

Міністерство освіти і науки України
Державний ВНЗ «Національний гірничий університет»

Факультет інформаційних технологій
(факультет)

Кафедра програмного забезпечення комп'ютерних систем
(повна назва)

ПОЯСНЮВАЛЬНА ЗАПИСКА
дипломної роботи

магістра
(назва освітньо-кваліфікаційного рівня)

галузь знань *12 Інформаційні технології*
(шифр і назва галузі знань)

спеціальність *121 Інженерія програмного забезпечення*
(код і назва спеціальності)

спеціалізація *Програмне забезпечення систем*
(код і назва спеціалізації)

освітній рівень *Магістр*
(назва освітнього рівня)

кваліфікація *інженер-програміст*
(назва кваліфікації)

на тему: *Обґрунтування ефективності моніторингу гірничих комплексів на основі нейронних мереж*

Виконавець:

студент 2 курсу, групи 121М-16-1

(підпис)

Хмура О.А.

(прізвище та ініціали)

Керівники	Посада, прізвище, ініціали	Оцінка	Підпис
проекту	<i>д.т.н., проф. Мещеряков Л. І.</i>		
розділів:			
Спеціальний	<i>д.т.н., проф. Мещеряков Л. І.</i>		
Економічний	<i>к.е.н., доц. Касьяненко Л. В.</i>		

Рецензент			
-----------	--	--	--

Нормоконтроль	<i>к.т.н., доц. Коротенко Л.М.</i>		
---------------	------------------------------------	--	--

Дніпро
2018

**Міністерство освіти і науки України
Державний вищий навчальний заклад
«Національний гірничий університет»**

ЗАТВЕРДЖЕНО:
завідувач кафедри

програми забезпечення комп'ютерних систем
_____ (повна назва)

_____ (підпис) І.М. Удовик (прізвище, ініціали)

« » _____ 20 ____ року

ЗАВДАННЯ

на виконання кваліфікаційної роботи магістра

спеціальності _____ 121 Інженерія програмного забезпечення
(код і назва спеціальності)

студенту _____ 121М-16-1 _____ Хмурі Олексію Андрійовичу
(група) (прізвище та ініціали)

Тема дипломної роботи _____ Обґрунтування ефективності моніторингу
гірничих комплексів на основі нейронних мереж

1 ПІДСТАВИ ДЛЯ ПРОВЕДЕННЯ РОБОТИ

Наказ ректора Державного ВНЗ «НГУ» від 26. 12 .2017 р. № 2127 -л

2 МЕТА ТА ВИХІДНІ ДАНІ ДЛЯ ПРОВЕДЕННЯ РОБІТ

Об'єкт досліджень – технологічний стан шахтної підйомної установки

Предмет досліджень – штучні нейронні мережі та метод групового урахування аргументів

Мета НДР – підтвердження ефективності використання методу групового урахування аргументів та нейронних мереж для прогнозування аварійних ситуацій на шахтних підйомних установках

Вихідні дані для проведення роботи: експериментальні сигнали у вигляді графіків

3 ОЧІКУВАНІ НАУКОВІ РЕЗУЛЬТАТИ

Наукова новизна полягає у проведенні аналізу та виявленні недоліків традиційного підходу до розробки систем захисту та у створенні нової системи

попередження аварійних ситуацій на основі поєднання нейронних мереж та методу групового урахування аргументів.

Практична цінність результатів полягає у: розробленні нейронних мереж для створення системи, що може попереджувати можливі аварійні ситуації та поламаки.

4 ВИМОГИ ДО РЕЗУЛЬТАТІВ ВИКОНАННЯ РОБОТИ

Програмний додаток дослідження прогнозування станів гірничих підйомних комплексів

5 ЕТАПИ ВИКОНАННЯ РОБІТ

Найменування етапів робіт	Строки виконання робіт (початок – кінець)
Формування та формалізація завдань магістерської роботи	21.04.2017-01.05.2017
Аналіз сфери застосування шахтних підйомних установок	18.08.2017-21.09.2017
Проектування та розробка програмного продукту, що вирішує поставлені в дипломній роботі задачі	24.09.2017-04.10.2017
Тестування розробленого програмного продукту на експериментальних даних і аналіз отриманих результатів	06.10.2017-20.11.2017

6 РЕАЛІЗАЦІЯ РЕЗУЛЬТАТІВ ТА ЕФЕКТИВНІСТЬ

Економічний ефект від реалізації результатів роботи очікується позитивним завдяки попередженню аварійних ситуацій з високою вірогідністю удосконалення процесу дослідження стану гірничих машин.

Соціальний ефект від реалізації результатів роботи очікується позитивним завдяки створенню програмного продукту, що робить умови праці на гірничих комплексах більш безпечними та захищеними.

7 ДОДАТКОВІ ВИМОГИ

Завдання видав

_____ (підпис)

Мещеряков Л.І.

_____ (прізвище, ініціали)

Завдання прийняв до виконання

_____ (підпис)

Хмура О.А.

_____ (прізвище, ініціали)

Дата видачі завдання 12.03.2017р.

Термін подання дипломного проекту до ДЕК 19.01.2018

Реферат

Пояснювальна записка: 135 с., 32 рис., 3 додатків., 72 джерела.

Об'єкт дослідження: є шахтні підйомні установки.

Мета магістерської роботи: Обґрунтування ефективності моніторингу гірничих комплексів на основі поєднання нейронних мереж та методу групового урахування аргументів.

Методи дослідження. При рішенні поставленої задачі використовувалися наукові досягнення в областях аналізу даних, моделювання штучних нейронних мереж, Data Science.

Наукова новизна результатів, що очікуються, полягає у проведенні аналізу та виявленні недоліків традиційного підходу до розробки систем захисту від аварійних ситуацій.

Практична цінність результатів полягає у розробленні методик для створення, розгортання та масштабування систем прогнозування аварій в часових рядах.

Область застосування. Розроблена методика може застосовуватися на гірничо-промислових підприємствах.

Значення роботи та висновки. Розроблена система дозволяє проектувати системи які можуть попереджувати загрозу, що підтверджується розробленим програмним продуктом в даній магістерській роботі.

Прогнози щодо розвитку досліджень. На основі розробленого проекту можна створити системи захисту та прогнозування поламок не тільки підйомних машин, а й усього шахтного устаткування взагалі, що значно підвищить безпеку праці в галузі.

У розділі «Економіка» проведені розрахунки трудомісткості розробки програмного забезпечення, витрат на створення ПЗ й тривалості його розробки, а також провести маркетингові дослідження ринку збуту створеного програмного продукту.

Список ключових слів: нейронні мережі, прогнозування, метод групового урахування аргументів, шахтні підйомні установки, аварії.

The abstract

Explanatory note: 135 p., 32 fig., 3 applications, 72 sources.

Object of research: is mine hoisting installations.

The purpose of the degree project: Rationale for the effectiveness of monitoring of mountain complexes based on neural networks.

Methods of research. When solving this problem, scientific achievements were used in the fields of data analysis, simulation of artificial neural networks, Data Science.

The scientific novelty is expected to analyze and identify shortcomings in the traditional approach to the development of emergency protection systems.

The practical value of work is to develop techniques for creating, deploying and scaling systems for forecasting accidents in time series.

The scope. The developed technique can be applied at mining and industrial enterprises.

The value of the work and conclusions. The developed system allows you to design systems that can warn the threat is confirmed by the developed software product in this master's work.

Projections on development research. On the basis of the developed project it is possible to create systems of protection and forecasting of breakdowns not only of lifting machines, but also of all mine equipment in general, which will significantly improve labor safety in the industry.

In section "Economics" calculated the complexity of software development, the cost of creating the software and the duration of its development, and marketing studies market created by the software.

List of keywords: neural networks, forecasting, method of group consideration of arguments, mine lifting installations, accidents.

Реферат

Пояснительная записка: 135 с., 32 рис., 3 приложений., 72 источника.

Объект исследования: есть шахтные подъемные установки.

Цель магистерской работы: Обоснование эффективности мониторинга горных комплексов на основе нейронных сетей.

Методы исследования. При решении поставленной задачи использовались научные достижения в областях анализа данных, моделирования искусственных нейронных сетей, Data Science.

Научная новизна результатов, ожидается, заключается в проведении анализа и выявлении недостатков традиционного подхода к разработке систем защиты от аварийных ситуаций.

Практическая ценность исследования заключается в разработке методик для создания, развертывания и масштабирования систем прогнозирования аварий во временных рядах.

Область применения. Разработанная методика может применяться на горно-промышленных предприятиях.

Значение работы и выводы. Разработанная система позволяет проектировать системы которые могут предупреждать угрозу подтверждается разработанным программным продуктом в данной магистерской работе.

Прогнозы по развитию исследований. На основе разработанного проекта можно создать системы защиты и прогнозирования поломок не только подъемных машин, но и всего шахтного оборудования вообще, что значительно повысит безопасность труда в отрасли.

В разделе «Экономика» проведены расчеты трудоемкости разработки программного обеспечения, затрат на создание ПО и длительности его разработки, а также провести маркетинговые исследования рынка сбыта созданного программного продукта.

Список ключевых слов: нейронные сети, прогнозирование, метод группового учета аргументов, шахтные подъемные установки, аварии.

ЗМІСТ

Перелік скорочень.....	9
Вступ.....	10
РОЗДІЛ 1. ДОСЛІДЖЕННЯ ТА АНАЛІЗ АВАРІЙНИХ СИТУАЦІЙ НА ШАХТНИХ ПІДЙОМНИХ УСТАНОВКАХ.....	13
1.1. Несправності механічної частини підйомних установок	13
1.2. Несправності електричної частини підйомних установок.....	25
1.2.1. Основні несправності приводних асинхронних двигунів.....	25
1.2.2. Основні несправності схем управління шахтними підйомними установками.....	26
1.3. Система контролю нормального режиму роботи і існуючі методи виявлення несправностей.....	27
1.4. Засоби і програма контролю.....	28
РОЗДІЛ 2. ПРОГНОЗУВАННЯ ЧАСОВИХ РЯДІВ.....	40
2.1. Постановка задачі.....	40
2.1.1. Регресійна постановка задачі прогнозування.....	40
2.1.2. Постановка задачі прогнозування як задачі класифікації.....	41
2.1.3. Задача безмодельного прогнозування процесів.....	42
2.2. Огляд методів.....	43
2.2.1. Регресійний аналіз.....	43
2.2.2. ARIMA.....	48
2.2.3. Метод групового урахування аргументів.....	49
2.2.4. Штучні нейронні мережі.....	51
2.3. Основні проблеми, що виникають при вирішенні задачі прогнозування	52
2.4. Моделі випадкових перешкод.....	53
2.4.1. Білий шум.....	53
2.4.2. Кольорові шуми.....	54
2.5. Первинна обробка даних.....	55

2.6. Постановка задачі.....	57
2.7. Класифікація методів.....	57
2.8. Кластеризація даних.....	59
2.9. Методика перетворення методів для врахування неоднорідності даних.....	60
РОЗДІЛ 3. ПРОГНОЗУВАННЯ ЧАСОВИХ РЯДІВ З ВИКОРИСТАННЯМ ШТУЧНИХ НЕЙРОННИХ МЕРЕЖ.....	62
3.1. Використання тільки ШНМ.....	62
3.2. Об'єднання підходів ШНМ і МГУА.....	64
3.3. Комплексування декількох методів прогнозування.....	72
3.3.1. Розбиття вихідної вибірки на підвибірки.....	73
3.3.2. Отримання моделей за допомогою перебору методів.....	74
3.4. Програмна система прогнозування часових рядів.....	76
3.4.1. Структура системи.....	76
3.4.2. Схема алгоритму роботи системи.....	78
3.5. Приклад роботи з додатком.....	79
РОЗДІЛ 4. ЕКОНОМІЧНИЙ РОЗДІЛ.....	83
4.1. Визначення трудомісткості розробки програмного забезпечення.....	83
4.2. Витрати на створення програмного забезпечення.....	86
4.3. Маркетингові дослідження ринку збуту розробленого програмного продукту.....	87
4.4. Оцінка економічної ефективності впровадження програмного забезпечення.....	89
ВИСНОВКИ.....	90
Перелік використаних джерел.....	92
Додаток А. Код програми.....	101
Додаток Б. Відгук.....	134
Додаток В. Рецензія.....	135

Перелік скорочень

ПК –	персональний комп'ютер
ОС –	операційна система
ПЗ –	програмне забезпечення
БД –	база даних
СУБД –	Системи управління базами даних
MDA –	Model Driven Architecture
UML –	Unified Modeling Language (уніфікована мова моделювання)
MOF –	MetaObject Facility (мова опису метамodelей)
CWM –	Common Warehouse Metamodel (стандарт опису інформаційних взаємодій між хранилищами даних)
PIM –	Platform Independent Model (платформено-незалежна модель)
PSM –	Platform Specified Model (платформено-залежна модель)
CASE –	Computer-Aided Software Engineering (набір інструментів та методів програмної інженерії для проектування програмного забезпечення)
BMDA –	Borland Model Driven Architecture

ВСТУП

Прогнозування є одним з найбільш складних і затребуваних завдань, що цікавлять людство. Людям завжди було цікаво знати, що ж чекає їх у майбутньому. Однак, не тільки інтерес робить це завдання настільки затребуваним – в умовах глобалізації та прискорення ритму життя, коли за долю секунди відбуваються тисячі угод, можливість як-небудь передбачити розвиток ринку хоча б на найближчий час може стати результатом мільйонних прибутків. З іншого боку, дії, засновані на помилковому прогнозі, можуть призвести до ще більших втрат. Усе це свідчить про високу актуальність задачі прогнозування.

З точки зору математики, завдання прогнозування зводиться до задачі ідентифікації об'єкта – де об'єктом є прогнозований процес. Тобто передбачається, що прогнозований процес має деяку «модель поведінки», яка взаємодіє із зовнішнім світом і, знаючи яку ми б змогли оцінити виходи моделі в майбутньому. Тут важливим є слово «оцінити» – воно спочатку попереджає, що ми можемо лише передбачити деяку оцінку майбутніх виходів моделі.

Рішення більшості прикладних проблем, пов'язаних із завданнями вибору, управління і проектування, полягає в побудові математичної моделі, в якій відображається взаємозв'язок найбільш важливих і суттєвих для розв'язуваної задачі характеристик об'єкта дослідження. В якості об'єкта дослідження може виступати, наприклад, технічний пристрій, фізичний чи технологічний процес, економічна систем, тощо. Подібні об'єкти дослідження можуть бути охарактеризовані сукупністю суттєвих властивостей, які можуть бути об'єктивно виміряні.

Рішення задач дискретної оптимізації пов'язані з труднощами принципового характеру – повний перебір точок допустимої множини, як правило, нездійсненний через занадто великий обсяг обчислювальної роботи. Існують різні евристичні прийоми скорочення перебору в задачах дискретної оптимізації. Задача дискретної оптимізації – це задача знаходження

екстремуму функції заданої на дискретній безлічі точок. Якщо область визначення функції складається з кінцевого числа точок, то задача дискретної оптимізації можна вирішити перебором всієї множини. Однак на практиці ця множина може бути дуже великою, так що методи перебору не ефективні.

Одним з підходів до вирішення такого типу задач є використання апарату штучних нейронних мереж (ШНМ). Теорія нейронних мереж є перспективним напрямком математики та інформатики і представляє цікаву область для дослідження. Багато вчених проводили теоретичні та практичні дослідження, щодо створення нейронних мереж з різною динамікою для вирішення завдань лінійної, квадратичної, нелінійної, комбінаторної оптимізації [42]. Методи, засновані на використанні штучних нейронних мереж, дозволяють значно підвищити оперативність рішення даного класу задач, забезпечуючи достатню точність результату [38]. Тому необхідно розробити моделі та алгоритми вирішення задачі про мінімальне покриття, яка є класичною задачею оптимізації на основі теорії нейронних мереж.

На сьогоднішній день розроблений дуже солідний математичний апарат для вирішення задачі прогнозування. Однак навряд чи можна назвати єдиний метод прогнозування, який варто застосовувати при вирішенні усього розмаїття можливих практичних завдань прогнозування – кожен метод найкраще підходить для вирішення деякого кола завдань, що володіють певними властивостями. Найбільш універсальними виявилися методи, що використовують елементи штучного інтелекту – інтелектуальні системи прогнозування. Найвідоміший представник цієї групи методів – штучні нейронні мережі (ШНМ), що є системою взаємозалежних штучних нейронів. Кількість завдань, для вирішення яких були успішно застосовані ШНМ, дійсно вражає, як і різноманітність цих завдань:

Завдання прогнозування, розпізнавання, управління і багато інших входять в цей список. Саме тому головним фокусом цієї роботи є ШНМ.

Метою роботи є дослідження існуючих інтелектуальних методів та систем прогнозування і розробка системи нейронних мереж для прогнозування аварійних ситуацій на шахтах.

Об'єктом дослідження є системи прогнозування для шахтних підйомних машин.

Предмет дослідження – шляхи покращення існуючих методів та систем попередження аварій.

Основні завдання роботи:

- дослідження та аналіз сучасних інтелектуальних методів прогнозування;
- розробка нових підходів до прогнозування на основі використання елементів штучного інтелекту;
- розробка програмного комплексу, що забезпечуватиме просте використання існуючих та розроблених методів для вирішення задачі прогнозування часових рядів.

Наукова новизна роботи:

- запропонована нова постановка задачі прогнозування, що враховує можливу неоднорідність даних;
- запропоновано декілька нових гібридних методів прогнозування, заснованих на ШНМ;
- запропонована методика модифікації методів прогнозування, які володіють певними властивостями, що поліпшує точність прогнозу цих методів на неоднорідних даних.

РОЗДІЛ 1.

ДОСЛІДЖЕННЯ ТА АНАЛІЗ АВАРІЙНИХ СИТУАЦІЙ НА ШАХТНИХ ПІДЙОМНИХ УСТАНОВКАХ

1.1. Несправності механічної частини підйомних установок

Шахтні підйомні посудини є складовим елементом підйомної установки. Їх працездатність і надійність залежать від ряду причин, в тому числі жорсткості і міцності конструкції підйомної посудини, стану армування стовбура і типу посадних пристроїв, розташування і типу напрямних, своєчасного виявлення та якісного усунення несправностей, що виникають в процесі експлуатації. У зв'язку з підвищенням продуктивності підйомних установок за рахунок збільшення швидкості руху посудин і корисною їх місткості вимоги до останніх неухильно підвищуються.

Найбільшого поширення отримали шахтні вертикальні одноповерхові і двоповерхові кліті, використовувані для підйому і спуску людей, підйому породи або вугілля на поверхню і спуску порожняка в шахту, спуску в шахту кріпильних і допоміжних матеріалів, спуску і підйому різного шахтного устаткування.

Кліті обладнуються парашутними і підвісними пристроями. Металоконструкції клітей поділяються на чотиристійкові безраскосне, що застосовуються в індивідуальних клітях, і з розкосами, застосовуваними в уніфікованих клітях.

Найбільшого поширення набули кліті клепаної конструкції.

Основні несправності металоконструкції кліті: ослаблення клепаних з'єднань, місцеві деформації в силових елементах каркаса, в обшивці, даху, підлоги, дверях.

Термін служби металоконструкції кліті багато в чому залежить від швидкості посадки кліті на кулаки. Зазвичай швидкість посадки кліті на кулаки змінюється від 0,1 до 0,5 м / с і залежать від технічних можливостей системи управління робочим гальмом і навичок машиніста підйому. При швидкості кліті 0,2-0,3 м/с посадка на кулаки відбувається практично без удару і наступних коливань. Динамічність навантаження враховується введення коефіцієнта

динамічності. При нормальній швидкості посадки (0,2 м/с) коефіцієнт динамічності не перевищує 1,07.

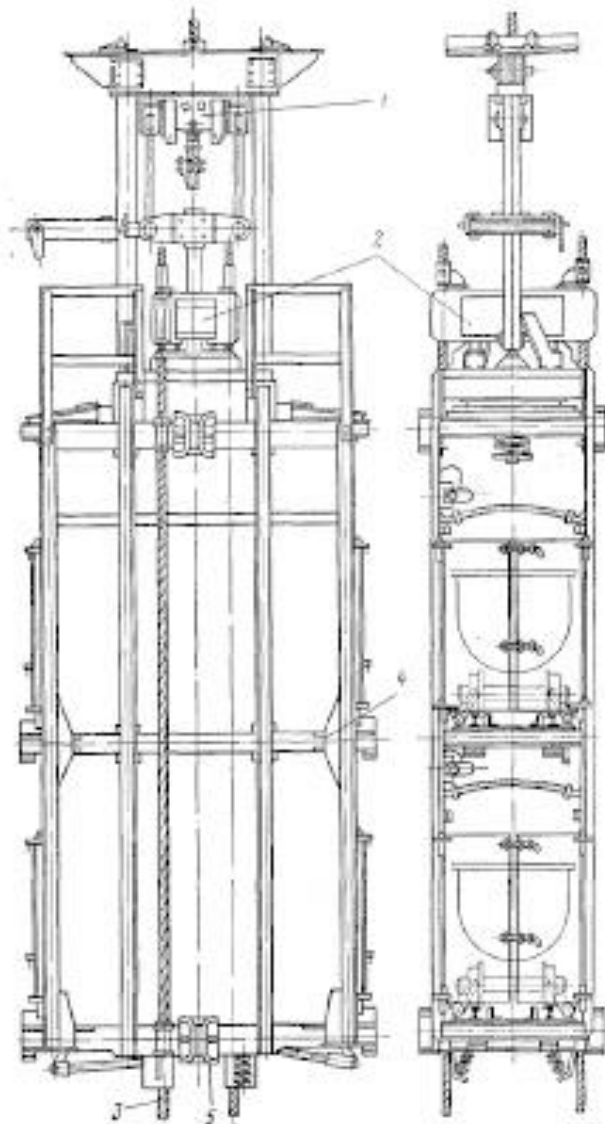


Рис. 1.1. Загальний вигляд неопрохіднимі двоповерхової кліті для рейкових провідників двостороннього розташування з парашутом ТК

Одним з відповідальних елементів, схильних до найбільш частих відмов, є Клітьові стопори, призначені для автоматичного стопорення вагонеток в кліті під час здійснення операцій з обміну на приймальних майданчиках і для надійного утримання їх під час руху кліті. З огляду на важкі умови роботи кліті, що виникають динамічні навантаження, Клітьові стопори щодня оглядаються з метою контролю їх справного стану.

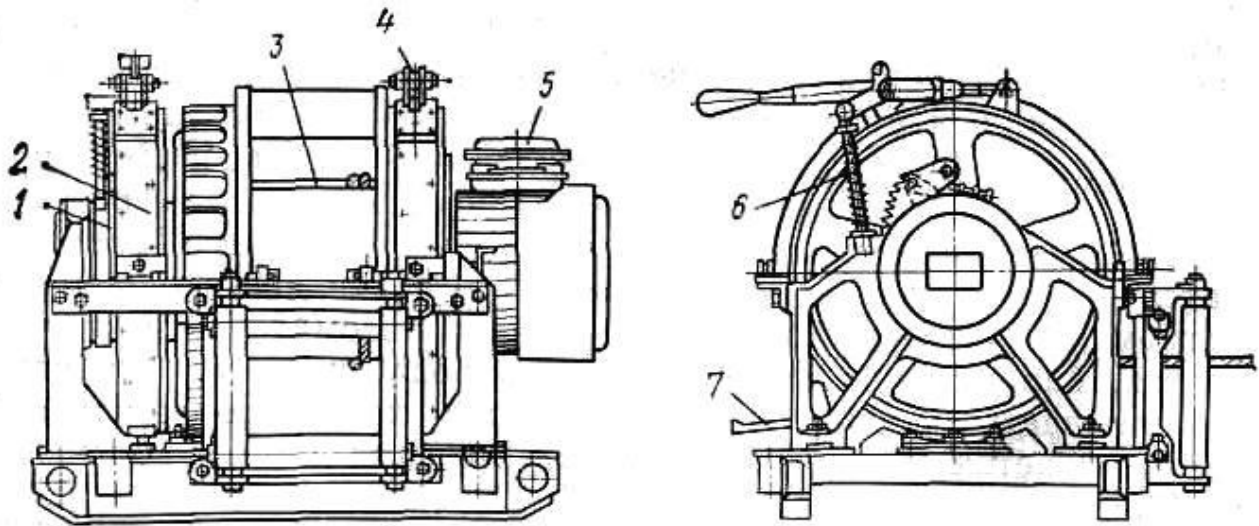


Рис. 1.2. Клітьові стопори

Найбільшого поширення на шахтах отримали два типи скіпів: з бічним розвантаженням і секторних затвором і з донною розвантаженням при отклоняющемся кузові.

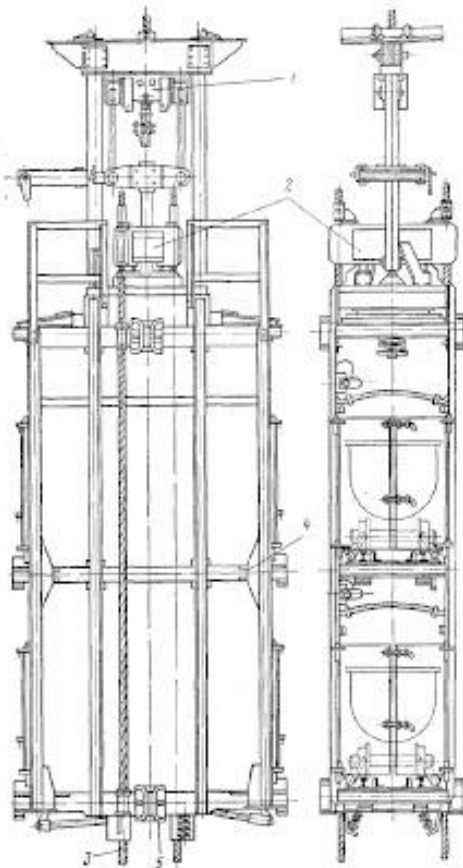


Рис. 1.3. Скіп вертикальний з нерухомим кузовом

Основною причиною обмеженого терміну служби скіпів є підвищений знос, пов'язаний з виключно важкими умовами експлуатації і значними динамічними навантаженнями, що виникають взаємодії скіпа з направляючими

провідниками тощо і вході в розвантажувальні криві. У зв'язку з цим до скіпів пред'являється ряд вимог, в тому числі підвищеної міцності і конструктивної жорсткості. До числа елементів, що відрізняються низькою надійністю, відносяться направляючі опори; футерування кузова скіпа, що має недостатню міцність; секторні затвори та ін.

Одним з найбільш небезпечних явищ при роботі підйомних установок є заклинювання скіпа в розвантажувальних кривих. Тому перевірка роботи затвора скіпа, стану розвантажувальних кривих повинна проводитися щодоби. Для попередження заклинювання необхідно: не допускати попадання в скіп сторонніх негабаритних предметів, здатних перешкодити закриттю затвора при виході скіпа з кривих; не допускати заклинювання напрямних роликів; попереджати перекіс кузова в кривих, не допускаючи великих зносів вкладишів підшипників і осі обертання. Величина сумарного зносу осей і охоплюють їх отворів не повинна перевищувати 3 мм; знос провідників в зоні розвантажувальних кривих - значень, наведених в правилах безпеки у вугільних і сланцевих шахтах; систематично міняти і очищати смуги розвантажувальних кривих; після заміни провідників проводити перевірку руху скіпа в розвантажувальних кривих; в зимовий час систематично очищати від льоду скіпи, розвантажувальні криві і провідники; при заміні кривих перевіряти їх виставку із залученням маркшейдерської служби шахти.

Органи навивки. Барабани і канатоведучих шківів є найбільш відповідальними елементами шахтних підйомних машин, що сприймають високі статичні та динамічні навантаження. Від правильно прийнятої і розрахованої конструкції барабанів і умов їх експлуатації залежить ресурс механічної частини підйомної машини.

Форма, число і розміри барабанів відповідають чинним стандартам або ТУ на підйомні машини.

Удосконалення конструкції барабанів йде по шляху підвищення надійності зчленування труби жорсткості з валом. У перших зразках машин зварна труба жорсткості прифланцовувалась болтами до посадженим на вал литим маточини.

Велика кратність нагрів ЗКІ і наявність зазору між маточиною і фланцем труби жорсткості приводили до швидкого виходу з ладу болтів. Для зменшення навантажень на болти в подальших конструкціях передбачаються установка в трубі жорсткості одній або двох додаткових опор, попередня затягування болтів, а також поліпшення технології складання, що забезпечує відсутність зазору між маточинами і фланцем труби жорсткості.

Нижче перераховані найбільш поширені несправності, привід яких до аварій, органів завивки.

Деформація обичайки. Причинами деформації обичайки є перевантаження або динамічні зусилля; значний знос футеровки, призначеної крім свого основного призначення розвантажувати оболонку барабана.

Скрипіння барабана. Причинами скрипіння барабанів є ослаблення заклепок, розхитування швів, знос або ослаблення кріплення болтів лобовини до маточини.

Тріщини в обечайке. Тріщини і розриви оболонки барабана і елементів жорсткості можуть відбуватися в наслідок деформації через слабшання болтових з'єднань рознімних частин барабана. Найменш надійними є окремі елементи металоконструкції барабанів типу БЦК, у яких з'являються тріщини в місцях концентрації напружень, перепаду жорсткостей і ін.

Стук в барабані під час роботи, підвищений осьовий розбіг. Причинами зазначених явищ є ослаблення шпонок, знос вкладишів переставної барабана.

Скрипіння і нерівномірний знос футеровки. Причинами нерівномірного зносу футеровки є ослаблення кріплень футерування, різну напругу підйомних канатів.

Зісковзування каната на барабанах БЦКБ. Причиною зісковзування каната на барабанах є незадовільний стан жолобів.

Знос незакріпленої маточини. У більшості заклиненому барабанів тільки одна лобовина жорстко з'єднана з валом. Друга, як правило, посаджена по ковзній посадці, забезпечена мастильним пристроєм, а іноді втулкою ковзання. При цьому крутний момент барабана передається тільки через обечайку.

Передача через обечайку знакозмінної навантаження, до того ж змінюється за величиною, призводить до зносу незакріпленої маточини.

Гальмівний обід. Від технічного стану гальмівного обода багато в чому залежать надійність роботи гальма, довговічність гальмівних колодок. Робоча поверхня гальмівних колодок повинна бути чистою, гладкою.

В результаті експлуатації на гальмівному ободі можуть з'явитися подряпини, шорсткості, тріщини в місцях з'єднання двох половин обода. При наявності подряпин і шорсткостей обід шліфується з доведенням чистоти поверхні до $R_a = 0,5$. При виявленні тріщин в зварних з'єднаннях половин гальмівного обода зварений шов за місцем тріщини видаляють і знову заварюють.

При появі еліпсності гальмівний обід проточується і потім шліфується.

Еліпсність обода або місцеві опуклості негативно оповіди ються на роботі гальмівної системи. При проходженні опуклою частини обода під гальмівний колодкою при пригальмовуванні машини збільшується гальмівний момент, потім після виходу цієї частини з-під колодки гальмівний момент зменшується. Це вносить труднощі в роботу машиніста. Крім того еліпсність обода викликає підвищені зазори між гальмівними колодками і ободом, що збільшує тривалість холостого ходу і час спрацьовування гальма, а також величини ходу поршнів гальмівних циліндрів.

Биття гальмівного обода вимірюється індикатором. Якщо биття гальмівного обода знову змонтованої машини перевищує допустимі значення, проводиться його проточка (табл. 1.1).

Таблиця 1.1. Допустимі значення биття обода барабана

Діаметр гальмівного обода, мм	Допустима величина биття обода барабана, мм	
	Заклиненного	Переставного
2000 до 2500	0,35	0,50
2500 до 3000	0,40	0,60
3000 до 4000	0,45	0,70
4000 до 5000	0,50	0,80

На машинах, які перебувають в експлуатації, допускаються великі величини биття, які можуть становити: до 1,2 мм - для барабанних машин з діаметром барабана понад 3,5 м і паралельним переміщенням гальмівних колодок, до 0,8 мм - для інших підйомних машин. При цьому повинен забезпечуватися достатній запас ходу поршня кожного гальмівного циліндра або демпфера при затормаживанні, а сумарний зазор між гальмівним ободом і колодками не повинен перевищувати 4 мм. Якщо ця умова не виконується, необхідно проточити і прошлифувати гальмівний обід.

Величина уступів в стиках різних гальмівних ободів не повинна перевищувати 0,1 мм. При великих зсувах слід проточити виступаючу кромку стику урівень або округляти виступ до радіуса не менше 100 мм. При значному зносі гальмівного обода останній повинен бути посилений спеціальними накладками або замінений новим. Мінімально допустима товщина гальмівного обода повинна бути узгоджена з заводом-виробником.

Неприпустимо попадання масла або води на робочу поверхню гальмівних ободів. Дослідженнями встановлено залежність коефіцієнта тертя від зволоження гальмівного поля (табл. 2)

Не допускається надмірне нагрівання гальмівного обода. Його температура не повинна перевищувати 70 ° С. При надмірному нагріванні гальмівного обода і прес-масових колодок також знижується коефіцієнт тертя між ними. З цієї причини мали місце аварії.

Таблиця 1.2. Коефіцієнт тертя гальмівного обода

Число грузів	Коефіцієнт тертя			
	При запобіжному гальмуванні		При робочому гальмуванні	
	сухі колодки	зволожений обід	сухі колодки	зволожений обід
12	0,305	0,13	0,366	0,17
10	0,308	0,129	0,366	0,186
8	0,318	0,13	0,4	0,176
6	0,312	0,13	0,43	0,182

Надмірний нагрів гальмівних ободів може бути викликаний поганою підгонкою гальмівних колодок, неправильною регулюванням виконавчого органу гальма, неповним оттормаживанні машини в слідстві неправильному регулюванні гальмівного приводу, а також опуклістю обода при місцевому нагріванні. Перераховані дефекти повинні бути усунені.

Якщо надмірний нагрів гальмівних ободів викликаний інтенсивним участю гальма в робочих режимах, рекомендується для зниження участі гальма користуватися електродинамічним гальмуванням.

Характерною несправністю гальмівних колодок, особливо дерев'яних, на машинах НКМЗ є їх вібрація при затормаживании машини. Вібрація може бути викликана нерівномірним прилеганием колодок до обода в слідстві їх неточною підгонки. Однак і при точній підгонці колодок на машинах НКМЗ при дерев'яних колодках можлива їх вібрація. Це пояснюється тим, що при високому коефіцієнті тертя колодок об обід і великому куті охоплення гальмівного обода колодками можливо явище самозахвативанія колодок гальмівних ободом. Якщо вібрацію дерев'яних колодок не вдається усунути ніякими заходами, рекомендується замінити їх прес-масовими, мають дещо менший коефіцієнт тертя.

Основні несправності вантажного приводу гальма. Різке гальмування, коливання системи важелів при гальмуванні. Трапляється через велику

перепускного отвору в обвідний требе при повітряному демпфері, що не заглушені отвори в поршні масляного демпфера, відсутність масла в демпфері або недостатність його рівня, низька в'язкість масла.

Великий час холостого ходу гальма, повільне падіння вантажу. Причини: висока в'язкість масла, кількість масла, залитого в демпфер, більше допустимого, чи не забезпечується необхідний запас ходу поршня до дна демпфера. Причини: неправильне вихідне стан демпфера після монтажу. Демпфер повинен бути встановлений таким чином, щоб при нормальному робочому ході поршня запас його до дна циліндра був не менше 100 мм, а для гальмівних колодок з електрогідравлічним вантажним приводом - не менше 50 мм.

Відмова гальмівної колодки, вантажі не падають. Причини: робота гальма з не повністю піднятим вантажем, відсутня або неправильно мають позначку, проти якої встановлюється палець при нормальному верхньому положенні вантажу; не працює блокування від неповного підйому гальмівних вантажів; неправильний вибір положення кулачка скидає важеля, регулюється зміною довжини тяги і перестановкою пальця, що з'єднує важіль з тягою; неправильна виставка гальмівний колонки, зміщення, перекис.

Мимовільне спрацьовування гальм через незадовільний стан захоплень і головки.

Несправності електромагніту: підвищений шум якоря, швидке нагрівання котушки. Несправність має місце після монтажу або заміни магніту і неправильного зустрічного увімкнення я котушок, при перекосі якоря, про слаблені елементів кріплення; якір магніти не відпадає. Звісно вантажу через заїдання важеля механізму гальма; затримка спрацьовування магніту постійного струму - знос немагнітної прокладки або зминання стику муздраттеатру.

Несправності гідравлічного приводу гальма:

1. При оттормажіванії машини спостерігається вібрація поршня вантажем акумулятора тиску - в циліндр акумулятора потрапило повітря.

2. Вібрація плунжера приводу гальма, гальмівних вантажів і стрілки манометра - потрапляння повітря в гальмівний циліндр приводу гальма.

3. Неповна або повільне розгальмовування і загальмування - неповне відкривання отвору золотником регулятора тиску.

4. Неправильно відрегульована величина робочого ходу золотника триходового крана. Максимальний робочий хід золотника без впливу зворотного зв'язку повинен бути не більше 34 мм. Якщо хід золотника без впливу зворотного зв'язку більш 30-34 мм, а зміни довжини тяг або перестановки на інший отвір у важелі рукоятки не дає необхідної величини ходу, то необхідно змінити довжину плеча важеля, висвердливши додатковий отвір. Відстань від осі рукоятки до нового отвору визначається за формулою:

$$l = (30/40) * L/h_{зол},$$

де L - відстань від існуючого отвору до осі рукоятки, мм;

$h_{зол}$ - фактичний хід золотника без впливу зворотного зв'язку, мм.

5. Повільна робоче та аварійне гальмування - велика в'язкість масла в акумуляторі тиску.

6. Неповне відкривання отворів кранів управління.

7. Повільно опускається якір електромагніту - закрито дросельний отвір в демпфері електромагніту.

8. Різке запобіжне гальмування на похилих підйомних установках - велике дросельний отвір в дросельної шайби.

9. Недостатня плавність роботи робочого гальма - сильно затягнута пружина і немає необхідного зазору.

10. Витоку масла з гальмівної системи - порушення ущільнень.

11. Неприпустимі коливання тиску масла в нагнітальному трубопроводі - дуже щільна або задерта манжета, пошкодження дотичних поверхонь поршня і циліндра, перекіс або відхилення від вертикалі осей циліндра і поршні і т.п.

12. Зниження подачі масла насосів гальмівної системи. Вібрація і надмірний шум під час роботи.

13. Несправність маслофільтра - засміченість фільтра.

14. Несправності приводного циліндра.

15. Незадовільний кріплення гальмівних вантажів. Відсутність необхідних запасів до ґрунту - кріплення вантажів здійснюється з відхиленнями від вимог заводу-виготовлювача.

16. Несправності триходового крана - туге переміщення золотника; викривлення штока, що викликає заїдання в прохідній кришці крана.

17. Несправності чотириходового крана аналогічні несправностей триходового.

Основні несправності грузопневматического приводу гальма:

1. Неодновременное рух поршнів циліндрів робочого і запобіжного гальм - неправильна затяжка сальників на робочих циліндрах.

2. Поршень робочого гальма стукає об дно циліндра - неправильно відрегульована гальмівна система. При розгальмованій гальмівні балки повинні стикнутися зі своїми упорами-обмежувачами до того, як поршень досягне дна циліндра.

3. Удари при установці поршня в верхньому положенні. Руйнування фундаменту під гальмівний майданчиком.

4. Поршень заклинює в верхньому положенні - значна корозія робочих поверхонь поршня і циліндра; зміщення осей поршня або циліндра і кришок, відсутність мастила.

5. Знос манжет сальникової набивки.

6. Несправність стопорення різьбових з'єднань приводу гальма машини НКМЗ.

7. Надмірний тиск стисненого повітря в повітрозбірнику.

Несправності елементів і схем управління гальмом:

1. Відсутня плавність регулювання - надмірно туга посадка золотника або його забруднення.

2. Мимовільне гальмування машини. Значні витоку повітря через глушник - знос золотника або втулки, наявність вибоїн або подряпин на їх робочих поверхнях.

3. Витоку повітря через пробку регулятора - знос манжети штока.

4. Уповільнена відсічення заданого тиску повітря в циліндрі робочого гальмування або значне його збільшення - засмічення часткове або повне отворів в золотнику регулятора тиску.
5. Наявність постійного невеликого тиску в циліндрі.
6. Збільшення мертвого ходу рукоятки - вільною не затиснута пружина.
7. Поява максимального тиску при неповному ході рукоятки управління робочим гальмом - занадто жорстка пружина.
8. Вібрація важеля управління робочим гальмом, що відчувається на рукоятці управління.
9. Погіршення статичної та динамічної характеристик регулятора - залипання і перекося рухомого якоря електромагнітної головки управління.
10. Чи не відбувається повного розгальмовування підйомної машини - не повністю відкривається дросельний отвір електромагнітного управління.
11. При зміні струму в обмотці управління регулятором спостерігається запізнювання зміни тиску, гістерезис або тиск не регулюється зовсім - заїдання рухомого якоря електромагніту управління.
12. Величина тиску в робочому циліндрі значно відрізняється від встановленої для відповідного положення рукоятки робочого гальма - вийшов з ладу один з діодів випрямного моста.
13. Коливання тиску при постійному струмі управління - заїдання золотника.
14. Витоку повітря через регулятор тиску - незадовільна якість притирання якоря і поверхні прилягання електромагніту.
15. Заклинювання або заїдання золотника у втулці - деформація корпусу регулятора при обтягуванні болтів на сполучних фланцях.
16. Відсутність тиску при першій ступені запобіжного гальмування - заїдання рухомого якоря електромагніту першого ступеня.
17. Нестабільність першого ступеня - неправильне прилягання якоря електромагніту управління.
18. Не включається контактор запобіжного гальма при початковому положенні машини - обрив ланцюга харчування обмотки управління.

19. Залипання якоря електромагніту - попадання масла на активну поверхню електромагніту при виході з ладу діафрагм.

20. Витоку повітря - знос поверхні втулки і золотника.

21. Пошкодження діафрагми.

22. Неповна перестановка клапанів. Інтенсивна витік повітря - тиск в повітророзподільних мережі нижче $4 * 10^5$ Па.

1.2. Несправності електричної частини підйомних установок

1.2.1. Основні несправності приводних асинхронних двигунів

Двигун при старті не розгоняється, гуде - відсутня напруга в одній з фаз статора, переплутані початок і кінець фази, немає струму в одній з фаз ротора, ротор примагнічується до статора в слідстві нерівномірного зазору між ними, заклинений підйомна посудина в стовбурі, машина повністю не расторможен. При відсутності струму в одній з фаз ротора двигун може не розвернутися з номінальним навантаженням, але на холостому ходу або з зменшеним навантаженням він буде розвертатися нормально. В цьому випадку при роботі двигуна на природній характеристиці величина струму в кожній фазі статора буде коливатися з частотою, рівній частоті струму ротора.

Перегрів підшипників. Підвищена вібрація двигуна - двигун недостатньо жорстко закріплений на фундаментній плити або плита на фундаменті, неправильна центровка двигуна або порушення первісної центрування в слідстві опади фундаменту, підвищена невірноваженість ротора, неплотная посадка напівмуфт, небаланс муфт, торцеве їх биття, нежорсткий фундамент, незадовільна заливка фундаментної плити.

Перегрів обмотки статора - двигун перевантажений, засмічені вентиляційні канали сердечників статора, міжфазова замикання в обмотці статора, междувітковое замикання або подвійне замикання однієї фази на землю. В останніх двох випадках виникають місцеві перегриви обмотки статора, супроводжувані гудінням двигуна і нерівномірністю струму в фазах.

Низький опір ізоляції і низький коефіцієнт абсорбція - забруднення і зволоження ізоляції.

Іскріння щіток - недостатній тиск щіток на контактні кільця, забруднення або нерівності на контактних кільцях.

Перебиття контактних кілець електричною дугою - контактні кільця і щітковий апарат забруднилися пилом, маслом; щіткові дроти сусідніх фаз знаходяться на близькій відстані; порушилося кріплення щіток проводів до шин траверси; є обрив в з'єднаннях між ротором і пусковими опорами; підвищилася вологість навколишнього повітря; спостерігається старіння ізоляції втулок і струмоведучих шпильок, ушкодилася різьба шпильок.

Пошкодження сердечників статора - неправильна експлуатація, неякісне виготовлення.

Порушення паяних з'єднань обмотки ротора - невисока надійність паяних з'єднань, перегрів двигуна.

Розмотування бандажів ротора або окремих витків - низька якість виготовлення, перевищення нормальної швидкості двигуна більш ніж удвічі.

1.2.2. Основні несправності схем управління шахтними підйомними установками

В процесі експлуатації має місце вихід з ладу напівпровідникових діодів і силового трансформатора, що пов'язано в ряді випадків з відсутністю дугового блокування в схемі управління.

Для усунення зазначеного недоліку в ланцюг захисту підйомної установки включаються паралельно один одному контакти перемикача і динамічного гальмування, які в нульовому положенні замкнуті.

Застосовується також схема, що передбачає заміну блок-контактів ланцюга реверсують контакторів В і Н контактами реле контролю струму, що забезпечує включення контакторів лише після згасання дуги по постійного струму відключення реле. При цьому забезпечується чітка дугова блокування.

Зазначена схема зводиться до простого схемного зміни в межах одного блоку КДТВ і забезпечує дугову блокування в найбільш небезпечному режимі - при переході від динамічного гальмування до рухового режиму.

Напівпровідникові діоди виходять з ладу також внаслідок їх перевантаження при неправильно обраної робочої точці магнітного підсилувача та при відсутності контролю ланцюга обмотки магнітного підсилувача. Для контролю стану контакту ДТ в ланцюзі обмотки в типовою схемою здійснюється Perez'єднання контакту реле, в результаті чого досягається контроль стану контакту в ланцюзі обмотки і підвищується рівень контролю з боку реле.

Оскільки пробою силових діодів відбувається, як правило, без обриву ланцюга, вторинна обмотка трансформатора закорачивається в один з напівперіодів, що призводить до різкого збільшення струму в три і більше разів, в цьому ланцюзі.

Пульти керування шахтними підйомними установками мають механічне блокування рукоятки управління рідинним реостатом з ручкою перемикача реверсора. Перемикач його здійснюється установкою рукоятки управління в початкове положення, що відповідає повністю введеному опору рідинного реостата. Якщо підйомні установки оснащені пультами управління застарілої конструкції, схема повинна бути доповнена електричним блокуванням нульового положення рідинного реостата.

1.3. Система контролю нормального режиму роботи і існуючі методи виявлення несправностей

Поява складних автоматичних систем управління і регулювання підйомних установок вимагає підвищення рівня експлуатації. Одним з найбільш дієвих шляхів підвищення рівня експлуатації підйомних установок є підвищення ступеня їх контролю. Під контролем мається на увазі комплекс вимірювальних та логічних операцій, спрямованих на визначення ступеня працездатності та виявлення несправностей в системах підйому і її елементах.

Контроль працездатності підйомної установки має на меті встановити факт втрати працездатності та причини появи несправностей.

Методи виявлення несправностей включають в себе індикацію і пошук. У першому випадку встановлюється певна кількість датчиків, які в разі виникнення несправності забезпечують її автоматичну індикацію. Датчики можуть ставитися до контрольованої системи або контрольно-вимірювального обладнання.

1.4. Засоби і програма контролю

Датчики контролю положення. Для забезпечення нормальних режимів роботи підйомної установки необхідно контролювати стан різних її елементів, судин в стовбурі, дозаторів, зміна положення гальма при зносі колодок.

Всі ці функції виконуються за допомогою різних датчиків контролю положення, в якості яких застосовуються контактують і не контактують вимикачі і реле.

Датчики контролю положення, побудовані на принципі безпосередньої механічної дії до контролюваного об'єкта на датчик, недостатньо надійні в важких штатних умовах, вимагають систематичного догляду та ремонту. У зв'язку з цим необхідне створення датчиків положення, які працюють без механічного або електричного контакту з об'єктом. Такі датчики мають істотно більшою надійністю і безвідмовністю в роботі, дозволяють створювати апаратуру з іскробезпечними параметрами і з малим споживанням енергії.

В даний час застосовуються безконтактні екранні датчики контролю положення.

Апаратура контролю температури. Безпечна і надійна експлуатація підйомних установок неможлива без системи теплового контролю: підшипників, двигуна, стисненого повітря на виході компресора, електроліту в рідинних реостатах. У ряді випадків необхідний контроль температури повітря в системах охолодження електричних машин.

Поряд з відомими приладами візуального контролю зазначені параметри повинні контролюватися датчиками, які мають вихідні контакти для впливу на систему сигналізації і захисту.

Для контролю температури підшипників використовується температурне реле, температурний датчик і манометричні термометри.

Апаратура контролю температури призначена для захисту від перегріву обмоток електричних машин і підшипників. Датчиком температури в цій апаратурі є термістор.

Електронний автоматичний многоточечний міст призначений для автоматичного контролю температури і захисту від перегріву підшипників агрегатів, що працюють в нормальній середовищі.

Система захисту від перегріву перевіряється відповідно до інструкцій на встановлену апаратуру. Необхідно періодично знімати характеристики з теплових датчиків шляхом їх тарировки. Балони термосигналізатори повинні повністю перебувати в контрольованій середовищі.

Апаратура контролю мастила. Припинення подачі масла в систему централізованого змащування може вивести з ладу підшипник, викликати підвищений знос зубчастих передач редуктора. Тому подача масла повинна контролюватися з використанням спеціальної апаратури.

Для контролю тиску і витрати масла використовується реле тиску, електроконтактний манометр і струменеві реле.

Зазначені пристрої мають пристосування для регулювання величини установки спрацьовування і контакти, які сигналізують про досягнення контрольованих рівнянь.

Апаратура контролю ізоляції. Зниження контролю ізоляції електричних ланцюгів викликає в них коротке замикання. Подвійні однополюсні замикання на землю можуть привести до відмови елементів захисту, помилкової подачі сигналу, самовключення контакторів та ін. Для контролю ізоляції в ланцюгах змінного струму застосовуються спеціальні реле.

Пристрій контролю ізоляції призначене для відключення від мережі напругою 127 В. з ізолюваною централлю при збільшенні витоку на землю внаслідок дотику людини до струмоведучих частин, погіршення ізоляції і т.д. Пристрій спрацьовує при опорі загальної витоку мережі щонайменше 2500 Ом, а в одній фазі - не менше 2200 Ом. При появі витоку запалюється сигнальна лампа і подається імпульс на звуковий сигнал, і тільки після зупинки підйомної машини подається імпульс на відключення живлення ланцюгів сигналізації.

Пристрій контролю ізоляції ланцюгів постійного струму вип про ляється різні м і способами (рисунок 1.4).

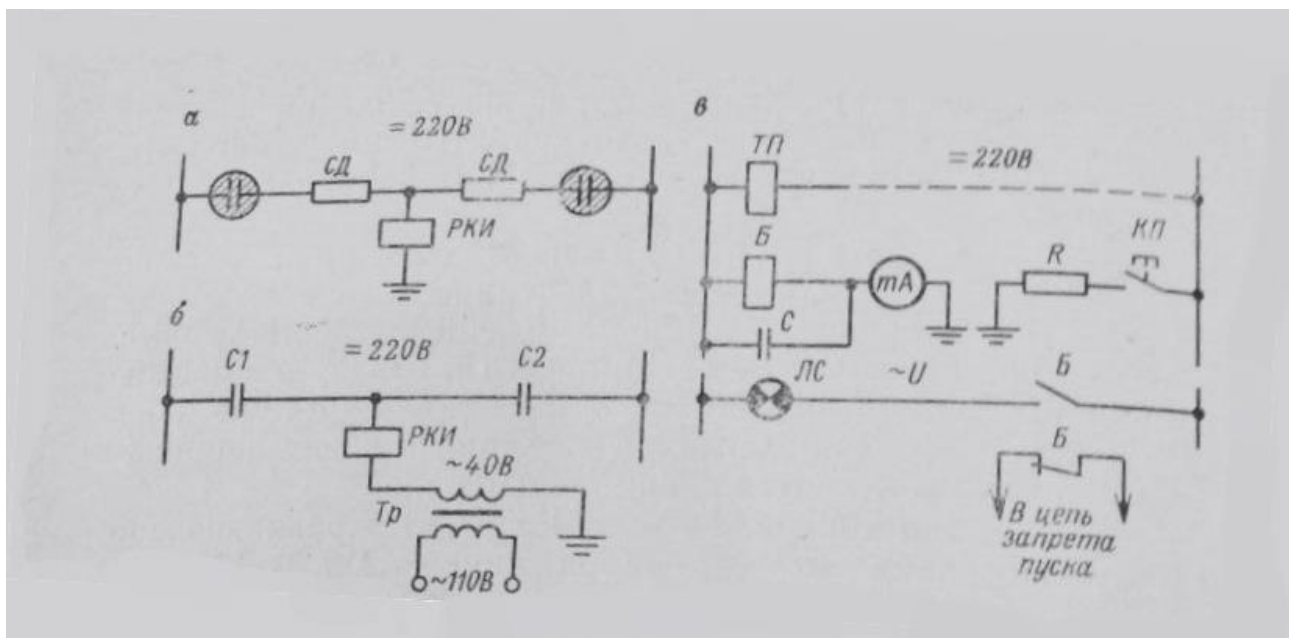


Рис. 1.4. Схеми контролю ізоляції ланцюгів постійного струму:

а - двофазна; б - з джерелом змінного струму; в - однофазна

Схеми не реагують на симетричне зниження опору ізоляції обох фаз і мають низьку чутливістю. Схема використовує автономне джерело змінного струму для контролю ізоляції ланцюгів управління.

Котушка реле підключається одним виходом на той же полюс оперативних ланцюгів постійного струму, що і контактор, а іншим виходом через міліамперметр на землю. Якщо струм витоку на землю іншого полюса оперативних ланцюгів постійного струму досягне 0,005 А, то реле включиться і

випаде його прапорець, який сигналізує про погіршення стану ізоляції. Контакти реле включаються відповідно в ланцюзі сигнальної лампочки і заборони пуску підйомної машини.

Контроль цілісності кінематичних ланцюгів. В апараті АЗК за допомогою реле передбачається контроль цілісності кінематичного ланцюга від вала підйомної машини до блоку датчиків, що вказують глибини і обмежувача швидкості. Це реле забезпечує також контроль справності ланцюгів прив'язки тахогенератора до підйомної машини. У схему контролю входять два тахогенераторних датчика і транзисторне реле обертання.

В основу принципової схеми покладено здійснення логічної функції «рівнозначність», причому незалежними логічними змінними є сигнали датчиків обертання, один з яких жорстко пов'язаний з валом приводного двигуна (або редуктором підйомної машини), а інший - з вихідним валом контрольованого механізму.

Якір вихідного реле притягнутий тільки в тому випадку, якщо наявні чи відсутні одночасно обидва сигнали датчиків обертання. Наявність тільки одного сигналу свідчить про аварію, коли вихідна реле знеструмлюється.

Схема реле контролю обертання, представлена на малюнку 1.5, розроблена в Донецькому налагоджувальному управлінні, дозволила усунути ряд недоліків заводської схеми. Однією з основних причин нестійкої роботи реле є наявність двох установок спрацювання реле, що створює штучну асиметрію схеми. Незважаючи на правильну настройку реле, при якій відключення і включення реле визначає спрацювання другого реле, все ж мають місце випереджаючі відключення і включення другого реле. Це пояснюється великим розкидом параметрів включення і відключення самих реле, а також наявністю пульсацій вхідної напруги на малих швидкостях. Задовільні результати отримані при виконанні симетричною схеми включення з однаковою їх налаштуванням, при якій здійснюється перехресна зворотний зв'язок. При цьому обидва реле налаштовуються на однакову швидкість включення.

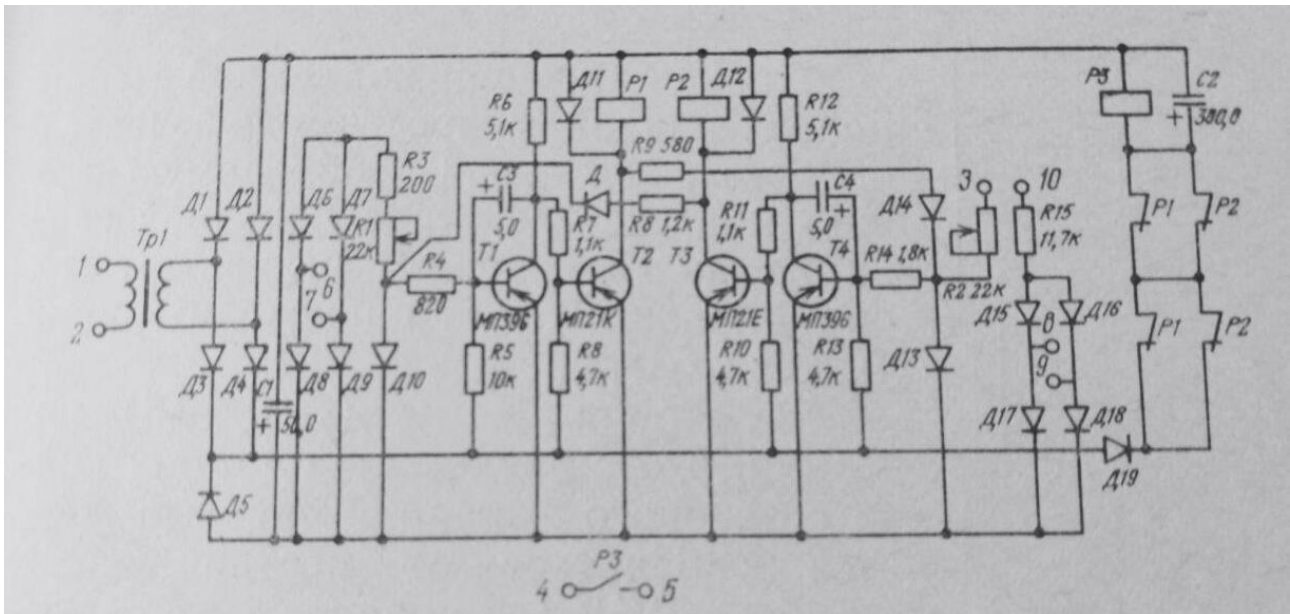


Рис. 1.5. Схема реле контролю обертання

При досягненні підйомної машини зазначеної швидкості відключається одне з реле, даючи необхідну добавку за рахунок дії зворотного зв'язку на інше реле. В результаті їх спрацювання відбуватиметься практично одночасно. Випробування схеми реле контролю обертання показали, що з введенням зворотного зв'язку з другого каналу на перший вдається позбутися від нестабільності в роботі реле і з'являється можливість легкої його налаштування.

Перевірка справності реле контролю обертання. Для перевірки реле доцільно в ланцюзі обох каналів включити розмикаючих контакти кнопок (рисунок 1.6). Натискаючи по черзі встановлені кнопки можна імітувати втрату кінематичної цілісності ланцюга від підйомної машини до програмного пристрою і тахогенератора обмежувача швидкості. При цьому на швидкості 1,0 - 1,5 м/с реле контролю обертання повинно включити запобіжне гальмо. Така перевірка дозволяє регулярно і ефективно стежити за працездатністю реле.

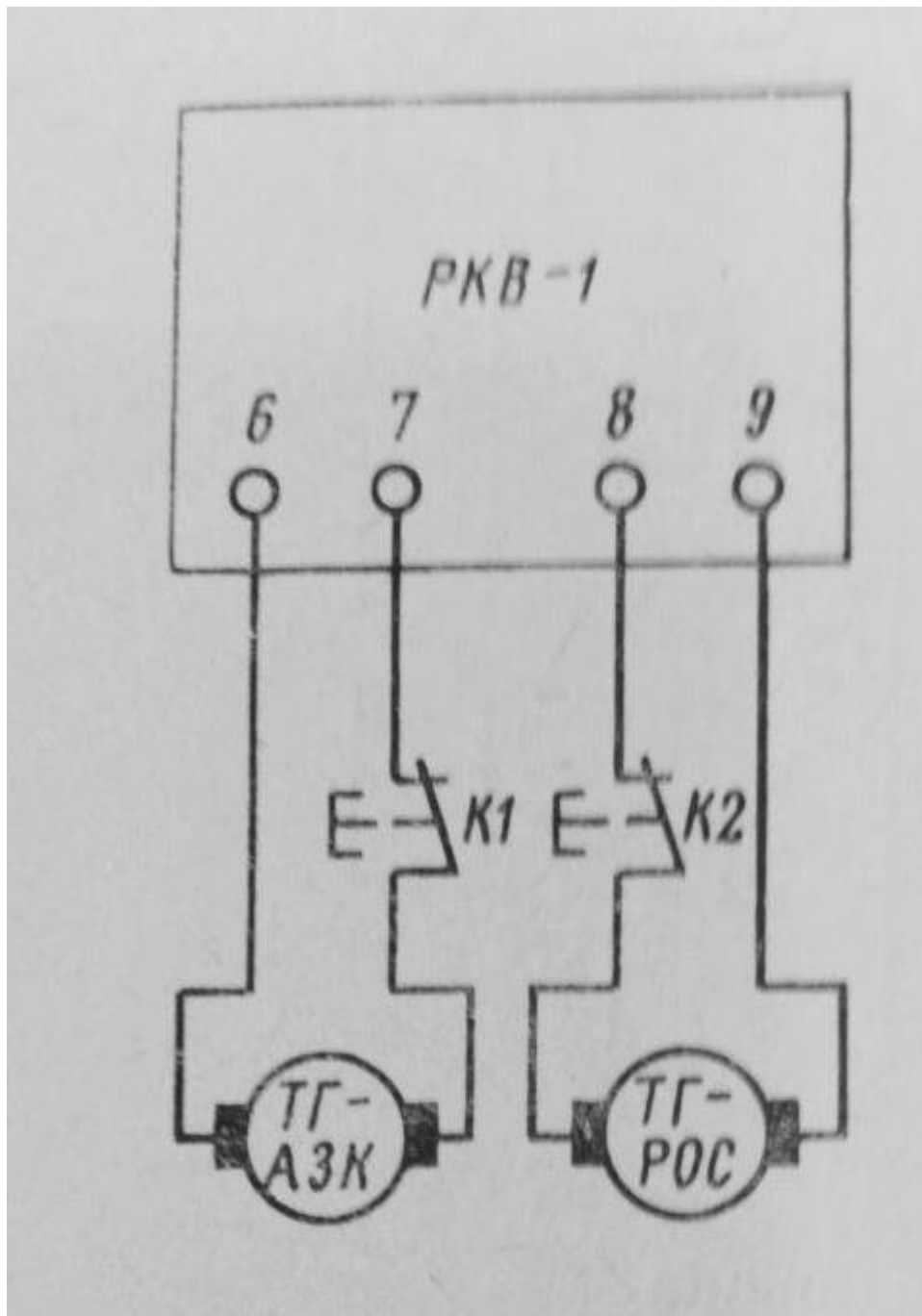


Рис. 1.6. Перевірка справності реле

Контроль напрямку обертання підйомної машини. У схемах підйомних установок часто виникає необхідність в пристрої, що контролює в широкому діапазоні напрямок обертання швидкості і видає необхідні команди в схеми керування, захисту та сигналізації.

Так, пристрій контролю напрямку обертання необхідно в схемі електричного обмежувача швидкості для підключення максимальної установки на час розгону підйомної машини. Пристрій контролю використовується в

схемах автоматики для дублювання захистів и від зворотного ходу. Крім того пристрій може іпользоваться в різного роду блокуванні, наприклад при русі машини воно може накладати заборону на спрацьовування посадочних пристроїв (куркулів або гойдаються майданчиків).

Основним недоліком застосовуваних способів контролю напрямку обертання (реле РКС, релейних схем зі зміною полярності тахогенератора при зміні напрямку обертання) є висока початкова швидкість підйомних посудин (0,8 м/с - 1,0 м/с), при якій відбувається спрацьовування апарату контролю. Крім того, виникають труднощі прив'язки реле до підйомної машини, а схема з реле постійного струму типу РЕВ, включених на напругу тахогенератора, споживає значну потужність.

В даний час є необхідність створення багатоцільового пристрої контролю для підйомних установок, що відрізняється простотою, надійністю, яке споживає мінімальну потужність, що не потребує додаткових джерел живлення, крім тахогенератора, що охоплює весь необхідний діапазон швидкостей. Цим умовам відповідає схема, представлена на малюнку 1.7, що складається з двох реле, транзисторів, стабілітронів, діодів і резисторів.

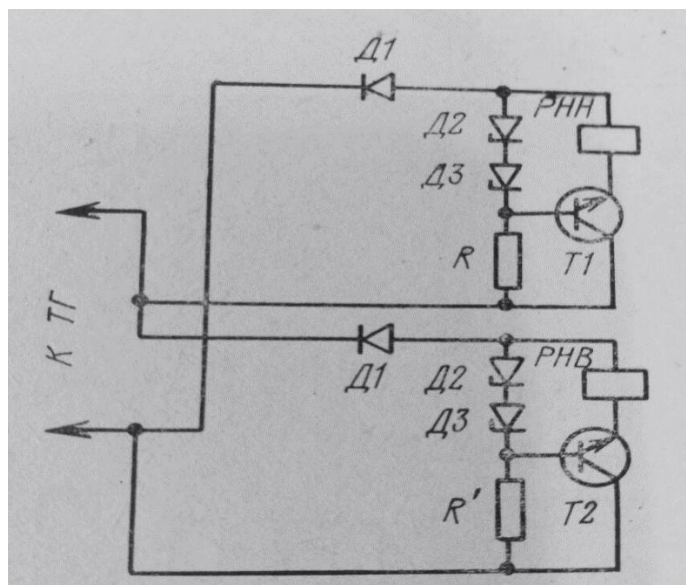


Рис. 1.7. Схема контролю напрямку обертання

Схема включається на затискачі тахогенератора, що змінює свою напругу від 0 до 200 В. При напрузі в 10 В транзистор відкривається і одне з реле, в залежності від полярності тахогенератора втягується. Опір $R = 10$ кОм використовується в схемі в якості зсуву. При досягненні опорного напруги на стабілітронах транзистор прикривається і збільшення струму в колекторної ланцюга не відбувається. Включення реле здійснюється при напрузі тахогенератора близько 10 В і струмі через реле 0,006 - 0,008 А. При наростанні швидкості аж до максимальної струм через реле збільшується лише вдвічі завдяки зворотного зв'язку через стабілітрони.

Для отримання пристрої контролю напрямку руху підйомних посудин, що володіє підвищеною чутливістю (при швидкості 0,1 - 0,2 м/с), з валом відомою циліндричної шестерні показчика глибини (або будь-яким іншим валом, пов'язаним жорстко з органом навивки) з'єднують перехідний вал (рисунок 1.8) зі склянкою, на який встановлюють кільцевої постійний магніт. При русі підйомної машини стакан разом з кільцевим магнітом починає обертатися. Важіль кінцевого вимикача перекидається в відповідну сторону профілем, встановленим на магніті, переміщення якого обмежується упорами.

Контроль справної роботи ретардуючих дисків. З ретардіруючими дисками показчика глибини пов'язані такі апарати, як обмежувач швидкості, регулятор ходу, кінцеві вимикачі перепідйому, шляхові кінцеві вимикачі та інші. Дія цих апаратів погоджено з положенням підйомної посудини в стовбурі. Несправний стан (зупинка) ретардіруючих дисків показчика глибини під час руху машини може призвести до помилкової інформації про стан підйомної посудини і, отже, до важкої аварії. Найбільш простий і надійний спосіб контролю справності ретардіруючих дисків використовується Донецьким налагоджувальних управлінням, представленим на малюнку 1.9. На спеціальному кронштейні закріплюються шків, які можуть вільно обертатися на своїх осях. Шків насаджується на важіль кінцевого вимикача і також може вільно обертатися. Між ретардіруючими дисками і перерахованими шківом натягується тонкий сталевий трос, кінці якого закріплюються на ободах ретардіруючих дисків. Ось

шківа залишається нерухомою при синхронній роботі обох дисків. Будь-яке порушення передач до одного з дисків призводить до повороту важеля кінцевого вимикача і спрацьовує захист. При цьому контролюється будь-яка несправність ретардіруючих дисків, а також самого пристрою. При використанні обмежувача швидкості Донецького налагоджувального управління контроль справності роботи ретардіруючих дисків здійснюється схемою самого обмежувача.

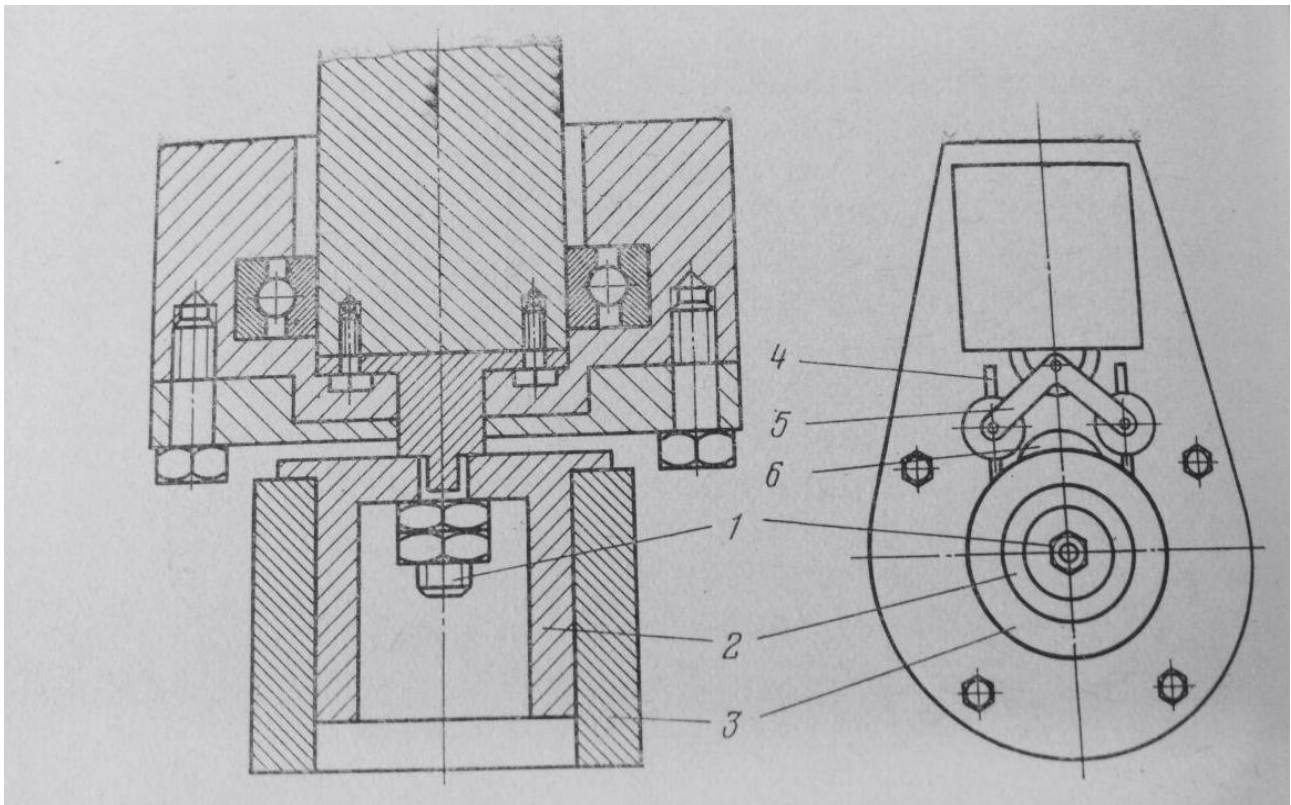


Рис. 1.8. Пристрій для контролю напрямку обертання:

1 - перехідний вал; 2 - стакан; 3 - постійний магніт; 4 - упор; 5 - важіль; 6 - профіль

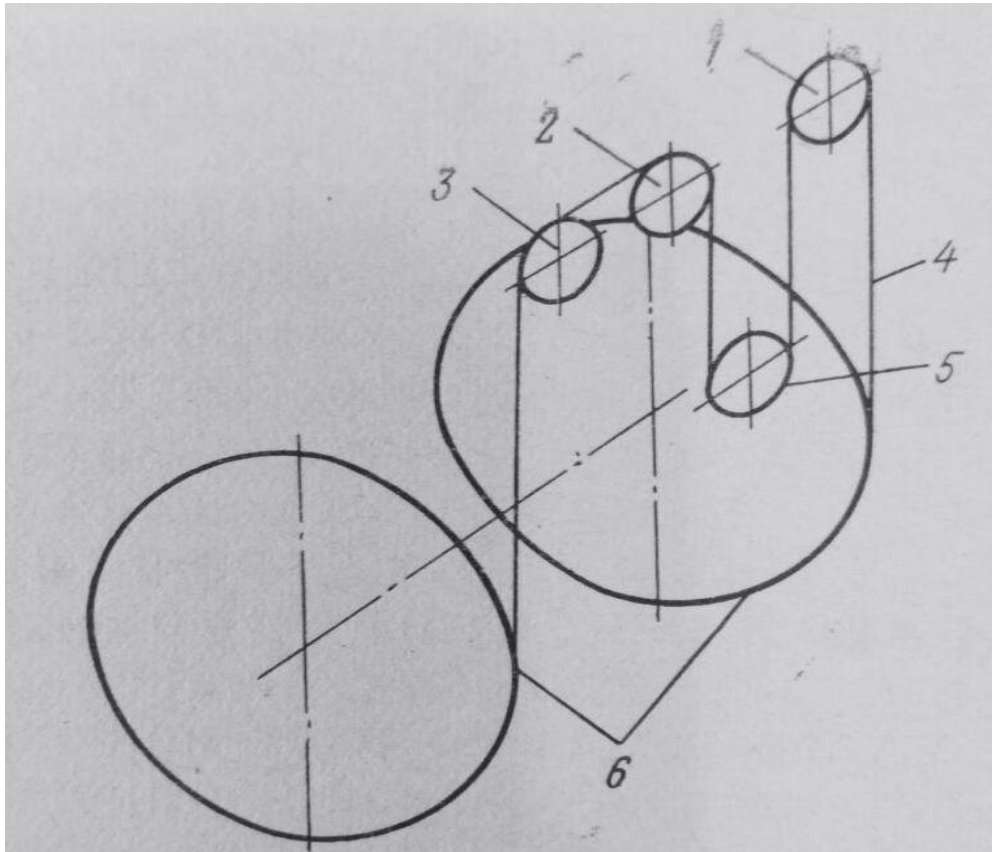


Рис. 1.9. Контроль справної роботи ретардіруючих дисків:

1, 2, 3, 5 - шківів; 4 - трос; 6 - ретардіруючий диск

Контроль замикання в обмотках двигунів. Для визначення замикань в обмотці між витками в одній котушці або проводами різних фаз користуються спеціальними приладами. Контроль обмоток всіх типів значно вдосконалений завдяки використанню електронних апаратів, які живляться від мережі змінного струму і посиляють в випробовувану обмотку хвилю імпульсу з амплітудою до 600 В при струмі в декілька міліампер. На екрані електронно-променевої трубки з'являється перетин у вигляді яскравої точки. За допомогою двох вертикальних пластин промінь розгортається в смугу, що світиться. Дві горизонтальні пластини, до яких приєднуються дві гілки випробовуваної обмотки, змушують промінь коливатися в вертикальному положенні і на екрані з'являється крива. При несправності однієї з двох гілок обмотки на екрані з'являється зображення двох кривих. Апарати дозволяють контролювати замикання в обмотках і міцність міжвиткової ізоляції електричних машин без їх розбирання.

Контроль завантаження підйомної посудини автоматизованої підйомної установки. На скіпових підйомних установках можливі випадки, коли по ряду причин скіп не завантажується. На автоматизованій установці це призводить до непродуктивного прогону порожніх скіпів. Для контролю завантаження скіпа застосовуються різні схеми.

Для контролю завантаження підйомної посудини на автоматизованих підйомних установках застосовуються датчики початку уповільнення. Принцип дії датчика заснований на активній потужності підйомного двигуна і встановлення відповідної витримки часу на відключення підйомного двигуна після подачі відключає імпульсу, тобто датчик забезпечує автоматичне перенесення точки початку уповільнення в залежності від завантаження підйомної посудини.

Датчик початку уповільнення складається з вимірювального органу і вихідного пристрою, службовця для отримання витягу часу в функції вихідного струму датчика активної потужності.

Контроль рівня. Для контролю та автоматизації заповнення бункерів і течок в системі підйому використовується іскробезпечне реле контролю опору. При зіткненні гірської маси з електродом через перехідний опір маси відбувається замикання ланцюга і спрацьовування вихідного реле.

Зазвичай схема включення реле не задовольняє вимогам самоконтролю, так як при будь-якій несправності в схемі контакт реле в ланцюгах захисту або управління залишається замкненим. Завдяки перез'єднання в схемі досягається її самоконтрольованого. В якості основних елементів рівня вугілля або породи в бункерах або скіпах при розвантаженні використовуються ізотопні гамма-реле.

Висновки

В даному розділі були проаналізовані основні складові частини шахтних підйомних машин, випадки їх поламки та можливі методи полагодження. На основі цього аналізу були сформульовані наступні задачі, для подальшого дослідження:

- Дослідження оптимізації існуючих систем контролю та попередження аварій.
- Технічний розгляд основних пристроїв сигналізації.
- Аналіз способів рішення проблеми та розгляд шаблонів проектування аварійних систем.

РОЗДІЛ 2. ПРОГНОЗУВАННЯ ЧАСОВИХ РЯДІВ

2.1. Постановка задачі

У більшості постановок задачі прогнозування вважається, що значення прогнозованого процесу $y(t)$ в будь-який момент часу t однозначно визначається як деяка функція h від виходу істинної моделі процесу в цей момент часу $f(t)$, і деякої випадкової складової $\varepsilon(t)$:

$$y(t) = h(f(t), \varepsilon(t)).$$

Найчастіше, у якості функції h використовують або функцію підсумовування $h(x, y) = x + y$, або мультиплікації $h(x, y) = x * y$.

Передбачається, що випадкова величина (ВВ) це білий шум [12], що має кінцеву дисперсію, але будь-які інші характеристики даної ВВ нам невідомі. У разі наявності будь-якої додаткової інформації про $\varepsilon(t)$ її слід обов'язково враховувати, оскільки це може значно поліпшити точність прогнозу.

Розглянемо основні постановки задачі прогнозування часових рядів.

2.1.1. Регресійна постановка задачі прогнозування

Задана тимчасова вибірка даних виду $\vec{x}^T = \{x(t)\}, t=1, \dots, T; x(t) \in \mathbb{R}$, де x це прогнозований випадковий процес. Також можуть бути задані вибірки $z_i^{\vec{\cdot}}$ випадкових процесів $z_i: z_i^{\vec{\cdot}} = \{z_i(t)\}, t=1, \dots, T; i=1, \dots, N$, від яких, можливо, залежить прогнозований процес. Ці додаткові процеси називаються екзогенними процесами або змінними. Важливо, що прогнозована величина є безперервною, у той час як екзогенні змінні можуть бути будь-якого типу: безперервні, дискретні, нечіткі і т.д. Потрібно побудувати модель процесу, що залежить як від значень самого прогнозованого процесу, значень процесів, що впливають на

прогнозований, так і від часу t , яка б з задовільною точністю давала оцінку істинного значення $x(T + T_y)$:

$$\hat{x}_{T+T_y} = f(x^{\rightarrow}, z_1^{\rightarrow}, \dots, z_N^{\rightarrow}, t),$$

де \hat{x}_{T+T_y} – оцінка моделі, а область значень моделі – вся множина дійсних чисел:

$$E(f) = \mathbb{R}.$$

2.1.2. Постановка задачі прогнозування як задачі класифікації

Класифікація вибірки даних означає поділ її на класи, число яких визначено заздалегідь – тобто в результаті класифікації кожна точка з вибірки буде віднесена до певного класу. Класифікацію можна розглядати як дискретний опис об'єкта.

Завдання класифікації вибірки даних може виникати при прогнозуванні деякої дискретної величини, що має обмежений набір значень.

Наприклад, потрібно передбачити наявність у пацієнта раку. У даному випадку мається навчальна вибірка даних про пацієнтів, що включає обидва класи пацієнтів: ті, у кого є рак, і ті, у кого немає, і потрібно визначити приналежність даного конкретного пацієнта до одного з класів.

Також, при прогнозуванні сильно зашумленої величини, зважаючи на більшу складність прогнозу, можливою є розбивка набору всіх прийнятих на вихідній вибірці значень на класи, і подальше прогнозування не самого значення величини, а його приналежності до деякого класу.

Вихідні дані ті ж, що й у регресійній постановці завдання прогнозування, тільки прогнозований процес є дискретним (як і будь-які його вибірки): $x^{\rightarrow} = \{x(t)\}, t = 1, \dots, T; x(t) \in C_k; k = 1, \dots, K$, де C_k – k -ий клас, K – кількість класів.

Потрібно побудувати модель процесу, що залежить від наявної інформації, яка б визначала належність до певного класу деякого майбутнього значення в момент часу: $T + T_y$:

$$\hat{x}_{T+T_y} = f(\vec{x}, \vec{z}_1, \dots, \vec{z}_N, t),$$

де область значень моделі – множина всіх можливих класів:

$$E(f) = \{1, \dots, K\}.$$

2.1.3. Задача безмодельного прогнозування процесів

Як було відмічено в попередніх постановках, стан складного об'єкта може описуватися вектором характеристичних змінних представлених у вибірці даних спостережень, де одне спостереження в момент часу t буде мати вигляд $o_{t \rightarrow} = [z_1(t), z_2(t), \dots, z_N(t)]^T$, а прогнозованою величиною $x(t)$ буде вихід об'єкта. І якщо розмірність вектора характеристичних змінних велика, то для здійснення прогнозу за поточним спостереженням $o_{t \rightarrow}$ може виявитися достатнім використання методу аналогії: необхідно знайти один або кілька близьких спостережень в передісторії і подивитися що було далі.

При цьому модель об'єкта не потрібна, її роль виконує сам об'єкт. Прогноз за аналогами з передісторії розроблявся головним чином в метеорології для прогнозу погоди. Формалізована постановка задачі безмодельного прогнозування за аналогією.

1. Є навчальна вибірка, що складається з N векторів спостережень характеристичних змінних об'єкта $o_{t \rightarrow} = [z_1(t), z_2(t), \dots, z_N(t)]^T, t = 1, \dots, N$ та N та відповідних значень виходу об'єкта $x(t), t = 1, \dots, N$.

2. Необхідно задати::

а. Критерій близькості спостережень $L(\vec{o}_i, \vec{o}_j), i, j = 1 \dots, T; L \geq 0$ – зазвичай його задають у вигляді відстані, і чим менша відстань – тим більша близькість.

б. Кількість аналогів k , що враховуються при прогнозі. Під аналогами маються на увазі спостереження, найближчі до заданого. У результаті, матимемо вектор з k векторів спостережень:

$$\vec{r} = [\vec{\sigma}_{w_1}, \dots, \vec{\sigma}_{w_k}]^T,$$

де $\vec{w} = [w_1, \dots, w_k]^T$ – вектор індексів найближчих k спостережень в наявній вибірці.

в. Функцію «усереднення» виходів об'єкта для знайдених аналогів:

$$\hat{x} = f(x_{w_1}, x_{w_2}, \dots, x_{w_k}),$$

яка буде використовуватися для отримання остаточного прогнозу.

2.2. Огляд методів

Наведемо огляд основних методів, які розроблені і застосовуються для вирішення задачі прогнозування.

2.2.1. Регресійний аналіз

У разі застосування регресійного аналізу [13] для вирішення задачі прогнозування регресором буде або час t , або деякі незалежні змінні z_i , а критеріальною (залежною) змінною – сам прогнозований процес.

У випадку, коли регресором є час, вихідними даними є вибірка

$$\mathbf{x} = \{x_t\}, t = 1, \dots, T,$$

і потрібно знайти рівняння регресії виду $x = f(t)$. У найпростішому випадку використовується лінійна регресія виду

$$x = b_0 + b_1 * t + b_2 * t^2 + \dots + b_n * t^n,$$

де вектор коефіцієнтів $\mathbf{b} = [b_0, \dots, b_n]^T$ знаходиться за допомогою методу найменших квадратів [13]. Відповідно, для прогнозування значення процесу в деякий момент часу t_{T+T_y} , потрібно просто підставити цей час

в рівняння регресії.

У другому випадку вихідними даними буде вибірка виду $Z = \{z_{ik}\}; i = 1, \dots, N; k = 1, \dots, K$, і набір відповідних значень $x = \{x_k\}$, де $N - N$ – кількість незалежних змінних, K – кількість точок у вибірці (кожен вектор виду $[z_{1k}, \dots, z_{Nk}]^T$ є однією точкою вибірки). Потрібно знайти рівняння регресії виду $x = f(\mathbf{z}_1, \dots, \mathbf{z}_N)$. Знову ж таки, у загальному випадку використовується лінійна регресія $x = b_0 + b_1 * \mathbf{z}_1 + b_2 * \mathbf{z}_2 + \dots + b_N * \mathbf{z}_N$, де коефіцієнти знаходяться аналогічно. Для прогнозування значення $x(\mathbf{z}_1, \dots, \mathbf{z}_N)$ при деяких певних значеннях незалежних змінних $\mathbf{z}_1, \dots, \mathbf{z}_N$ потрібно підставити ці значення в рівняння регресії.

Згідно з теоремою Гаусса-Маркова, якщо виконується певний набір умов, то оцінки параметрів лінійної моделі, отримані за допомогою методу найменших квадратів, будуть ефективні (тобто будуть володіти найменшою дисперсією) в класі незміщених оцінок (тобто таких оцінок, математичне очікування яких дорівнює істинним значенням параметрів). Ось 5 умов, необхідних для отримання ефективних оцінок при використанні методу найменших квадратів:

Перша умова: правильно задана модель, тобто істинна залежність між вхідними і вихідною змінними дійсно лінійна, помилка впливає на вихід моделі за адитивним законом і відсутня недовизначеність (упущені важливі фактори) або перевизначеність (враховані непотрібні фактори).

Друга умова: всі вхідні змінні детерміновані і не всі рівні між собою.

Третя умова: помилки не носять систематичного характеру. Випадковий член може бути іноді позитивним, іноді негативним, але він не повинен мати систематичного зсуву ні в якому з двох можливих напрямків. Якщо рівняння регресії включає постійний член, то ця умова найчастіше виконується автоматично, так як постійний член відображає

будь-яку систематичну, але постійну складову у вихідної змінної, якою не враховують пояснюючі змінні, включені в рівняння регресії.

Четверта умова: дисперсія помилок однакова. Однаковість дисперсії помилок також прийнято називати гомоскедастичністю. Не повинно бути апіорної причини для того, щоб випадковий член породжував більшу помилку в одних спостереженнях, ніж в інших. Так як $E(\varepsilon_i) = 0 \forall i$ і теоретична дисперсія відхилень ε_i дорівнює $E(\varepsilon^2)$, то цю умову можна записати так:

$$E(\varepsilon^2) = \sigma^2 \forall i.$$

Одне із завдань регресійного аналізу полягає в оцінці стандартного відхилення випадкового члена. Якщо умова, що розглядається, не виконується, то коефіцієнти регресії, знайдені за методом найменших квадратів, будуть неефективні, а більш ефективні результати будуть виходити шляхом застосування модифікованого методу оцінювання (зважений МНК або оцінка коваріаційної матриці за формулою Уайта [14] або Девідсона-Маккінона).

П'ята умова: ε_i розподілені незалежно від ε_j при $i \neq j$. Ця умова передбачає відсутність систематичного зв'язку між значеннями випадкового члена в будь-яких двох спостереженнях. Якщо один випадковий член великий і позитивний в одному напрямку, не повинно бути систематичної тенденції до того, що він буде таким же великим і позитивним (те ж можна сказати і про малі, і про негативні залишки). Теоретична коваріація $\sigma_{\varepsilon_i, \varepsilon_j}$ повинна дорівнювати нулю, оскільки:

$$\sigma_{\varepsilon_i, \varepsilon_j} = E\{[\varepsilon_i - E(\varepsilon_i)][\varepsilon_j - E(\varepsilon_j)]\} = E(\varepsilon_i * \varepsilon_j) - E(\varepsilon_i) * E(\varepsilon_j) = 0.$$

Теоретичні середні для ε_i і ε_j дорівнюють нулю в силу третьої умови теореми. При невиконанні цієї умови оцінки, отримані за методом найменших квадратів, будуть також неефективні.

Перевагами використання лінійної регресії як моделі і методу найменших квадратів для знаходження параметрів цієї моделі є простота,

обчислювальна та практична ефективність. Під практичною ефективністю розуміється гарна якість прогнозу (з точки зору мінімуму середньоквадратичної помилки (СКП)), отриманого з використанням цього методу для багатьох реальних задач прогнозування.

Недоліками даного підходу є необхідність повного перерахунку параметрів при появі нової інформації і необхідність явного вибору моделі прогнозування – при невідповідності обраної та істинної моделі отримані параметри будуть зміщеними – тобто їх математичне очікування буде відрізнятися від дійсних значень параметрів. Для ілюстрації цього недоліку розглянемо наступний приклад.

1) Згенеруємо навчальні дані, задавши справжню модель як $y = 0.8 * x + 2$, для всіх $x = 1, 2, \dots, 100$, додаючи нормальну випадкову перешкоду ε з параметрами $E(\varepsilon) = 0$, $\sigma(\varepsilon) = 5$. Графік одного з нескінченної кількості можливих прикладів буде виглядати наступним чином (рис. 2.1):

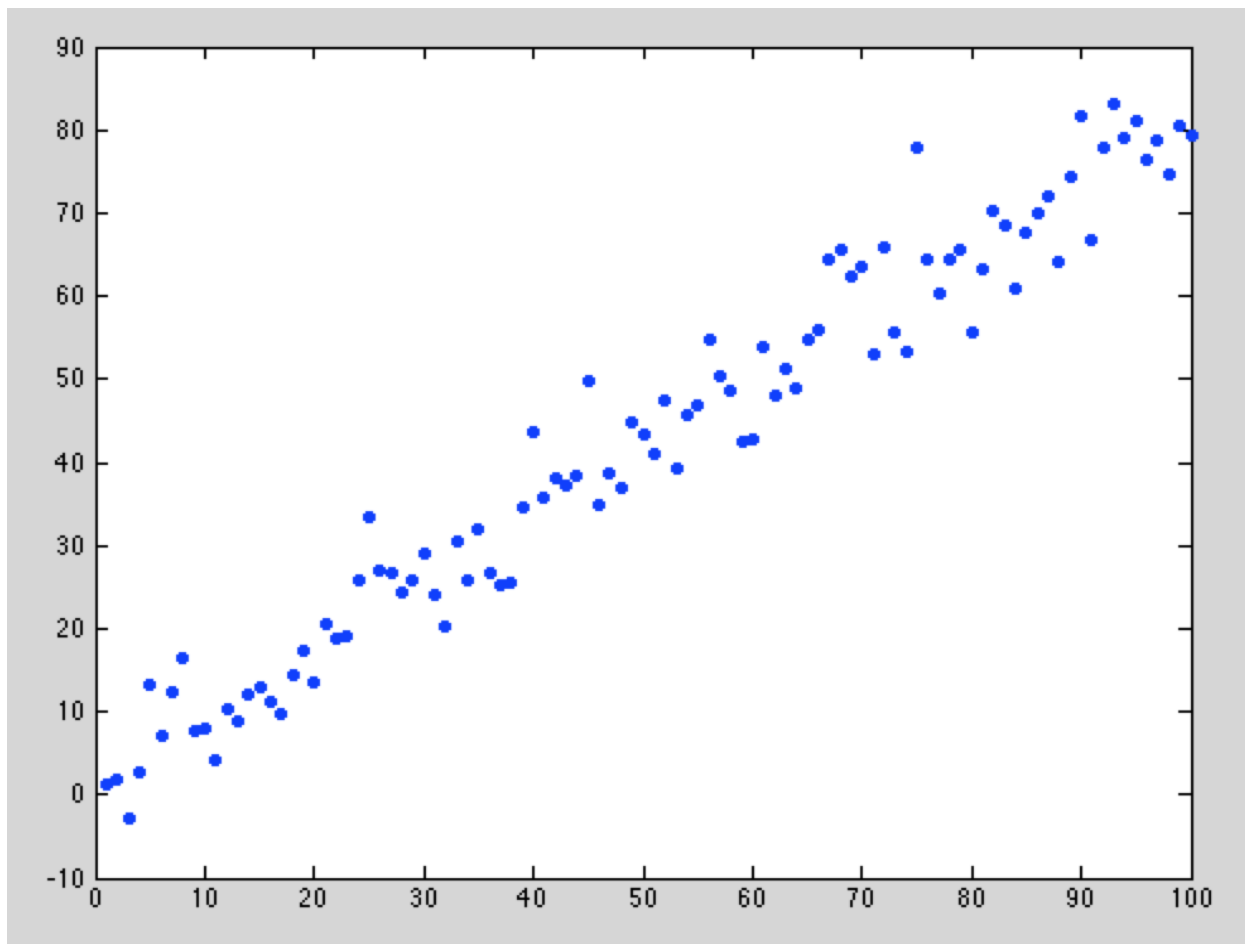


Рис. 2.1. Графік згенерованих навчальних даних

2) Для згенерованого прикладу отримаємо параметри двох моделей, використовуючи метод найменших квадратів:

$$y_1 = a_{11} * x + a_{10},$$

$$y_2 = a_{23} * x^3 + a_{22} * x^2 + a_{21} * x + a_{20}.$$

3) Запам'ятемо отримані параметри, і повторимо пункти 1-2 ще 9999 разів, щоразу запам'ятовуючи параметри.

4) Усереднимо отримані вектори параметрів за всіма експериментами. Таким чином, ми отримаємо оцінку математичного очікування параметрів обраних моделей. Результати:

$$a_{11} = 0.8, a_{10} = 2.0039$$

$$a_{23} = -0.00001, a_{22} = 0.0000, a_{21} = 0.7985, a_{20} = 2.0211$$

Як добре видно з результатів, математичне очікування оцінок параметрів моделі y_2 дійсно злегка зміщене від істинних значень, у той час як математичне очікування оцінок параметрів моделі y_1 (яка збігається з істинною) збігається з істинними значеннями параметрів.

2.2.2. ARIMA

Інтегрована модель авторегресії-ковзаючого середнього (модель Бокса-Дженкінса, англ. AutoRegressive Integrated Moving Average) [15] – одна з математичних моделей, що використовуються для аналізу і прогнозування стаціонарних часових рядів в статистиці. Є розширенням моделей ARMA (AutoRegressive Moving Average) [15] для нестаціонарних часових рядів, які можна зробити стаціонарними взяттям різниць деякого порядку від вихідного часового ряду (так звані інтегровані або різницево-стаціонарні часові ряди). Модель $ARIMA(p, d, q)$ означає, що різниці часового ряду порядку d підкоряються моделі $ARMA(p, q)$.

Моделлю $ARMA(p, q)$, де p і q – цілі числа, що задають порядок моделі, називається наступний процес генерації часового ряду $\{x_t\}$:

$$x_t = c + \sum_{i=1}^p \alpha_i x_{t-i} + \sum_{i=1}^q \beta_i \varepsilon_{t-i},$$

де c – константа, $\{\varepsilon_t\}$ – білий шум, тобто послідовність незалежних і однаково розподілених випадкових величин (як правило, нормальних), з нульовим середнім, а $\alpha_1, \dots, \alpha_p$ і β_1, \dots, β_q – дійсні числа, авто регресійні коефіцієнти і коефіцієнти змінного середнього, відповідно.

Також дана модель інтерпретується як $ARMA(p + d, q)$ -модель з d одиничними рішеннями. При $d = 0$ маємо звичайні ARMA-моделі.

ARIMA-моделі дозволяють моделювати інтегровані або різницево-стаціонарні часові ряди (DS-ряди, difference stationary).

Часовий ряд X_t називається інтегрованим порядку k (зазвичай пишуть $X_t \sim I(k)$), якщо різниці ряду порядку k , тобто $\Delta^k X_t$ є стаціонарними, у той час як різниці меншого порядку (включаючи нульового порядку, тобто сам часовий ряд) не є стаціонарними щодо деякого тренду рядами (TS-рядами, trend stationary). Зокрема $I(0)$ – це стаціонарний процес.

Порядок інтегрованості часового ряду i є порядок d моделі $ARIMA(p, d, q)$.

Для побудови ж моделі ARMA за серією спостережень необхідно визначити порядок моделі (числа p і q), а потім і самі коефіцієнти. Для визначення порядку моделі може застосовуватися дослідження таких характеристик часового ряду, як його автокореляційна функція і приватна автокореляційна функція. Для визначення коефіцієнтів застосовуються такі методи, як метод найменших квадратів і метод максимальної правдоподібності [16].

2.2.3. Метод групового урахування аргументів

Метод групового урахування аргументів (МГУА) [17] – сімейство індуктивних алгоритмів для математичного моделювання мультипараметричних даних. Метод заснований на рекурсивному селективному відборі моделей, на основі яких будуються більш складні моделі. Точність моделювання на кожному наступному кроці рекурсії збільшується за рахунок ускладнення моделі.

Найпростіший ітераційний багаторядний алгоритм МГУА полягає в наступних етапах.

1. Маємо дані спостережень: x^T, y . Необхідно побудувати найкращу

в певному сенсі модель $Y(x_1, \dots, x_n)$.

2. Важливим етапом МГУА є розбиття вихідної вибірки \vec{x} на дві (в деяких випадках три) підвибірки: навчальну і перевірочну (іноді ще виділяють екзаменаційну вибірку). Навчальна вибірка буде надалі використовуватися для знаходження параметрів моделей, перебираються, а перевірочна – для перевірки їх придатності та відбору найкращих моделей (за наявності екзаменаційної вибірки на ній буде перевірятися остаточна модель). Існує кілька основних способів розділення даних на перевірочну і навчальну вибірки.

- У навчальну вибірку відбираються перші $n * k$ прикладів, n – загальна кількість прикладів, k – деякий коефіцієнт, $0 < k < 1$; в перевірочну вибірку – усі приклади, що залишилися.

- Всі приклади ранжуються за дисперсією, і в навчальну вибірку відбираються $n * k$ вузлів з найбільшою дисперсією, а в перевірочну – решта вузлів.

- Вузли відбираються в перевірочну і навчальну вибірку через один. У загальному випадку, те, яким чином вихідна вибірка розбивається на підвибірки, може значно вплинути на прогнозуючі якості майбутньої моделі

3. Використовуючи опорні функції будуються різні варіанти моделей для деяких або всіх аргументів. Наприклад, будуються поліноми з однією змінною, поліноми з усіма можливими парами змінних, поліноми з усіма можливими трійками змінних, і т.д., поліном з усіма змінними. Для кожної моделі визначаються її коефіцієнти методом регресійного аналізу.

4. Серед усіх моделей вибираються кілька найкращих. Якість моделей визначається коефіцієнтом детермінації, або середньоквадратичним відхиленням помилки, або кореляцією Y і вихідних даних.

5. Якщо знайдена достатньо «хороша» модель або досягнута максимально допустима складність моделей, то алгоритм закінчується.

Інакше, знайдені на 3-му кроці моделі використовуються як аргументи (x_1, \dots, x_n) для опорних функцій наступного етапу ітерації (перехід на 2-ий пункт). Тобто вже знайдені моделі беруть участь в формуванні більш складних.

Зазвичай ступінь полінома опорної функції вибирається не вище $N-1$, де N – кількість точок вибірки. Часто буває достатнім використати в якості опорних функції поліноми другого ступеня. У такому випадку на кожному кроці ітерації ступінь результуючого полінома подвоюється.

2.2.4. Штучні нейронні мережі

Штучні нейронні мережі (ШНМ) [18] являють собою систему з'єднаних і взаємодіючих між собою простих процесорів – штучних нейронів.

Нейронні мережі не програмуються у звичному розумінні цього слова, вони навчаються. З математичної точки зору, навчання нейронних мереж – це багатопараметричне завдання нелінійної оптимізації. Можливість навчання – одна з головних переваг нейронних мереж в порівнянні з традиційними алгоритмами. Технічно навчання полягає в знаходженні коефіцієнтів зв'язків між нейронами. У процесі навчання нейронна мережа здатна виявляти складні залежності між вхідними даними і вихідними, а також виконувати узагальнення.

Здібності нейронної мережі до прогнозування безпосередньо впливають з її здатності до узагальнення і виділення прихованих залежностей між вхідними та вихідними даними.

Великою перевагою ШНМ перед іншими методами прогнозування є непотрібність явно задавати рівняння моделі прогнозованого процесу, що значно спрощує процес прогнозування, а також прибирає можливість вибору неправильного рівняння, яке погано б відповідало реальному процесу.

Другим вагомим аргументом у бік вибору ШНМ для вирішення задачі прогнозування є їх універсальність: апарат нейронних мереж успішно

використовується як для прогнозу часових рядів, так і для прогнозу значень залежної змінної від вектора значень незалежних змінних, кластеризації вибірки даних і багатьох інших завдань. Це дозволяє використовувати єдиний математичний апарат ШНМ для вирішення декількох різних завдань прогнозування (схожою універсальністю володіє МГУА, але в ньому потрібно явно задавати рівняння приватних описів).

Виходячи з вищевикладеного, у якості основних методів для дослідження були обрані ШНМ і МГУА – обидва методи дозволяють знаходити нелінійні залежності між даними без необхідності явного задавання моделі.

2.3. Основні проблеми, що виникають при вирішенні задачі прогнозування

Вирішення задачі прогнозування супроводжується великою кількістю різноманітних проблем, які, власне, і роблять цю задачу складною. Перечислимо основні:

- 1) неможливо врахувати всі чинники, що впливають на процес, який ми намагаємося прогнозувати; більше того, їх вплив може змінюватися з плином часу - фактор, що не був важливий сьогодні, може відіграти важливу роль завтра;
- 2) завжди є багато (іноді нескінченне число) можливих моделей, що добре відповідають навчальним даними - ми повинні вирішити, яку модель або множину моделей використовувати, і такі рішення зазвичай дуже схильні до помилок;
- 3) часто буває важко (якщо не неможливо) знайти оптимальну складність моделі.

Якщо проаналізувати перераховані існуючі методи, то можна зробити висновок, що більшість методів ніяк не вирішують описані проблеми – тільки МГУА певним чином вирішує третю проблему, знаходячи оптимальну складність обраної моделі шляхом перебору усіх можливих

«варіацій» моделі починаючи з самої простої.

2.4. Моделі випадкових перешкод

Як було сказано при розгляді постановок задачі прогнозування, вибірка, що використовується для прогнозування, майже завжди містить деяку випадкову перешкоду, яка взаємодіє з «чистим» сигналом за деяким законом (найчастіше аддитивним або мультиплікативним). Далі розглянемо найбільш поширені моделі випадкових перешкод.

2.4.1. Білий шум

Білий шум – стаціонарний шум, спектральні складові якого рівномірно розподілені по всьому діапазону задіяних частот.

Безперервний у часі випадковий процес $w(t)$, де $t \in \mathbb{R}$, є білим шумом тоді і тільки тоді, коли його математичне очікування і автокореляційна функція задовольняють наступні рівняння відповідно:

$$\mu_w(t) = E\{w(t)\} = 0$$

$$R_{ww}(t_1, t_2) = E\{w(t_1)w(t_2)\} = \sigma^2 \delta(t_1 - t_2)$$

Тобто, це випадковий процес з нульовим математичним очікуванням, що має автокореляційну функцію, яка є дельта-функцією Дірака. Така автокореляційна функція передбачає наступну спектральну щільність потужності:

$$S_{ww}(w) = \sigma^2$$

так як перетворення Фур'є дельта-функції дорівнює одиниці на всіх частотах. Через те, що спектральна щільність потужності однакова на всіх частотах, білий шум і отримав свою назву (за аналогією з частотним спектром білого світла).

Багато розроблених методів прогнозування та обробки часових рядів є оптимальними за умови, якщо перешкода, що впливає на чистий сигнал, є білим шумом (або деяким його підвидом, наприклад нормальним білим

шумом). Як приклад можна привести метод найменших квадратів і фільтр Калмана [19]. Саме тому, якщо вважається, що перешкода є білим шумом, зазвичай достатньо просто використовувати відповідний метод без будь-якої фільтрації.

2.4.2. Кольорові шуми

Рожевий шум або шум $1/f$ (іноді також званий флікер-шум) є сигналом або процесом з таким спектром частот, що спектральна щільність потужності (енергія або потужність на Гц) обернено пропорційна частоті сигналу. При рожевому шумі, кожна октава містить рівну кількість потужності шуму.

У науковій літературі термін рожевий шум іноді використовується трохи більш вільно і може відноситися до будь-якого шуму зі спектральною щільністю потужності у вигляді $S(f) \sim 1/f^\alpha$

де f – частота і $0 < \alpha < 2$, з показником α зазвичай близько до 1. Ці схожі на рожевий шуми широко поширені в природі.

Через те, що потужність рожевого шуму велика на низьких частотах, його практично неможливо відфільтрувати, не «зачепивши» чистий інформаційний сигнал, оскільки тренд інформаційного сигналу зазвичай «займає» низькі частоти.

Броунівський (червоний) шум

Спектральна щільність червоного шуму пропорційна $1/f^2$, де f – частота. Це означає, що на низьких частотах цей шум має навіть більше енергії, ніж рожевий шум.

Червоний шум може бути отриманий шляхом інтегрування білого шуму. Тобто, тоді як білий (дискретний) шум може бути отриманий шляхом випадкового вибору кожного відліку незалежно один від одного, червоний шум може бути отриманий шляхом додавання до кожного відліку сигналу випадкової величини для отримання наступного відліку:

$$x_k = x_{k-1} + \mu_k,$$

де μ_k – білий шум.

Чорний шум

Під чорним шумом зазвичай розуміють випадковий процес з спектральною щільністю виду $1/f^\beta$, де $\beta > 2$. Цю модель часто використовують для моделювання катастроф природного походження.

2.5. Первинна обробка даних

При вирішенні задачі прогнозування даних етап часто є важливішим від етапу складання моделі прогнозу. Неправильна обробка даних (або її відсутність) може значно погіршити прогнозуючі якості побудованої моделі, і навпаки.

Розглянемо постановку задачі попередньої обробки даних для подальшого прогнозування.

1) Постановка задачі

Задана вибірка даних виду $\mathbf{x} = \{x_t\}, t = 1, \dots, T$, де \mathbf{x} – прогнозований процес, і передбачається, що значення прогнозованого процесу в будь-який момент часу однозначно визначається деякою функцією, залежною від значення істинної моделі процесу в цей момент час f_t , і деякої випадкової складової ε_t :

$$y_t = h(f_t, \varepsilon_t).$$

Загальна задача попередньої обробки даних розбивається на 2 підзадачі:

1) по можливості враховуючи інформацію про випадкову величину ε_t , спробувати таким чином перетворити вектор вихідних даних

\mathbf{x} , щоб максимально позбавитися від випадкової складової;

2) враховуючи особливості обраного методу розв'язання задачі прогнозування, перетворити вектор \mathbf{x}' , отриманий внаслідок вирішення попередньої підзадачі, щоб максимально збільшити ефективність використання обраного методу.

Особливо важливою є попередня обробка даних при використанні ШНМ в якості методу прогнозування. Правильна обробка може значно прискорити процес навчання мережі і навіть допомогти їй знайти більш глибокий мінімум СКП. Також, у роботі [20] показано вплив неврахованих викидів у вихідній вибірці даних на прогнозні властивості мережі (зокрема на СКП прогнозу на перевіірочній вибірці, яка значно збільшується за наявності викидів в більш ніж 10% вихідних даних).

Далі будуть розглянуті основні методи попередньої обробки даних для ШНМ.

Поліпшення інформативності вибірки

Оскільки нейронна мережа навчається, то кожен приклад в навчальній вибірці даних вносить свій внесок у зміну ваг між нейронами мережі. Як наслідок, наявність великої кількості однакових прикладів призведе до того, що мережа фактично підлаштується під ці приклади. Тому для поліпшення інформативності вибірки вихідних даних слід виключити часто повторювані однакові приклади.

Масштабування вихідних даних

Даний метод використовується при більшості основних методів прогнозування, оскільки він спочатку «зрівнює» вплив незалежних змінних на прогнозований процес.

Існує кілька основних методів центрування і нормалізації змінних.

1) $x_{in} = x_i - \bar{x}\sigma_x$, де x_{in} , x_i – нормовані і вихідні значення змінної відповідно, \bar{x} – середнє значення, σ_x – середньоквадратичне відхилення.

2) $x_{in} = x_i - \bar{x}$.

Згладжування вихідних даних

Згладжування вихідних даних застосовується для спроби позбавлення від випадкової складової і викидів в прогнозованому процесі. Існує безліч методів згладжування – перерахуємо основні.

2.6. Постановка задачі

При постановці завдання прогнозування найчастіше робиться припущення, що поведінка прогнозованого процесу постійна, і, таким чином, завдання можна вирішити шляхом знаходження потрібної моделі та оцінки її параметрів, використовуючи всі наявні дані. Проте, найчастіше поведінка прогнозованого процесу мінлива, що робить таку постановку задачі неправильною. Розглянемо постановку задачі прогнозування, яка явно враховує проблему непостійної поведінки прогнозованого процесу.

2.7. Класифікація методів

Усі методи прогнозування можна поділити на 3 групи.

1) *Глобальні методи.* До цієї групи входить найбільша кількість методів: усі методи даної групи знаходять єдиний набір параметрів деякої прогнозуючої моделі, використовуючи всі наявні дані. Найбільш відомі методи в даній групі: лінійна регресія, ARIMA, нейронні мережі, МГУА.

2) *Локальні методи* є протилежністю глобальним: замість однієї глобальної будується кілька локальних моделей, параметри для яких підбираються на деяких підмножинах вихідних даних. Дані методи враховують проблему непостійної поведінки прогнозованого процесу, але й мають свої недоліки:

- вони більш вимогливі до кількості навчальних даних – чим більше локальних моделей, тим більше потрібно даних;
- втрачається глобальна інформація – кожна локальна модель навчається тільки на своїх даних, що може призвести до зайвої «спеціалізації» та перенавчання моделей.

Найбільш відомі методи в даній групі: локальна регресія, кластеризація з регресією, локальна апроксимація та ін.

3) *Глобально-локальні методи.* Методи з цієї групи намагаються об'єднати найкраще з обох підходів: всі вихідні вхідні дані використовуються

для отримання глобальної інформації про прогнозований об'єкт (часто званої трендом), у той час як деякі підмножини даних використовуються для знаходження локальних особливостей. У роботі [51] в якості прикладу глобально-локального методу названий метод MARS. Авторами також був розроблений підхід «перетворення» глобальних параметричних методів у глобально-локальні методи [3]. Суть підходу полягає в навчанні спочатку однієї глобальної моделі, після чого навчається безліч локальних моделей з додатковим критерієм близькості параметрів локальних моделей до параметрів глобальної моделі; на стадії прогнозу просто вибираються параметри кількох «найближчих» локальних моделей, потім вони усереднюються і застосовуються до нових входів (аналогічно деяким локальним методам). Завдяки такому підходу, а саме використанню критерію близькості параметрів локальної та глобальної моделей, при побудові локальних моделей не втрачається глобальна інформація і самим локальним моделям не потрібна велика кількість даних. Даний підхід був застосований до лінійної регресії, а отриманий метод названий «лінійна регресія з динамічними вагами» – ще один приклад глобально-локального методу.

Найбільш перспективними є глобально-локальні методи, так як вони позбавлені недоліків інших 2 груп:

- на відміну від глобальних методів, параметри прогнозуючої моделі залежать від значень вхідних змінних – тобто використовується локальна інформація про прогнозований процес;
- на відміну від локальних методів, не губиться глобальна інформація, що дозволяє уникнути перенавчання локальних моделей і знизити необхідну кількість навчальних даних.

Саме тому розроблений авторами підхід дозволяє переходити від глобальних до глобально-локальних методів, дає можливість отримувати якіснішу прогнозуючу модель (з точки зору мінімуму оцінки помилки прогнозу моделі), застосовуючи його до існуючих глобальних методів. Для демонстрації розглянемо глобально-локальний метод, отриманий шляхом

застосування описаного підходу до глобального методу прогнозування з використанням нейронних мереж прямого поширення – найбільш відомою і застосовуваною архітектурою нейронних мереж для вирішення задачі прогнозування [44-46],[7].

2.8. Кластеризація даних

Дуже важливим кроком у локальних і глобально-локальних методах є розбиття вихідної вибірки на підвибірки, які будуть використані для побудови локальних моделей. Очевидно, що в найкращому випадку кожна підвибірка буде містити набір прикладів, схожих між собою і які відрізняються від прикладів з інших груп. Така неформальна постановка задачі повністю відповідає завданню кластеризації. Для формалізації завдання досить ввести лише функцію відстані між двома прикладами $D = f: X \times X \rightarrow \mathbb{R}^+$ і задати деякий критерій оптимальності розбиття на кластери I , який би враховував відстань між прикладами одного кластера і між прикладами з різних кластерів.

Одним з перших і найпростіших алгоритмів кластеризації є алгоритм k-середніх (k-means [52]). Не дивлячись на його простоту, алгоритм дає дуже хороші результати для реальних практичних проблем.

Алгоритм є версією EM-алгоритму (expectation maximization algorithm), застосовуваного також для поділу суміші Гауссіан. Він розбиває безліч елементів векторного простору на заздалегідь відоме число кластерів.

Основна ідея полягає в тому, що на кожній ітерації перерозраховується центр мас для кожного кластера, отриманого на попередньому кроці, потім вектори розбиваються на кластери знову відповідно до того, який з нових центрів виявився ближче за обраною метрикою.

Алгоритм завершується, коли на якийсь ітерації не відбувається зміни центру мас кластерів. Це відбувається за кінцеве число ітерацій, так як кількість можливих розбиттів кінцевої множини звичайна, а на кожному кроці сумарне квадратичне відхилення V не збільшується, тому зациклення

неможливе.

Інші відомі алгоритми кластеризації: нейронні мережі Кохонена, ієрархічна кластеризація, генетичні алгоритми та ін. У даному алгоритмі для кластеризації прикладів використовувався саме алгоритм k-середніх через його простоту та ефективність.

2.9. Методика перетворення методів для врахування неоднорідності даних

Для оцінки запропонованого методу було виконано його порівняння з наступними методами:

1. «Метод-предок» – нейронна мережа прямого поширення.
2. Повністю локальний варіант методу прогнозування з використанням нейронних мереж – розбиття всіх даних на кластери з подальшою побудовою і навчанням нейронної мережі для кожного кластера.
3. Покращений варіант чисто локального методу прогнозування з використанням нейронних мереж. Спочатку навчається нейронна мережа на всіх даних, і запам'ятовуються її ваги; після цього виконується розбиття даних на кластери та навчання нейронної мережі для кожного кластера, при цьому ваги кожної нейронної мережі будуть ініціалізовані запам'ятованою вагою мережею, навченою на всіх даних.

Порівняння виконувалося шляхом перевірки якості прогнозу вищеназаних методів на 11 різних наборах даних (часових рядах), які знаходяться у відкритому доступі в мережі Інтернет (більшість наборів було взято з сайту [53]). Для перевірки якості прогнозу кожного окремого методу використовувалися наступні параметри:

- кількість попередніх значень часового ряду використовуються в якості вхідних змінних (розмірність вкладення): 5;
- співвідношення розміру навчальної вибірки до розміру всієї вибірки: 0.7;
- метод розбиття вибірки на навчальну і перевіірочну: випадковим

чином;

- величина попередження прогнозу – на 2 значення вперед.

Всі методи були поставлені в рівні умови: використовувалися нейронні мережі однакової архітектури, однакові алгоритми навчання і один і той же метод розбиття на кластери – k-means.

Висновки

У цьому розділі було розглянуто проблему прогнозування при неоднорідних даних – тобто коли поведінка прогнозованого об'єкту змінюється (можливо, декілька разів) на спостережуваному періоді. В результаті було запропоновано нову постановку задачі прогнозування, що явно враховує можливу неоднорідність даних, та нову методику прогнозування, що дозволяє використовувати існуючі алгоритми прогнозування, які не враховують неоднорідність даних у навчальній вибірці, коли ця неоднорідність існує. Запропонована методика була застосована до нейронних мереж і перевірена на реальних даних: отримана середньоквадратична помилка була менша ніж у таких методів, як ШНМ, локальні ШНМ та інші.

РОЗДІЛ 3. ПРОГНОЗУВАННЯ ЧАСОВИХ РЯДІВ З ВИКОРИСТАННЯМ ШТУЧНИХ НЕЙРОННИХ МЕРЕЖ

3.1. Використання тільки ШНМ

Оскільки у загальному випадку істинна модель прогнозованого процесу невідома, то використання ШНМ, здатних за минулим спостереженням відтворити нелінійне відображення [25-33] виду:

$$x(k)=F(x(k-1),x(k-2),\dots,x(k-n_A))+e(k)=\hat{x}(k)+e(k),$$

у якості методу прогнозування є доволі логічним вибором (тут $\hat{x}(k)$ – оцінка-прогноз значення $x(k)$, отримана на виході нейронної мережі, що в даному випадку представляє собою нелінійну авторегресійну NAR [x] модель, $e(k)$ – помилка прогнозу).

Можливість і ефективність використання NAR-моделі в задачах прогнозування визначається теоремою Такенса про дифеоморфізм [34], яка встановлює існування порядку моделі n_A , який забезпечує як завгодно мале значення помилки $e(k)$, і універсальними апроксимуючими властивостями ШНМ.

Згідно з монографією [35], при використанні ШНМ у якості методу прогнозування, найбільш розповсюдженими є 2 підходи:

1. Використання багат шарових мереж прямого розповсюдження [36], вхідний шар яких складається з елементів чистої затримки z^{-1} з відводами.

На рис. 3.1 приведена архітектура багат шарової мережі прямого розповсюдження:

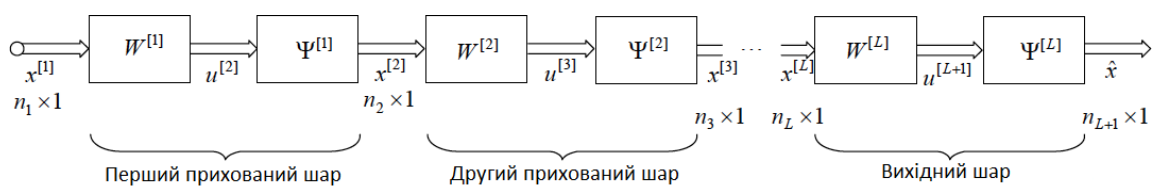


Рис. 3.1. L -шарова нейронна мережа прямого розповсюдження

На рис. 3.2 приведена схема стандартного формального статического нейрона l -го шару, $l=1,2,\dots,L$:

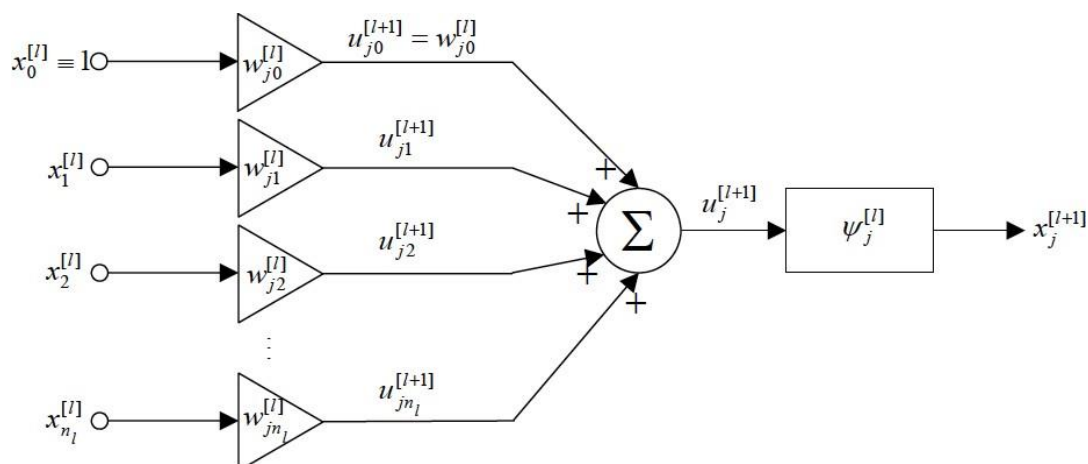


Рис. 3.2. Статичний нейрон

Така ШНМ складається зі стандартних статичних нейронів, що реалізують наступне нелінійне відображення:

$$x_j^{[l+1]} = \psi_j^{[l]}(u_j^{[l+1]}) = \psi_j^{[l]} \left(\sum_{i=0}^{n_l} u_{ji}^{[l+1]} \right) = \psi_j^{[l]} \left(\sum_{i=0}^{n_l} w_{ji}^{[l]} x_i^{[l]} \right).$$

Загальним недоліком прогнозуючих нейронних мереж на статичних нейронах є надто велика кількість параметрів, що налаштовуються, і низька швидкість навчання.

У зв'язку з цим, Е. Ваном було запропоновано [38-41] у прогнозуючих нейронних мережах замість статичних нейронів використовувати їх динамічні аналоги, у яких синаптичні ваги є цифровими адаптивними нерекурсивними фільтрами с кінцевою імпульсною характеристикою (КІХ-фільтри, FIR-фільтри) [42] так, як це показано на рис. 3.3.

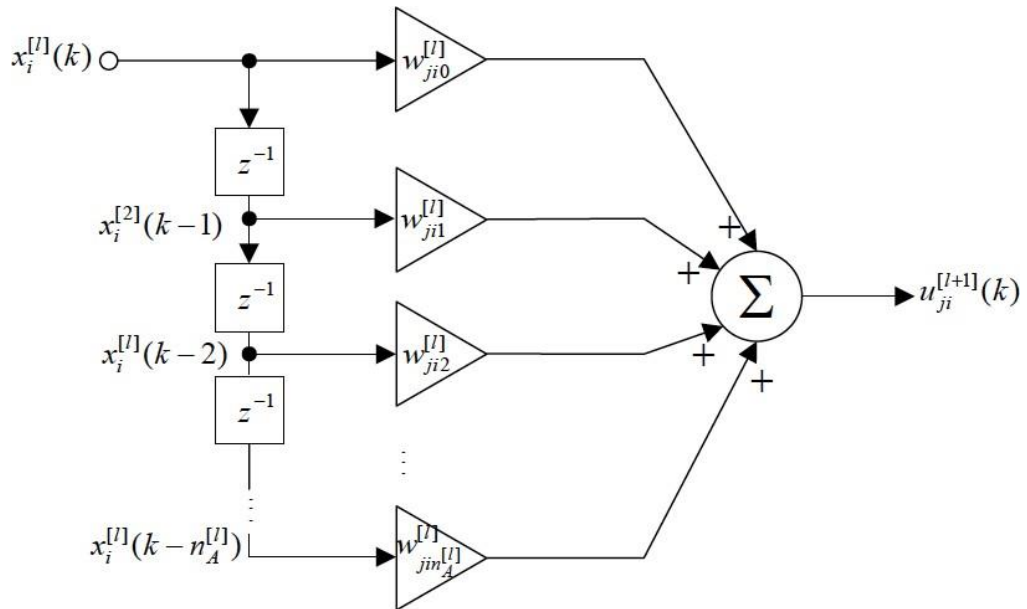


Рис. 3.3. Синапс–КІХ-фільтр

Тоді динамічний нейрон, що використовує ці синаптичні КІХ-ваги буде мати вигляд, як показано на рис. 3.4.

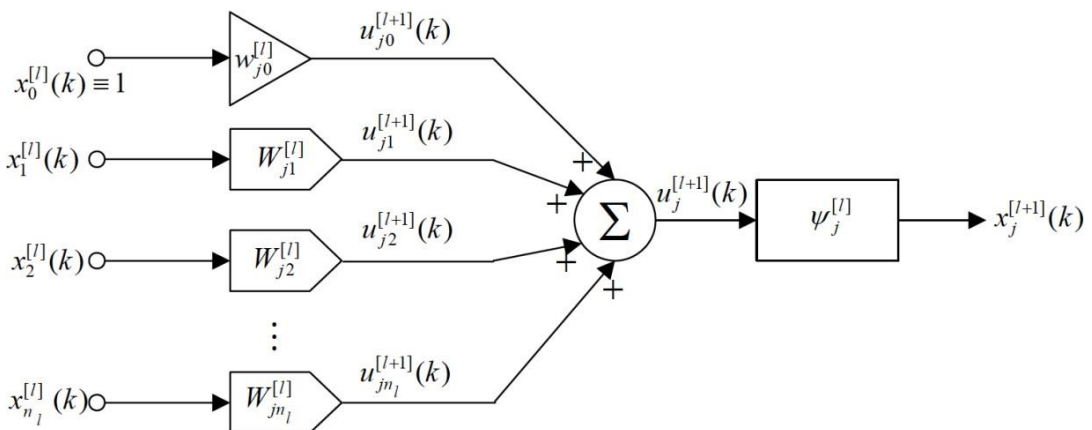


Рис. 3.4. Динамічний КІХ-нейрон

І хоча динамічний нейрон складається з $n_l (n_A^{[l]} + 1) + 1$ параметрів, що перевищує кількість синаптичних ваг звичайного нейрону, мережа, побудована з таких вузлів, має набагато менше параметрів, ніж мережа зі стандартною архітектурою на статичних нейронах з лініями затримки на вході. У роботі [39] було доведено, що в мережі на статичних нейронах з лініями затримки кількість параметрів зростає у геометричній залежності від

n_A , у той час як в ШНМ на динамічних нейронах кількість налаштовуваних параметрів є лінійною функцією від n_A і L . Для навчання ШНМ на динамічних нейронах в роботі [40] була введена градієнтна процедура, що отримала назву зворотного поширення помилки у часі.

3.2. Об'єднання підходів ШНМ і МГУА

Після аналізу різних існуючих на даний момент методів побудови прогнозуючої моделі, найбільш «гнучкими» методами можна однозначно назвати ШНМ і МГУА. Під гнучкістю розуміється відсутність необхідності визначення експертом чіткої моделі прогнозування, як наприклад, у лінійної регресії або методології Бокса-Дженкінса. Однак, при використанні ШНМ постає питання вибору оптимальної архітектури мережі, а при використанні МГУА – оптимальних опорних функцій.

Переваги та недоліки запропонованого алгоритму

Перевагою даного алгоритму в порівнянні з МГУА є те, що не потрібно явно задавати вид опорних функцій, необхідну залежність знаходитимуть нейронні мережі, які, як відомо, дуже добре справляються з цим завданням.

Даний алгоритм також позбавлений відомого недоліку ШНМ: при побудові мережі заздалегідь невідома її оптимальна складність, і занадто проста нейронна мережа може погано змоделювати процес, а надто складні мережі схильні до так званого «оверфітінгу», або перенавчання, у результаті якого мережу починає моделювати шум, присутній у навчальній вибірці, і як наслідок він показує погані результати на перевірочній вибірці. Пропонований же алгоритм на кожній ітерації використовує прості мережі, не схильні до перенавчання, але за рахунок каскадного ускладнення здатний прогнозувати дуже складні процеси.

Головним недоліком запропонованого методу є його ресурсомісткість, так як на кожній ітерації будується деяка кількість нейронних мереж, що може зайняти чимало часу. Але за рахунок використання чисельно-

оптимізованих алгоритмів навчання нейронних мереж і загальної оптимізації алгоритму, можна досягти порівняно швидкої роботи методу.

- **Визначення параметрів запропонованого методу**

Щоб побудувати остаточну прогнозуючу модель, використовуючи запропонований метод, необхідно визначити його параметри. Оскільки даний алгоритм є сумішшю підходу МГУА та штучних нейронних мереж, то завдання визначення його параметрів розпадається на дві підзадачі:

- 1) задача структурно-параметричного синтезу нейронної мережі;
- 2) задача визначення потрібних параметрів МГУА.

Для вирішення першої підзадачі необхідно більш детально розглянути апарат ШНМ.

Як вже говорилося вище, нейронні мережі складаються зі штучних нейронів. Штучний нейрон зазвичай представляють як деяку нелінійну функцію від єдиного аргументу – лінійної комбінації всіх вхідних сигналів. Дану функцію називають функцією активації або функцією спрацьовування, передавальною функцією (рис. 3.5).

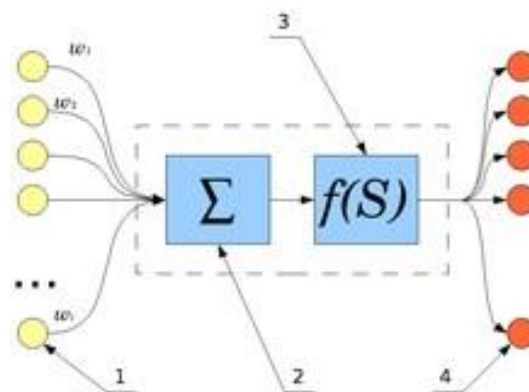


Рис. 3.5. Схема штучного нейрона: 1. Нейрони, вихідні сигнали яких надходять на вхід даному нейрону. 2. Суматор вхідних сигналів. 3. Обчислювач передавальної функції. 4. Нейрони, на входи яких подається вихідний сигнал даного нейрона

Математично нейрон являє собою зважений суматор, єдиний вихід

якого визначається через його входи і матрицю ваг наступним чином:

$$y = f(u), \text{ де } u = \sum_{i=1}^n \omega_i * x_i + \omega_0 * x_0.$$

Тут x_i і ω_i – сигнали на входах нейрона і ваги входів відповідно, функція u називається індукованим локальним полем, а $f(u)$ – активаційною функцією. Додатковий вхід x_0 і відповідна йому вага ω_0 використовуються для ініціалізації нейрона.

Оскільки ШНМ – це мережа взаємопов'язаних нейронів, то одним з основних параметрів, що характеризують нейронну мережу, є те, як саме здійснюється зв'язок між нейронами, тобто топологія мережі.

Далі розглянемо основні топології ШНМ.

Багатошаровий перцептрон (БП). Ця топологія являє собою мережу, що складається з декількох послідовно з'єднаних шарів нейронів, причому кожен нейрон шару i з'єднаний з усіма нейронами шару $i - 1$, де i – номер шару. Найперший шар є входами мережі, у процесі навчання або роботи мережі на нього подаються вихідні дані. За ним слідує один або кілька прихованих шарів, і останній шар

– вихідний шар, який являє собою виходи мережі (рис. 3.6).

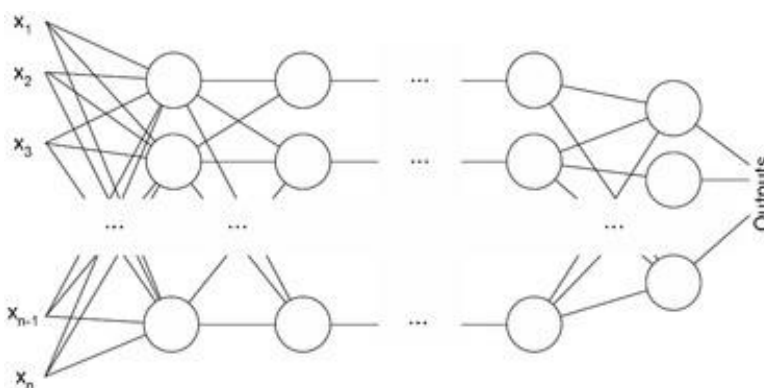


Рис. 3.6. Архітектура багатошарового перцептрона

Переваги: найбільша вивченість, простота, стійкість.

Недоліки: розглядає навчальні приклади незалежно один від одного,

тим самим упускаючи можливий зв'язок між прикладами.

Рекурентні нейронні мережі.

У рекурентних мережах зв'язки між нейронами можуть бути не тільки послідовними, а й зворотними, у тому числі зв'язок може бути між нейроном і самим собою.

Найбільш популярним окремим випадком рекурентних мереж є мережа Елмана [43]. Рекурентна мережа Елмана характеризується частковою рекурентністю у вигляді зворотного зв'язку між прихованим і вхідним шаром, реалізованою за допомогою одиничних елементів запізнювання (рис. 3.7).

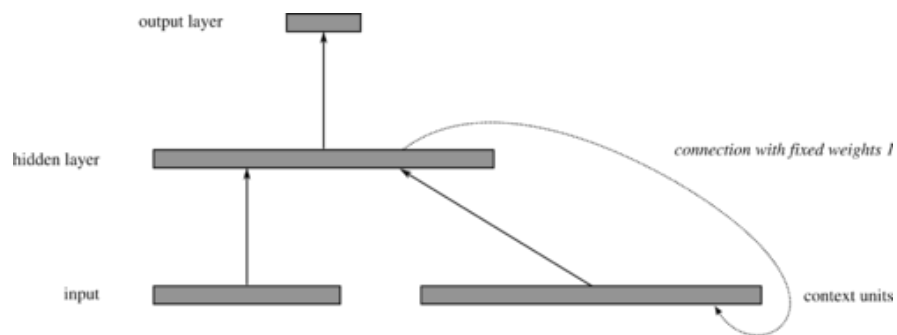


Рис. 3.7. Архітектура нейронної мережі Елмана

Переваги: рекурентні ШНМ враховують можливий зв'язок між прикладами, як наслідок краще підходять для моделювання послідовностей; їх здатність до узагальнення більша, ніж у БП.

Недоліки: нестійкість, складність, навчання в загальному випадку більш повільне, ніж у БП.

Постановка задачі структурно-параметричного синтезу нейронних мереж для вирішення задачі прогнозування

Побудова ШНМ полягає в послідовному вирішенні наступних завдань:

- вибір топології;
- структурний синтез ШНМ: визначення кількості і типів (числові,

бінарні, нечіткі) входів мережі, кількості прихованих шарів і кількості нейронів в них, кількості виходів мережі;

- вибір алгоритму навчання;
- параметричний синтез ШНМ: вибір типу активаційних функцій для шарів (сигмоїда, лінійна, порогова), визначення значень вагових коефіцієнтів.

Вирішення задачі структурно-параметричного синтезу нейронних мереж для вирішення задачі прогнозування

- У роботах [44] – [46] обґрунтовується вибір БП в якості топології нейронної мережі при вирішенні задачі прогнозування, а в роботах [47], [43] – вибір рекурентних нейронних мереж. У даній роботі в якості топології нейронної мережі була обрана топологія багатошарового персептрона через більшу її вивченість, а також стійкість і простоту.

- Теоретично було доведено [48], що багатошаровий персептрон з одним прихованим шаром і достатньою кількістю нейронів у ньому може апроксимувати як завгодно складну безперервну функцію. Таким чином, для завдання прогнозування не має сенсу вибирати більш складну структуру.

Кількість і тип входів може змінюватися в залежності від розміру навчальної вибірки і типу змінних в ній. Так як у запропонованому методі кількість входів мережі зазвичай (якщо не вводити додаткові змінні) дорівнює розміру передісторії, тобто кількості попередніх значень, що враховуються при прогнозі майбутнього значення, то починати слід з малої кількості входів ($\approx 0.05 * m \dots 0.15 * m$, де m – кількість прикладів в навчальній вибірці). Це дозволить побудувати більш просту прогнозуючу модель, а більшість практичних випробувань показують, що простіша модель зазвичай менш схильна до великих помилок прогнозу (хоча і здається, що більша кількість інформації має дати більш точну модель).

Кількість виходів в загальному випадку дорівнює одиниці.

Кількість нейронів в прихованому шарі може варіюватися, але знову ж,

бажано не вибирати дуже велику кількість, тому що це, по-перше, ускладнить структуру мережі, а по-друге, швидше за все, призведе до перенавчання. Так як у запропонованому алгоритмі ускладнення моделі відбувається з ряду в ряд аналогічно підходу МГУА, то кількість нейронів в прихованому шарі зазвичай вибирається малим ($\approx 3 \dots 10$).

- Найбільш відомий і поширений алгоритм навчання ШНМ – це алгоритм зворотного поширення помилки, який ґрунтується на методі градієнтного спуску. Однак, на сьогоднішній день цей алгоритм в чистому вигляді використовувати небажано, тому що розроблено безліч куди більш ефективних методів. У даній роботі в якості алгоритму навчання буде використовуватися алгоритм Левенберга-Марквардта [49].

Для задачі прогнозування прийнято для нейронів в прихованому шарі вибирати сигмоїду в якості активаційної функції, а для нейрона вихідного шару – лінійну активаційну функцію, оскільки значення прогнозованого процесу в загальному випадку можуть змінюватися від $(-\infty; +\infty)$, у той час як область значень сигмоїд лежить в межах $(0; 1)$.

Значення ж вагових коефіцієнтів зв'язків між нейронами визначаються в результаті навчання мережі.

Постановка задачі визначення параметрів МГУА для вирішення задачі прогнозування запропонованим методом

У рамках даного методу потрібно визначити такі параметри МГУА:

- конкретний алгоритм МГУА (як було сказано, МГУА – це сімейство алгоритмів);
- зовнішній критерій, за яким відбиратимуться «кращі» моделі для переходу на наступну ітерацію, і за яким буде визначатися момент зупинки алгоритму;
- вид опорних функцій.

Вирішення задачі визначення параметрів МГУА для вирішення задачі прогнозування запропонованим методом

- Два основних алгоритми МГУА – це комбінаторний алгоритм і багаторядний алгоритм.

1) Комбінаторний алгоритм.

У комбінаторних алгоритмах виконується перебір всіляких моделей із заданого базису з вибором кращої з цих моделей за заданим критерієм селекції.

При переборі складність часткових моделей, тобто число аргументів поступово нарощується від 1 до максимального числа n (числа аргументів базисного набору функцій).

Таким чином, загальна схема комбінаторного алгоритму включає наступні операції. Визначаються коефіцієнти всіх приватних моделей при складності $s=1..n$.

- Для кожної з них обчислюється значення зовнішнього індивідуального або комбінованого критерію складності селекції.

- Єдина модель оптимальної складності вибирається за мінімальним значенням критерію.

2) Багаторядний алгоритм.

Багаторядні алгоритми працюють за наступною схемою.

1-ий ряд – на основі даних таблиці спостережень будуються часткові описи об'єкта (найчастіше попарні), що наближають вихідну змінну q :

$$y_1=f_1(x_1,x_2),y_2=f_2(x_1,x_3),\dots,y_k=f_k(x_{n-1},x_n).$$

З цих моделей вибирається деяке число кращих за зовнішнім критерієм.

2-ий ряд – отримані змінні приймаються як аргументи другого ряду, і знову будуються всі часткові описи:

$$z_1=\varphi_1(y_1,y_2),z_2=\varphi_2(y_1,y_3),\dots,z_l=\varphi_l(y_{F1-1},y_F).$$

З них за зовнішнім критерієм відбирається F_2 кращих моделей в якості змінних наступного ряду і т.д. Ряди нарощуються до тих пір, поки

знижується значення зовнішнього критерію.

У даній роботі пропонується використовувати багаторядний підхід МГУА, оскільки йому віддається перевага при невеликій кількості прикладів в навчальній вибірці, і до того ж, він дозволяє використовувати ШНМ в якості часткових описів. Порівняння якості прогнозу методу і ШНМ

Для перевірки алгоритму використовуємо публічно доступні дані про продажі літаків (загальна кількість проданих в США літаків за роками) з 1947 по 2011 рік [50]. Дані попередньо опрацюємо за допомогою алгоритму Tukey 53H.

При побудові моделі за допомогою алгоритму використовувалися наступні параметри:

- розмірність, що використовується при вкладенні часового ряду для отримання відповідних матриць прикладів: $k = 5$;
- співвідношення розмірів навчальної і перевіркової вибірок: $0.7m / 0.3 m$, де m – загальна кількість прикладів. Для отримання навчальної вибірки використовувалися перші $0.7 m$ прикладів;
- структура нейронних мереж: один нейрон у вихідному шарі, один прихований шар з 3 нейронами, 5 нейронів у вхідному шарі (відповідно до розмірності вкладення);
- при відборі мереж СКП розраховувалася на всіх наявних прикладах;
- для створення матриці прикладів для наступної ітерації використовувалися виходи 3 мереж з мінімальною СКП і дві вхідні змінні, які були входами мережі з найменшою СКП.

Для порівняння якості прогнозу алгоритму використовуємо прогноз, отриманий за допомогою БП з одним вихідним нейроном, одним прихованим шаром з 10 нейронами і 10 вхідними нейронами. БП, так як і алгоритм, навчався використовуючи приклади тільки з навчальної вибірки. Для навчання використовувався алгоритм Левенберга-Марквардта. Відомо, що

під час навчання, ШНМ схильні «застрягати» в локальних мінімумах функції помилки, тому було навчено близько 10 ШНМ і обрано мережу з мінімальною СКП.

Порівняння прогнозів, отриманих за допомогою ШНМ і алгоритму, показано на рис. 3.8.

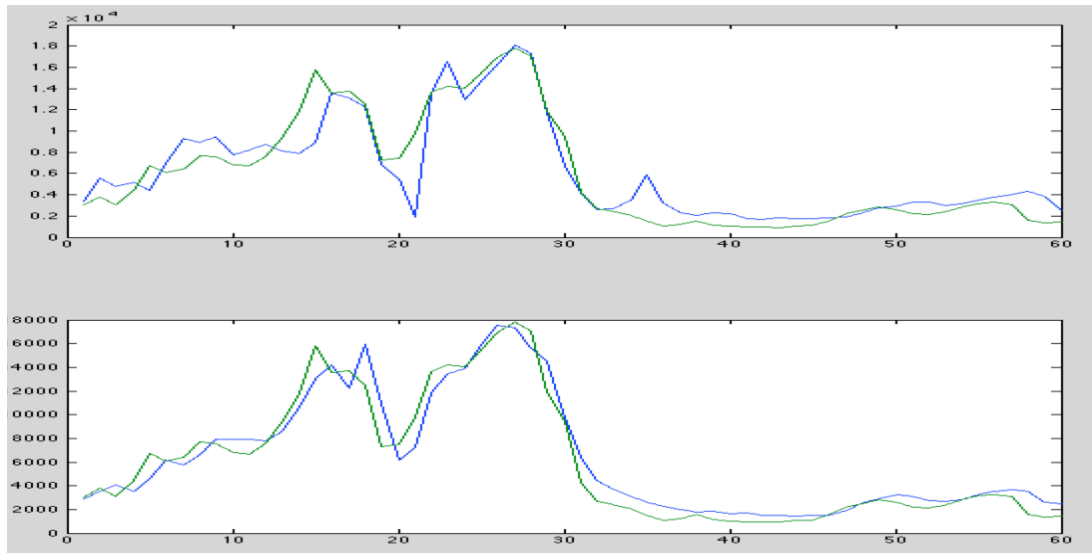


Рис. 3.8. Порівняння прогнозів, отриманих за допомогою ШНМ (зверху) і алгоритму (знизу). Зелений графік – вихідні дані, сині графіки – прогнози

Значення СКП: ШНМ – 0.0613, алгоритму – 0.0255.

На графіках можна помітити, що навчений персептрон прогнозує значення перевірконої вибірки гірше (права частина графіків), у той час як помилки прогнозу, отриманого за допомогою алгоритму, практично однакові для обох – навчальної та перевірконої – вибірок, що зазвичай свідчить про більш точну модель прогнозованого процесу. Алгоритм також більш точно спрогнозував «екстремальні» значення, що є дуже важливим фактором при прогнозуванні попиту.

3.3. Комплексування декількох методів прогнозування

Для поліпшення якості прогнозу можливо використовувати комплексування оцінок, отриманих за допомогою різних моделей. Під

комплексуванням розуміється зважена сума оцінок, отриманих за допомогою згенерованого набору моделей. Вагові коефіцієнти визначаються за допомогою зовнішнього критерію оптимальності моделей – дисперсії на екзаменаційній вибірці. Множина моделей отримується перебором варіантів розбиття вихідної вибірки на підвибірки і перебором різних методів прогнозування. Таким чином, маючи k_1 варіантів розбиття на підвибірки і k_2 методів прогнозування, отримуємо $k_1 * k_2$ різних моделей. Для отримання остаточного прогнозу, маючи вектор вхідних даних, необхідно:

Подати цей вектор на вхід кожної моделі, таким чином отримавши вектор оцінок $\mathbf{y} = [y^1, \dots, y^{k_1 * k_2}]$;

1. Отримати остаточний прогноз \hat{y} як зважену суму елементів вектора оцінок \mathbf{y} .

Ці кроки вимагають визначення вагових коефіцієнтів $a_i, i = 1 \dots k_1 * k_2$.

3.3.1. Розбиття вихідної вибірки на підвибірки

- По порядку.

У навчальну вибірку відбираються перші $C_1 * N$ точок, у перевірочну – наступні $C_2 * N$ точок, і в екзаменаційну – точки, що залишилися, тобто $(1 - C_1 - C_2) * N$ (де N – загальна кількість точок, $C_1 + C_2 < 1; C_1, C_2 > 0$ – коефіцієнти, зазвичай вибирають $C_1 = 0.6, C_2 = 0.2$).

- Випадковим чином.

Аналогічно попередньому методу, тільки точки відбираються не по порядку, а випадковим чином, але пропорції між підвибірками зберігаються.

- Кожен i -й.

У перевірочну вибірку відбирається кожна i -а точка, з решти точок в екзаменаційну відбирається кожна j -а, всі точки, що залишилися, відбираються в навчальну вибірку (зазвичай вибирають $i = 3, j = 4$)

- За дисперсією.

Усі точки ранжуються за дисперсією (під точкою розуміється один приклад) і потім відбираються в вибірки аналогічно першому методу.

3.3.2. Отримання моделей за допомогою перебору методів

Для кожного з відібраних методів розбиття на вибірки і методів прогнозування будується модель прогнозованого процесу. У роботі [5] використовуються такі методи прогнозування: ШНМ, МГУА, комбінація МГУА та ШНМ. Таким чином, при 4 способах розбиття на вибірки і 3 методах прогнозування отримуємо $4 * 3 = 12$ різних моделей.

Графік штучного часового ряду з істинною моделлю та доданим нормальним шумом наведений на рис. 3.9.

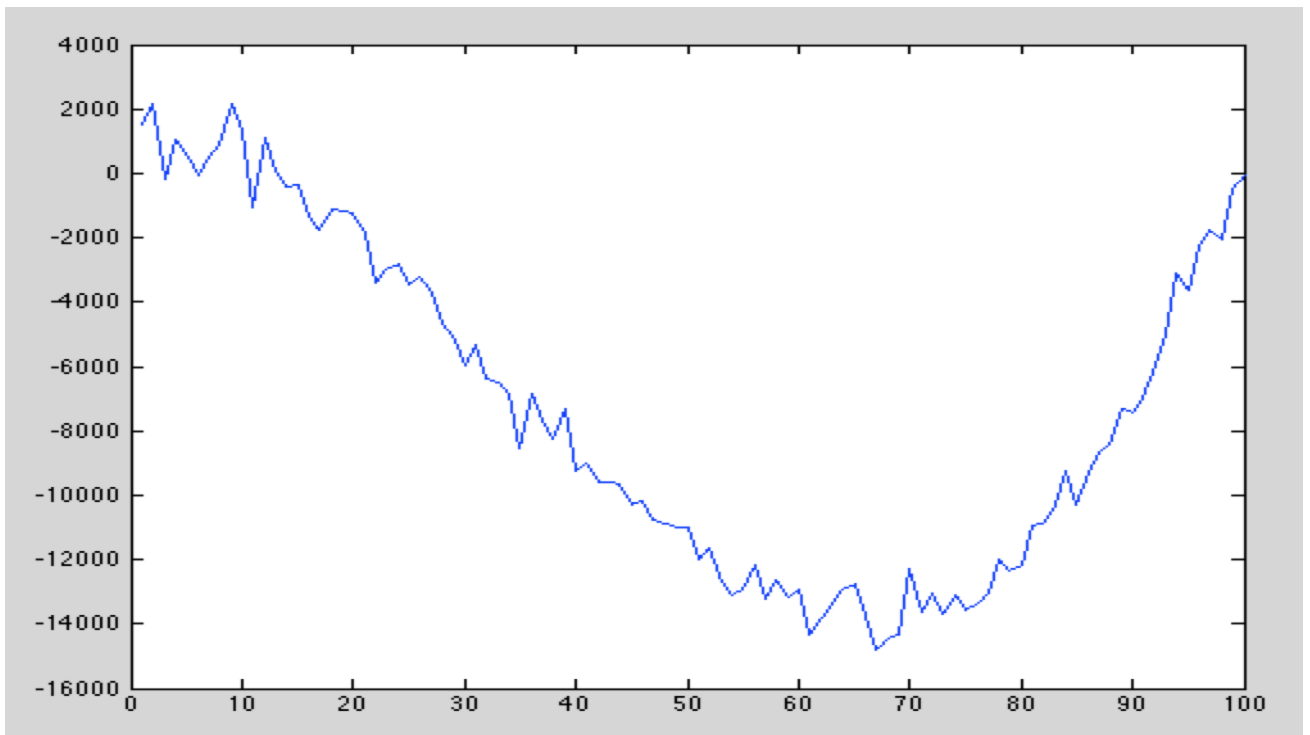


Рис. 3.9. Штучний часовий ряд

Після цього побудуємо прогнозуючу модель із застосуванням алгоритму МГУА + ШНМ та із застосуванням комплексування оцінок декількох методів. НСКП прогнозу, отриманого за допомогою алгоритму МГУА + ШНМ, дорівнює 0.0038 (рис.3.10).

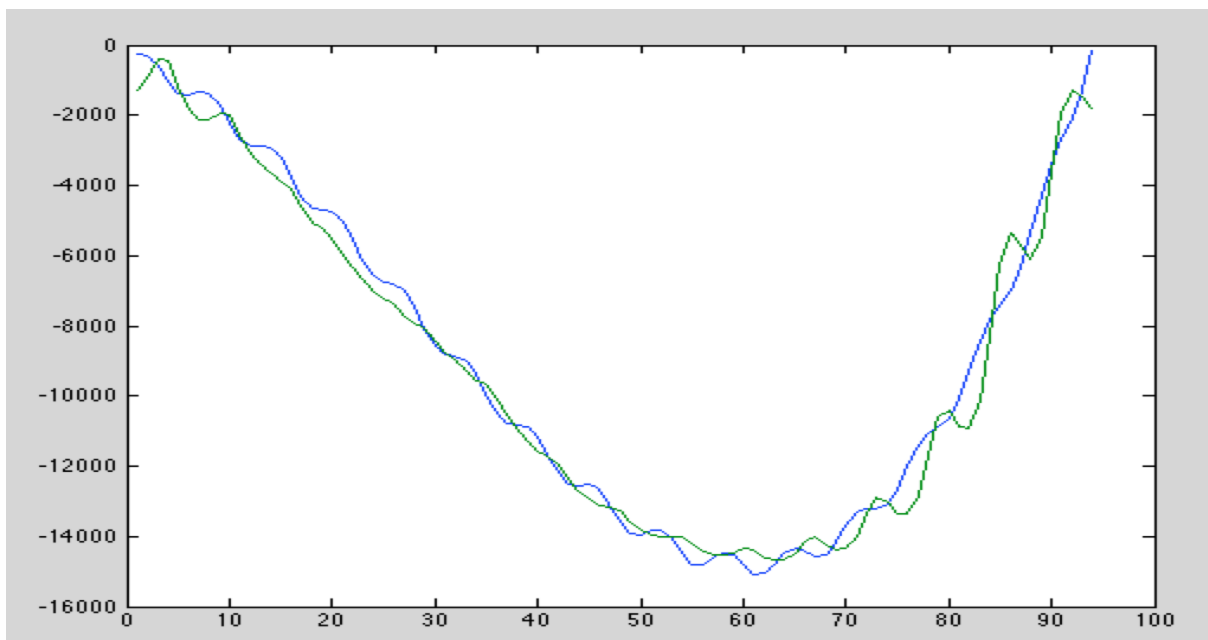


Рис. 3.10. Прогноз, отриманий за допомогою алгоритму МГУА + ШНМ

НСКП прогнозу, отриманого за допомогою комплексування декількох алгоритмів (МГУА + ШНМ, МГУА, ШНМ), дорівнює 0.0020 (рис. 3.11).

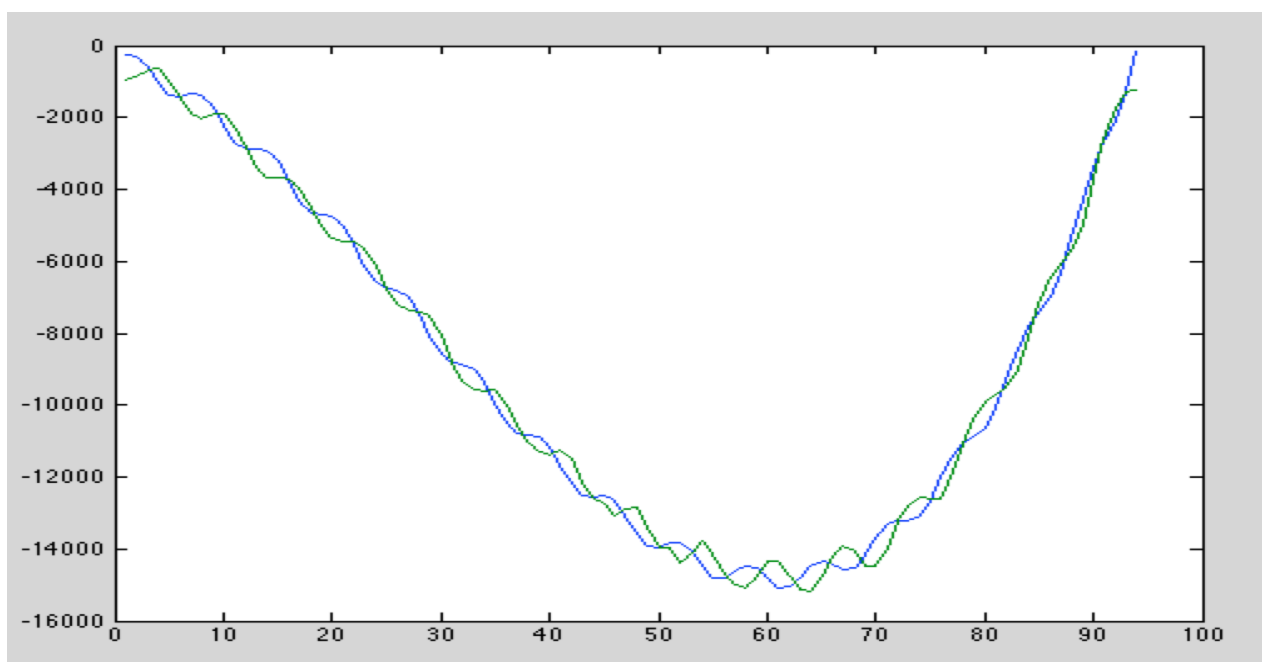


Рис. 3.11. Прогноз, отриманий за допомогою комплексування декількох алгоритмів

Як добре видно з рисунків, комплексування дійсно значно покращує якість отриманого прогнозу.

3.4. Програмна система прогнозування часових рядів

Використовуючи вищеописані методи прогнозування була розроблена програмна система прогнозування часових рядів (ПСПЧР).

Дана система складається з п'яти основних компонентів.

- Компонент попередньої обробки даних – ці компоненти дають користувачеві можливість застосовувати до даних різні перетворення та / або фільтри (такі як фільтр змінного середнього, медіанний фільтр і так далі).
- Компоненти побудови прогнозуючої моделі – дані компоненти дозволяють будувати прогнозуючу модель, використовуючи різні методи прогнозування (ШНМ, МГУА).
- Компонент побудови графіків – компонент дозволяє візуалізувати часові ряди, у тому числі кілька рядів на одному графіку.
- Компонент завантаження даних – дозволяє завантажувати часові ряди з файлів.
- Компонент збереження даних – дозволяє зберігати часові ряди в файл.

Далі будуть описані: структура системи, список основних вирішуваних завдань, опис входів-виходів для кожного завдання, схема алгоритму роботи системи, опис інтерфейсу користувача і приклади роботи з системою.

3.4.1. Структура системи

Структура системи відповідає стандартній трирівневій схемі «Інтерфейс користувача» \Leftrightarrow «Логіка додата» \Leftrightarrow «Доступ до даних», де кожен рівень може «спілкуватися» тільки з сусідніми рівнями. Також, для спілкування між рівнями використовуються «інтерфейси» рівнів – тобто, не кожен клас або компонент певного рівня може безпосередньо спілкуватися з будь-яким класом (компонентом) рівня-сусіда – це призвело б до занадто сильної і непередбачуваної пов'язаності між компонентами додатка – а тільки

одному спеціально призначеному «інтерфейсному» класу дозволено спілкуватися з інтерфейсними класами рівнів-сусідів.

Високорівнева структура додатка виглядає наступним чином.

- Контролер головної форми додатка реагує на дії користувача (перетягування елементів, з'єднання елементів, кліки по елементах), і передає (при необхідності) інформацію про дії користувача в інтерфейсний клас рівня

«Логіка додатка».

- Інтерфейсний клас рівня «Логіка додатка» також реагує на дії користувача і сигналізує (при необхідності) контролеру головної форми про необхідність оновити форму. Для реагування на дії користувача інтерфейсний клас рівня «Логіка додатка» може також використовувати інтерфейсний клас рівня «Доступ до даних» або інтерфейси внутрішніх компонентів рівня «Логіка додатка» – кожен елемент («Вхідні дані»,

«Обробка», «Побудова моделі») має власний інтерфейс, і викликається (при необхідності) інтерфейсним класом рівня «Логіка додатка». Цей клас також відповідає за злагоджену роботу всіх компонентів рівня «Логіка додатка» в цілому, оскільки самі компоненти не знають про існування інших компонентів – вони являють собою відокремлені одиниці, які можна використовувати повторно (reusable components).

- Інтерфейс рівня «Доступ до даних» взагалі «не знає» нічого про логіку додатка, він лише надає методи виду «Дістати дані з файлу», «Завантажити дані в файл» і т.д.

- Форма налаштування кожного елемента працює тільки з відповідним елементом за чітко визначеним інтерфейсному виду «Встановити значення за ключем» і «Отримати значення за ключем». Таким чином, коли користувач налаштовує значення певного параметра, контролер форми налаштування просто викликає інтерфейсний метод «Встановити значення за ключем» для відповідного ключа, а вже сам

компонент реагує потрібним чином.

3.4.2. Схема алгоритму роботи системи

Найбільш інтенсивне застосування всієї структури програми починається після запуску змодельованої користувачем системи на виконання. При цьому:

- інтерфейсний клас рівня «Логіка додатку» перевіряє побудовану користувачем схему на наявність помилок. У випадку, якщо помилки наявні

– запуск припиняється і користувачу повідомляється про відповідні помилки; в іншому випадку, інтерфейсний клас рівня «Логіка додатку» відшукує всі

«вхідні» (ті, що тільки генерують дані) елементи зі списку всіх елементів, і асинхронно викликає у них метод «Виконати» (всі елементи реалізують відповідний інтерфейс базового елементу);

- вхідні елементи відпрацьовують (при цьому вони можуть викликати методи інтерфейсного класу «Доступ до даних»), кожен елемент після закінчення сигналізує інтерфейсному класу «Логіка додатка», що він закінчив. Якщо під час виконання виникає виняткова ситуація, то робота елемента припиняється негайно, і про це також сигналізується

інтерфейсному класу «Логіка додатку». Крім того, якщо в результаті виконання елемента користувач повинен отримати певну інформацію – у вигляді повідомлення, або графіку – ця інформація відображається на екрані;

- інтерфейсний клас рівня «Логіка додатку» реагує на закінчення роботи наступного елемента, і якщо відбувся виняток – причина виключення запам'ятовується, після чого він очікує на закінчення всіх вже запущених елементів, і повідомляє контролеру головної форми про «аварійне» закінчення разом з його ймовірною

причиною. Якщо ж робота закінчилася нормальним чином – відшуковуються всі елементи, які чекали закінчення цього елемента і якщо вони готові почати – у них також асинхронно викликається метод «Виконати». Якщо елемент, який закінчив роботу, останній – про це повідомляється контролеру головної форми, щоб він міг відключити режим очікування.

3.5. Приклад роботи з додатком

Складемо найпростішу схему, у якій перші 180 точок деякого часового ряду будуть використані для побудови моделі, використовуємо простий багатосаровий персептрон, а 120 точок, що залишилися, будуть прогнозуватися побудованою моделлю. Так само, для позбавлення від непотрібних шумів слід попередньо обробити часовий ряд медіанним фільтром.

Отже, спочатку подивимося, як будуть виглядати вихідні і згладжені медіанним фільтром дані. Для цього перетягнемо компоненти «Джерело даних», «Обробка даних» та «Графік» і з'єднаємо їх потрібним чином (рис.3.12).

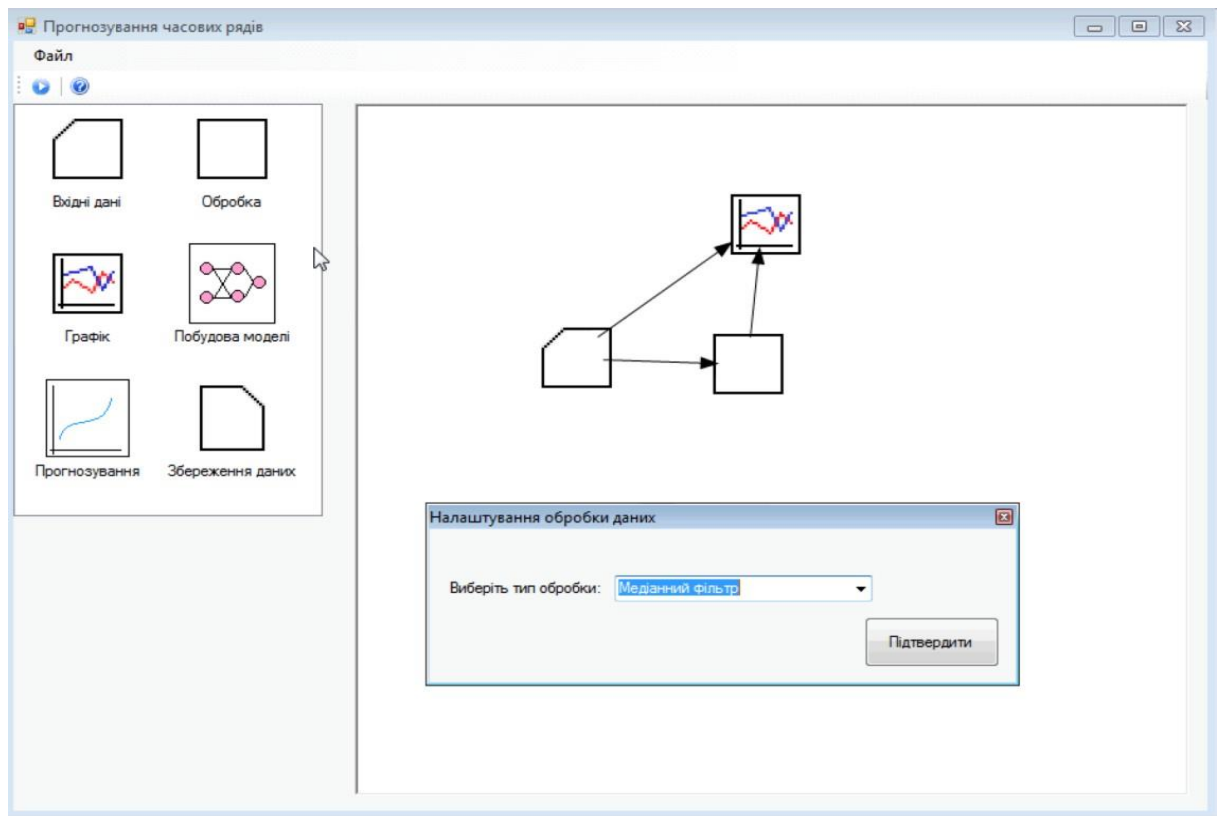


Рис. 3.12. Приклад роботи з додатком – побудова графіка

Після цього запусимо побудовану схему і отримаємо наступний графік (рис. 3.13).

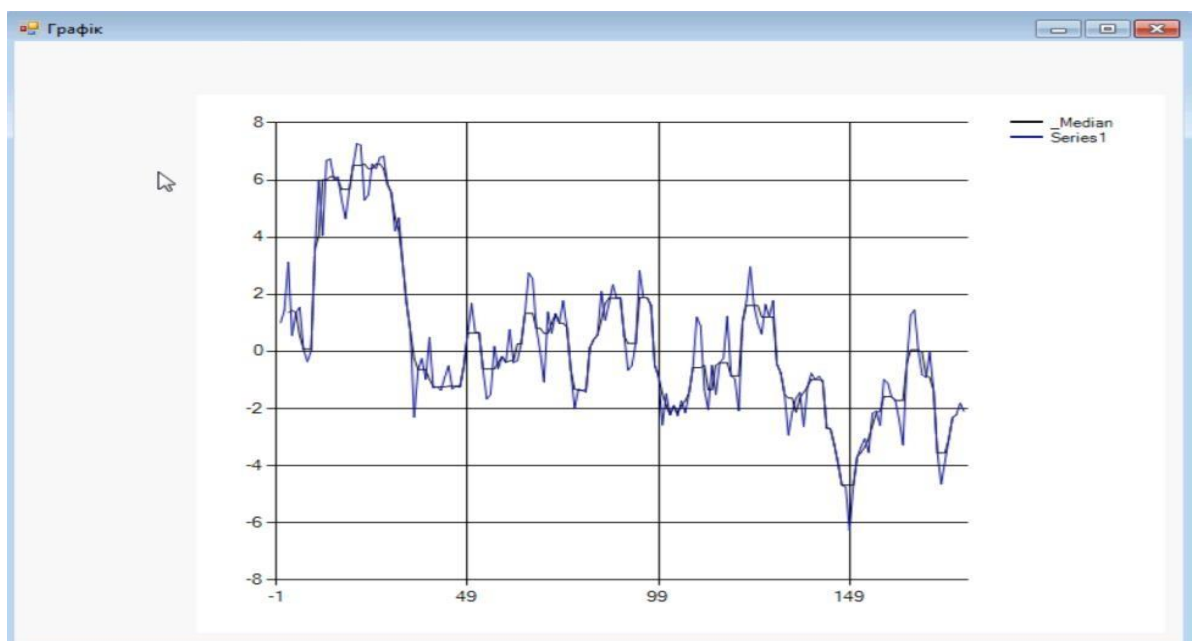


Рис. 3.13. Приклад роботи з додатком – побудований графік

Далі добудуємо схему таким чином, щоб модель будувалася за допомогою БП з цих 180 точок, а прогноз робився за рештою 120 точкам, які знаходяться в іншому файлі (рис.3.14).

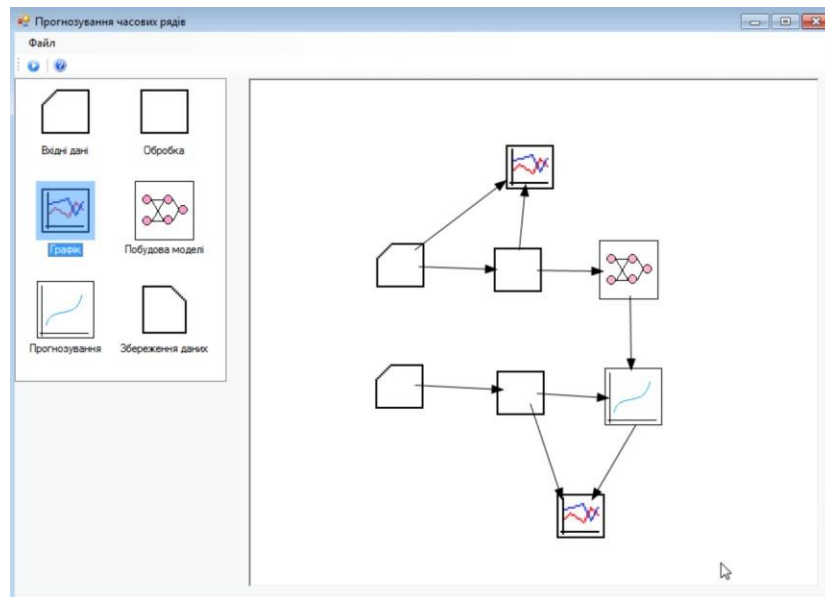


Рис. 3.14. Приклад роботи з додатком – побудова системи для прогнозування часового ряду

Запустимо побудовану схему на виконання і отримаємо наступний результат (рис.3.15).

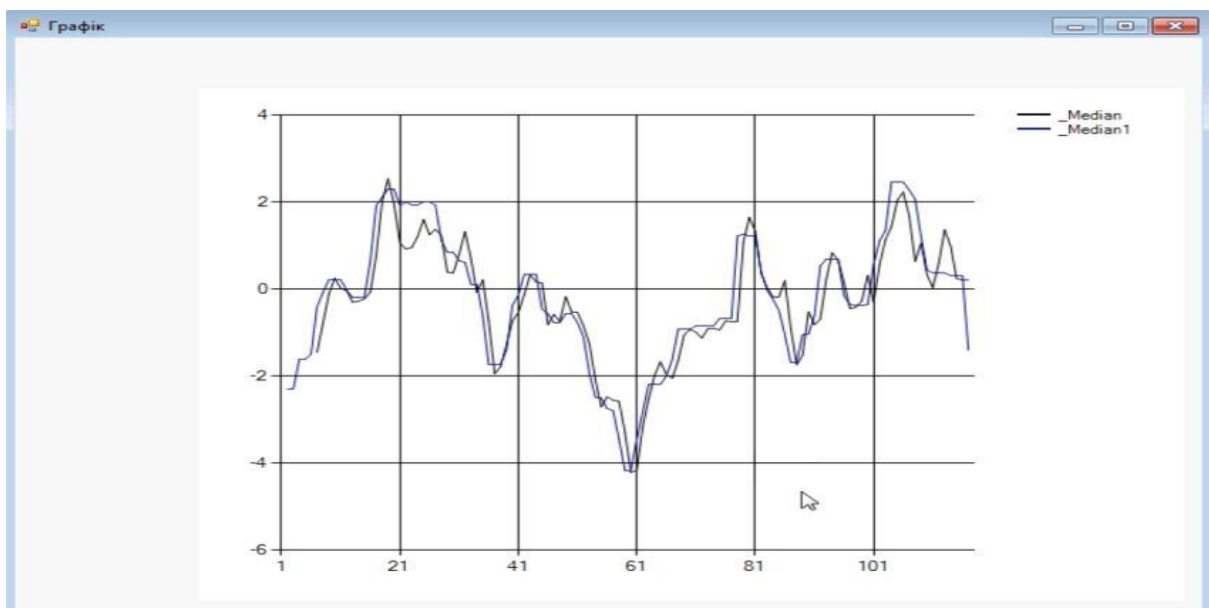


Рис. 3.15. Приклад роботи з додатком – результат прогнозу

Як бачимо, побудована модель дійсно є адекватною і добре виконала узагальнення даних.

Висновки

У цьому розділі було розглянуто певні існуючі методи прогнозування, що базуються на використанні ШНМ та запропоновано два нових:

метод, що об'єднує підходи ШНМ та МГУА для знаходження оптимальної структури нейронної мережі для конкретної задачі прогнозування;

комплексування декількох прогнозуючих моделей, отриманих шляхом розбиття наявних даних на певну кількість вибірок і використання декількох різноманітних методів прогнозування для навчання моделей (кожна модель навчається на одній з отриманих вибірок).

Вхідні дані були перевірені першим методом, та дали середньоквадратичну помилку, меншу, ніж у таких існуючих методів як ШНМ та МГУА.

РОЗДІЛ 4. ЕКОНОМІЧНИЙ РОЗДІЛ

4.1. Визначення трудомісткості розробки програмного забезпечення

- 1) передбачувана кількість операторів - 3200
- 2) коефіцієнт складності програми - 1,8
- 3) коефіцієнт редагування програми у ході її розробки - 0,09
- 4) часова заробітна платня програміста, грн/год - 65,0
- 5) вартість машиночасу, грн/год - 19,0

Нормування праці в процесі створення ПЗ істотно ускладнено в силу творчого характеру праці програміста. Тому трудомісткість розробки ПЗ може бути розрахована на основі системи моделей з різною точністю оцінки. Трудомісткість розробки ПЗ можна розрахувати за формулою:

$$t = t_o + t_u + t_a + t_n + t_{отл} + t_d, \text{ ЛЮДИНО-ГОДИН,}$$

де t_o - витрати праці на підготовку й опис поставленої задачі (приймається 50);

t_u - витрати праці на дослідження алгоритму рішення задачі;

t_a - витрати праці на розробку блок-схеми алгоритму;

t_n - витрати праці на програмування по готовій блок-схемі;

$t_{отл}$ - витрати праці на налагодження програми на ЕОМ;

t_d - витрати праці на підготовку документації.

Складові витрати праці визначаються через умовне число операторів у ПЗ, яке розробляється. Умовне число операторів (підпрограм):

$$Q = q \cdot C \cdot (1 + p),$$

де q - передбачуване число операторів;

C - коефіцієнт складності програми;

p - коефіцієнт кореляції програми в ході її розробки.

Звідки:

$$Q = 3200 \cdot 1,8 (1 + 0,09) = 6278$$

Витрати праці на вивчення опису задачі ти визначається з урахуванням уточнення опису і кваліфікації програміста:

$$t_u = \frac{Q \cdot B}{(75..85) \cdot k}, \text{ людино-годин}$$

де B - коефіцієнт збільшення витрат праці внаслідок недостатнього опису задачі;

k - коефіцієнт кваліфікації програміста, обумовлений від стажу роботи з даної спеціальності.

Звідки:

$$t_u = \frac{6278 \cdot 1,3}{77 \cdot 1,2} = 88,3$$

Витрати праці на розробку алгоритму рішення задачі:

$$t_a = \frac{Q}{(20..25) \cdot k}, \text{ людино-годин}$$

Звідки:

$$t_a = \frac{6278}{22 \cdot 1,2} = 237,8$$

Витрати на складання програми по готовій блок-схемі:

$$t_n = \frac{Q}{(20..25) \cdot k}, \text{ людино-годин}$$

Звідки:

$$t_n = \frac{6278}{22 \cdot 1,2} = 237,8$$

Витрати праці на налагодження програми на ЕОМ:

- за умови автономного налагодження одного завдання:

$$t_{отл} = \frac{Q}{(20..25) \cdot k}, \text{ люДИНО-ГОДИН}$$

- за умови комплексного налагодження завдання:

$$t_{отл}^k = 1.5 \cdot t_{отл}, \text{ люДИНО-ГОДИН}$$

Звідки:

$$t_{отл} = \frac{6278}{22 \cdot 1,2} = 237,8$$

$$t_{отл}^k = 1.5 \cdot 237,8 = 356,7$$

Витрати праці на підготовку документації:

$$t_d = t_{др} + t_{до}, \text{ люДИНО-ГОДИН},$$

де $t_{др}$ - трудомісткість підготовки матеріалів і рукопису.

$$t_{др} = \frac{Q}{(15..20) \cdot k}, \text{ люДИНО-ГОДИН}$$

$t_{до}$ - трудомісткість редагування, печатки й оформлення документації

$$t_{до} = 0.75 \cdot t_{др}, \text{ люДИНО-ГОДИН},$$

Звідки:

$$t_d = \frac{6278}{17 \cdot 1,2} + \frac{0,75 \cdot 6278}{17 \cdot 1,2} = 307,7 + 230,8 = 538,5$$

Звідки отримуємо, що трудоміткість розроблення програмного забезпечення складає:

$$t = 50 + 88,3 + 237,8 + 237,8 + 237,8 + 538,5 = 1390,2 \text{ людино-години.}$$

4.2. Витрати на створення програмного забезпечення

Витрати на створення ПЗ $K_{\text{по}}$ включають витрати на заробітну плату виконавця програми $Z_{\text{зп}}$ і витрат машинного часу, необхідного на налагодження програми на ЕОМ

$$K_{\text{по}} = Z_{\text{зп}} + Z_{\text{мв}}, \text{ грн.}$$

Заробітна плата виконавців визначається за формулою:

$$Z_{\text{зп}} = C_{\text{пр}} \cdot t = 65 \cdot 1390,2 = 90363 \text{ грн,}$$

де: t - загальна трудоміткість, людино-годин;

$C_{\text{пр}}$ - середня годинна заробітна плата програміста, грн/година.

Вартість машинного часу, необхідного для налагодження програми на ЕОМ:

$$Z_{\text{мв}} = t_{\text{отл}} \cdot C_{\text{мч}} = 237,8 \cdot 19 = 4518,2 \text{ грн,}$$

де $t_{\text{отл}}$ - трудоміткість налагодження програми на ЕОМ, год.

$C_{\text{мч}}$ - вартість машино-години ЕОМ, грн/год.

Звідки отримуємо, що витрати на створення програмного забезпечення складає:

$$K_{\text{ПО}} = 4518,2 + 90363 = 94881,2 \text{ грн.}$$

Визначені в такий спосіб витрати на створення програмного забезпечення є частиною одноразових капітальних витрат на створення АСУП.

Очікуваний період створення ПЗ:

$$T = \frac{t}{B_k \cdot F_p}, \text{ місяців,}$$

де B_k - число виконавців;

F_p - місячний фонд робочого часу (при 40 годинному робочому тижні $F_p=176$ годин).

Звідки:

$$T = \frac{1390,2}{1 \cdot 176} = 7,89 \approx 8 \text{ місяців.}$$

4.3. Маркетингові дослідження ринку збуту розробленого програмного продукту

Розроблена система виконує функцію часового аналізу та передбачення аварійних ситуацій. Програма зчитує показники датчиків контролю на шахтах та за допомогою алгоритмів нейронних мереж передбачає у яких часових рядах які прилади можуть відмовити, що в свою чергу дозволяє своєчасно замінити їх та визначити причини поломки і уникнути аварії.

За допомогою маркетингового дослідження було виявлено, що розроблена програма на сьогодні не має аналогів. Даний програмний продукт має дуже вузьку цільову аудиторію, а саме гірничо-промислові компанії та підприємства, діяльність котрих пов'язана з шахтними видобутками.

Подібні програми націлені на підвищення безпеки життя і праці співробітників вище названих організацій. Навіть незначне відхилення в роботі таких підприємств несе за собою величезні витрати грошей і часу. Програма попереджає аварійні ситуації від незначних до катастрофічних, що дає змогу своєчасно звернути увагу на наступаючу проблему, завдяки чому її можна уникнути до проявлення і таким чином обійтися без форс-мажорів і наслідків небезпечних ситуацій. Це, в свою чергу дозволяє уникнути в найкращому випадку незапланованих суттєвих витрат, і, в найгіршому випадку людських жертв.

На сьогодні неможливо розробити проект з таким функціоналом для широкого користування, тому що вхідними даними для програми виступають сигнали з вимірних та контролюючих приладів. Такі прилади компанії закупають в залежності від бюджету, географічного розташування і інших параметрів, таких як глибина шахти, її протяжність, тип робіт, тощо.

Окреме підприємство замовляє програмний продукт згідно з чітким технічним завданням, у якому детально розписані вимоги до програми, до її функціональних можливостей та надає перелік апаратного забезпечення, з яким має взаємодіяти програма. Після цього аналітики з компанії, що надає послуги розраховують час, необхідний на розробку та ціну на продукт.

Цей метод розробки програмного забезпечення ефективний тим, що ми бачимо, що потрібно замовнику, та з умовами договору, після досягнення мети максимально швидко продамо розроблений програмний продукт.

Також дуже важливою складовою є захист інформації від злодіїв. Для захисту та адміністрування комп'ютерної мережі підприємства, що займається розробкою програмного забезпечення, важливо попередити запис інформації на носії та встановлення з них сторонніх програм. За допомогою автоматизованої системи контролю з'ємних носіїв

адміністратор може контролювати доступ користувачей до всіх вбудованих та з'ємних носіїв машини, адаптерам WiFi та Bluetooth, а також до USB, WireFire, інфрочервоним, COM та LPT-портам.

При продажі проекту буде використовуватися метод прямих продажів. Прямі продажі являють собою форму організації роздрібної торівлі поза стаціонарних торгівельних точок. Можуть практикуватися як окремими торгівцями, так і організаціями.

4.4. Оцінка економічної ефективності впровадження програмного забезпечення

Програмне забезпечення націлене на зменшення ризику аварійних ситуацій на шахтах. Щорічно на Україні трапляється декілька аварійних випадків, які несуть за собою витрати на відновлення підприємства. Неможливо точно вирахувати суму збитків, яка залежить від масштабів ремонтних робіт і кількості пошкодженої техніки та частоту аварій. Через це для розрахунків не вистачає параметрів і складно оцінити економічний ефект програмного забезпечення, тому воно має соціальний ефект, а саме забезпечення безпеки для працівників гірничої промисловості.

Висновки

Ціна розробки даного програмного забезпечення без урахування ПДВ складає 94881,2 грн (дев'яносто чотири тисячі вісімсот вісімдесят одна гривня, двадцять копійок).

Час, використаний на розробку складає приблизно вісім місяців.

Проект має соціальний ефект, через що складно вирахувати економічну вигоду та строк окупності.

ВИСНОВКИ

У ході роботи над магістерською дисертацією було зроблено наступні висновки та досягнуто наступних результатів.

1. Найбільш перспективними методами прогнозування для подальших досліджень є ШНМ та МГУА: обидва методи дозволяють знаходити складні нелінійні залежності між вхідними та вихідними змінними та не потребують явного вибору моделі. Крім того, ШНМ мають перевагу у вигляді можливості навчання, що дозволяє використовувати їх для онлайн-прогнозування, а МГУА дозволяє знаходити оптимальну структуру моделі.
2. Виділено основні проблеми, що виникають при вирішенні задачі прогнозування, а саме:
 - неможливо врахувати всі чинники, що впливають на процес, який ми намагаємося прогнозувати; більше того, їх вплив може змінюватися з плином часу - фактор, що не був важливий сьогодні, може відіграти важливу роль завтра;
 - завжди є багато (іноді нескінченне число) можливих моделей, що добре відповідають навчальним даними - ми повинні вирішити, яку модель або множину моделей використовувати, і такі рішення зазвичай дуже схильні до помилок;
 - часто буває важко (якщо не неможливо) знайти оптимальну складність моделі.
3. На основі аналізу проблем, що виникають при вирішенні задачі прогнозування було розроблено:
 - нову постановку задачі прогнозування, яка явно враховує можливу непостійність впливу вхідних факторів на поведінку об'єкту, що прогнозується;

- новий метод прогнозування, що об'єднує підходи ШНМ і МГУА, та дозволяє знаходити оптимальну структуру нейронної мережі для даної конкретної задачі прогнозування;
 - нову методику прогнозування, що дозволяє використовувати існуючі методи прогнозування для практичних задач прогнозування, де наявна проблема неоднорідності даних.
4. Проведено тестування нових розроблених методів на наборі реальних даних, і отримана помилка прогнозу (у вигляді середньоквадратичної помилки) менша ніж у таких методів, як ШНМ, лінійна регресія та інші.
 5. Реалізовано програмний комплекс прогнозування часових рядів. Основні характеристики та особливості розробленого комплексу:
 - інтуїтивно-зрозумілий інтерфейс, що використовує принцип drag- n-drop;
 - складається з набору компонентів (таких як компонент завантаження вхідних даних, компонент побудови прогнозуючої моделі та інші), які користувач може з'єднувати потрібним чином для побудови складних багаторівневих систем прогнозування;
 - у якості алгоритмів, що використовуються для побудови прогнозуючої моделі, можуть використовуватися як існуючі алгоритми, так і нові розроблені алгоритми.
 6. Була доведена ефективність моніторингу гірничих комплексів на основі нейронних мереж. Це дало змогу в числовому ряді прогнозувати можливу поламку шахтної апаратури і, як наслідок, попереджувати аварії.

ПЕРЕЛІК ПОСИЛАНЬ

1. An algorithm for solving the problem of forecasting / Victor Sineglazov, Elena Chumachenko, Vladyslav Gorbatiuk // Aviation. -2013. №1(17), p. 9-13.
2. Victor Sineglazov, Elena Chumachenko & Vladyslav Gorbatiuk (2014) Using a mixture of experts' approach to solve the forecasting task, Aviation, 18:3, p. 129-133.
3. Gorbatiuk, Vladyslav; Sineglazov, Victor; Chumachenko, Olena. A method for building a forecasting model with dynamic weights. Eastern-European Journal of Enterprise Technologies, [S.l.], v. 2, n. 4(68), p. 4-8, apr. 2014. ISSN 1729-4061.
4. Алгоритм решения задачи прогнозирования / Е.И. Чумаченко, В.С. Горбатюк // Штучний інтелект. – 2012. – № 2., С. 24-31.
5. Метод решения задачи прогнозирования на основе комплексирования оценок / В. М. Синеглазов, Е. И. Чумаченко, В. С. Горбатюк // Індуктивне моделювання складних систем, випуск 4, 2012, С. 214-223.
6. Интеллектуальная система прогнозирования рисков послеоперационных осложнений / Синеглазов В. М., Чумаченко Е. И., Горбатюк В. С. // ISDMCI'2013, С. 289-291.
7. Использование искусственных нейронных сетей для задачи прогнозирования / Е. И. Чумаченко, В. С. Горбатюк // Електроніка та системи упр. – 2012. – № 1, С. 113-119.
8. Applying Different Neural Network's Topologies to the Forecasting Task – Gorbatiuk, Vladyslav; Sineglazov, Victor; Chumachenko, Olena. – International Conference in Inductive Modelling ICIM' 2013, С. 217-220.
9. One approach for the forecasting task solution – Gorbatiuk, Vladyslav; Sineglazov, Victor; Chumachenko, Olena. – V Всесвітній конгрес

«Авіація у ХХІ столітті» – «Безпека в авіації та космічні технології», р.
3.5.49-3.5.53.

10. Method for predicting failure risk of UAV navigation systems (2012)
– Gorbatiuk, Vladyslav; Chumachenko, Olena. – Methods and systems of
navigation and motion control (MSNMC 2012), р. 63-65.

11. Алгоритм решения задачи прогнозирования (2012) – Чумаченко,
Горбатюк – Міжнародна наукова конференція «Інтелектуальні системи
прийняття рішень і проблеми обчислювального інтелекту ISDMCI-2012», С.
423-425.

12. Francis X. Diebold (2007), Elements of Forecasting, 4th edition.

13. Ивахненко А.Г., Ивахненко Г.А. Обзор задач, решаемых по
алгоритмам Метода Группового Учета Аргументов (МГУ А)//
<http://www.gmdh.net>.

14. В.Феллер. Глава XI. Целочисленные величины. Производящие
функции // Введение в теорию вероятностей и её приложения = An introduction
to probability theory and its applicatons, Volume I second edition / Под ред. Е. Б.
Дынкина. – 2-е изд. – М.: Мир, 1964. – С. 270 – 272.

15. Box, George; Jenkins, Gwilym (1970). Time Series Analysis:
Forecasting and Control. San Francisco: Holden-Day.

16. Fisher, R. A. (1922). On the mathematical foundations of theoretical
statistics. Philos. Trans. Roy. Soc. London Ser. A 222 309–368.

17. Ивахненко А. Г. Индуктивный метод самоорганизации моделей
сложный систем. – Киев: Наукова думка, 1982. – 245с.

18. Мак-Каллок У. С., Питтс В., Логическое исчисление идей,
относящихся к нервной активности // В сб.: «Автоматы» под ред. К. Э.
Шеннона и Дж. Маккарти. – М.: Изд-во иностр. лит., 1956. – С. 363-384.

19. Kalman, R.E. (1960). "A new approach to linear filtering and prediction
problems". Journal of Basic Engineering 82 (1): pp. 35–45

20. Vogel, L., Owens, A., and Walsh, M., *Iskustvennyj Intellekt i Evolyutsyonnoe Modelirovanie. (Artificial Intelligence and Evolutive Modeling)*. Moscow: Nauka, 1969.
21. Алесинская Т.В. Методы скользящего среднего и экспоненциального сглаживания / Т.В. Алесинская // *Экономико-математические методы и модели : уч. пособие по решению задач по курсу*. – Таган-рог : Изд-во ТРТУ, 2002. – 153 с.
22. Klevecka Irina. Pre-Processing of Input Data of Neural Networks: The Case of Forecasting Telecommunication Network Traffic / Irina Klevecka, Janis // *Teletronikk* 3/4.2008. – С. 168-178.
23. Зорич В. А. *Математический анализ*. – М.: Физматлит, 1984. – 544 с.
24. Addison P.S. *The Illustrated Wavelet Transform Handbook*. – IOP, 2002.
25. Wong F. S. Time series forecasting using backpropagation neural networks // *Neurocomputing*. – 1990/91. – 2. – P. 147-159.
26. Groot de C., Wuertz D. Analysis of univariate time series with connectionist nets: a case study of two classical examples // *Neurocomputing*. – 1991. – 3. – P. 177-192.
27. Connor J. T., Martin R. D., Atlas L. E. Recurrent neural networks and robust time series prediction // *IEEE Trans. on Neural Networks*. – 1994. – 5. – P. 240–254.
28. Saxen H. Nonlinear time series analysis by neural networks. A case study // *Int. J. Neural Systems*. – 1996. – 7. – P. 195-201.
29. Madhavan P. G. A new recurrent neural network learning algorithm for time series prediction // *J. of Intelligent Systems*. – 1997. – 7. – P. 103 – 116.
30. Conway A. J., Macpherson K. P., Brown J. C. Delayed time series predictions with neural networks // *Neurocomputing*. – 1998. – 18. – P. 81 – 89.
31. Бодяньський Є. В., Кулішова Н. Є., Руденко О. Г. Рекурентна

прогнозуюча штучна нейронна мережа: архітектура та алгоритми навчання //

Адаптивні системи автоматичного управління. – Дніпропетровськ: Системні технології, 1999. – Вип. 2 (22). – С. 129-137.

32. Бодянский Е. В., Попов С. В., Штефан А. Нейросетевой упредитель многомерных стохастических последовательностей // Праці Міжнар. конф. з автоматичного управління «Автоматика-2000»:Т.2 – Львів: ДНДІ інформаційної інфраструктури, 2000. – С. 40-42.

33. Bodyanskiy Ye., Kolodyazniy V., Kulishova N. Generalized forecasting Sigma-Pi neural network // In “Intelligent Technologies – Theory and Applications”.–Amsterdam: IOS Press, 2002. – P.29-33.

34. Packard N., Crutchfield J., Farmer J., Shaw R. Geometry from a time series // Phys. Rev. Lett. – 1980. – 45. – P. 712-716.

35. Бодянский, Е. В. Искусственные нейронные сети: архитектуры, обучение, применения / Е. В. Бодянский, О. Г. Руденко // Харьков : ТЕЛЕТЕХ, 2004. – 369 с. : ил.

36. Rosenblatt, Frank. x. Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms. Spartan Books, Washington DC, 1961.

37. Rumelhart, David E.; Hinton, Geoffrey E.; Williams, Ronald J. (8 October 1986). "Learning representations by back-propagating errors". Nature 323 (6088): 533–536.

38. Wan E. Temporal backpropagation for FIR neural networks // Int. Joint Conf. on Neural Networks. – V.1. – San Diego, 1990. – P. 575 – 580.

39. Wan E. A. Time series prediction by using a connectionist network with integral delay lines / Eds. by A. Weigend, N. Gershenfeld “Time Series Prediction. Forecasting the Future and Understanding the Past”. – SFI Studies in the Sciences of Complexity. – V. XVII. – Reading: Addison-Wesley, 1994. – P. 195 – 218.

40. Back A. D., Wan E. A., Lawrence S., Tsoi A. C. A unifying view of some training algorithms for multilayer perceptrons with FIR filter synapses / Eds. by J. Vlontzos, J. Hwang, E. Wilson “Neural Networks for Signal Processing 4”. – N.Y.: IEEE Press, 1994. – P. 146 – 154.

41. Yu H.-Y., Bang S.-Y. An improved time series prediction by applying the layer-by-layer learning method to FIR neural networks // Neural Networks. – 1997. – 10. – P. 1717 – 1729.
42. Уидроу Б., Стирнз С. Адаптивная обработка сигналов. – М.: Радио и СВЯЗЬ, 1989. – 440 с.
43. Elman, J.L. Finding structure in time. // Cognitive Science. – 1990. – С. 179-211.
44. Amir F. A. A comparison between neural-network forecasting techniques / F. A. Amir, I. S. Samir – case study: river flow forecasting. // IEEE Transactions on neural networks. –Vol. 10, No. 2. – 1999. – p. 402–409.
45. Mohsen H. Artificial neural network approach for short term load forecasting for Illam region / H. Mohsen, S. Yazdan // World Academy of Science, Engineering and Technology 28 2007. – p. 280–284.
46. Short term load forecasting with multilayer perceptron and recurrent neural networks / Muhammad Riaz Khan, Cestm Ondrusek // Journal of ELECTRICAL ENGINEERING, VOL. 53, NO. 1-2, 2002, p. 17-23.
47. Jerome T. C. Recurrent neural networks and robust time series prediction / T. C. Jerome, R. M. Douglas, L. E. Atlas // IEEE transactions on neural networks. – Vol. 5, No. 2. –1994. – p. 240–254.
48. Горбань А. Н., Обобщенная аппроксимационная теорема и вычислительные возможности нейронных сетей, Сибирский журнал вычислительной математики, 1998. Т.1, № 1. С. 12-24.
49. Kenneth Levenberg (1944). «A Method for the Solution of Certain Non-Linear Problems in Least Squares». Quarterly of Applied Mathematics 2: p. 164–168.
50. U.S. General Aviation Aircraft Shipments and Sales 2012: <http://www.bga-aeroweb.com/database/Data3/US-General-Aviation-Aircraft-Sales-and-Shipments.xls>.
51. Friedman, J. H. (1991). "Multivariate Adaptive Regression Splines".

The Annals of Statistics 19: 1.

52. MacQueen, J. B. (1967). Some Methods for classification and Analysis of Multivariate Observations. Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability 1. University of California Press. pp. 281–297.
53. Data Sets for Time-Series Analysis [online] 2005. Available from Internet: <http://tracer.uc3m.es/tws/TimeSeriesWeb/repo.html>.
54. Lendasse, A., Oja, E., Simula, O., Verleysen, M. (2004). Time Series Prediction Competition: The CATS Benchmark. *International Joint Conference on Neural Networks, Budapest (Hungary), IEEE*, 1615-1620.
55. Гвоздева Т.В., Баллод Б.А. Проектирование информационных систем. – Ростов н/Д.: Феникс, 2009. – 512 с.
56. Головчинер М.Н. Проектирование информационных систем. – Томск: ТГУ, 2015. – 110 с.
57. Гома Х. UML–проектирование систем реального времени параллельных и распределенных приложений. – М.: ДМК–Пресс, 2011. – 704 с.
58. Грекул В.И., Денищенко Г.Н., Коровкина Н.Л. Проектирование информационных систем. – М.: Интернет–Университет Информационных Технологий «Интуит», 2016. – 570 с.
59. Грибачев К.Г. Delphi и Model Driven Architecture. Разработка приложений баз данных.– СПб.: Изд–во Питер, 2004. – 352 с.
60. Коцюба И.Ю., Чунаев А.В., Шиков А.Н. Основы проектирования информационных систем. Учебное пособие. – СПб.: Университет ИТМО, 2015. – 206 с.
61. Кузнецов М.Б. Трансформация UML–моделей и ее применение в технологии MDA. – М.: Институт системного программирования РАН, 2005. – 13 с.
62. Культин Н. Основы программирования в Delphi 2010. – СПб.: БХВ–Петербург, 2010. – 438 с.
63. Ларман Крэг. Применение UML 2.0 и шаблонов

проектирования. 3-е изд. – М.: Вильямс, 2013. – 736 с.

64. Новикова Г.М. Основы разработки корпоративных инфокоммуникационных систем. Учеб. пособие. М.: РУДН, 2008. – 152 с.

65. Осипов Д.Л. Базы данных и Delphi. Теория и практика. – СПб.: БХВ–Петербург, 2011. – 855 с.

66. Осипов Д. Delphi XE2. – СПб.: БХВ–Петербург, 2012. – 912 с.

67. Осипов Д. Delphi. Программирование для Windows, OS X, iOS и Android. – СПб.: БХВ–Петербург, 2014. – 464 с.

68. Рубанцев Валерий. Delphi на пальцах. Осваиваем новые технологии Windows 8: Touch и Gesture. Учебно–популярное издание. – М.: Издательство RVGames, 2013 – 81 с.

69. Скиена С. Алгоритмы. Руководство по разработке. 2-е изд.: Пер. с англ. – СПб.: БХВ–Петербург. 2011. – 720 с.

70. Сухарев М.В. Основы Delphi. Профессиональный подход. – СПб.: Наука и техника, 2004. – 600 с.

71. Хомоненко А.Д., Гофман В.Э., Мещеряков Е.В. Delphi 7. – СПб.: БХВ–Петербург, 2010. – 1136 с.

72. Якунин Ю.Ю. Технологии разработки программного обеспечения. – Красноярск: ИПК СФУ, 2008. – 225с.

```

import os
import sys
import time
import numpy
import theano

import theano.tensor as T

from theano.tensor.shared_randomstreams import RandomStreams
from logistic_sgd import LogisticRegression, load_data

from mlp import HiddenLayer

from rbm import RBM
# start-snippet-1

class DBN(object):
def __init__(self, numpy_rng, theano_rng=None, n_ins=784,

hidden_layers_sizes=[500, 500], n_outs=10):

self.sigmoid_layers = []

self.rbm_layers = []

self.params = []

self.n_layers = len(hidden_layers_sizes)
assert self.n_layers > 0
if not theano_rng:

theano_rng = RandomStreams(numpy_rng.randint(2 ** 30))
# allocate symbolic variables for the data

self.x = T.matrix('x') # the data is presented as rasterized images

self.y = T.ivector('y') # the labels are presented as 1D vector

# of [int] labels

# end-snippet-1

# The DBN is an MLP, for which all weights of intermediate

# layers are shared with a different RBM. We will first

```

```

# construct the DBN as a deep multilayer perceptron, and when
# constructing each sigmoidal layer we also construct an RBM
# that shares weights with that layer. During pretraining we
# will train these RBMs (which will lead to changing the
# weights of the MLP as well) During finetuning we will finish
# training the DBN by doing stochastic gradient descent on the
# MLP.
for i in xrange(self.n_layers):

# construct the sigmoidal layer
# the size of the input is either the number of hidden
# units of the layer below or the input size if we are on
# the first layer

if i == 0:

input_size = n_ins

else:

input_size = hidden_layers_sizes[i - 1]
# the input to this layer is either the activation of the
# hidden layer below or the input of the DBN if you are on
# the first layer

if i == 0:

layer_input = self.x

else:

layer_input = self.sigmoid_layers[-1].output
sigmoid_layer = HiddenLayer(rng=numpy_rng,

input=layer_input,

```

```

n_in=input_size,

n_out=hidden_layers_sizes[i],

activation=T.nnet.sigmoid)
# add the layer to our list of layers

self.sigmoid_layers.append(sigmoid_layer)
# its arguably a philosophical question... but we are

# going to only declare that the parameters of the

# sigmoid_layers are parameters of the DBN. The visible

# biases in the RBM are parameters of those RBMs, but not

# of the DBN.

self.params.extend(sigmoid_layer.params)
# Construct an RBM that shared weights with this layer

rbm_layer = RBM(numpy_rng=numpy_rng,

theano_rng=theano_rng,

input=layer_input,

n_visible=input_size,

n_hidden=hidden_layers_sizes[i],

W=sigmoid_layer.W,

hbias=sigmoid_layer.b)
self.rbm_layers.append(rbm_layer)
# We now need to add a logistic layer on top of the MLP

self.logLayer = LogisticRegression(

input=self.sigmoid_layers[-1].output,

n_in=hidden_layers_sizes[-1],

n_out=n_outs)

self.params.extend(self.logLayer.params)

```



```

# compute the cost for second phase of training, defined as the
# negative log likelihood of the logistic regression (output) layer

self.finetune_cost = self.logLayer.negative_log_likelihood(self.y)
# compute the gradients with respect to the model parameters

# symbolic variable that points to the number of errors made on the
# minibatch given by self.x and self.y

self.errors = self.logLayer.errors(self.y)
def pretraining_functions(self, train_set_x, batch_size, k):

# index to a [mini]batch

index = T.lscalar('index') # index to a minibatch

learning_rate = T.scalar('lr') # learning rate to use
# number of batches

n_batches = train_set_x.get_value(borrow=True).shape[0] / batch_size

# beginning of a batch, given `index`

batch_begin = index * batch_size

# ending of a batch given `index`

batch_end = batch_begin + batch_size

pretrain_fns = []

for rbm in self.rbm_layers:
# get the cost and the updates list

# using CD-k here (persistent=None) for training each RBM.

# TODO: change cost function to reconstruction error

cost, updates = rbm.get_cost_updates(learning_rate,
persistent=None, k=k)
# compile the theano function

```

```

fn = theano.function(

inputs=[index, theano.Param(learning_rate, default=0.1)],

outputs=cost,

updates=updates,

givens={

self.x: train_set_x[batch_begin:batch_end]

}

)

# append `fn` to the list of functions

pretrain_fns.append(fn)
return pretrain_fns
def build_finetune_functions(self, datasets, batch_size, learning_rate):
(train_set_x, train_set_y) = datasets[0]

(valid_set_x, valid_set_y) = datasets[1]

(test_set_x, test_set_y) = datasets[2]
# compute number of minibatches for training, validation and testing

n_valid_batches = valid_set_x.get_value(borrow=True).shape[0]

n_valid_batches /= batch_size

n_test_batches = test_set_x.get_value(borrow=True).shape[0]

n_test_batches /= batch_size
index = T.lscalar('index') # index to a [mini]batch
# compute the gradients with respect to the model parameters

gparams = T.grad(self.finetune_cost, self.params)

# compute list of fine-tuning updates

updates = []

```

```

for param, gparam in zip(self.params, gparams):

updates.append((param, param - gparam * learning_rate))
train_fn = theano.function(

inputs=[index],

outputs=self.finetune_cost,

updates=updates,

givens={

self.x: train_set_x[

index * batch_size: (index + 1) * batch_size

],

self.y: train_set_y[

index * batch_size: (index + 1) * batch_size

]

}

)
test_score_i = theano.function(

[index],

self.errors,

givens={

self.x: test_set_x[

index * batch_size: (index + 1) * batch_size

],

self.y: test_set_y[

index * batch_size: (index + 1) * batch_size

```

```

]
}

)
valid_score_i = theano.function(

[index],

self.errors,

givens={

self.x: valid_set_x[

index * batch_size: (index + 1) * batch_size

],

self.y: valid_set_y[

index * batch_size: (index + 1) * batch_size

]

}

)
# Create a function that scans the entire validation set

def valid_score():

return [valid_score_i(i) for i in xrange(n_valid_batches)]
# Create a function that scans the entire test set

def test_score():

return [test_score_i(i) for i in xrange(n_test_batches)]
return train_fn, valid_score, test_score
def test_DBN(finetime_lr=0.1, pretraining_epochs=100,

pretrain_lr=0.01, k=1, training_epochs=1000,

dataset='mnist.pkl.gz', batch_size=10):
datasets = load_data(dataset)
train_set_x, train_set_y = datasets[0]

```

```

valid_set_x, valid_set_y = datasets[1]

test_set_x, test_set_y = datasets[2]

# compute number of minibatches for training, validation and testing

n_train_batches = train_set_x.get_value(borrow=True).shape[0] / batch_size
# numpy random generator

numpy_rng = numpy.random.RandomState(123)

print '... building the model'

# construct the Deep Belief Network

dbn = DBN(numpy_rng=numpy_rng, n_ins=28 * 28,

hidden_layers_sizes=[1000, 1000, 1000],

n_outs=10)
# start-snippet-2

#####

# PRETRAINING THE MODEL #

#####

print '... getting the pretraining functions'

pretraining_fns = dbn.pretraining_functions(train_set_x=train_set_x,

batch_size=batch_size,

k=k)
print '... pre-training the model'

start_time = time.clock()

## Pre-train layer-wise

for i in xrange(dbn.n_layers):

# go through pretraining epochs

```

```

for epoch in xrange(pretraining_epochs):

# go through the training set

c = []

for batch_index in xrange(n_train_batches):

c.append(pretraining_fns[i](index=batch_index,

lr=pretrain_lr))

print 'Pre-training layer %i, epoch %d, cost ' % (i, epoch),

print numpy.mean(c)
end_time = time.clock()

# end-snippet-2

print >> sys.stderr, ('The pretraining code for file ' +

os.path.split(__file__)[1] +

' ran for %.2fm' % ((end_time - start_time) / 60.))

#####

# FINETUNING THE MODEL #

#####

# get the training, validation and testing function for the model

print '... getting the finetuning functions'

train_fn, validate_model, test_model = dbn.build_finetune_functions(

datasets=datasets,

batch_size=batch_size,

learning_rate=finetune_lr

)

print '... finetuning the model'

```

```

# early-stopping parameters

patience = 4 * n_train_batches # look as this many examples regardless

patience_increase = 2. # wait this much longer when a new best is

# found

improvement_threshold = 0.995 # a relative improvement of this much is

# considered significant

validation_frequency = min(n_train_batches, patience / 2)

# go through this many

# minibatches before checking the network

# on the validation set; in this case we

# check every epoch
best_validation_loss = numpy.inf

test_score = 0.

start_time = time.clock()
done_looping = False

epoch = 0

while (epoch < training_epochs) and (not done_looping):

    epoch = epoch + 1

    for minibatch_index in xrange(n_train_batches):
        minibatch_avg_cost = train_fn(minibatch_index)

        iter = (epoch - 1) * n_train_batches + minibatch_index
        if (iter + 1) % validation_frequency == 0:
            validation_losses = validate_model()

            this_validation_loss = numpy.mean(validation_losses)

    print(

```

```

'epoch %i, minibatch %i/%i, validation error %f %%'

% (

epoch,

minibatch_index + 1,

n_train_batches,

this_validation_loss * 100.

)

)

# if we got the best validation score until now

if this_validation_loss < best_validation_loss:

#improve patience if loss improvement is good enough

if (

this_validation_loss < best_validation_loss *

improvement_threshold

):

patience = max(patience, iter * patience_increase)

# save best validation score and iteration number

best_validation_loss = this_validation_loss

best_iter = iter

# test it on the test set

test_losses = test_model()

test_score = numpy.mean(test_losses)

print((' epoch %i, minibatch %i/%i, test error of '

'best model %f %%' ) %

```



```

(epoch, minibatch_index + 1, n_train_batches,

test_score * 100.))
if patience <= iter:

done_looping = True

break

end_time = time.clock()

print(

(

'Optimization complete with best validation score of %f %%, '

'obtained at iteration %i, '

'with test performance %f %%'

) % (best_validation_loss * 100., best_iter + 1, test_score * 100.)

)

print >> sys.stderr, ('The fine tuning code for file ' +

os.path.split(__file__)[1] +

' ran for %.2fm' % ((end_time - start_time)

/ 60.))
if __name__ == '__main__':

test_DBN()

import os
import sys
import time
import numpy
import theano

import theano.tensor as T

```

```

from theano.tensor.shared_randomstreams import RandomStreams
from logistic_sgd import LogisticRegression, load_data

from mlp import HiddenLayer

from dA import dA

def __init__(

self,

numpy_rng,

theano_rng=None,

n_ins=784,

hidden_layers_sizes=[500, 500],

n_outs=10,

corruption_levels=[0.1, 0.1]

):
self.sigmoid_layers = []

self.dA_layers = []

self.params = []

self.n_layers = len(hidden_layers_sizes)
assert self.n_layers > 0
if not theano_rng:

theano_rng = RandomStreams(numpy_rng.randint(2 ** 30))

# allocate symbolic variables for the data

self.x = T.matrix('x') # the data is presented as rasterized images

self.y = T.ivector('y') # the labels are presented as 1D vector of

# [int] labels

# end-snippet-1

```

```

# The SdA is an MLP, for which all weights of intermediate layers
# are shared with a different denoising autoencoders

# We will first construct the SdA as a deep multilayer perceptron,
# and when constructing each sigmoidal layer we also construct a
# denoising autoencoder that shares weights with that layer

# During pretraining we will train these autoencoders (which will
# lead to changing the weights of the MLP as well)

# During finetuning we will finish training the SdA by doing
# stochastic gradient descent on the MLP
# start-snippet-2

for i in xrange(self.n_layers):

# construct the sigmoidal layer
# the size of the input is either the number of hidden units of

# the layer below or the input size if we are on the first layer

if i == 0:

input_size = n_ins

else:

input_size = hidden_layers_sizes[i - 1]
# the input to this layer is either the activation of the hidden

# layer below or the input of the SdA if you are on the first

# layer

if i == 0:

layer_input = self.x

else:

```

```

layer_input = self.sigmoid_layers[-1].output
sigmoid_layer = HiddenLayer(rng=numpy_rng,

input=layer_input,

n_in=input_size,

n_out=hidden_layers_sizes[i],

activation=T.nnet.sigmoid)

# add the layer to our list of layers

self.sigmoid_layers.append(sigmoid_layer)

# its arguably a philosophical question...

# but we are going to only declare that the parameters of the

# sigmoid_layers are parameters of the StackedDAA

# the visible biases in the dA are parameters of those

# dA, but not the SdA

self.params.extend(sigmoid_layer.params)
# Construct a denoising autoencoder that shared weights with this

# layer

dA_layer = dA(numpy_rng=numpy_rng,

theano_rng=theano_rng,

input=layer_input,

n_visible=input_size,

n_hidden=hidden_layers_sizes[i],

W=sigmoid_layer.W,

bhid=sigmoid_layer.b)

self.dA_layers.append(dA_layer)

```

```

# end-snippet-2

# We now need to add a logistic layer on top of the MLP

self.logLayer = LogisticRegression(

input=self.sigmoid_layers[-1].output,

n_in=hidden_layers_sizes[-1],

n_out=n_outs

)

self.params.extend(self.logLayer.params)

# construct a function that implements one step of finetuning

# compute the cost for second phase of training,

# defined as the negative log likelihood

self.finetune_cost = self.logLayer.negative_log_likelihood(self.y)

# compute the gradients with respect to the model parameters

# symbolic variable that points to the number of errors made on the

# minibatch given by self.x and self.y

self.errors = self.logLayer.errors(self.y)

```

Відгук

на дипломний проект магістра студента групи 121м-16-1 Хмури Олексія Андрійовича на тему: «Обґрунтування ефективності моніторингу гірничих комплексів на основі поєднання нейронних мереж та методу групового урахування аргументів»

Ціль дипломного проекту – створення нейронної мережі для прогнозування аварійних ситуацій на шахтних підйомних машинах для доведення ефективності моніторингу гірничих комплексів.

Відсутність у традиційному підході до систем безпеки на виробництві аналогів обумовлює актуальність роботи.

Тема дипломної роботи безпосередньо пов'язана з об'єктом діяльності магістра спеціальності 121 «Інженерія програмного забезпечення» області знань 12 «Інформаційні технології» - створення, дослідження та реалізація моделей та програмних засобів.

Наукова новизна результатів, що очікуються, полягає у: проведенні аналізу та виявленні недоліків традиційного підходу до розробки систем захисту та у створенні нової системи попередження аварійних ситуацій використовуючи штучні нейронні мережі та метод групового урахування аргументів.

Для реалізації програмного додатку використані такі технології як DOCKER, Python та штучні нейронні мережі, що підтверджує оригінальність технічних рішень.

Практичне значення результатів роботи полягає у розробленні нейронних мереж для створення системи, що може попереджувати можливі аварійні ситуації та поламаки.

Для реалізації проекту була використана мова програмування Python та суміжні до неї бібліотеки для роботи зі штучними нейронними мережами, операційною системою, математичним стеком, графічним інтерфесом та відображенням даних.

Оформлення матеріалів дипломного проекту виконано на сучасному рівні і відповідає вимогам, які пред'являються до робіт даної кваліфікації.

Ступінь самостійності виконання дипломного проекту висока.

Даний дипломний проект в цілому заслуговує оцінки «відмінно», а студент Хмура О. А. - присвоєння кваліфікації «Інженер-програміст».

Керівник дипломного
проекту магістра,
д.т.н.,
проф. кафедри ПЗКС

Мещеряков Л.І.

Рецензія

на дипломний проект магістра на тему:
**«Обґрунтування ефективності моніторингу гірничих комплексів на основі
нейронних мереж»**

Студента групи 121М-16-1 Хмури Олексія Андрійовича

Зараз машинне навчання набирає значну популярність. Саме тому вибрана тема дипломного проекту є досить актуальною.

Тема дипломної роботи безпосередньо пов'язана з об'єктом діяльності магістра спеціальності 121 «Інженерія програмного забезпечення» області знань 12 «Інформаційні технології» - створення, дослідження та реалізація моделей та програмних засобів.

Наукова новизна результатів, що очікуються, полягає у: проведенні аналізу та виявленні недоліків традиційного підходу до розробки систем захисту та у створенні нової системи попередження аварійних ситуацій на основі поєднання нейронних мереж та методу групового урахування аргументів.

Використані технології розробки безпосередньо пов'язані з об'єктом діяльності магістра.

Студент Хмура О.А. добре розібрався зі специфікаціями проектування, розгортання та тестування систем, пов'язаних з машинним навчанням та штучними нейронними мережами.

З огляду на вищевикладене, можна зробити висновок, що даний проект цілком відповідає вимогам, що пред'являються до кваліфікаційних робіт рівня магістра.

Ступінь опрацювання компонентів даного проекту дозволяє оцінити роботу на «відмінно» і рекомендувати присвоїти О.А. Хмура кваліфікацію «Інженер-програміст».

Рецензент, (ст. вик.)