. Таким чином, модуль інтерпретації моделей машинного навчання в структурі інтелектуальної системи належить до блоку машинного навчання та реалізується у вигляді підсистеми інструментальних засобів виведення рішень.

Визначення якості сурогатних моделей, збудованих у модулі інтерпретації, здійснюється у блоці валідації моделей. Завдання визначення якості інструментальних засобів побудови моделей є досить складним та традиційно вирішується за допомогою моделей якості програмного забезпечення, що будуються на основі вимог до інструментальних програмних засобів [24, 49, 60, 103, 105, 137].

У наступному параграфі розглядаються вимоги до програмного забезпечення на основі існуючих стандартів [119, 138, 143] з метою побудови ефективної моделей якості програмного забезпечення інтерпретації моделей машинного навчання.

4.1.2. Вимоги до інструментальних програмних засобів комп'ютерного моделювання. Інструментальні програмні засоби моделювання мають допоміжний характер під час проектування, розроблення та підтримки інтелектуальних систем. Отже, системи моделювання, до яких належить напрям інтерпретації моделей машинного навчання, можна віднести до типу допоміжного інструментального програмного забезпечення.

Нижче наведено класифікацію інструментальних програмних засобів за принципом величини спектра розв'язуваних задач: від більш універсальних до спеціалізованих систем [71, 80, 127, 139].

- Мови програмування, широко використовувані в наукових та інженерних дослідженнях, включно зі спеціалізованими бібліотеками для чисельного моделювання, аналізу даних, машинного навчання тощо.

- Середовища для чисельного моделювання, що широко застосовується для інженерного аналізу та моделювання, включаючи інструментальні засоби для багатьох прикладних галузей.

- Спеціалізовані середовища комп'ютерного моделювання застосовуються для вузькоспеціалізованих предметних галузей, зокрема візуалізації даних.

Вибір засобів розроблення спеціалізованого програмного забезпечення залежить від конкретних завдань, зокрема вимог до точності та продуктивності [14, 78, 119, 143, 145]. Як було показано в Розділі 1, дослідження в напрямі інтерпретації моделей машинного навчання нелінійних динамічних об'єктів є таким, що розвивається та ще не сформоване в єдину наукову галузь. Тому, для розроблення інструментальних засобів моделювання раціонально використовувати технології з найбільш широкими можливостями, не обмежуючись спеціалізованими програмними середовищами. Для таких інструментальних програмних засобів і будуть розглядатися та формулюватися вимоги.

Вимоги до якості програмного забезпечення ґрунтуються на наявних стандартах [62, 66]. Важливо зазначити, що вимоги можуть варіюватися залежно від призначення програмного забезпечення і предметної області. Вимоги забезпечення ефективності, безпеки та сумісності інтелектуальних систем регламентуються ДСТУ ISO/IEC 25010:2016 Інженерія систем і програмних засобів. Вимоги до якості систем і програмних засобів та її оцінювання (SQuaRE) [62]. Цей документ висуває вимоги до якості систем і програмного забезпечення та є частиною серії стандартів SQuaRE і містить у собі такі розділи: "Менеджмент якості" (ISO/IEC 2500n), "Модель якості" (ISO/IEC 2501n), "Вимірювання якості" (ISO/IEC 2502n) і встановлює загальну модель якості програмного забезпечення і її оцінки. Він описує характеристики якості та відповідні підхарактеристики. Ці вимоги можуть бути адаптовані до програмного забезпечення інтелектуальних систем залежно від специфіки завдання та вимог до проекту.

Вимоги, викладені в ISO/IEC 25010:2016, належать насамперед до категорії вимог до продукту, визначаючи характеристики якості, які слід враховувати під

час проектування, розроблення та тестування програмного продукту або системи. Хоча вимоги бізнесу та користувачів встановлюють цілі та очікування вищого рівня, у низці випадків їх також можна представити як пункти цього стандарту. Для систематизації вимог до програмного забезпечення інтелектуальних систем на рис. 4.2. наведено класифікацію наявних вимог.

Під час розроблення програмного забезпечення важливо мати механізми моніторингу для відстеження характеристик продукту та оцінки його роботи – метрики. Бізнес-вимоги до інтелектуальних систем можуть містити різні аспекти, орієнтовані на досягнення конкретних бізнес-цілей і завдань мети [14, 53, 143]. Програмні системи допомагають досягти ці бізнес-цілі за підтримки вимірних метрик для оцінки успіху. Розглянемо узагальнені категорії бізнес-вимог для інтелектуальних систем, представлені на рисунку 4.2. і відповідні їм метрики (табл. 4.1).

Таблиця 4.1 – Узагальнені категорії бізнес-вимог для інтелектуальних систем

№	Назва категорії	Метрика
1	Покращення бізнес-процесів	– значення цільового параметра (час, ресурс); – комплексного показника (ефективність)
2	Підвищення задоволеності клієнтів та конкурентоспроможності	– задоволеність клієнтів; – частка ринку
3	Управління ризиками	– частота виникнення ризикових ситуацій

Призначені для користувача вимоги до інтелектуальних систем фокусуються на тому, як система має відповідати потребам і очікуванням кінцевих користувачів. Нижче розглянуто узагальнені категорії користувацьких вимог [14, 61, 63] із відповідними метриками, що встановлюють наскільки система відповідає очікуванням користувачів, наскільки вона зручна у використанні, ефективна та безпечна (табл. 4.2).



Рисунок 4.2 – Вимоги до програмного забезпечення інтелектуальних систем на основі стандарту ISO/IEC 25010

Таблиця 4.2 – Узагальнені категорії вимог користувача для інтелектуальних систем

№	Назва категорії	Метрика
1	Задоволення потреб і очікувань користувачів	рівень задоволеності, оцінка зручності інтерфейсу
2	Адаптивність і персоналізація	рівень персоналізації системи
3	Ефективність і безпека використання	значення комплексного параметра (ефективність), рівень забезпечення безпеки та конфіденційності даних

Нарешті, вимоги до продукту можуть містити різноманітні аспекти, засновані на стандарті ISO/IEC 25010. Стандартом визначено два основні типи характеристик якості: функціональні та нефункціональні.

Функціональні вимоги до продукту описують, що система повинна робити. Це можуть бути певні завдання або функції, які система повинна виконувати (табл. 4.3). Нефункціональні вимоги до продукту описують, як система має працювати. Це можуть бути вимоги до продуктивності, надійності, безпеки та інші аспекти системи.

Під час розроблення інтелектуальних систем важливо враховувати також обмеження предметної області, тому що вони можуть істотно впливати на реалізацію проекту та його застосування в реальних умовах.

З урахуванням проведеного аналізу та класифікації існуючих вимог до програмних продуктів запропоновано модель якості та метрики її оцінювання для підсистеми моделювання та інтерпретації моделей машинного навчання інтелектуальних систем, що базуються на стандарті ISO/IEC 25010.

Таблиця 4.3 – Узагальнені категорії вимог користувача для інтелектуальних систем

№	Назва категорії	Метрика	Характеристики
1	2	3	4
1	Функціональність	кількість реалізованих функцій із технічного завдання	<ul style="list-style-type: none"> – <i>Здатність відповідності</i>: ПЗ повинно відповідати специфікаціям і очікуванням користувача, надаючи необхідні функції для досягнення цілей системи. – <i>Точність</i>: важливо, щоб результати роботи ІС були точними і відповідали очікуванням.
2	Продуктивність	час відгуку системи	<ul style="list-style-type: none"> – <i>Швидкодія</i>: ПЗ має забезпечувати достатню швидкість виконання та відгуку, щоб задовольнити потреби користувачів. – <i>Ефективність</i>. ПЗ має бути оптимізоване для ефективного використання обчислювальних ресурсів, особливо при інтенсивних обчисленнях, як у випадку з ІС.
3	Сумісність	кількість успішних інтеграцій з іншими системами	<ul style="list-style-type: none"> – <i>Сумісність</i> із різними операційними системами, браузерами, БД тощо. ПЗ має враховувати можливі зміни у майбутньому. – <i>Інтеграція</i> з іншими системами та пристроями. Це забезпечує безпроблемну взаємодію з наявною інфраструктурою і технологіями.
4	Надійність	час безвідмовної роботи	<ul style="list-style-type: none"> – <i>Надійність</i> стосується здатності ПЗ послідовно і надійно виконувати свої функції в різних умовах. Вона включає в себе такі аспекти, як час безвідмовної роботи системи, обробка помилок і можливість коректного відновлення після збоїв

1	2	3	4
5	Зручність	результати користувацького тестування	<ul style="list-style-type: none"> – ПЗ має бути легким у використанні та розумінні для кінцевих користувачів. – ПЗ має бути зрозумілим для користувачів, а користувацький інтерфейс інтуїтивно зрозумілим
6	Ефективність	швидкість досягнення цілей, використання ресурсів	<ul style="list-style-type: none"> – Сюди входять такі аспекти, як час відгуку, час виконання завдання і загальна ефективність. – Програмне забезпечення має використовувати ресурси (наприклад, пам'ять, процесорний час) ефективно.
7	Підтримка змін	час, необхідний на внесення змін	<ul style="list-style-type: none"> – ПЗ має бути легким для обслуговування та модифікації. – Програмний код має бути зрозумілим для розробників, щоб забезпечити легкість супроводу та внесення змін. – Система має бути розбита на модулі, щоб полегшити внесення змін і підтримку. – ПЗ має бути відкритим для доповнень і розширень. Це дає змогу впроваджувати нові функції, модулі та алгоритми, забезпечуючи тим самим стійкість до мінливих вимог.
5	Переносимість	кількість підтримуваних платформ і операційних систем	<ul style="list-style-type: none"> – Переносимість означає здатність ПЗ переносити з одного середовища в інше. Це може включати сумісність із різними операційними системами, БД або конфігураціями обладнання. – ПЗ має успішно працювати на різних платформах без значних змін. Треба передбачити можливість адаптуватися до різних оточень та умов експлуатації.

4.2. Модель якості програмного забезпечення інтерпретації моделей машинного навчання

Визначення та вимірювання якості програмного забезпечення інтелектуальних систем залишається складним завданням, яке прийнято вирішувати за допомогою моделей якості [24, 49]. Існуючі моделі якості програмного забезпечення інтелектуальних систем є занадто абстрактними, щоб бути реалізовані на практиці [24] і потребують додаткових інструкцій для їх застосування. Тому, далі в роботі пропонується модель якості ПЗ інтелектуальних систем, яка дозволяє перетворити абстрактну та важко вимірювану концепцію якості на практичний і дієвий інструмент управління якістю моделей машинного навчання та яку зручно реалізувати у складі програмних засобів інтелектуальної системи.

Процес побудови моделі якості для інтелектуальних систем зручно розділити на три кроки [116, 119, 137].

1. *Визначення мета-моделі якості.* На даному етапі визначаються компоненти базової структури моделі, які впливають на якість ПЗ інтелектуальної системи. Компоненти базової структури моделі обираються з повного переліку, визначеного існуючими стандартами, на основі контексту застосування моделі якості. Це дає основу для наповнення моделі якості найбільш релевантними компонентами, замість того, щоб намагатися скласти модель з повного переліку усіх компонент якості, що визначено діючими стандартами.

2. *Визначення метрик для кількісної оцінки кожного компонента мета-моделі.* Вимірювання ступеня якості окремих компонентів – це складне завдання, оскільки поняття якості може мати різні аспекти залежно від мети і встановлених вимог.

3. *Побудова багатофакторної моделі якості на основі мета-моделі, визначеної на першому кроці та метрик, прийнятих на другому кроці.*

4.2.1. Визначення компонентів мета-моделі якості програмного забезпечення та метрик для їх кількісної оцінки. Модель якості ПЗ інтелектуальних систем та метрики оцінювання її компонентів базуються на аналізі існуючих вимог в області системної та програмної інженерії, регламентованих стандартами SQuaRE.

Так, для побудови моделі якості ПЗ інтелектуальних систем, по-перше, визначаються значущі компоненти якості: характеристики, що визначають функціональність, надійність і безпеку, під час розв'язання конкретної прикладної задачі.

З урахуванням обраних компонент модель якості ПЗ інтелектуальних систем у загальному вигляді приймає наступний вигляд:

$$Q = \langle Q_1, Q_2, \dots, Q_s \rangle, \quad (4.1)$$

де Q_s – компонента якості, $s=1,2,\dots,r$, r – кількість обраних компоненти якості.

Модель якості (4.1) дає змогу адекватно порівнювати моделі машинного навчання між собою, тому, має розв'язувати існуючі суперечливі вимоги до інтелектуальних систем, найбільш поширеними з яких є наступні.

– *Функціональність-складність інтерпретації моделі*: з одного боку, модель із меншою кількістю параметрів є кращою з погляду ресурсів та інтерпретації; з іншого боку, така модель поступатиметься в точності.

– *Функціональність-складність розгортання*: налаштування GPU-сервера для розгортання більш функціональних моделей (наприклад, нейромережевих моделей) може значно перевершувати за складністю налаштування звичайного сервісу для навчання простіших моделей (наприклад, логістичної регресії) засобами CPU.

– *Функціональність-ефективність у використанні ресурсів*: вибір між моделями з істотною відмінністю часу навчання: наприклад, час навчання моделі градієнтного бустингу і більш функціональної нейронної мережі може відрізнитися на порядок.

– *Функціональність-продуктивність*: зазвичай більш функціональні моделі працюють значно повільніше.

– *Функціональність-надійність*: складні моделі вразливі до adversarial-атак, коли додають невеликий шум до вихідних даних, що може призвести до зміни моделлю свого рішення; прості моделі більш стійкі до таких атак.

Вирішення одного з перелічених протиріч, а саме, “*Функціональність-складність інтерпретації моделі*”, що відповідає встановленій в роботі меті, здійснюється на прикладі задачі моделювання тестового нелінійного динамічного об’єкту за допомогою НМ з часовими затримками. З огляду на поставлену задачу, виконано аналіз стандартів ISO/IEC 25059:2023, ISO/IEC TS 25058:2024 та виокремлено наступні значущі характеристики і визначено метрики для їх вимірювання.

1. *Функціональність*: система повинна забезпечувати відтворення вихідних сигналів на тестовому датасеті.

Метрика Q_F : точність відтворення вихідного сигналу, наприклад, msa та mae .

2. *Складність*: чим простіша структура моделі, менша кількість змінних, рівнянь та елементів, використовуваних для опису поведінки системи, тим більше ступінь її розуміння (інтерпретованості). Загалом, складність моделі часто є компромісом між точністю, обсягом інформації та ресурсами, необхідними для її створення, аналізу та використання.

Метрика Q_C : кількість параметрів моделі.

Найпоширеніші програмно-орієнтовані критерії оцінки складності моделей [2, 36, 105]:

- кількість дійсних множень моделі;
- кількість операцій множення з урахуванням розрядності (бітова точність вхідних даних, ваг та функції активації);
- кількість операцій з фіксованою комою (врахування операцій зсуву та суматори).

У роботі для підвищення властивості інтерпретованості моделей прийнято використовувати спрощений критерій на основі визначення кількості параметрів

моделі. Враховуючи, що структура моделі машинного навчання визначається з точністю до параметрів, кількість дійсних множень при реалізації моделі можна замінити на кількість параметрів моделі, що істотно спрощує визначення складності моделі в цілому. Це припущення особливо корисно при визначенні складності моделей у вигляді НМ.

Так, критерій оцінки складності моделі машинного навчання у вигляді повнозв'язної НМ враховує вагові коефіцієнти та коефіцієнти зсуву моделі та приймає вигляд:

$$Q_C = n_k + \sum_{i=1}^{k-1} n_i (n_{i+1} + 1), \quad (4.2)$$

де k – кількість шарів НМ; n_i – кількість нейронів у i -му шарі.

Для окремого випадку тришарової повнозв'язної нейронної мережі з часовими затримками, яка має один вихід та використовуються для ідентифікації нелінійних динамічних об'єктів, вираз (4.2) отримує спрощеного вигляду:

$$Q_C = (M + 1)(K + 1) + K, \quad (4.3)$$

де M – кількість нейронів у вхідному шарі НМ; K – кількість нейронів у прихованому шарі.

З урахуванням обраних компонент модель якості ПЗ інтелектуальних систем приймає наступний вигляд:

$$Q = \langle Q_F, Q_C \rangle, \quad (4.4)$$

де Q_F – точність відтворення вихідного сигналу, Q_C – кількість параметрів у моделі.

Модель якості (4.4) має задовольняти двом суперечливим вимогам: функціональність-складність – здатна якнайкраще відтворювати поведінку об'єкту і при цьому бути зручною для інтерпретації користувачем (мати якомога простішу структуру моделі). Підвищення відповідності моделі даним, як правило, пов'язане з її ускладненням, а що складніша модель, то нижча її інтерпретованість. Тому, під час вибору між простою і складною моделлю остання має значимо збільшувати відповідність моделі даним, щоб виправдати зростання складності та

відповідне зниження інтерпретованості. Якщо ця умова не виконується, то слід обрати простішу модель.

Таким чином, щоб оцінити, наскільки зміна певного показника якості впливає на якість всієї системи доцільно розробити критерії порівняння якості моделей. У цьому разі загальновідомі та популярні метрики якості моделей, такі як *mse* та *mae*, не можуть бути застосовані, бо не враховують складність моделі.

В наступному розділі пропонується багатофакторна модель якості, яка враховує одразу декілька складових показників якості.

4.2.2. Побудова багатофакторної моделі якості моделей машинного навчання. Для вирішення задачі оцінки якості моделей машинного навчання широко використовуються критерії – *aic* та *bic*, які засновані на показнику правдоподібності. Для моделей, заснованих на інших показниках, подібні критерії вибору моделі відсутні. Крім того, показник складності, що використовується в цих критеріях є занадто загальним і потребує уточнення у відповідності від виду моделі машинного навчання, що використовується при побудові інтелектуальної системи.

У загальному випадку *aic* обчислюється за формулою [1, 86, 99]:

$$aic=2k-2\ln(L), \quad (4.5)$$

де k – число параметрів моделі; L – значення функції правдоподібності моделі. Найкращою є модель, значення критерія *aic* для якої є найменшим.

З виразу видно, що зростання критерію зумовлене переважно збільшенням числа параметрів моделі k , та меншою мірою її помилкою. Таким чином, критерій *aic* полягає у виборі моделі з мінімальною кількістю параметрів моделі, які містять найбільшу кількість інформації.

В основі критерія *bic* знаходиться той факт, що при збільшенні кількості параметрів моделі значення функції правдоподібності зростає, але при цьому можливий ефект перенавчання. Коли параметрів моделі виявляється занадто багато, внесок кожного з них у значення функції правдоподібності моделі стає малою, і вони втрачають свою значущість.

Тому завдання вибору моделі полягає в тому, щоб включити в неї мінімум параметрів, які, тим не менш, робили б найбільший внесок у значення функції правдоподібності.

Значення критерію *bic* обчислюють за формулою [54]:

$$bic = k \cdot \ln(n) - 2 \ln(L), \quad (4.6)$$

де n – об'єм навчальної вибірки.

Обидва критерії широко застосовується для аналізу часових рядів і розв'язання задач регресії.

Розвиваючи цей підхід, в роботі запропоновано метрику для оцінки якості моделей машинного навчання у вигляді НМ. Метрика складається з показників функціональності та складності і має наступний вигляд:

$$Q = \ln(Q_c) - \ln(1/l), \quad (4.7)$$

де l – значення функції втрат моделі.

Найкращою обирається модель, для якої значення метрики Q є мінімальним.

Критерій якості моделі машинного навчання у вигляді повнозв'язної нейронної мережі можна отримати, якщо підставити у вираз (4.7) значення Q_c з виразу (4.2):

$$Q = \ln\left(n_k + \sum_{i=1}^{k-1} (n_i + 1)n_{i+1}\right) - \ln(1/l) \quad (4.8)$$

В окремому випадку, для поширеної архітектури тришарової нейронної мережі із часовими затримками, що використовується для ідентифікації нелінійних динамічних об'єктів вираз (4.8) набуває наступного вигляду, якщо замінити значення Q_c виразом (4.3):

$$Q = \ln((M+1)(K+1)+K) - \ln(1/l), \quad (4.9)$$

Для порівняння двох моделей Q_1 та Q_2 можна використовувати відношення:

$$q = Q_2 / Q_1. \quad (4.10)$$

Якщо q менш за 1, то модель Q_2 працює краще за модель Q_1 , якщо q дорівнює 1, то моделі однакові за показником якості, якщо значення q більше 1, то модель Q_1 працює краще, за модель Q_2 .

Запропонована модель (4.9) дозволяє кількісно визначати оцінку якості моделей машинного навчання, а вираз (4.10) допомагає порівнювати декілька моделей між собою. Завдяки цим моделям стає можливим автоматизація процесу оптимізації моделей машинного навчання за цільовим критерієм.

Випробування запропонованого багатофакторного критерія якості моделей машинного навчання здійснюється на прикладі тестового об'єкта з нелінійними динамічними характеристиками, розглянутого в Розділі 2 [39, 108, 174].

В цій задачі для побудови сурогатних моделей тестових нелінійних динамічних об'єктів треба визначити найкращі значення параметрів M та K , які визначають архітектуру TDNN. Для цього збудовано ряд НМ з різною кількістю нейронів M та K у вхідному і прихованому шарах відповідно.

Залежність усередненої функції втрат, визначеної за результатами 5 експериментів для кожного сполучення M та K , від кількості нейронів у вхідному і прихованому шарах представлено на рис. 2.3. В якості функції втрат використовується критерій mse .

За виразом (4.9) обчислюється критерій якості кожної збудованої моделі. Залежність критерія Q від кількості нейронів у вхідному і прихованому шарах представлено на рис. 4.3. Для порівняння, на рис. 4.4 наведено залежність критерія aic від кількості нейронів M та K у вхідному і прихованому шарах відповідно.

За результатами експериментального дослідження запропонованого критерія (4.9) з використанням в якості похибки ε відтворення вихідного сигналу функції mse (1.9) в якості моделі нелінійного динамічного об'єкту обрано нейромережеву структуру з кількістю нейронів у вхідному та прихованому шарах $M=60$ та $K=80$ відповідно. Така структура НМ забезпечує компроміс між точністю ($loss = 0.3$) та складністю збудованої НМ – найменше значення критерію (4.9).

При цьому, нейромережева модель тестового об'єкту, обрана за показником aic , має показник $loss = 0.8$ та структуру з кількістю нейронів у вхідному шарі $M=100$ та кількістю нейронів у прихованому шарі $K=20$ (в загальному випадку, aic не залежить від K). Критерій bic має значення, близькі до значень функції aic , але має тенденцію віддавати перевагу простішим моделям, ніж aic .

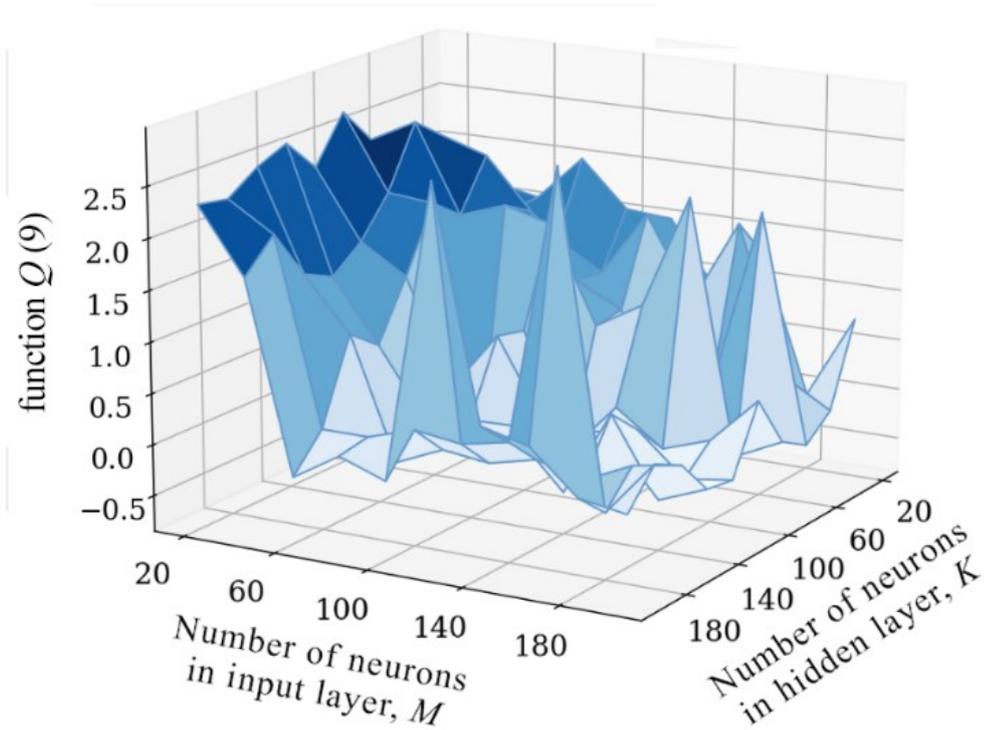


Рисунок 4.3 – Залежність критерія Q від кількості нейронів у вхідному і прихованому шарах

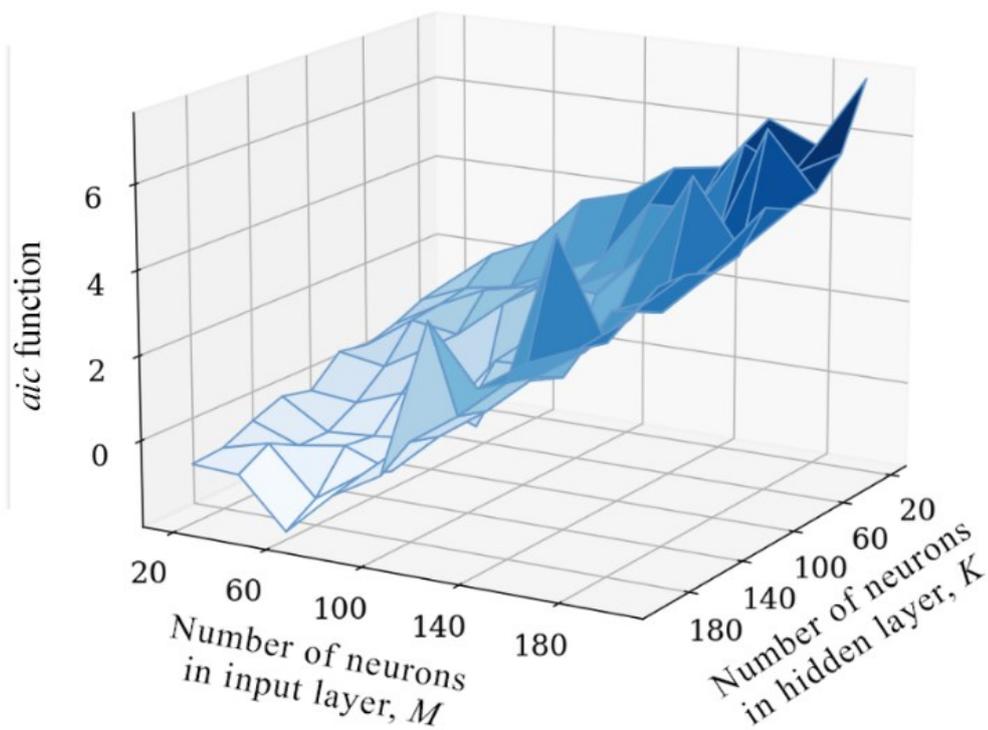


Рисунок 4.4 – Залежність критерія aic від кількості нейронів у вхідному і прихованому шарах

Результати обчислювального експерименту демонструють переваги використання запропонованого критерія (4.9) в точності перед існуючими інформаційними критеріями *aic* та *bic* при порівнянні складності тришарової НМ.

Перевага критерію (4.9) зумовлена більш детальним врахуванню компонента моделі якості, що відповідає за складність моделі машинного навчання.

При збільшенні складності моделі машинного навчання чутливість запропонованого критерія (4.9) зменшується і він наближається до критерію *bic*, за умови незмінного об'єму навчальної вибірки. В цьому випадку складова моделі якості, яка відповідає за точність, впливає на оцінку якості значно більшою мірою.

Область ефективного використання запропонованого критерія обмежена класом моделей машинного навчання у вигляді повнозв'язної тришарової НМ. Для оцінки якості повнозв'язних нейромережевих моделей з іншою структурою можна використовувати запропонований критерій (4.8).

В наступному параграфі, виходячи зі сформульованих вимог здійснюється вибір інструментальних програмних засобів для реалізації розробленого методу інтерпретації моделей машинного навчання.

4.3. Розробка інструментальних програмних засобів інтерпретації моделей машинного навчання

4.3.1. Модульно-орієнтована архітектура інструментальних засобів. Для задоволення наявних вимог необхідно розробити відповідну структуру програмного забезпечення (ПЗ), використовуючи сучасні технології розробки програмного забезпечення в сфері машинного навчання та інтелектуальних систем [12, 75, 77, 94, 128, 145]. У сучасних умовах ринкового тиску розробка ПЗ істотно зорієнтована на хмарні технології та модульність, що має низку значних переваг: робота кількох команд розробників, що використовують різні мови програмування та платформи розробки, інтенсивно використовують готові технології та сервіси.

Під модулем в даному випадку слід розуміти автономну реалізацію певної задачі (етапу розробки структури інтелектуальної системи) з повнофункціональним інтерфейсом, який встановлює правила обміну інформацією при взаємодії модуля з зовнішнім середовищем.

Цей підхід обрано при проектуванні та розробці інструментальних засобів підсистеми інтерпретації моделей машинного навчання. Це дозволяє уникнути початку роботи з нуля, створюючи нові програмні модулі вищого рівня, використовуючи вже готові модулі, доступні в екосистемі проекту, що істотно прискорює процес розробки, дає можливість використання розроблених модулів у подальших проектах.

У зв'язку з викладеним, перспективним напрямом у розробці сучасного програмного забезпечення спеціального призначення стає побудова складної мережі модулів від різних постачальників у вигляді розподіленої системи. Використання сучасних технологій, таких як API-орієнтована архітектура. Цей підхід наголошує на важливості розробки API як основного будівельного блоку програми. Різні модулі можуть взаємодіяти один з одним через певні API, що забезпечує ефективне створення та керування такою складною розподіленою системою.

Основна концепція розробки в цьому випадку полягає в інкапсуляції функціональності великомасштабних додатків у модулях, що поширюються в хмарних середовищах. Ці модулі взаємодіють між собою за допомогою стандартних інтерфейсів, що забезпечує прозору роботу та кросплатформне рішення. Архітектуру програмних засобів в даному випадку можна визначити як множину слабо пов'язаних модулів. Доступ до модулів здійснюється через мережу відповідно до інтерфейсів, чітко визначених цими модулями.

Таким чином, застосування архітектури, орієнтовані на модульність, дозволяє розмістити складові системи по різних вузлах мережі та використовувати їх як незалежні, слабо пов'язані сервіси зі стандартизованим доступом. Це дозволяє досягти чіткого розподілу функціональності системи та інтероперабельності між ними (здатності модулів взаємодіяти між собою та з

іншими системами), прозорості розташування. Цей підхід забезпечує гнучкість при налаштуванні та ефективність при вирішенні практичних задач.

Сучасним стандартом розробки модулів є технологія REST API. Вона встановлює декілька архітектурних принципів для проектування модулів системи:

- робота інтерфейсів за допомогою HTTP/HTTPS для забезпечення взаємодії з різноманітними мережевими програмами, написаними різними мовами програмування;

- додаткова інформація, моделі, параметри та керуючі команди передаються у форматі JSON.

З використанням технології взаємодії на основі стандарту REST, спроектовано структурну схему програмних засобів інтелектуальної системи з інтерфейсами для взаємодії модулів між собою та зовнішніми сервісами, включаючи локальні та хмарні бази даних, настільні та мобільні застосування, IoT пристрої, веб та хмарні сервіси. Узагальнену структурну схему програмних засобів інтелектуальної системи представлено на рис. 4.5.

Наведена узагальнена структурна схема програмних засобів інтелектуальної системи складається з множини модулів, об'єднаних у окремі блоки.

Блок підготовки даних. Для побудови датасету використовуються результати експерименту «вхід-вихід», що зберігаються в базі даних (БД). Інформація отримана з БД після попередньої обробки та нормалізації стає основою для формування навчального та тестового датасетів.

Отримані в кожному блоці результати вимірювань, простори ознак, моделі також зберігаються в БД для подальшого аналізу та використання.

Блок машинного навчання. Для отримання моделі машинного навчання використовуються певні алгоритми побудови простору ознак та навчання нейронної мережі в цьому просторі.

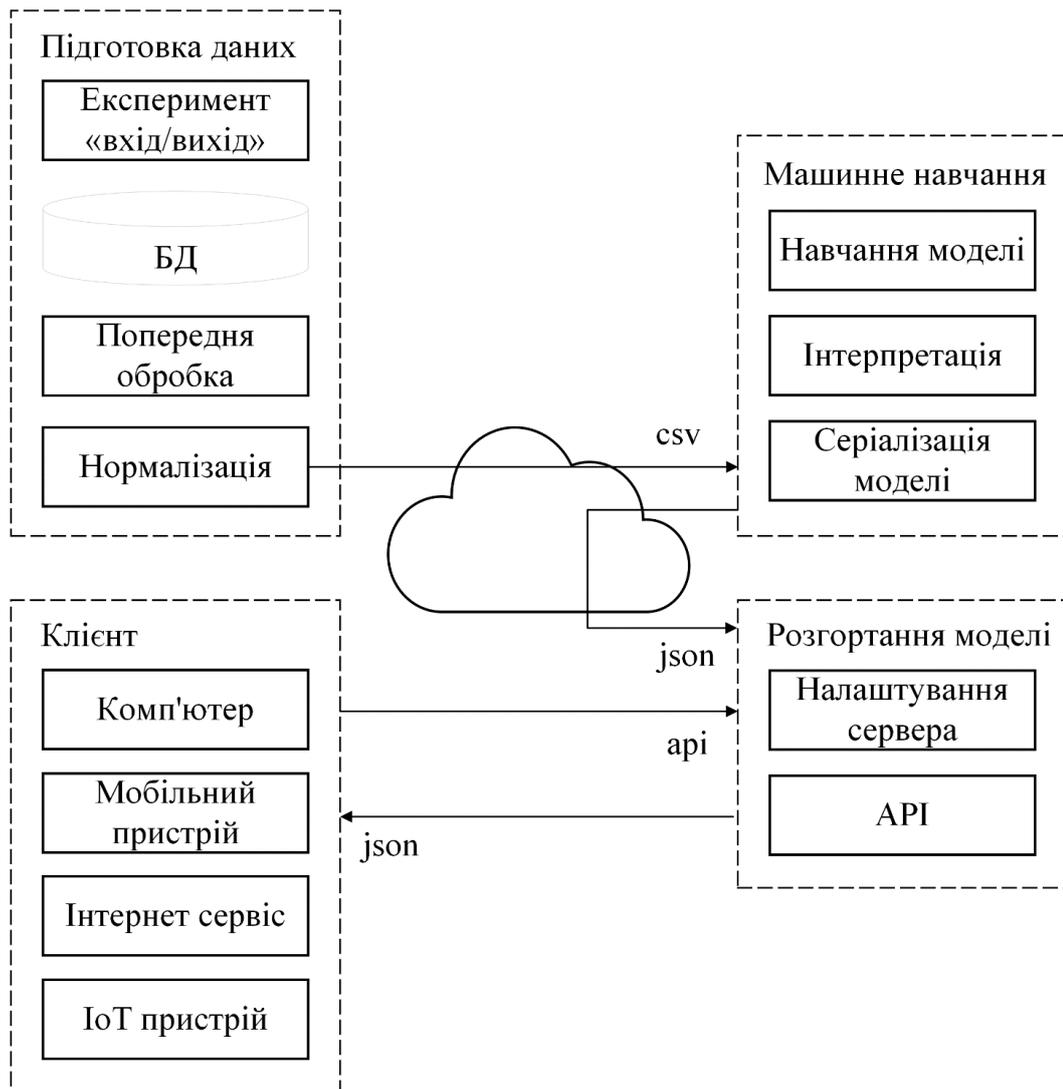


Рисунок 4.5 – Узагальнена структурна схема програмних засобів інтелектуальної системи

Інтерпретація. Для отриманої моделі в цьому модулі будується сурогатна модель. Обидві моделі (нейромережева та сурогатна) серіалізуються для подальшого розгортання та використання.

Блок розгортання моделі. Для використання моделей та організації роботи за REST-інтерфейсом розгортається сервер та розробляються API-функції для забезпечення повнофункціональної взаємодії з моделями.

Блок клієнтського програмного забезпечення. Доступ до моделей за API-функціями можуть здійснювати будь-які клієнти: персональні комп'ютери, сервери, мобільні пристрої, хмарні сервіси та IoT-пристрої.

Наведена структурна схема програмних засобів інформаційної системи має чітко виражену хмарну структуру і відповідає модульно-орієнтованій архітектурі програмного забезпечення.

Область інтересів даної роботи поширюється на етапи машинного навчання та розгортання моделі. На основі запропонованих моделей та розвинутого методу інтерпретації моделей машинного навчання безперервних нелінійних динамічних об'єктів розробляються базові інструментальні засоби ідентифікації та інтерпретації моделей машинного навчання. Для реалізації інструментальних засобів відповідних модулів обґрунтовується вибір технологій розробки програмного забезпечення.

4.3.2. Технології розробки програмного забезпечення розгортання моделей. Технологіями розробки, які найбільше відповідають вимогам, сформульованим у розділі 4.1.2, є хмарні рішення. Існує кілька типів хмарних сервісів, які надають різні рівні абстракції та функціональності для користувачів. Основні типи хмарних сервісів включають:

IaaS (Infrastructure as a Service). IaaS надає віртуальні обчислювальні ресурси через Інтернет. Це включає віртуальні машини, сховище даних і мережеві ресурси. Користувачі можуть керувати та налаштовувати інфраструктуру, встановлювати програмне забезпечення та розгортати свої програми.

PaaS (Platform as a Service). PaaS надає платформу та інструменти для розробки та розгортання власних додатків. Користувачі зазвичай не керують інфраструктурою, а зосереджуються на розробці програм. Це включає бази даних, засоби розробки та інструменти для автоматизації процесів.

SaaS (Software as a Service). SaaS надає доступ через Інтернет до готових до використання програм. Користувачі не керують інфраструктурою, а просто використовують функціональність програми.

FaaS (Function as a Service). FaaS надає середовище виконання функцій (або "функціональних блоків") без необхідності управління інфраструктурою. Код виконується на вимогу та автоматично масштабується.

CaaS (Containers as a Service). CaaS надає контейнери та оркестрацію для їхнього управління. Користувачі можуть запакувати свої програми в контейнери, які легко переносяться і масштабуються.

DaaS (Data as a Service). DaaS надає доступ до даних через Інтернет. Це може включати хмарні бази даних, сховища даних та інші сервіси для обробки та аналізу даних.

Щоб поєднати переваги різних типів хмарних сервісів та відповідати унікальним вимогам проекту розглянуті технології можуть поєднуватися в комбіновані та гібридні структури.

Для обґрунтованого обрання типу хмарних технологій для проектування та розробки програмних засобів виконано порівняльний огляд даних технологій (табл. 4.4).

Таблиця 4.4 – Порівняльний огляд хмарних технологій для проектування та розробки програмних засобів інтерпретації моделей машинного навчання

Характеристика	IaaS	PaaS	SaaS	FaaS	CaaS	DaaS
1	2	3	4	5	6	7
Рівень керування	Повний (інфраструктура)	Частковий (платформа)	Мінімальний (додаток)	Низький (функції)	Середній (контейнери)	Залежить від служби даних
Управління інфраструктурою	Користувальницьке	Обмежене	Не вимагається	Мінімальне (автоматичне)	Часткове (контейнери)	Не вимагається
Складність розгортання	Висока	Середня	Низька	Низька	Середня	Низька
Масштабованість	Користувальницька	Обмежена	Автоматична	Автоматична	Автоматична	Залежно від служби даних
Платня	За ресурси	За платформу	За програму	За функції	За контейнери	За дані
Застосування	Інфраструктурні компоненти	Розробка та розгортання додатків	Готові до використання програми	Мікросервіси, обробка подій	Управління програмами	Обробка, аналіз та надання даних

1	2	3	4	5	6	7
Переваги	Повний контроль над інфраструктурою	Прискорення розробки	Мінімальні зусилля щодо обслуговування	Висока масштабованість	Ізоляція додатків, переносимість	Зручність доступу та обробки даних
Недоліки	Складність управління	Обмежений вибір інструментів	Обмежені можливості налаштування	Обмежені можливості для довготривалих функцій	Необхідність керування контейнерами	Обмежений контроль за інфраструктурою
Приклади	Amazon EC2, Azure Virtual Machines	Google App Engine, Heroku	Google Workspace, Salesforce, Dropbox	Google Cloud Functions, Azure Functions, AWS Lambda	Kubernetes, Docker, Swarm	Amazon RDS, Google Cloud SQL, BigQuery

Прийнявши до уваги вимоги до проекту та необхідний рівень контролю, розробку інструментальних засобів ідентифікації та інтерпретації моделей машинного навчання доцільно здійснювати за допомогою PaaS-технологій з підтримкою мови *Python*. Використання цієї технології поширено в галузі розробки інтелектуальних систем та машинного навчання з ряду причин, і має кілька значних переваг.

– Велика та розвинена екосистема бібліотек машинного навчання для *Python*: *TensorFlow*, *PyTorch*, *Scikit-learn*, *Keras* та інші. Це робить обране середовище розробки дуже привабливим для розробників, тому що вони можуть легко використовувати безліч інструментів для створення та навчання моделей.

– Простота та читаність коду: *Python* має простий та лаконічний синтаксис, який спрощує написання та розуміння коду. Це особливо важливо в галузі машинного навчання, де код часто піддається дослідницькому та експериментальному аналізу.

- Підтримка обробки даних: *Python* активно застосовується при аналізі та обробці даних, що є важливим етапом у створенні моделей машинного навчання. Бібліотеки, такі як *Pandas* та *NumPy*, надають потужні засоби для роботи з даними.
- Інтеграція з BigData та хмарними технологіями: *Python* легко інтегрується з технологіями BigData та хмарними сервісами, що особливо важливо у сучасній розробці інтелектуальних систем.
- Платформонезалежність: *Python* є кросплатформною мовою, що забезпечує платформонезалежність. Проекти, створені на *Python*, можуть бути запущені в різних операційних системах без значних змін.
- Широка підтримка спільноти: *Python* має велику кількість ресурсів, бібліотек та документації для підтримки проектів машинного навчання.
- Інтеграція з іншими мовами: *Python* легко інтегрується з іншими мовами програмування, що корисно, якщо потрібно використовувати бібліотеки або компоненти, написані іншими мовами.
- Широке застосування в науці та дослідженнях: *Python* є популярним вибором у науковій спільноті, що сприяє обміну ідеями, кодом та результатами досліджень у галузі штучного інтелекту.

Ці переваги роблять PaaS-технологій з підтримкою мови *Python* одним з найпопулярніших середовищ програмування для розробки інтелектуальних систем та машинного навчання.

В якості PaaS-сервісу для *Python*-розробки обрано Google Colab. Ця платформа надає хмарне оточення для запуску коду в блокнотах *Jupyter* без керування інфраструктурою із використанням обчислювальних ресурсів Google. У цьому оточенні користувачі можуть писати, виконувати та спільно редагувати код, а також проводити експерименти з моделями машинного навчання, використовуючи обчислювальні ресурси у хмарі. Платформа широко використовується в машинному навчанні та дослідницьких проектах.

Важливим аспектом Google Colab є доступ до графічних процесорів (GPU) та тензорних процесорів (TPU) Google для прискорення навчання моделей машинного навчання. Користувачі можуть використовувати обчислювальні

потужності Google без керування віртуальними машинами або інфраструктурою, що відповідає основним принципам моделі PaaS.

Для програмної реалізації навчання моделей у вигляді нейронних мереж використовується бібліотека *Keras*. Це одна з найбільш популярних високорівневих бібліотек глибокого навчання, яка є офіційним API для *TensorFlow*. *Keras* надає широкий функціонал та інтуїтивно зрозумілий інтерфейс для створення нейронних мереж.

Завдяки принципу модульності вдається комбінувати різні шари та функції активації для побудови різноманітних архітектур нейронних мереж. Бібліотека забезпечує розширені можливості для налаштування та навчання моделей, включаючи різні функції втрат, оптимізатори та метрики для оцінки продуктивності моделей.

Для побудови TDNN-мережі з прямим розповсюдженням сигналу з *Keras* можна використовувати будь-яку кількість послідовних шарів попередньо визначених типів: вхідний (Input), повнозв'язний (Dense) та функціональний (Activation).

Існує декілька технологій та готових платформ для розгортання моделей машинного навчання. Для обґрунтування обрання певного рішення в табл. 4.5 розглянуто кілька популярних технологій розгортання моделей машинного навчання.

Таблиця 4.5 – Порівняльний огляд технологій розгортання моделей машинного навчання

Технологія	Опис	Переваги	Недоліки
1	2	3	4
Docker та Kubernetes	Контейнеризація та оркестрація контейнерів для масштабованого та керованого розгортання	- Масштабованість та керованість. - Ізоляція залежностей	- Необхідність у вивченні Docker та Kubernetes - Деяка складність налаштування

1	2	3	4
AWS SageMaker	Хмарна платформа для розробки, навчання та розгортання моделей машинного навчання на Amazon Web Services	- Інтеграція з іншими сервісами AWS - Управління життєвим циклом моделі	- Залежність від хмарної інфраструктури AWS
Azure Machine Learning	Хмарна платформа для побудови та розгортання моделей машинного навчання на Microsoft Azure	- Інтеграція з іншими сервісами Azure. - Інструменти для роботи з даними	- Залежність від хмарної інфраструктури Azure
Google AI Platform	Хмарна платформа Google Cloud для побудови та розгортання моделей машинного навчання	- Інтеграція з іншими сервісами Google Cloud	- Залежність від хмарної інфраструктури Google Cloud
TensorFlow Serving	Високопродуктивна бібліотека для розгортання моделей TensorFlow	- Висока продуктивність	- Потрібне використання моделей TensorFlow
Open Neural Network Exchange	Відкритий формат обміну моделями між різними фреймворками	- Підтримується різними фреймворками	- Обмежена підтримка певних функцій деякими фреймворками
Flask та FastAPI	Легкі веб-фреймворки для створення веб-сервісів для розгортання моделей	- Простота використання	- Не надають масштабованість та керування життєвим циклом моделей, як інші
MLflow	Платформа управління життєвим циклом машинного навчання із підтримкою розгортання моделей.	- Управління життєвим циклом моделей - Підтримка різних фреймворків	- Вимагає додаткового вивчення та інтеграції у процес розробки
Heroku	Хмарна платформа для розгортання та масштабування веб-застосунків, включаючи моделі машинного навчання.	- Простота розгортання веб-застосунків.	- Обмежені ресурси у безкоштовній версії

Вибір технології розгортання моделей здійснюється з огляду на вимоги до системи, що сформульовано в параграфі 4.1 з урахуванням продуктивності, масштабованості та інтеграції з іншими сервісами.

Так, в якості веб-сервісу для розгортання ML-моделей обрано Flask завдяки його перевагам:

- мінімальний, але гнучкий набір інструментів для створення веб-додатків;
- визначення URL-маршрутів та пов'язаних з ними функцій-обробників;
- автоматична генерація шаблонів HTML та інших форматів;
- можливість використання додаткових бібліотеки та розширення за необхідності.

Структурна схема програмних засобів розгортання моделей машинного навчання наведена на рис. 4.6.

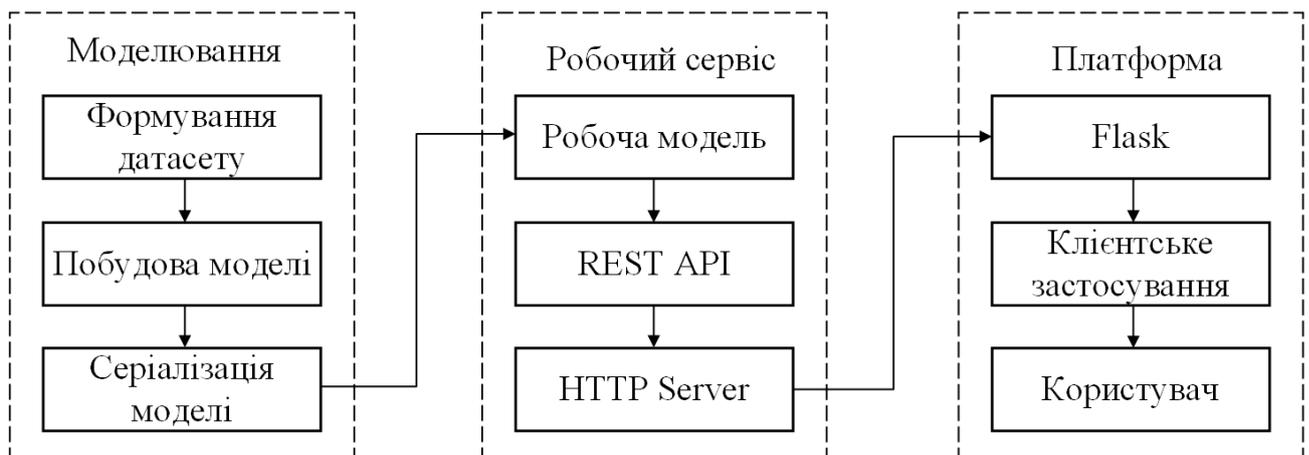


Рисунок 4.6 – Структурна схема програмних засобів розгортання моделей машинного навчання

4.4. Організація структур даних та моделей

4.4.1. Формати і типи даних. Поширеними структурами даних в сервісах машинного навчання є таблиці з числовими або текстовими даними. Модулі інформаційної системи (рис. 4.1) оперують з вихідними даними та перетворюють їх у інші векторні або матричні дані. В залежності від алгоритмів обробки даних

та технологій, що використовуються в процесі перетворення даних, на кожному етапі перетворення даних використовуються різні формати зберігання даних, зручні для виконання певних операцій.

Так, зручною практикою збереження даних, отриманих в результаті експерименту вимірювань є числові дані, які можуть зберігатися у текстових, бінарних файлах або у БД. У сервісах машинного навчання популярними форматами представлення даних є Series та DataFrames. Передача даних між модулями інтелектуальної системи за допомогою REST API здійснюється форматом даних JSON. В табл. 4.6. наведено формати даних, що використовуються в програмних засобах інтелектуальної системи моделювання на кожному конвеєрі з обробки даних.

Таблиця 4.6 – Формати і типи даних, що використовуються в програмних засобах інтелектуальної системи моделювання

Етап	Формат	Введення даних	Передача даних	Серіалізація даних та моделей
Підготовка даних	CSV (Comma-Separated Values)	Текстовий формат	Обмін даними між модулями	Серіалізація табличних даних
	Protobuf (Protocol Buffers)	Бінарний формат	Передача даних між системами	Зберігає дані та моделі в бінарній формі
	SQL (Structured Query Language)	Запити до БД	Взаємодія з реляційними БД	Зберігання та вилучення даних з БД
Перетворення даних	Series та DataFrame (pandas)	Структури даних для аналізу даних	Перетворюється в інші формати для передачі даних	Серіалізація в різні формати даних
Виведення даних	TensorFlow SavedModel	Зберігання в форматі TensorFlow	Зберігання та відновлення моделей в TensorFlow	Зберігання структури моделі та її ваги

4.4.2. Серіалізація моделей. Серіалізація моделей машинного навчання є ключовим етапом у процесі розробки та розгортання інтелектуальних систем. Цей процес дає змогу зберігати стан навчених моделей, забезпечуючи їхню переносимість і можливість повторного використання.

Серіалізація являє собою процес перетворення структури даних у формат, який можна легко зберегти і передати. У контексті машинного навчання, це означає збереження навчених моделей, їхніх параметрів і ваг у формат, який дає змогу відновити модель на іншій пристрої або в іншому оточенні.

Принципи серіалізації включають у себе такі показники.

- Портативність: серіалізовані моделі мають бути переносимими між різними платформами та мовами програмування.

- Ефективність: процес серіалізації має бути ефективним з погляду використання ресурсів, щоб мінімізувати витрати часу та пам'яті.

- Безпека: серіалізація має забезпечувати захист від змін або пошкодження даних.

Існує безліч форматів для серіалізації моделей машинного навчання (табл. 4.6). Серіалізація моделей застосовується на різних етапах життєвого циклу машинного навчання:

1. Навчання та збереження моделей. Після навчання моделі її параметри серіалізуються для подальшого використання. Для збереження навченої моделі та її подальшого використання в додатку Flask під час розробки використовується бібліотека `joblib` для серіалізації моделі.

```
from flask import Flask, jsonify, request
import joblib
app = Flask(__name__)
# Збереження навченої моделі model
model_filename = 'trained_model.joblib'
joblib.dump(model, model_filename)
@app.route('/smodel', methods=['POST'])
def smodel ():
```

...

У цьому прикладі Flask застосунок надає API *endpoint* `/smodel`, який приймає POST-запит із даними для побудови сурогатної моделі. Модель завантажується зі збереженого файлу, і потім використовується в робочому сервісі (сервіс *smodel*).

2. Розгортання моделей. Серіалізовані моделі розгортаються на платформі у вигляді робочого сервісу для роботи в реальному часі.

```
def smodel():
    loaded_model = joblib.load(model_filename)
    data = request.json['data']
    res = loaded_model.predict([data])
    return jsonify({'res': int(res[0])})
```

3. Оновлення моделей. При внесенні змін до моделі, нові версії можуть бути серіалізовані й оновлені в робочому сервісі без перенавчання.

```
@app.route('/update_model', methods=['POST'])
def update_model():
    global current_model
    new_data = request.json['new_data']
    current_model.partial_fit(new_data['X'], new_data['y'])
    joblib.dump(current_model, 'trained_model.joblib')
    return jsonify({'status': 'Model updated successfully'})
```

У цьому коді створюється нова глобальна змінна *current_model*, яка ініціалізується під час старту програми. Потім додається новий *endpoint* `/update_model`, який приймає нові дані для навчання і оновлює модель, зберігаючи її у файл. Це дає змогу оновлювати модель без перезапуску програми та без перенавчання на всьому обсязі даних.

4.5. Інтерфейс системи інтерпретації моделей машинного навчання

З розвитком технологій машинного навчання і штучного інтелекту, створення ефективного і зручного інтерфейсу стає критично важливим компонентом для успішного впровадження систем машинного навчання в різні галузі. У цьому параграфі розглянуто підхід побудови інтерфейсу розроблюваної

системи машинного навчання з використанням сучасних тенденцій розробки. Проектований інтерфейс системи машинного навчання має відповідати всім вимогам, озвученим у параграфі 4.1:

- візуалізація результатів. Оскільки моделі машинного навчання можуть бути складними і важкозрозумілими, важливо надавати інтуїтивні та інформативні візуалізації результатів роботи моделі. Цю вимогу реалізовано шляхом використання потужної підсистеми графіки для візуалізації вхідних, вихідних сигналів, а також моделей на різних етапах роботи.

- інтерактивність. Інтерактивні інтерфейси дають змогу користувачам взаємодіяти з моделями, налаштовувати параметри та отримувати миттєвий зворотний зв'язок. Це особливо корисно під час навчання моделей і проведення експериментів.

- зрозумілість і пояснюваність. Підвищення зрозумілості моделей є активною галуззю досліджень. Інтерфейси мають забезпечувати доступ до пояснень ухвалених моделлю рішень, що покращує розуміння користувачів і підвищує їхню довіру до системи.

Реалізація наведених вимог має проводитися з урахуванням припущення про те, що інтерфейси, які проектуються, розраховані на користувачів без глибоких знань у сфері машинного навчання, тобто фахівців у предметних галузях.

Застосування принципів дизайну користувацького досвіду (User Experience, UX) і інтерфейсного дизайну (User Interface, UI) стає стандартом для створення зручних і ефективних інтерфейсів. Це охоплює інтуїтивний розподіл елементів, колірну гаму, анімацію тощо.

З урахуванням наведеного, розробка UI/UX програмного забезпечення інтерпретації моделей машинного навчання ґрунтується на використанні наявних бібліотек, що спрощує розробку та забезпечує сумісність з екосистемою машинного навчання.

Основна увага при проектуванні UI/UX приділяється організації окремих задач та допоміжних функцій в єдиний конвеєр для обробки вхідних даних з

мінімальними втратами ресурсів. На основі структурної схеми інтелектуальної системи з модулями ідентифікації та інтерпретації (рис.4.1) розроблено конвеєр руху даних всередині інтелектуальної системи інтерпретації моделей машинного навчання нелінійних динамічних об'єктів, який наведено на рис. 4.7.

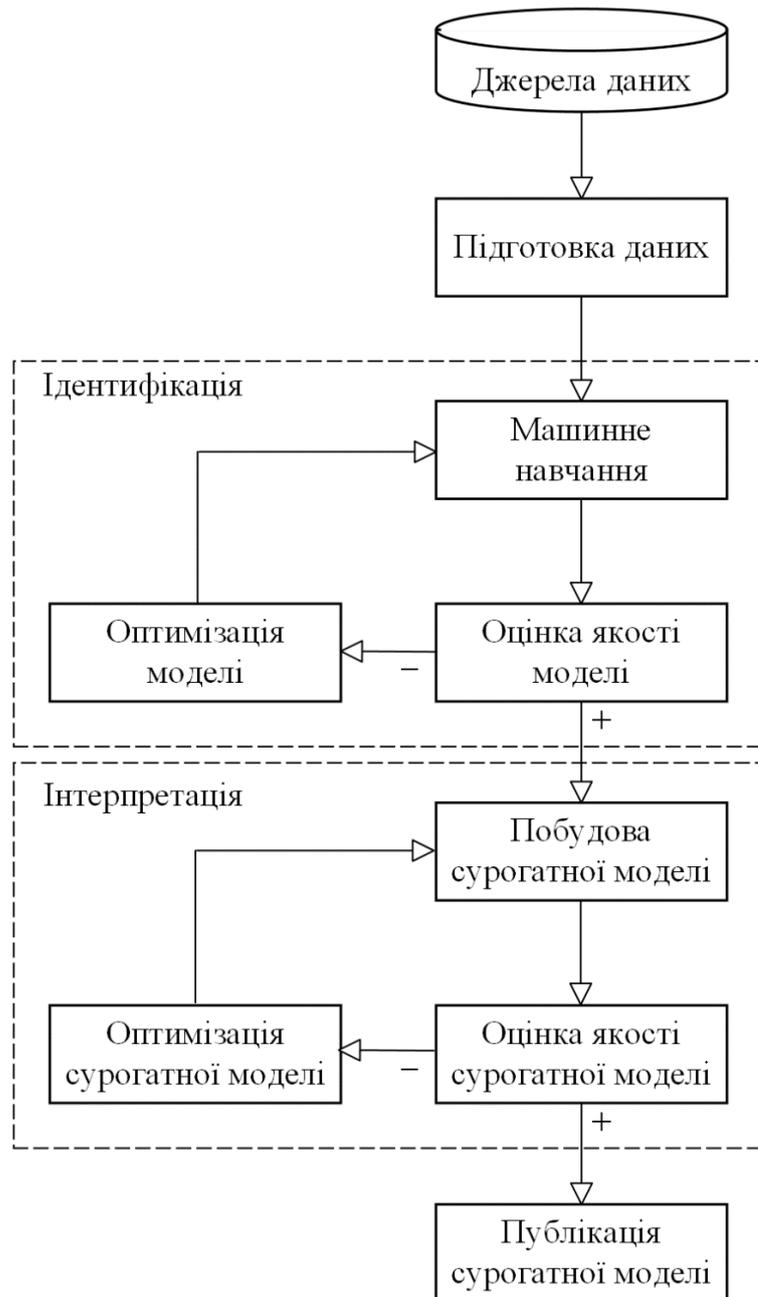


Рисунок 4.7 – Конвеєр руху даних всередині інтелектуальної системи інтерпретації моделей машинного навчання нелінійних динамічних об'єктів

Конвеєр руху даних всередині інтелектуальної системи інтерпретації моделей машинного навчання складається з наступних етапів.

1. Джерела даних
2. Підготовка даних
3. Ідентифікація
4. Інтерпретація
5. Публікація моделі

Проектування інтерфейсу для інтелектуальної системи, що ґрунтується на описаному вище конвеєрі руху даних при інтерпретації моделей машинного навчання, відіграє важливу роль у забезпеченні зручності використання та розуміння роботи системи користувачами.

Проектування інтерфейсу на основі принципу конвеєру руху даних всередині інтелектуальної системи допомогло зробити систему більш доступною та ефективною для користувачів, а також підвищить її прозорість та розуміння.

В результаті спроектовано та розроблено простий та ефективний інтерфейс інтелектуальної системи, наведеної на рис. 4.1.

Система штучного інтелекту з повним циклом роботи, що охоплює навчання та інтерпретацію моделей машинного навчання, вимагає сучасного та інтуїтивного користувацького інтерфейсу, здатного ефективно взаємодіяти з користувачами на всіх етапах роботи з даними та моделями. За допомогою розробленого інтерфейсу відтворено траєкторію руху користувача впродовж циклу побудови моделей машинного навчання (рис. 4.8).

Відкрита програмна архітектура в системах моделювання та інженерних розрахунків сприяє інноваціям, співпраці та більш ефективному використанню інженерних ресурсів. Відкрита програмна архітектура розроблених інструментальних засобів надає низку переваг.

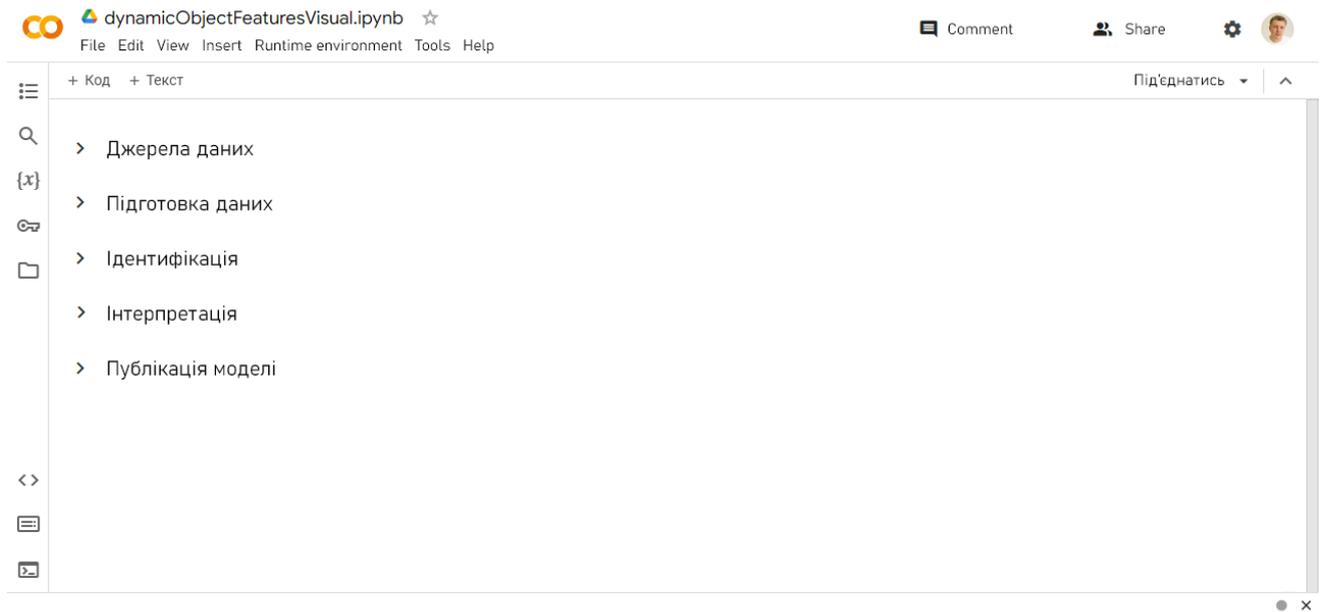


Рисунок 4.8 – Інтерфейс програмних засобів інтерпретації моделей машинного навчання нелінійних динамічних об’єктів

Наступні переваги сприяють більш ефективному та гнучкому процесу розробки та використання ПЗ в інженерній галузі. Інтерфейс взаємодії з програмним кодом в розробленій системі спроектовано з урахуванням відповідності до наступних вимог.

1. Гнучкість та розширюваність. Відкрита архітектура дозволяє користувачам легко розширювати та модифікувати функціональність програмних продуктів під свої потреби.

Можливість інтеграції з іншими програмами та бібліотеками забезпечує створення комплексних систем, здатних вирішувати різноманітні завдання.

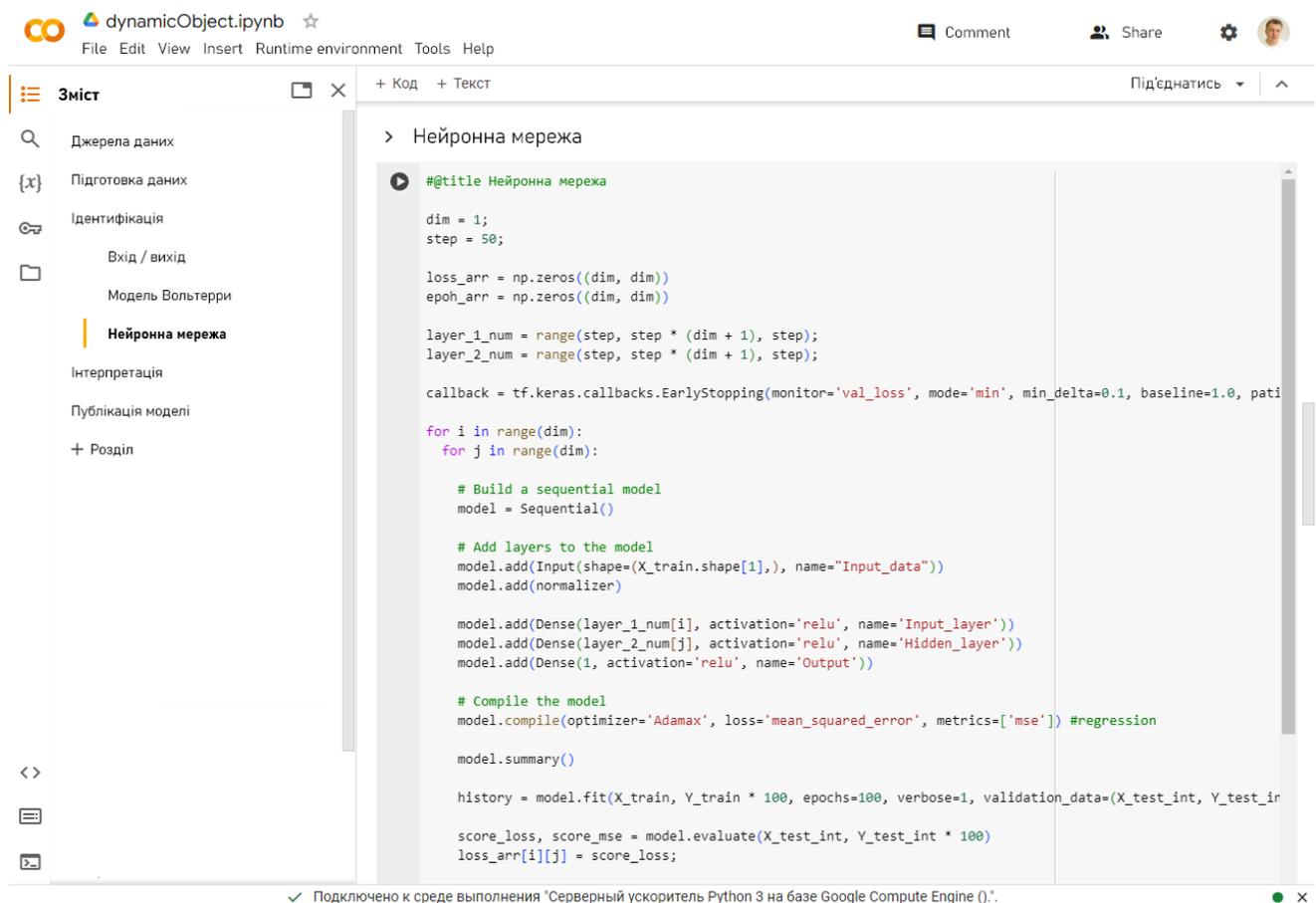
2. Прозорість та надійність. Відкриті програмні архітектури забезпечують прозорість у роботі систем, оскільки користувачі мають доступ до вихідного коду та можуть перевіряти його на безпеку та надійність.

Можливість самостійного контролю за програмними компонентами сприяє швидкому виявленню та усуненню помилок.

3. Ефективне використання ресурсів. Відкриті системи зазвичай надають користувачеві можливість вибору лише необхідних компонентів, що дозволяє оптимізувати використання ресурсів.

З можливістю модифікації та оптимізації коду користувачі можуть адаптувати програмне забезпечення під свої вимоги, скорочуючи надмірність та покращуючи продуктивність.

Інтерфейс відкритої програмної архітектури розроблених програмних засобів наведено на рис. 4.9.



```

#@title Нейронна мережа

dim = 1;
step = 50;

loss_arr = np.zeros((dim, dim))
epoch_arr = np.zeros((dim, dim))

layer_1_num = range(step, step * (dim + 1), step);
layer_2_num = range(step, step * (dim + 1), step);

callback = tf.keras.callbacks.EarlyStopping(monitor='val_loss', mode='min', min_delta=0.1, baseline=1.0, pati

for i in range(dim):
    for j in range(dim):

        # Build a sequential model
        model = Sequential()

        # Add layers to the model
        model.add(Input(shape=(X_train.shape[1],), name="Input_data"))
        model.add(normalizer)

        model.add(Dense(layer_1_num[i], activation='relu', name='Input_layer'))
        model.add(Dense(layer_2_num[j], activation='relu', name='Hidden_layer'))
        model.add(Dense(1, activation='relu', name='Output'))

        # Compile the model
        model.compile(optimizer='Adamax', loss='mean_squared_error', metrics=['mse']) #regression

        model.summary()

        history = model.fit(X_train, Y_train * 100, epochs=100, verbose=1, validation_data=(X_test_int, Y_test_ir

        score_loss, score_mse = model.evaluate(X_test_int, Y_test_int * 100)
        loss_arr[i][j] = score_loss;

```

Рисунок 4.9 – Інтерфейс відкритої програмної архітектури програмних засобів

Нижче описано основні компоненти та принципи інтерфейсу реалізованої системи.

1. Джерела даних

Джерела даних, передбачені для використання в розробленій системі – це текстові та медійні файли даних, API сторонніх сервісів, сенсори та IoT пристрої

2. Підготовка даних

Система надає інструменти для очищення, масштабування, опрацювання пропущених значень і кодування категоріальних ознак.

3. Ідентифікація

Користувачі можуть обирати метод ідентифікації залежно від типу завдання: моделі Вольтерри або нейронні мережі. Для кожної моделі передбачена можливість аналізувати параметри якості ідентифікації, відображення процесів у графічному форматі, реалізована можливість візуалізації прогнозів моделі та порівняння їх з реальними значеннями.

4. Інтерпретація

Користувачі можуть аналізувати роботу збудованих моделей за допомогою графічних засобів інтерпретації, SHAP або будувати сурогатні моделі у вигляді поліномів Вольтерри.

Цей процес дозволяє більш глибоко зрозуміти, як модель приймає рішення, виділяти ключові ознаки та вживати заходів для покращення її продуктивності.

5. Публікація моделі

Істотною зручністю розробленої системи є функціонал серіалізації моделі та її публікації.

Так, для ефективного вирішення задач ідентифікації складних об'єктів та вирішення множини допоміжних задач розроблено інтуїтивний, потужний і гнучкий інтерфейс програмних засобів інтерпретації моделей машинного навчання нелінійних динамічних об'єктів, що забезпечує користувачам простоту використання та широкі можливості аналізу і взаємодії з даними і результатами моделей на протязі всього циклу вирішення задач моделювання. Усі вищеописані компоненти разом створюють комплексний користувацький досвід, що сприяє успішному впровадженню та використанню системи штучного інтелекту.

4.6. Висновки за розділом

Розглянуто вимоги до програмного забезпечення згідно стандарту ISO/IEC 2500n. Запропоновано модель оцінки якості програмного забезпечення інтерпретації моделей машинного навчання.

Для побудови моделі якості ПЗ в контексті інтерпретації результатів машинного навчання визначено релевантні для сурогатних моделей компоненти якості функціональність та складність та визначені метрики для їх оцінки.

Запропонована модель оцінки якості ПЗ інтерпретації моделей машинного навчання згідно з моделлю ISO/IEC 25010, яка дозволяє одночасно врахувати метрики функціональності та складності, що дозволяє кількісно виразити здатність до інтерпретації для формального порівняння сурогатних моделей, збудованих різними методами та автоматизації відбору моделей машинного навчання та сурогатних моделей. Запропонована модель оцінки якості ПЗ є важливою ланкою для забезпечення високої продуктивності та відповідності стандартам у сфері моделювання нелінійних динамічних об'єктів.

Спроектровано структуру програмних засобів комп'ютерного моделювання у складі інтелектуальної системи. Структура забезпечує високу обчислювальну потужність за рахунок використання хмарних обчислень та швидкість пошуку розв'язків, що дозволяє підвищити точність та оперативність моделювання нелінійних динамічних об'єктів. Використання модельно-орієнтованого підходу робить інтелектуальну систему гнучкою та здатною до масштабування.

Розроблені модулі інтелектуальної системи містять ефективні обчислювальні алгоритми побудови моделей машинного навчання та сурогатних моделей нелінійних динамічних об'єктів.

Розроблено інструментальні програмні засоби комп'ютерного моделювання у складі інтелектуальної системи. Розробка програмного забезпечення виконана з використанням сучасних хмарних технологій на платформі Google Colab, що забезпечує високу обчислювальну потужність та швидкість пошуку розв'язків за рахунок використання хмарних сервісів.

Застосування розроблених програмних засобів дозволяє скоротити витрати часу на численні операції з підготовки, передачі та перетворення даних між різними етапами життєвого циклу моделі машинного навчання, забезпечує ефективну взаємодію модулів при конвеєрній обробці даних і організацію простого програмного інтерфейсу між компонентами системи.

Матеріали розділу відображені в публікаціях [36, 89, 94, 158].

ЗАГАЛЬНІ ВИСНОВКИ

В дисертаційній роботі вирішено актуальну науково-технічну задачу підвищення точності сурогатних моделей шляхом розвитку методу інтерпретації моделей машинного навчання нелінійних динамічних об'єктів за допомогою непараметричних динамічних моделей на основі інтегро-степеневих поліномів Вольтерри та реалізації запропонованого методу у вигляді програмно-алгоритмічних засобів ідентифікації неперервних об'єктів у складі інтелектуальних систем.

В рамках проведеного дослідження отримано наступні теоретичні та практичні результати.

1. Виконано аналіз сучасного стану проблеми інтерпретації моделей машинного навчання в системах ідентифікації неперервних нелінійних динамічних об'єктів, систематизовано методи побудови сурогатних моделей та окреслено особливості їх використання. Показано, що застосування існуючих сурогатних моделей орієнтовано на формування якісної картини інтерпретації та не забезпечує достатньої точності ідентифікації моделей машинного навчання нелінійних динамічних об'єктів.

На основі проведеного аналізу обрано та обґрунтовано підхід до інтерпретації моделей машинного навчання шляхом побудови сурогатних моделей у вигляді інтегральних непараметричних динамічних моделей на основі інтегро-степеневих поліномів Вольтерри для аналітичного опису нелінійних динамічних об'єктів.

2. Запропоновано використання непараметричних динамічних моделей на основі інтегро-степеневих поліномів Вольтерри в якості сурогатних моделей в задачах інтерпретації нейронних мереж з часовими затримками.

Встановлено зв'язок в аналітичному вигляді між нелінійними динамічними моделями у вигляді нейронних мереж з часовими затримками та поліномами Вольтерри. Розроблено метод оцінки ядер Вольтерри за допомогою нейронних мереж з часовими затримками з активацією у вигляді лінійного випрямляча.

Перевагою розробленого методу у порівнянні з існуючими є можливість оцінки ядер Вольтерри без застосування вхідних сигналів спеціальної форми, при функціонуванні об'єкту в робочому режимі.

В задачі інтерпретації TDNN моделі тестового нелінійного динамічного об'єкту отримані сурогатні моделі другого порядку шляхом оцінки ядер Вольтерри за параметрами навченої нейронної мережі. Збудовані моделі мають точність вище на 10-12% за лінійну модель.

3. Розвинуто метод інтерпретації нейронних мереж з часовими затримками шляхом побудови сурогатних моделей у вигляді непараметричних динамічних моделей на основі інтегро-степеневих поліномів Вольтерри для опису нелінійних динамічних об'єктів із забезпеченням прийнятної точності моделювання. Перевагою розробленого методу у порівнянні з існуючими є підвищення точності ідентифікації динамічних об'єктів за рахунок найбільш повного урахування нелінійних характеристик в моделях інтегро-степеневих поліномів Вольтерри.

Для визначення області ефективного використання розробленого методу інтерпретації моделей машинного навчання організовано та проведено імітаційні експерименти з використанням даних тестових динамічних об'єктів з гладкими нелінійностями та насиченням. Отримані результати моделювання свідчать про переваги застосування розробленого методу при ідентифікації тестових динамічних об'єктів зі значними нелінійностями в точності на 5-7% у порівнянні з моделями у вигляді інтегрозепеневих рядів Вольтерри.

4. Розроблено інформаційну технологію інтерпретації моделей машинного навчання нелінійних динамічних об'єктів, що ґрунтується на використанні нейронної мережі з часовими затримками для ідентифікації нелінійної динамічної моделі об'єкта за даними експерименту «вхід–вихід» та побудови сурогатної моделі у вигляді непараметричних динамічних моделей на основі інтегро-степеневих поліномів Вольтерри із забезпеченням прийнятної точності моделювання для інтерпретації нейронної мережі.

5. Спроектовано та розроблено програмно-алгоритмічні засоби комп'ютерного моделювання неперервних нелінійних динамічних об'єктів у

складі інтелектуальних систем. Розроблені засоби забезпечують високу обчислювальну потужність та швидкість побудови нейромережових моделей нелінійних динамічних об'єктів та сурогатних моделей для їх інтерпретації при вирішенні ресурсномістких задач.

З використанням розроблених інструментальних засобів комп'ютерного моделювання вирішено прикладні задачі інтерпретації моделей машинного навчання: збудовано сурогатні моделі для динамічного об'єкта з гладкою нелінійністю та суттєвою нелінійністю у вигляді насичення.

Результати дослідження впроваджені в ТОВ «Дискрет» (м. Одеса) та в навчальному процесі Національного університету «Одеська політехніка».

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Agiakloglou, C., Tsimpanos, A. Evaluating the performance of AIC and BIC for selecting spatial econometric models. *J Spat Econometrics*. 2023; Vol. 4: No. 2.
2. A Novel Dynamic Neuro-Space Mapping Approach for Nonlinear Microwave Device Modeling / L. Zhu et al. *IEEE Microwave and Wireless Components Letters*. 2016. Vol. 26, no. 2. P. 131-133.
3. A review of neural network based techniques for nonlinear microwave device modeling / W. Liu et al. *2016 IEEE MTT-S International Conference on Numerical Electromagnetic and Multiphysics Modeling and Optimization (NEMO)*, Beijing, China, 2016. P. 1-2.
4. A Review of Wiener-Type Dynamic Neural Network for Nonlinear Device Modeling / W. Liu et al. *2022 IEEE MTT-S International Microwave Workshop Series on Advanced Materials and Processes for RF and THz Applications (IMWS-AMP)*, Guangzhou, China, 2022. P. 1-3.
5. A survey of methods for explaining black box models / R. Guidotti et al. *ACM Computer Surveys*. 2018. Vol. 51, no. 5. P. 1-42.
6. A time delay neural network based technique for nonlinear microwave device modelling / L. Wenyuan et al. *Micromachines*, Basel. 2020. Vol. 11, no. 9. P. 831.
7. A Wiener-Type Dynamic Neural Network Approach to the Modeling of Nonlinear Microwave Devices / W. Liu et al. *IEEE Transactions on Microwave Theory and Techniques*. 2017. Vol. 65, no. 6. P. 2043-2062.
8. A Wiener-Type Dynamic Neural Network Approach to the Modeling of Nonlinear Microwave Devices and Its Applications / W. Liu et al. *2020 IEEE MTT-S International Conference on Numerical Electromagnetic and Multiphysics Modeling and Optimization (NEMO)*, Hangzhou, China, 2020. P. 1-3.
9. Agresti A. Foundations of Linear and Generalized Linear Models. Wiley series in probability and statistics, 2017. 427 p.

10. An overview of Neuro-space mapping techniques for microwave device modeling / W. Liu et al. *2016 IEEE MTT-S Latin America Microwave Conference (LAMC)*, Puerto Vallarta, Mexico, 2016. P. 1-3.
11. Analysis of the Priorities and Perspectives in Artificial Intelligence Implementation / Y. Kondratenko et al. *The 13th IEEE International Conference on Dependable Systems, Services and Technologies, DESSERT'2023*, Athens, Greece. 2023. P. 1-8.
12. Analytic Solutions Unified Method. Implementation with Agile Principles. 2019. URL: <https://public.dhe.ibm.com/software/data/sw-library/services/ASUM.pdf>.
13. Automated Machine Learning: Methods, Systems, Challenges / Hutter F., Kotthoff L., Vanschoren J. (eds.). Springer. 2019. 242 p.
14. Belani H., Vukovic M., Car Z. Requirements Engineering Challenges in Building AI-Based Complex Systems. *2019 IEEE 27th International Requirements Engineering Conference Workshops (REW)*, 2019. P. 252–255.
15. Biecek P., Burzykowski T. Explanatory Model Analysis. Explore, Explain, and Examine Predictive Models. Chapman and Hall/CRC, New York, 2021. 324 p.
16. Borys A. Consideration of Volterra Series With Excitation and/or Impulse Responses in the Form of Dirac Impulses. *IEEE Transactions on Circuits and Systems II: Express Briefs*. 2010. Vol. CAS 57. Iss. 6. P. 466–470.
17. Borys A. Nonlinear Aspects of Telecommunications: Discrete Volterra Series and Nonlinear Echo Cancellation. Florida, USA: CRC Press. 2000. 300 p.
18. Boyd S., Chua L. O., Desoer C. Analytical foundations of Volterra series. *IMA Journal of Mathematical Control and Information*. 1984. Vol. 1. P. 243–282.
19. Canonical Mathematical Model and Information Technology for Cardio-Vascular Diseases Diagnostics / I. Atamanyuk et al. *Proceedings of 14th The Experience of Designing and Application of CAD Systems in Microelectronics "CADSM 2017"*, Lviv-Poljana, 2017. P. 438-440.
20. Chepynoha V.V., Chepynoha A.V., Palahin V.V. Polynomial estimation of data model parameters with negative kurtosis. *Radio Electronics, Computer Science, Control*. 2023. Vol. 3. C. 64-64.

21. Chua L.O., Liao Y. Measuring Volterra Kernels (III), How to Estimate the Highest Significant Order? *Int. J. Circuit Theory and Applications*. 1991. Vol. 19, no 2. P. 189–209.
22. Computational Complexity Evaluation of Neural Network Applications in Signal Processing / J. Freire Pedro et al. URL: <https://arxiv.org/abs/2206.12191>.
23. Computational modelling for decision-making: where, why, what, who and how / M. Calder et al. *Royal Society Open Science*. 2018, 5, 172096.
24. Construction of a quality model for machine learning systems / J. Siebert et al. *Software Qual J*. 2022. 30. P. 307–335.
25. CRISP-DM Twenty Years Later: From Data Mining Processes to Data Science Trajectories / F. Martinez-Plumed et al. *IEEE Transaction on Knowledge and Data Engineering*. 2020. Vol. 33, no. 8. P. 3048-3061.
26. Deep neural networks for nonlinear model order reduction of unsteady flows / H. Eivazi et al. *Physics of Fluids*. 2020, vol. 32, no. 105104.
27. Definitions, methods, and applications in interpretable machine learning / W. J. Murdoch et al. *Proceedings of the National Academy of Sciences*. 2019. 116(44). P. 22071-22080.
28. Development of models in resilient computing / O. Drozd et al. *2019 10th International Conference on Dependable Systems, Services and Technologies (DESSERT)*. 2019. P. 1-6.
29. Different Modelling Purposes / B. Edmonds et al. *Journal of Artificial Societies and Social Simulation*. 2019. 22(3) 6.
30. Doyle F. J., Pearson R. K., Ogunnaike B. A. Identification and Control Using Volterra Models. Published Springer Technology & Industrial Arts. 2001. 314 p.
31. Explaining Deep Neural Networks and Beyond: A Review of Methods and Applications / W. Samek et al. *Proceedings of the IEEE*. 2021. Vol. 109, no. 3. P. 247-278.
32. Fainzilberg L.S. New Approaches to the Analysis and Interpretation of the Shape of Cyclic Signals. *Cybernetics and Systems Analysis*. 2020. Vol. 56. P. 665-674.

33. Fault Diagnosis Approach Based on Volterra Models / H. Tang et al. *Mechanical Systems and Signal Processing*. 2010. Vol. 24. P. 1099-1113.
34. Fault Diagnosis: Models, Artificial Intelligence, Applications / J. Korbicz, J. M. Kościelny, Z. Kowalczyk, W. Cholewa (Eds.) Berlin: Springer. 2004. 920 p.
35. Favier G., Kibangou A. Y., Bouilloc, T. Nonlinear system modelling and identification using Volterra-PARAFAC models. *International journal of adaptive control and signal processing*. Wiley. 2012. Vol. 26, no. 1, P. 30-53.
36. Fomin O.O., Krykun, V.A. Assessment of the Quality of Neural Network Models Based on a Multifactorial Information Criterion. *Вісник сучасних інформаційних технологій*. 2024. Том 7, № 1. С. 13–23.
37. Fomin O. O., Ruban O. D., Rudkovskyi O. V. Method for Construction the Diagnostic Features Space of Switched Reluctance Motors Based on Integral Dynamic Models. *Problems of the regional energetics*. 2020. Vol. 4 (48). P. 35-44.
38. Fomin O., Pavlenko V. Construction of diagnostic features space using Volterra kernels moments. *20th International Conference «Methods and Models in Automation and Robotics (MMAR)»*. Miedzzydroje, 2015. P. 1022-1027.
39. Fomin O., Masri M., Pavlenko V. Intelligent Technology of Nonlinear Dynamics Diagnostics using Volterra Kernels Moments. *International journal of mathematical models and methods in applied sciences*. 2016. Vol. 10. P. 158-165.
40. Fomin O., Masri M., Pavlenko V. Construction intelligent computing systems for diagnostics complex dynamical objects using Volterra kernels. *Intellectual systems for decision making and problems of computational intelligence (ISDMCI'2015)*. Kherson, 2015. P. 241-243.
41. Fomin O. O., Pavlenko V. D. Model-oriented method for construction of intelligent information systems of diagnosing based on Volterra kernels. *Електротехнічні та комп'ютерні системи*. 2016. P. 146-153.
42. Fomin O., Medvedev A., Pavlenko V. Technology of Intelligent Diagnostics Based on Volterra Kernels Moments. *IEEE 8th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*. Warsaw, 2015. P. 796-801.

43. Fomin O., Pavlenko V., Ruban O. Construction of the diagnostic model based on combining spectral characteristics of nonlinear dynamic objects. *Applied Aspects of Information Technology*. 2020. Vol. 3, no.1. P. 431-442.
44. General Data Protection Regulation (GDPR). URL: <https://gdpr-info.eu/>
45. Giannakis G. B., Serpedin E. A Bibliography on Nonlinear System Identification and its Applications in Signal Processing, Communications and Biomedical Engineering. *Signal Processing – EURASIP*. 2001. Vol. 81, no 3. P. 533-580.
46. Gomolka Z., Dudek-Dyduch E., Kondratenko Y. P. From homogeneous network to neural nets with fractional derivative mechanism. *International Conference on Artificial Intelligence and Soft Computing, ICAISC-2017*, Rutkowski, L. et al. (Eds), Part I, Zakopane, Poland, 2017. LNAI 10245, Springer, Cham, 2017. P. 52-63.
47. Gopika N., Meena Kowshalaya M. E. Correlation Based Feature Selection Algorithm for Machine Learning. *Proceedings of the International Conference on Communication and Electronics Systems (ICCES 2018)*. 2018. P. 692-695.
48. Gu J., Zhao R., Tresp V. Semantics for Global and Local Interpretation of Deep Convolutional Neural Networks. *2021 International Joint Conference on Neural Networks (IJCNN)*, Shenzhen, China, 2021. P. 1-8.
49. Guidelines for Quality Assurance of Machine Learning-based Artificial Intelligence / K. Hamada et al. *SEKE 2020: the 32nd International Conference on Software Engineering & Knowledge Engineering*, 2020. P. 335-341.
50. Hastie T., Tibshirani R., Friedman J. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2nd ed. Springer–Verlag. 2009. 746 p.
51. Hidden Technical Debt in Machine Learning Systems / D. Sculley et al. In: *Proceedings of the 28th International Conference on Neural Information Processing Systems*. 2015. P. 2503-2511.
52. Hleg A. High-Level Expert group on artificial intelligence: Ethics guidelines for trustworthy AI. European Commission. 2019. URL: <https://ec.europa.eu/futurium/en/ai-alliance-consultation.1.html>.

53. Horkoff, J. Non-Functional Requirements for Machine Learning: Challenges and New Directions. *27th International Requirements Engineering Conference (RE2019)*. IEEE Computer Society, Conference Publishing Services, Los Alamitos, California, 2019. P. 386-391.
54. Hossin M., Sulaiman M. N. A Review on Evaluation Metrics for Data Classification Evaluations. *IJDKP*. 2015. Vol. 5. P. 1–11.
55. How does Machine Learning Change Software Development Practices? / Z. Wan et al. *IEEE Transactions on Software Engineering*. 2019. P. 1-14.
56. Improved system identification using artificial neural networks and analysis of individual differences in responses of an identified neuron / A. C. Meruelo et al. *Neural Networks*. 2016. Vol. 75. P. 56-65.
57. Interpretation Method for Dynamic States Neural Network Models / S. Polozhaenko et al. *IEEE 3rd International Conference on System Analysis & Intelligent Computing (SAIC)*. Kyiv, Ukraine, 2022. P. 1-5. DOI: 10.1109/SAIC57818.2022.9923008.
58. Interpretation of Dynamic Models Based on Neural Networks in the Form of Integral-Power Series / O. Fomin et al. ; in: Arsenyeva, O., Romanova, T., Sukhonos, M., Tsegelnyk, Y. (eds). *Smart Technologies in Urban Engineering. STUE 2022. Lecture Notes in Networks and Systems*. Springer, Cham, 2022. Vol. 536. P. 258-265. DOI: 10.1007/978-3-031-20141-7_24.
59. Interpreting Black-box Machine Learning Models for High Dimensional Datasets / Md. Rezaul Karim et al. 2022. URL: <https://arxiv.org/abs/2208.13405>.
60. Ishikawa F. Concepts in Quality Assessment for Machine Learning – From Test Data to Arguments / Juan C. Trujillo et al. (eds.) *Conceptual modeling. 37th International Conference*. Xi'an, China, 2018. Springer, Cham, Switzerland. P. 536-544.
61. Ismail A., Truong H.-L., Kastner W. Manufacturing process data analysis pipelines: a requirements analysis and survey. *Journal of Big Data*. 2019. Vol. 6. P. 1-26.

62. ISO/IEC 25010. (2011). Systems and software engineering. Systems and software Quality Requirements and Evaluation (SQuaRE). System and software quality models. URL: <https://www.iso.org/standard/35733.html>.

63. It takes three to tango: Requirement, outcome/data, and AI driven development / J. Bosch et al. *SiBW 2018*. CEUR Workshop Proceedings. 2018. P. 177-192.

64. Ivanyuk V., Ponedilok V., Sterten Jo. Solving inverse problems of dynamics of non linear objects based on the Volterra series. *Computational problems of electrical engineering*. 2016. Vol. 6, no. 1. C. 9-16.

65. Juscafresa A. N. An introduction to explainable artificial intelligence with LIME and SHAP. Barcelona, 2022. 64 p.

66. Kästner C., Kang E. Teaching Software Engineering for AI-Enabled Systems. *The 42nd International Conference on Software Engineering (ICSE 2020). Software Engineering Education and Training*. 2020. URL: <https://arxiv.org/abs/2001.06691>.

67. Kiciński J. Non-linear Modelling of the Rotating Machine in Technical Diagnostics. The Concept of Adequacy Intervals and Weight Functions in the Identification Procedure. *Advances in Mechanism and Machine Science*. Springer, Cham. 2019. Vol. 73.

68. Potamias G., Papanikolaou E., Hatzigeorgiou A. Knowledge-based TDNN architectures for features recognition in DNA sequences. *Proceedings of IEEE International Joint Conference on Neural Networks*. Washington, DC, USA, 2001. P. 2327.

69. Korbicz, J., Kościelny J. M. Modeling, Diagnostics and Process Control: Implementation in the DiaSter System. Berlin: Springer. 2010. 384 p.

70. Krykun V. Improving the accuracy of the neural network models interpretation of nonlinear dynamic objects. Математичне та комп'ютерне моделювання. Серія: Технічні науки. 2023. Вип. 24. С. 45-55. DOI: 10.32626/2308-5916.2023-24.45-55.

71. Kumeno F. Software engineering challenges for machine learning applications: a literature review. *Intelligent Decision Technologies*. 2019. Vol. 13, no. 4. P. 463-476.
72. Kurgan L. A., Muslek P. A survey of knowledge discovery and data mining process models. *The Knowledge Engineering Review*. 2006. Vol. 21:1. P. 1-24.
73. Large-Scale Machine Learning Systems in Real-World Industrial Settings. A Review of Challenges and Solutions / L. E. Lwakatare et al. *Information and Software Technology*. 2020. Vol. 127. 106368.
74. Forecasting reliability of complex technology objects. Parameters optimization of their technical exploitation: monography / I. Tolok et al ; under edition S.V. Lenkov. Bielsko-Biala: Publishing house «BEL». Poland. 2018. 254 p.
75. Liubchenko V. Applications of Machine Learning in Software Defect Prediction: A Literature Review. *2023 IEEE 18th International Conference on Computer Science and Information Technologies (CSIT)*. Lviv, Ukraine, 2023. P. 1-4.
76. Liubchenko V. Software Metrics Visualization. *Proceedings of the 11th International Conference on Applied Innovations in IT, (ICAIIIT)*. 2023. Vol. 11, no 1. P. 81-87.
77. Liubchenko V. Specific Aspects of Software Development Process for AI/ML-based Systems. *2022 IEEE 17th International Conference on Computer Sciences and Information Technologies (CSIT)*. Lviv, Ukraine, 2022. P. 470-473.
78. Liubchenko V. The Machine Learning Techniques for Enhancing Software Requirement Specification: Literature Review. *2023 IEEE 12th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*. Dortmund, Germany, 2023. P. 10-14.
79. Lundberg S., Lee S.-I. A Unified Approach to Interpreting Model Predictions. *31st Conference on Neural Information Processing Systems (NIPS 2017)*. Long Beach, CA, USA, 2017. P. 1-10.
80. Machine Learning Model Development from a Software Engineering Perspective: A Systematic Literature Review / G. Lorenzoni et al. 2021. URL: <https://arxiv.org/abs/2102.07574>.

81. Machine Learning Techniques for Increasing Efficiency of the Robot's Sensor and Control Information Processing / Y. Kondratenko et al. *Sensors*. 2022. Vol. 22(3). P. 1062.
82. Machine Learning Testing: Survey, Landscapes and Horizons / J. M. Zhang et al. *IEEE Transactions on Software Engineering*, 2020. Vol. 48, no. 1. P. 1-36.
83. Mariscal G., Marbán Ó., Fernández C. A survey of data mining and knowledge discovery process models and methodologies. *The Knowledge Engineering Review*. 2010. Vol. 25. P. 137-166.
84. Marmarelis V. Z., Xiao Zh. Volterra models and three-layer perceptrons. *IEEE Transactions on neural networks*. 1997. Vol. 8, no. 6. P. 1421-1433.
85. Marselis R., Shaukat H. Machine Intelligence quality characteristics. How to measure the quality of Artificial Intelligence and robotics. Sogeti. Paris. 2018.
86. Marselis R., Shaukat H., Gansel T. Testing of Artificial Intelligence. Sogeti. 2017.
87. Marton, S., Lüdtke, S. and Bartelt, C. Explanations for Neural Networks by Neural Networks. *Appl. Sci*. 2022. Vol. 12. P. 980.
88. Marz N., Warren J. Big data. Principles and best practices of scalable real-time data systems. *Manning*. Shelter Island. 2015. 425 p.
89. Mathematical models of software quality assurance for interpretation of dynamic neural networks / O. Fomin, V. Krykun, A. Orlov et al. Вчені записки ТНУ імені В.І. Вернадського. Серія: Технічні науки. 2023. Том 34 (73), № 5. С. 250-256. DOI: 10.32782/2663-5941/2023.5/39.
90. Method of an optimal nonlinear extrapolation of a noisy random sequence on the basis of the apparatus of canonical expansions / I. Atamanyuk et al. *Proceedings of XVIII International Conference on Data Science and Intelligent Analysis of Information "ICDSIAI'2018"*, Kiev, Ukraine, 2018. P. 329-337.
91. Models and algorithms for prediction of electrical energy consumption based on canonical expansions of random sequences / Igor Atamanyuk et al. *Green IT Engineering: Social, Business and Industrial Applications* : monograph. Springer International Publishing, 2019. P. 397-421.

92. Microsoft. Team Data Science Process Documentation. 2019. URL: <https://docs.microsoft.com/en-us/azure/machine-learning/team-data-science-process/>.

93. Mišić J., Marković V., Marinković Z. Behavioral modeling by Volterra series and time-delay neural network approach. *12th Symposium on Neural Network Applications in Electrical Engineering (NEUREL)*. Belgrade, Serbia, 2014. P. 45-50.

94. Mitrea C. A., Lee C. K. M., Wu Z. A comparison between neural networks and traditional forecasting methods: case study. *International journal of engineering business management*. 2009. Vol. 1, no. 2. P. 19-24.

95. Modeling of the agricultural crops development using satellite imagery / O. Fomin et al. Біоінтенсивні та SMART-технології у тваринництві: матеріали II Міжнародної науково-практичної конференції науково-педагогічних працівників та молодих науковців. Одеса, Україна, 2023. С. 10–14.

96. Models of dynamic objects with significant nonlinearity based on time-delay neural networks / O. Fomin, V. Speranskyu, V. Krykun et al. Вісник черкаського державного технологічного університету. Технічні науки. 2023. № 3. С. 97-112. DOI: 10.24025/2306-4412.3.2023.288284.

97. Molnar C. Interpretable Machine learning: A Guide For Making Black Box Models Explainable. Leanpub. 2020. 320 p.

98. Mrugalski M., Korbicz J. Robust fault diagnosis via parameter identification of dynamical systems. *European Control Conference, ECC*. Budapest, Hungary, 2009. P. 2923-2928.

99. Nakajima, S. Quality Assurance of Machine Learning Software. *2018 IEEE 7th Global Conference on Consumer Electronics (GCCE)*. Piscataway, NJ. 2018. Vol. 9–12. P. 601–604.

100. Neuroevolution Methods for Organizing the Search for Anomalies in Time Series / S. Leoshchenko et al. *The Sixth International Workshop on Computer Modeling and Intelligent Systems (CMIS-2023)*. Zaporizhzhia, Ukraine. 2023. P. 164-176.

101. Neuromodeling of operational processes / S. A. Subbotin et al. *Radio Electronics, Computer Science, Control*. 2022. №1. P. 120-129.

102. Nistala P., Nori K. V., Reddy R. Software Quality Models: A Systematic Mapping Study. *Proceedings IEEE/ACM International Conference on Software and System Processes ICSSP 2019*. Montréal, Canada, 2019. P. 125–134.
103. Nonlinear Device Modeling Based on Dynamic Neural Networks: A Review of Methods / W. Liu et al. *2021 IEEE 4th International Conference on Electronic Information and Communication Technology (ICEICT)*. Xi'an, China, 2021. P. 662-665.
104. Neural Networks for Modelling and Control of Dynamic Systems / M. Nørgaard et al. Springer–Verlag, London, 2000. 336 p.
105. Operationalised product quality models and assessment: The Quamoco approach / A. Goeb et al. *Information and Software Technology*. 2015. Vol. 62. P. 101-123.
106. Parameter Estimation in Linear Dynamic Systems using Bayesian networks / M. Garan et al. *Proc. 22nd International Conference "Process Control"*. Strbske Pleso, Slovakia, 2019. P. 203-208.
107. Patton R. J., Fantuzzi C., Simani S. Model-based fault diagnosis in dynamic systems using identification techniques. New York, Springer-Verlag. 2003. 368 p.
108. Pavlenko V., Speranskyy V. Analysis of Identification Accuracy of Nonlinear System Based on Volterra Model in Frequency Domain. *American Journal of Modeling and Optimization*. 2013. Vol. 1, no. 2. P. 11-18.
109. Pavlenko V., Pavlenko S, Speranskyy V. Chapter 10: Identification of systems using Volterra model in time and frequency domain. *Advanced Data Acquisition and Intelligent Data Processing*. V. Haasz, K. Madani (Eds.). River Publishers. 2014. P. 233-270.
110. Pavlenko V., Massri M., Ilyin V. Computing of the Volterra Kernels of a Nonlinear System Using Impulse Response Data. *Proceedings of 9th International Middle Eastern Simulation Multiconference MESM'2008*. Philadelphia University, Amman, Jordan, 2008. P. 131-138.

111. Pavlenko V., Fomin O. Methods For Black-Box Diagnostics Using Volterra Kernels. *ICIM 2008: 2nd International Conference on Inductive Modelling*. Kyiv. 2008. P. 104-107.
112. Pavlenko V. D., Pavlenko S. V. Deterministic identification methods for nonlinear dynamical systems based on the Volterra Model. *Applied Aspects of Information Technology*. 2023. Vol. 1, no. 1. P. 11-32.
113. Petch J, Di S, Nelson W. Opening the Black Box: The Promise and Limitations of Explainable Machine Learning in Cardiology. *Canadian Journal of Cardiology*. 2022. Vol. 38, no. 2. P. 204-213.
114. Pomerat J., Segev A., Datta R. On Neural Network Activation Functions and Optimizers in Relation to Polynomial Regression. *2019 IEEE International Conference on Big Data (Big Data)*. Los Angeles, CA, USA, 2019. P. 6183-6185.
115. Povoroznyuk A., Povoroznyuk O., Skarga-Bandurova I. Application of a multiplicative model with linear partial descriptions in self-organization methods. *9th International Conference "Information Control Systems and Technologies", ICST 2020*. CEUR Workshop Proceedings. Odessa, Ukraine, 2020. Vol. 2711. P. 31-42.
116. Quality Assurance for Machine Learning – an approach to function and system safeguarding / A. Poth et al. *2020 IEEE 20th International Conference on Software Quality, Reliability and Security (QRS)*. 2020. P. 22–29.
117. Ramy E. Ali, Jinhyun So, Avestimehr A. Salman. On Polynomial Approximations for Privacy-Preserving and Verifiable ReLU Networks. 2021. URL: <https://arxiv.org/abs/2011.05530>.
118. Rao A. R., Reimherr M. Non-linear functional modelling using neural networks. 2021. URL: <https://arxiv.org/abs/2104.09371>.
119. Requirements-Driven Method to Determine Quality Characteristics and Measurements for Machine Learning Software and Its Evaluation / K. Nakamichi et al. *28th IEEE International Requirements Engineering Conference (RE)*. Zurich, Switzerland, 2020. P. 260-270.

120. Ruban O. Volterra neural network construction in the nonlinear dynamic systems modeling problem. *Herald of Advanced Information Technology*. 2019. Vol.2, no. 1. P. 24-28.

121. Rudin C., Radin J. Why are we using black box models in AI when we don't need to? A lesson from an explainable AI competition. *Harvard Data Science Review*. 2019. Vol. 2, no. 1.

122. Rugh W. J. Nonlinear System Theory. The Volterra–Wiener Approach. Baltimore: John Hopkins University Press. 1981. 338 p.

123. Schetzen M. The Volterra and Wiener Theories of Nonlinear Systems. New York : John Wiley & Sons. 1980. 389 p.

124. Schlesinger M.I., Vodolazskiy E.V. Minimax Deviation Strategies for Machine Learning and Recognition with Short Learning Samples. *Cybernetics and Systems Analysis*. 2022. Vol. 6. P. 862–875.

125. Schoukens J., Ljung L. Nonlinear System Identification: A User-Oriented Road Map. *IEEE Control Systems Magazine*. 2019. Vol. 39, no. 6. P. 28-99.

126. Sen J. Machine Learning – Algorithms, Models and Applications. London, United Kingdom, IntechOpen, 2021. 154 p.

127. Software Engineering Challenges of Deep Learning / A. Arpteg et al. *2018 44th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*. 2018. P. 50-59.

128. Software Engineering for Machine Learning: A Case Study / S. Amershi et al. *2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*. 2019. P. 291-300.

129. Sousa C. Neural network learning by the levenberg-marquardt algorithm with Bayesian regularization (part 1). – 2009. – URL: <https://www.codeproject.com/Articles/55691/Neural-Network-Learning-by-the-Levenberg-Marquardt>.

130. Stegmayer G., Chiotti O., Orengo G. A Neural Network that helps building a Nonlinear Dynamical model of a Power Amplifier. *ESANN 2005, 13th European*

Symposium on Artificial Neural Networks. Conference Proceedings. Bruges, Belgium, 2005. P. 539-544.

131. Subbotin, S. The Sample and Instance Selection for Data Dimensionality Reduction, Recent Advances in Systems, Control and Information Technology / S. Subbotin, A. Oliinyk. *Advances in Intelligent Systems and Computing*. 2017. Vol. 543. P. 97-103.

132. Sugiyama M., Sawai H., Waibel A. H. Review of TDNN (time delay neural network) architectures for speech recognition. *IEEE International Symposium on Circuits and Systems*. 1991, Vol. 1. P. 582-585.

133. The method of structural adjustment of neural network models to ensure interpretation / S. Leoshchenko et al. *Radio Electronics, Computer Science, Control*. 2021. No. 3. P. 86-96.

134. Time-domain neural network characterization for dynamic behavioral models of power amplifiers / G. Orengo et al. *European Gallium Arsenide and Other Semiconductor Application Symposium, GAAS 2005*. Paris, France, 2005. P. 189-192.

135. Todorovic N., Klan P. State of the Art in Nonlinear Dynamical System Identification using Artificial Neural Networks. *2006 8th Seminar on Neural Network Applications in Electrical Engineering*. Belgrade, Serbia, 2006. P. 103-108.

136. Towards a Volterra series representation from a neural network model / G. Stegmayer et al. *WSEAS Transactions on Circuits and Systems*. 2004. Archive 1. P. 55-61.

137. Towards Guidelines for Assessing Qualities of Machine Learning Systems / J. Siebert et al. In: Shepperd, M., Brito e Abreu, F., Rodrigues da Silva, A., Pérez-Castillo, R. (eds.) *Quality of Information and Communications Technology*. Springer International Publishing, Cham. 2020. 1266. P. 17–31.

138. Transient in the Software Systems. Untraditional Approach to Software Reliability / D. Maevsky et al. *Proceedings of the 33rd International Conference on Information Technologies (InfoTech-2019)*. 2019. P. 1-5.

139. Understanding Development Process of Machine Learning Systems: Challenges and Solutions / E. De Souza Nascimento et al. *2019 ACM/IEEE*

International Symposium on Empirical Software Engineering and Measurement (ESEM). 2019. P. 1–6.

140. Use of Dynamic Neural Networks for Modeling Nonlinear Objects with Significant Nonlinearity / O. Fomin et al. *18th Conference on Computer Science and Intelligence Systems*. Warsaw, Poland, 2023. P. 97-102. DOI: 10.15439/2023F3874.

141. Use of dynamic neural networks for modeling nonlinear objects with significant nonlinearity / S. Polozhaenko et al. Збірник тез IV Міжнародної науково-практичної Інтернет-конференції "Математика та інформатика в науці й освіті: виклики сучасності". Вінниця, Україна, 2023. С. 121-124.

142. Visualizing higher-layer features of a deep network / D. Erhan et al. University of Montreal. 2009. Vol. 1341, no. 3. P. 1-13.

143. Vogelsang A., & Borg M. Requirements Engineering for Machine Learning: Perspectives from Data Scientists. *2019 IEEE 27th International Requirements Engineering Conference Workshops (REW)*. P. 245-251.

144. Wang Y., Kim S.-Ph., Principe J. C. Comparison of TDNN training algorithms in brain machine interfaces. *Proceedings of 2005 IEEE International Joint Conference on Neural Networks*. Montreal, QC, Canada, 2005. Vol. 4. P. 2459-2462.

145. Zhang D., Tsai J. Machine learning and software engineering. In: Staff, I.C.S. (ed.) *14th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2002)*. IEEE Computer Society Press, Los Alamitos, 2002. P. 22-29.

146. Айвазян С. А., Енюков И. С., Мешалкин Л. Д. Прикладная статистика: Исследование зависимостей. Справочное издание. М.: Финансы и статистика. 1985. 487 с.

147. Айвазян С. А., Енюков И. С., Мешалкин Л. Д. Прикладная статистика: Основы моделирования и первичная обработка данных. Справочное издание. М.: Финансы и статистика. 1983. 471 с.

148. Борис А. Аналіз і синтез нелінійних пристроїв і засобів телекомунікації з використанням неперервного і дискретного рядів Вольтерри: Автореф. дис. д-ра техн. наук: 05.12.13. Нац. ун-т. "Львів. політехніка". 2005. 36 с.

149. Верлань А.Ф., Дячук О.А., Палагін В.В. Методи математичної редукції моделей динамічних систем. Київ: Наукова думка. 2019. 311с.
150. Верлань А. Ф., Москалюк С. С. Математическое моделирование непрерывных динамических систем. К.: Наукова думка, 1988. 287 с.
151. Верлань А. Ф., Сизиков В. С., Мосенцова Л. В. Метод вычислительных экспериментов для решения интегральных уравнений в обратной задаче спектроскопии. *Електронне моделювання*. 2011. 33, №2. С. 3-12.
152. Задірака В. К., Швідченко І. В. Про точність наближеного розв'язку задачі. *Вісник НАН України*. 2022. №10. С. 53-58.
153. Ивахненко А. Г., Юрачковский Ю. П. Моделирование сложных систем по экспериментальным данным. М.: Радио и связь, 1987. 117 с.
154. Ивахненко А. Г. Моделирование сложных систем: информационный подход. К.: Наукова думка, 1987. 136 с.
155. Іванюк В.А, Понеділок В.В. Дячук О.А Метод обернених операторів відновлення сигналів на вході лінійних динамічних систем, що задані передатними функціями. *Математичне та комп'ютерне моделювання. Серія: Технічні науки* : зб. наук. праць / Інститут кібернетики імені В. М. Глушкова Національної академії наук України, Кам'янець-Подільський національний університет імені Івана Огієнка ; [редкол.: О. М. Хіміч (відп. ред.) та ін.]. Кам'янець- Подільський: Кам'янець-Подільський національний університет імені Івана Огієнка, 2017. Вип. 15. С. 62-67.
156. Іванюк В.А., Федорчук В.А. Адаптивний метод ідентифікації моделей нелінійних динамічних систем інтегральними рядами Вольтерри. *Електронне моделювання*. 2019. Т. 41, № 3. С. 33-42.
157. Іванюк, В., Фомін О. Інтегральні динамічні моделі в задачах ідентифікації і діагностики. *Математичне та комп'ютерне моделювання. Серія: Фізико-математичні науки*. 2020, С. 42-53.
158. Крикун В. А. Математична модель оцінки якості програмного забезпечення інтерпретації моделей машинного навчання. *Комп'ютеризовані системи та програмні технології*. 2023. № 1. С. 7–11.

159. Крикун В. А., Фомін О. О. Інтерпретація динамічних моделей у вигляді нейронних мереж з часовими затримками. *Матеріали Дванадцятій Міжнародної наукової конференції студентів та молодих учених «Сучасні інформаційні технології – 2022»*. Одеса, Україна, 2022. С. 140-141.

160. Крикун В. А., Фомін О. О. Нелінійне моделювання об'єктів на основі динамічних нейронних мереж. *Матеріали Тринадцятій Міжнародної наукової конференції студентів та молодих учених «Сучасні інформаційні технології – 2023»*. Одеса, Україна, 2023. С. 153-155.

161. Метод и информационная технология построения непараметрической динамической модели глазо-двигательного аппарата / А. А. Фомин и др. *Східно-Європейський журнал передових технологій*. 2015. Том 4, вып. 2/9 (74). С. 64-69.

162. Моделирование динамических систем: аспекты мониторинга и обработки сигналов / Под ред. В. В. Васильева. Институт проблем моделирования в энергетике им. Г. Е. Пухова НАН Украины. Отделение гибридных моделирующих и управляющих систем в энергетике. К.: 2002. 344 с.

163. Мокін Б. І., Мокін В. Б., Мокін О. Б. Математичні методи ідентифікації динамічних систем: навчальний посібник. Вінниця: ВНТУ, 2010. 260 с.

164. Павленко В. Д. Идентификация нелинейных динамических систем в виде ядер Вольтерры на основе данных измерений импульсных откликов. *Електронне моделювання*. 2010. Т. 32. №3. С. 3-18.

165. Павленко В. Д., Череватий В. В. Метод идентификации нелинейных систем на основе рядов Вольтерра во временной области. *Проблеми інформатизації та управління: Збірник наукових праць НАУ*. К.: НАУ, 2004. Вип. 11. С. 161-164.

166. Підвищення оперативності побудови діагностичних моделей великих розмірностей в системах технічної діагностики / Рубан О.Д., Федорова .М., Кузін І.О., Фомін О.О. *VII Міжнародна науково-практична конференція «Інформатика. Культура. Технології» ІКТ–2019*. С. 23-25.

167. Поворознюк А. І., Поворознюк О.А. Врахування статистичної та експертної інформації в медичних системах підтримки прийняття рішень. *Proceedings of XVI International Conference Measurement and Control in Complex System (MCCS-2022)*. 2022. Р. 1-5.

168. Серков А. А., Бреславец В. С. Моделирование информационных систем с априорной неопределенностью. *Вісник ХДПУ. Сер.: Системний аналіз, управління і інформаційні технології*: Зб. наук. праць. 2000. Вип. 108. С. 55-59.

169. Субботин С.А., Бойченко К. Ю. Оценка моделей представления данных в системах обнаружения и распознавания объектов. *Искусственный интеллект*. 2009. №2. С. 136-142.

170. Тихоход В.О., Федорчук В.А. Паралельний алгоритм розв'язування систем інтегральних рівнянь Вольтерри II роду. *Електронне моделювання*. 2023. Том 45, Випуск 6. С. 3-10.

171. Файнзільберг Л. С. Інтелектуальні інформаційні технології оброблення сигналів з локально зосередженими ознаками (до 100-річчя з дня народження академіка В.М. Глушкова). *International Scientific Technical Journal "Problems of Control and Informatics"*. 2023. Vol. 68(3). Р. 61-73.

172. Федорчук В.А., Іванюк В.А., Понеділок В.В. Синтез макромоделей елементів керованих електромеханічних систем за інтегральними моделями їх структурних ланок. *Наукові праці кам'янець-подільського національного університету імені Івана Огієнка*. Кам'янець-Подільський: Кам'янець-Подільський національний університет імені Івана Огієнка. 2019. Вип. 15. С. 67-71.

173. Федорчук В.А., Канарчук Ю.В. Комп'ютерне моделювання динаміки одновимірних нелінійних об'єктів з розподіленими параметрами. *Математичне та комп'ютерне моделювання. Серія: Технічні науки*. 2018. Вип. 18. С. 91-102.

174. Фомін О. О. Інформаційна технологія вторинної спектрально-статистичної ідентифікації нелінійних інерційних об'єктів в системах технічної діагностики. Дис. докт. техн. наук. Одеса, 2019. 337 с.

175. Фомін О. О. Метод побудови простору діагностичних ознак на основі інтегральних динамічних моделей. *Вчені записки ТНУ імені В. І. Вернадського. Серія: Технічні науки*. 2018. Том 29 (68), № 1. С. 43-47.

176. Фомін О. О., Павленко В. Д. Модельно-орієнтована інформаційна технологія побудови систем діагностування нелінійних динамічних об'єктів. *Проблеми інформатики и моделирования (ПИМ-2017)*. Харьков-Одесса (Каролино-Бугаз). 2017. С. 91.

177. Фомін О. О. Формування простору діагностичних ознак на основі перетинів ядер Вольтерра. *Математичне та комп'ютерне моделювання. Серія: Технічні науки*. 2018. Вип. 17. С. 141-150.

Додаток А

Програма генерації датасету для імітаційної моделі тестового нелінійного динамічного об'єкта

Код програми:

```

clear all;
time = 0:0.01:3;
timeSize = size(time,2);
mSize = 15;
m = zeros(1, mSize-1);

for a=0.1:.1:1.0
    %step input
    input = ones(timeSize,1) .* a;

    %pulse input
    %input = zeros(timeSize,1);
    %input(1:1) = a;

    %sin input
    %input = zeros(timeSize,1);
    %for i=1:timeSize
    %    input(i) = a * sin(25 * pi * (i-1) / timeSize);
    %end

    %line input
    %input = zeros(timeSize,1);
    %halfTimeSize = round(timeSize / 2) - 1;
    %for i=1:halfTimeSize
    %    input(i) = a * i / halfTimeSize ;
    %end
    %for i=(halfTimeSize + 1):timeSize
    %    input(i) = a * ( 1 - (i - halfTimeSize) / (timeSize -
        halfTimeSize));
    %end

    simIn = [time' input];
    simOutT = dynamicObjectPowMdl(simIn);
    resFull = [m simOutX'];
    resCount = size(simOutX, 1);
    for i=1:resCount
        if ~exist('res','var')
            res = [m 0 0; resFull(i:i+mSize-1) simOutY(i)];
        else
            res = [res; resFull(i:i+mSize-1) simOutY(i)];
        end
    end
end
end
csvwrite('dynamicObject.csv',res)

```


Додаток Б

Реалізація засобів комп'ютерного моделювання для розв'язування задач інтерпретації моделей машинного навчання

Вигляд інтерфейсу ПЗ, що відтворює траєкторію руху користувача впродовж основних етапів циклу побудови моделей машинного навчання.

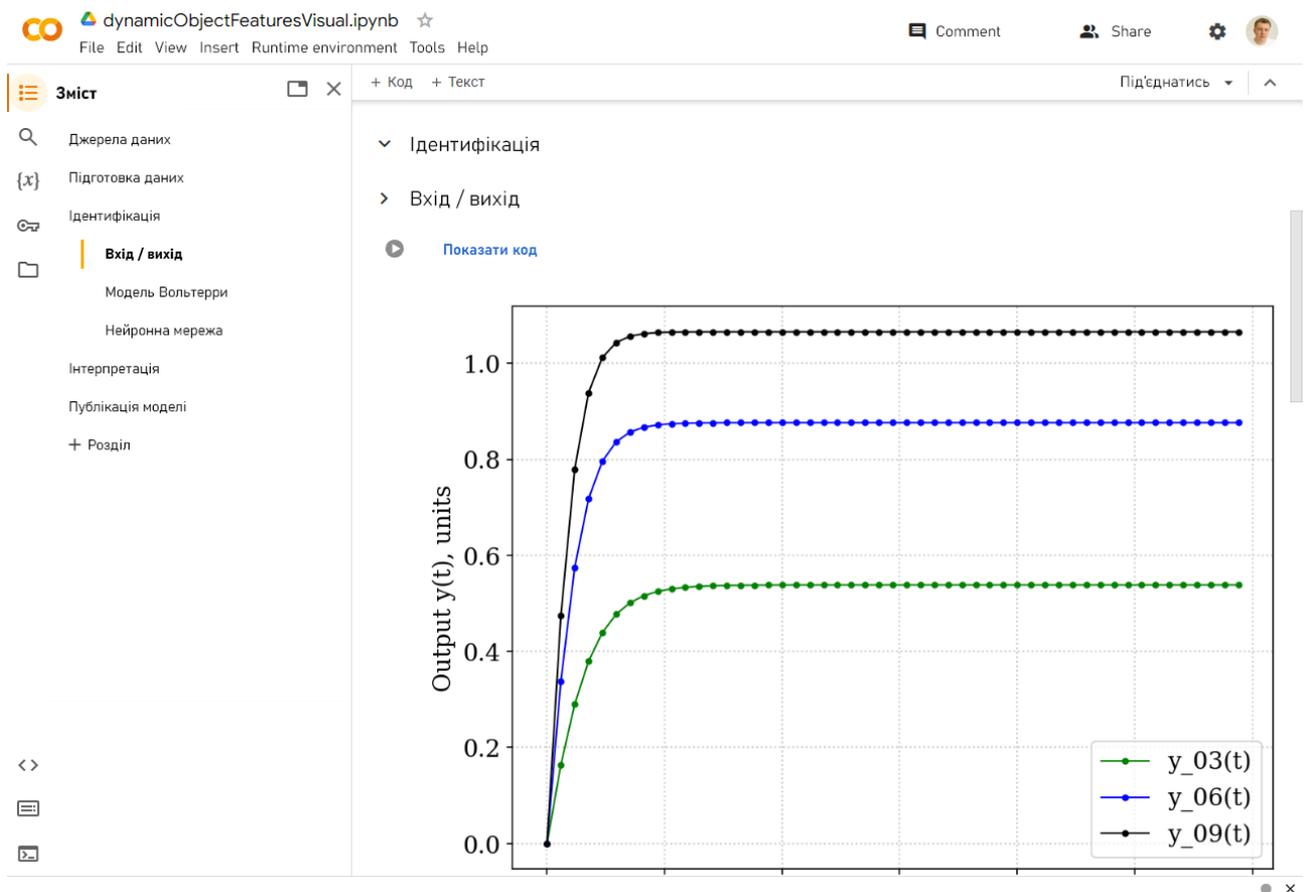


Рисунок Б.1 – Зовнішній вигляд сторінки «Ідентифікація - Вхід/вихід»

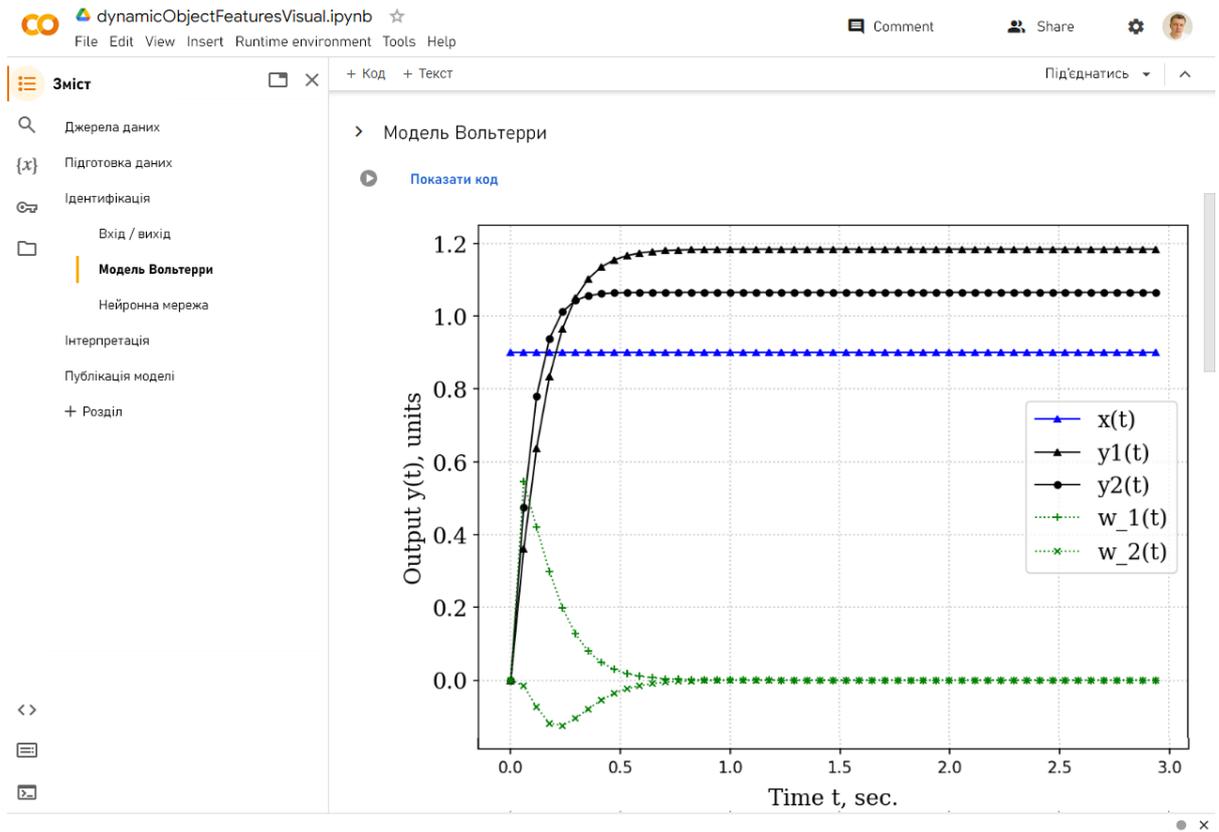


Рисунок Б.2 – Зовнішній вигляд сторінки «Ідентифікація – Модель Вольтерри»

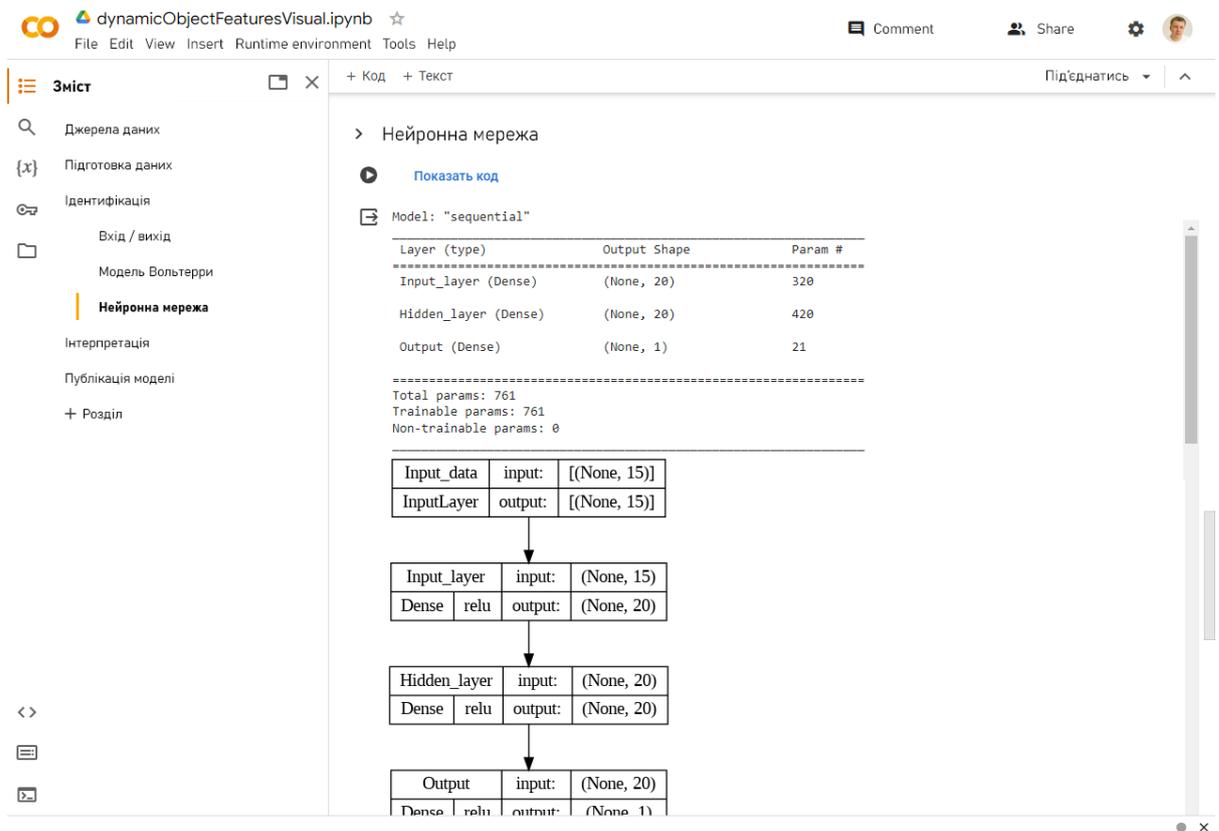


Рисунок Б.3 – Зовнішній вигляд сторінки «Ідентифікація – Нейронна мережа»

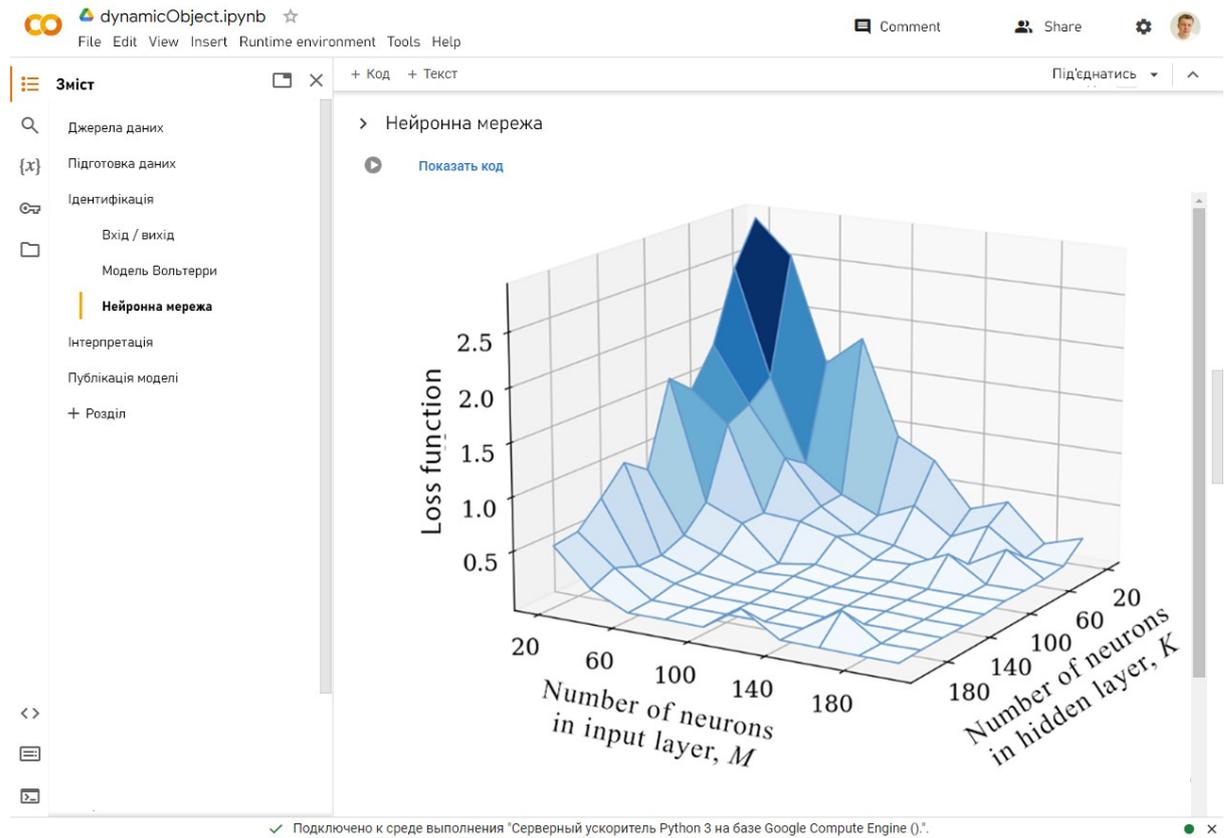


Рисунок Б.4 – Зовнішній вигляд сторінки «Ідентифікація – Оптимізація нейронної мережі»

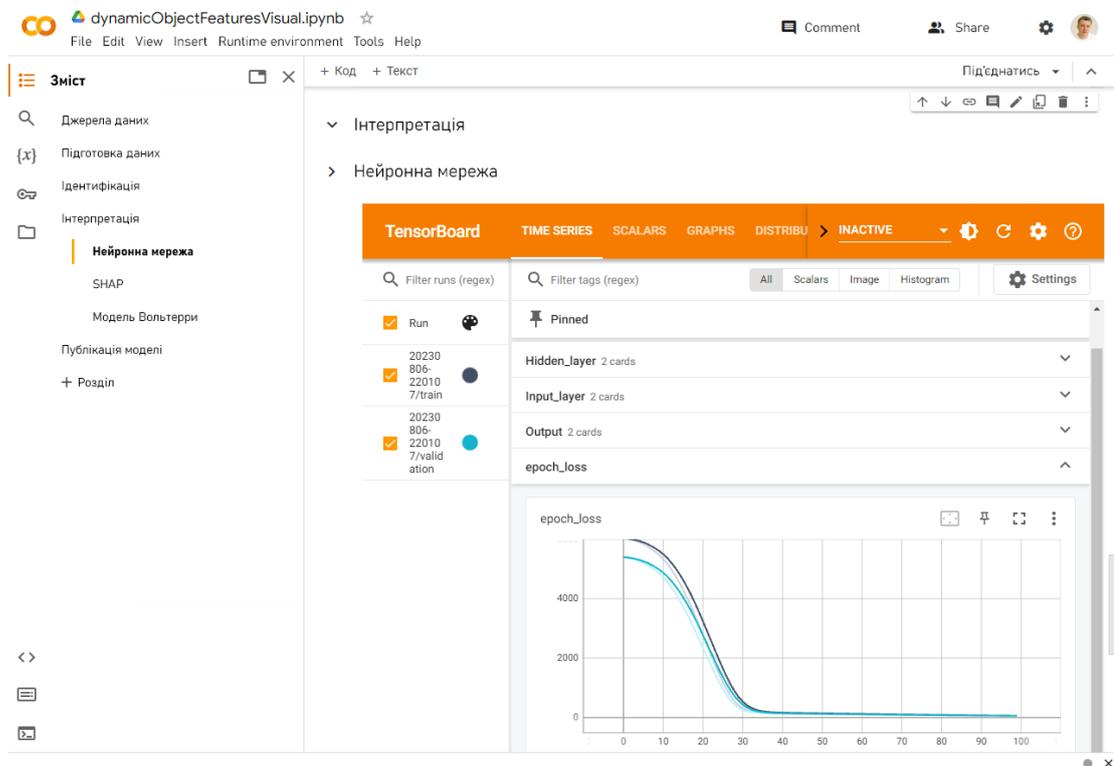


Рисунок Б.5 – Зовнішній вигляд сторінки «Інтерпретація – Навчання нейронної мережі»

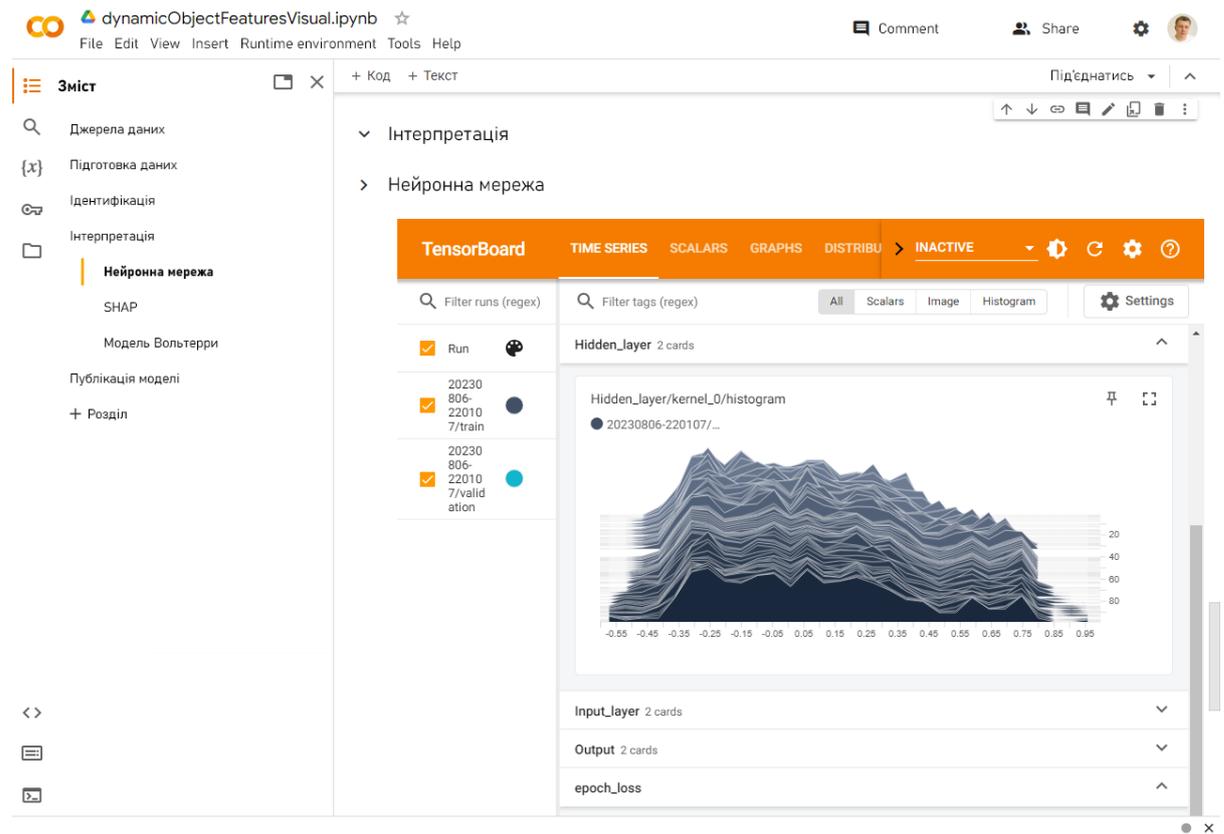


Рисунок Б.6 – Зовнішній вигляд сторінки «Інтерпретація – Нейронна мережа»

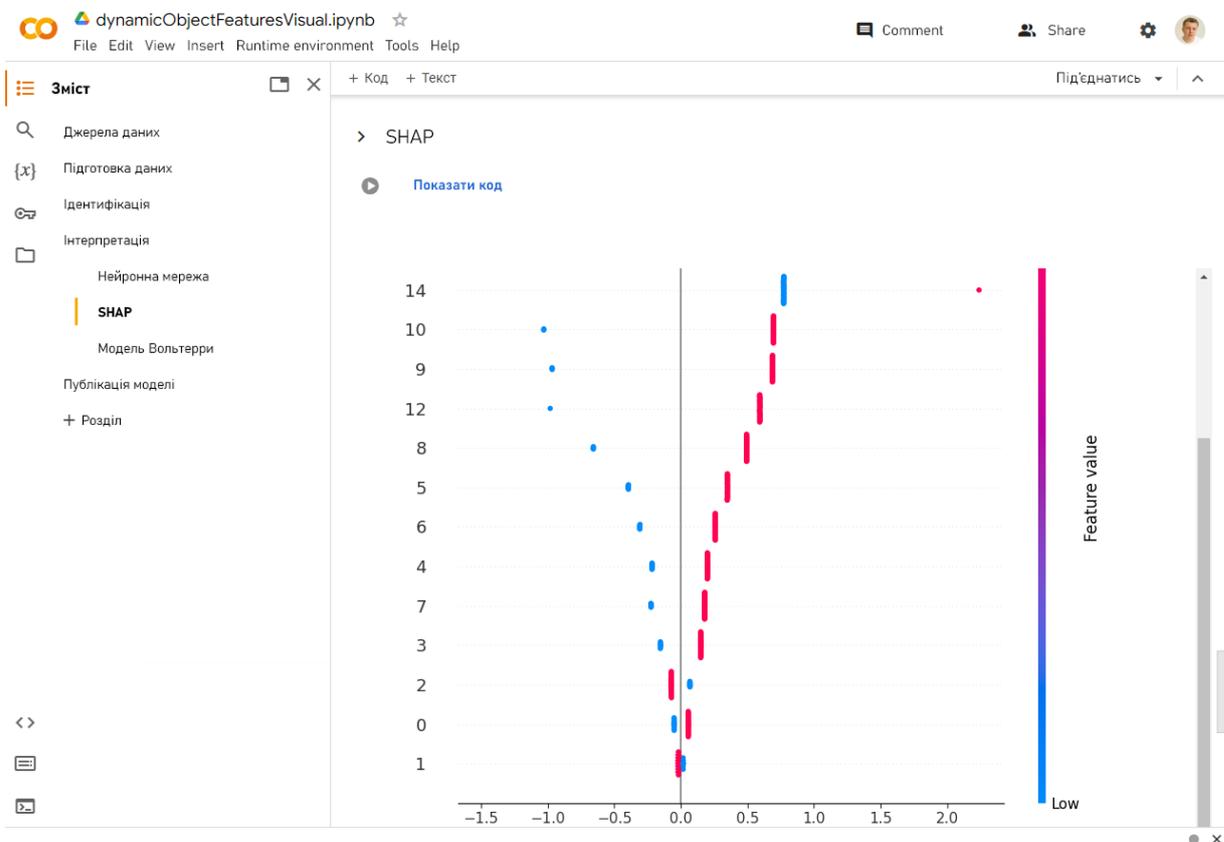


Рисунок Б.7 – Зовнішній вигляд сторінки «Ідентифікація - SHAP»

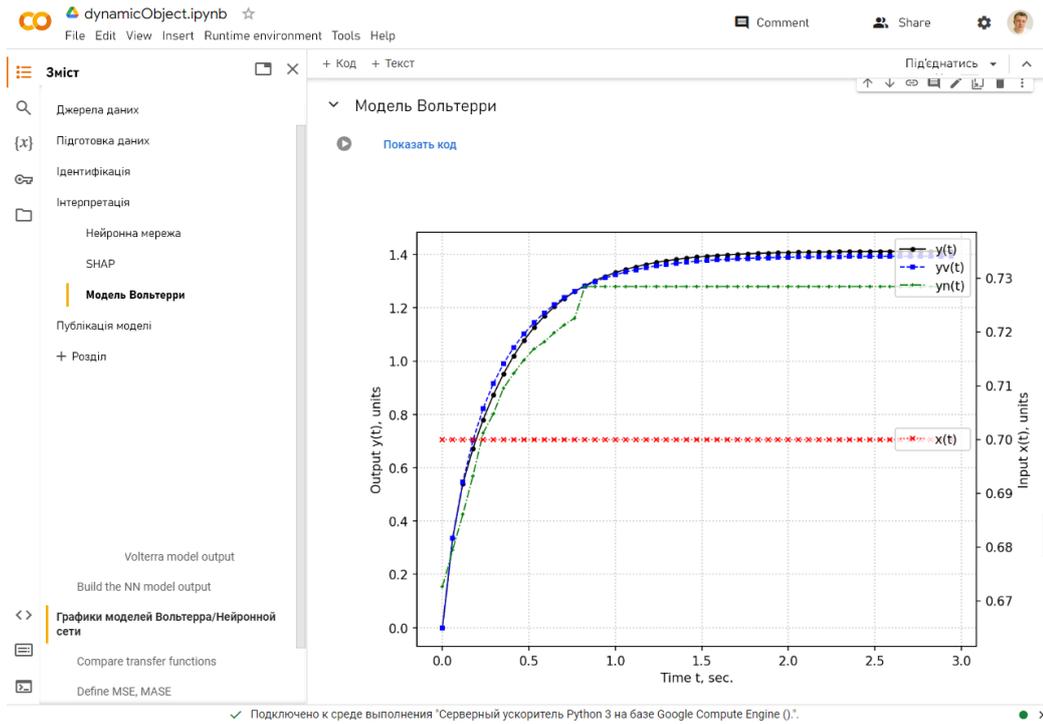


Рисунок Б.8 – Зовнішній вигляд сторінки «Ідентифікація – Модель Вольтерри»

Додаток В

Список публікацій здобувача

Наукові праці, в яких опубліковано основні наукові результати дисертації.

1. Krykun V. Improving the accuracy of the neural network models interpretation of nonlinear dynamic objects. *Математичне та комп'ютерне моделювання. Серія: Технічні науки*. 2023. Вип. 24. С. 45–55. DOI: 10.32626/2308-5916.2023-24.45-55. Видання включено до переліку наукових фахових видань України, категорія «Б».

<http://mcm-tech.kpnu.edu.ua/issue/view/17449>

2. Mathematical models of software quality assurance for interpretation of dynamic neural networks / O. Fomin, V. Krykun, A. Orlov et al. *Вчені записки ТНУ імені В.І. Вернадського. Серія: Технічні науки*. 2023. Том 34 (73), № 5. С. 250–256. DOI: 10.32782/2663-5941/2023.5/39. Видання включено до переліку наукових фахових видань України, категорія «Б».

<https://tech.vernadskyjournals.in.ua/34-73-5>

3. Models of dynamic objects with significant nonlinearity based on time-delay neural networks / O. Fomin, V. Speranskyu, V. Krykun et al. *Вісник черкаського державного технологічного університету. Технічні науки*. 2023. № 3. С. 97–112. DOI: 10.24025/2306-4412.3.2023.288284. Видання включено до переліку наукових фахових видань України, категорія «Б».

<http://vtn.chdtu.edu.ua/issue/view/16885>

4. Fomin O.O., Krykun, V.A. Assessment of the Quality of Neural Network Models Based on a Multifactorial Information Criterion. *Вісник сучасних інформаційних технологій*. 2024. Том 7, № 1. С. 13–23. DOI: 10.15276/hait.07.2024.1. Видання включено до переліку наукових фахових видань України, категорія «Б».

<https://hait.od.ua/index.php/journal/article/view/201>

Наукові праці, які засвідчують апробацію матеріалів дисертації.

5. Interpretation of Dynamic Models Based on Neural Networks in the Form of Integral-Power Series / O. Fomin et al. ; in: Arsenyeva, O., Romanova, T., Sukhonos, M.,

Tsegelnyk, Y. (eds). *Smart Technologies in Urban Engineering. STUE 2022. Lecture Notes in Networks and Systems*. Springer, Cham, 2022. Vol. 536. P. 258–265. DOI: 10.1007/978-3-031-20141-7_24. Видання проіндексовано у базах даних *Web of Science Core Collection* та *Scopus*.

<https://link.springer.com/book/10.1007/978-3-031-20141-7?page=2>

6. Interpretation Method for Dynamic States Neural Network Models / S. Polozhaenko et al. *IEEE 3rd International Conference on System Analysis & Intelligent Computing (SAIC)*. Kyiv, Ukraine, 2022. P. 1–5. DOI: 10.1109/SAIC57818.2022.9923008. Видання проіндексовано у базі даних *Scopus*.

<https://ieeexplore.ieee.org/xpl/conhome/9922952/proceeding?isnumber=9922911&pageNumber=2>

7. Крикун В. А., Фомін О. О. Інтерпретація динамічних моделей у вигляді нейронних мереж з часовими затримками. *Матеріали Дванадцятій Міжнародної наукової конференції студентів та молодих учених «Сучасні інформаційні технології – 2022»*. Одеса, Україна, 2022. С. 140-141.

<http://dspace.opu.ua/jspui/handle/123456789/12912>

8. Use of dynamic neural networks for modeling nonlinear objects with significant nonlinearity / S. Polozhaenko et al. *Збірник тез IV Міжнародної науково-практичної Інтернет-конференції "Математика та інформатика в науці й освіті: виклики сучасності"*. Вінниця, Україна, 2023. С. 121-124.

<https://press.vntu.edu.ua/index.php/vntu/catalog/book/791>

9. Крикун В. А., Фомін О. О. Нелінійне моделювання об'єктів на основі динамічних нейронних мереж. *Матеріали Тринадцятій Міжнародної наукової конференції студентів та молодих учених «Сучасні інформаційні технології – 2023»*. Одеса, Україна, 2023. С. 153-155.

<http://dspace.opu.ua/jspui/handle/123456789/14172>

10. Use of Dynamic Neural Networks for Modeling Nonlinear Objects with Significant Nonlinearity / O. Fomin et al. *18th Conference on Computer Science and Intelligence Systems*. Warsaw, Poland, 2023. P. 97–102. DOI: 10.15439/2023F3874.

<https://annals-csis.org/proceedings/2023/communication.html>

11. Modeling of the agricultural crops development using satellite imagery / O. Fomin et al. *Біоінтенсивні та SMART-технології у тваринництві: матеріали II Міжнародної науково-практичної конференції науково-педагогічних працівників та молодих науковців*. Одеса, Україна, 2023. С. 10–14.

<https://osau.edu.ua/wp-content/uploads/2023/11/TEZY-2023-II-Mizhn-konf-NNIBtaA-2906.pdf>

12. Крикун В. А. Математична модель оцінки якості програмного забезпечення інтерпретації моделей машинного навчання. *Комп'ютеризовані системи та програмні технології*. 2023. № 1. С. 7–11.

https://ccs.od.ua/wp-content/uploads/journal/cspt_1_2023.pdf

Додаток Г

Відомості про апробацію результатів дисертації

1. International Conference on Smart Technologies in Urban Engineering (STUE). Kharkiv, Ukraine, 2022. *Публікація тез та доповідь.*

<https://link.springer.com/book/10.1007/978-3-031-20141-7?page=2>

1. IEEE 3rd International Conference on System Analysis & Intelligent Computing (SAIC). Kyiv, Ukraine, 2022. *Публікація тез та доповідь.*

<https://ieeexplore.ieee.org/xpl/conhome/9922952/proceeding?isnumber=9922911&pageNumber=2>

2. Дванадцята Міжнародна наукова конференція студентів та молодих учених «Сучасні інформаційні технології – 2022». Одеса, Україна, 2022. *Публікація тез та доповідь.*

<http://dspace.opu.ua/jspui/handle/123456789/12912>

3. IV Міжнародна науково-практична Інтернет-конференція "Математика та інформатика в науці й освіті: виклики сучасності". Вінниця, Україна, 2023. *Публікація тез та доповідь.*

<https://press.vntu.edu.ua/index.php/vntu/catalog/book/791>

4. Тринадцята Міжнародна наукова конференція студентів та молодих учених «Сучасні інформаційні технології – 2023». Одеса, Україна, 2023. *Публікація тез та доповідь.*

<http://dspace.opu.ua/jspui/handle/123456789/14172>

5. 18th Conference on Computer Science and Intelligence Systems. Warsaw, Poland, 2023. *Публікація тез та доповідь.*

<https://annals-csis.org/proceedings/2023/communication.html>

6. II Міжнародна науково-практична конференція науково-педагогічних працівників та молодих науковців «Біоінтенсивні та SMART-технології у тваринництві». Одеса, Україна, 2023. *Публікація тез та доповідь.*

<https://osau.edu.ua/wp-content/uploads/2023/11/TEZY-2023-II-Mizhn-konf-NNIBtaA-2906.pdf>

7. Молодіжний науковий семінар кафедри комп'ютеризованих систем і програмних технологій для аспірантів, магістрів і студентів «Інтелектуальні інформаційні технології в системах контролю та управління». Одеса, Україна, 2023. *Доповідь*.

<https://ccs.od.ua/science/seminar/>

Додаток Д

Документи про впровадження результатів дисертації

**DISKRET**
НАУКОВО-ВИРОБНИЧЕ ОБ'ЄДНАННЯТел. (48) 7001225, Сайт: www.diskret.com.ua, E-mail: info@diskret.com.ua

АКТ

впровадження результатів дисертаційного дослідження на здобуття наукового ступеню доктора філософії за спеціальністю 121 Інженерія програмного забезпечення «Метод та програмні засоби інтерпретації моделей машинного навчання нелінійних динамічних об'єктів» здобувача Крикуна Валентина Андрійовича у науково-практичну діяльність ТОВ НВО «Діскрет»

Ми, що нижче підписалися, представник ТОВ НВО «Діскрет» директор Стаценко Г.С. з одного боку, та представники кафедри Комп'ютеризованих систем та програмних технологій (КСПТ) Національного університету «Одеська політехніка» Міністерства освіти і науки України: завідувач кафедри, д.т.н., проф. Положаєнко С.А., к.т.н., доц. Сперанський В.О., з другого боку, склали цей акт у підтвердження впровадження у науково-практичні розробки ТОВ НВО «Діскрет» результатів теоретичних та експериментальних досліджень, програмних засобів і результатів комп'ютерного моделювання, які містяться в дисертації здобувача Крикуна В.А., а саме:

1. Інформаційні моделі технологічних процесів ваговимірювальної техніки при зважуванні різних типів великовантажної автомобільної техніки у вигляді нейронних мереж з часовими затримками та сурогатні моделі, збудовані на основі інформаційних моделей для скорочення їх розмірності та спрощення їх реалізації у вигляді програмного забезпечення ваговимірювальної техніки, систем обліку та автоматизації.

2. Програмне забезпечення для комп'ютерного моделювання технологічних процесів різних видів облікових систем та ваговимірювальної техніки в системах промислової автоматизації та безпеки власного виробництва, що дозволяє визначати відхилення експлуатаційних параметрів та характеристик в устаткуванні та виконавчих механізмах, що знаходиться у функціональному режимі.

3. Методика проведення експериментальних досліджень «вхід-вихід» ваговимірювальної техніки в системах промислової автоматизації, підготовки



DISKRET

НАУКОВО-ВИРОБНИЧЕ ОБ'ЄДНАННЯ

Тел. (48) 7001225, Сайт: www.diskret.com.ua, E-mail: info@diskret.com.ua

експериментальних даних та побудови нейромережових моделей технологічних процесів ваговимірювальної техніки в системах промислової автоматизації.

Перелічені розробки використовуються ТОВ НВО «Діскрет» для розробки іновативних методів зважування великовантажної техніки та устаткування у функціональному режимі (під час руху) та їх впровадження при експлуатації ваговимірювальної техніки, систем обліку та автоматизації.

В процесі впровадження прикладних розробок, які містяться в дисертаційній роботі здобувача Крикуна В. А. з 06.11.2023 по 20.11.2023 встановлено, що надані програмні засоби комп'ютерного моделювання дозволяють підвищити точність зважування великовантажної автомобільної техніки у русі до 8% у порівнянні з діючим алгоритмом на основі безпосередньої обробки сигналу ваговимірювального устаткування.

Від ТОВ НВО «Діскрет»:

Директор


Гліб Стаценко

.. 20 .. 12 2023 р.

Адреса: 65003, Одеса,
вул. Балківська, 130
ТОВ НВО «Діскрет»
Тел. (8-048) 733-6860
Генеральний Директор
Г.С.Стаценко

Від Національного університету
«Одеська політехніка»:

Завідувач кафедри КСПТ, д.т.н., проф.
Сергій Положаско

Доцент кафедри КСПТ, к.т.н., доц.
Віктор Сперанський

“12” 21 2023 р.

Адреса: 65044, Одеса,
просп. Шевченко 1,
Національний університет
«Одеська політехніка».
Тел. (8-048) 705-8436

«ЗАТВЕРДЖУЮ»

Перший проректор, проректор з науково-педагогічної та виховної роботи Національного університету «Одеська політехніка»,
професор Сергій НЕСТЕРЕНКО



« 12 » грудня 2023 р.

ДОВІДКА

про впровадження результатів дисертації доктора філософії

КРИКУНА Валентина Андрійовича

**«Метод та програмні засоби інтерпретації моделей машинного навчання
нелінійних динамічних об'єктів»**

у навчальному процесі Національного університету «Одеська політехніка»

Чинна довідка видана в тому, що в навчальних курсах магістерської підготовки «Інтелектуальні системи» та «Математичне моделювання нелінійних динамічних процесів», які читаються студентам за спеціальністю 151 – «Автоматизація та комп'ютерно-інтегровані технології» на кафедрі комп'ютеризованих систем та програмних технологій Інституту комп'ютерних систем Національного університету «Одеська політехніка», використовуються наукові результати, одержані в дисертаційній роботі Крикуна В.А.

1. Суругатні моделі у вигляді непараметричної динамічної моделі на основі інтегродстепеневих поліномів Вольтерри для нейронних мереж з часовими затримками:

«Нейронні мережі для моделювання нелінійної динаміки» — дисципліни «Інтелектуальні системи»;

«Нелінійні динамічні моделі у часовій області. Огляд та класифікація» — дисципліни «Математичне моделювання нелінійних динамічних

процесів».

2. Метод інтерпретації нейронних мереж з часовими затримками у вигляді непараметричних динамічних моделей на основі інтегродіференціальних поліномів Вольтерри:

«Методи інтерпретації нейронних мереж» — дисципліни «Інтелектуальні системи».

3. При створенні лабораторного практикуму по ідентифікації нелінійних динамічних об'єктів за дисципліною «Математичне моделювання нелінійних динамічних процесів», використовуються запропоновані в дисертаційній роботі Крикуна В.А. програмні модулі, що реалізують розроблені обчислювальні алгоритми ідентифікації моделей нелінійних динамічних об'єктів.

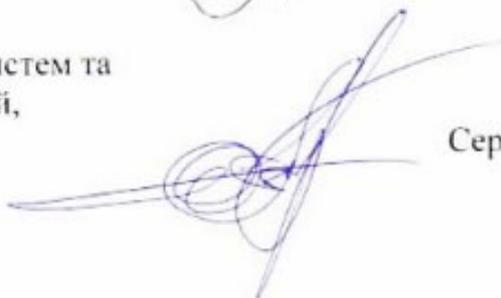
4. Навчальний процес підтримано підготовленими Крикуном В.А. розділами конспектів лекцій та методичними вказівками до виконання лабораторних робіт із зазначених дисциплін та курсової роботи з «Інтелектуальних систем».

Директор
Інституту комп'ютерних
систем, д.т.н., професор



Світлана Антощук

Завідувач кафедри
Комп'ютеризованих систем та
Програмних технологій,
д.т.н., професор



Сергій Положасенко