

Довідка
про відсутність плагіату у випускній кваліфікаційній роботі

Міністерство освіти і науки України
Український державний університет науки та технологій

Кафедра «Комп'ютерні інформаційні технології»

ДОВІДКА

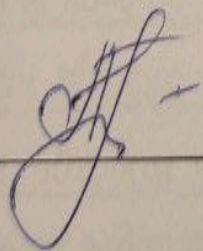
За результатами перевірки випускної кваліфікаційної роботи здобувача вищої освіти

Медведєвої Карини Віталіївни

(прізвище, ім'я, по батькові)

на тему: Дослідження часових рядів навантаження мережесевих систем
в роботі не виявлено порушень академічної доброчесності.

Керівник ВКР



Олександра ГОРБОВА

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Український державний університет науки і технологій

Кафедра Комп'ютерні інформаційні технології

«ДО ЗАХИСТУ»

Завідувач кафедри

В.Г. /Вадим ГОРЯЧКІН/

« 20 » 12 20 21 р.

ДИПЛОМНА РОБОТА

на здобуття освітнього ступеня «магістр»

Галузь знань **12 Інформаційні технології**

Спеціальність **121 Інженерія програмного забезпечення**

Тема **Дослідження часових рядів навантаження мережевих систем**

Theme **Time-series research of network systems overload**

Керівник дипломної роботи

доц. О.Г. Олександра ГОРБОВА

Нормоконтролер

доц. О.К. Олена КУРОП'ЯТНИК

Студентка групи ПЗ2021

К.М. Карина МЕДВЕДСЬКА

Student

Karyna MEDVEDIEVA

Дніпро – 2021

Дніпровський національний університет залізничного транспорту імені
академіка В.Лазаряна

Факультет Комп'ютерних технологій і систем
Кафедра Комп'ютерні інформаційні технології
Спеціальність Інженерія програмного забезпечення

«ЗАТВЕРДЖУЮ»

Завідувач кафедри

/В.І.Шинкаренко/

(підпис)

«25» грудня 2020 р.

ЗАВДАННЯ

до дипломної роботи на здобуття ОС магістр
(освітній ступінь)

студентки групи ПЗ2021 К.В. Медведєвої
(номер групи) (ПІБ)

1 Тема дипломної роботи: Дослідження часових часових рядів навантаження мережевих систем

затверджена наказом по університету від «18» листопада 2020 р. № 690.

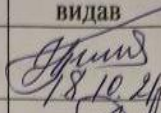
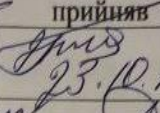
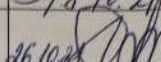
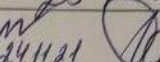
2 Термін подання студентом закінченої роботи 3 грудня 2020 р.

3 Вихідні дані до дипломної роботи маска підмережі, розмір тестового пакеті, час за який передаються дані

4 Зміст пояснювальної записки (перелік питань до розробки) аналіз сучасного стану дослідження та програмно-апаратного забезпечення, огляд методів та моделей прогнозування, проектування й розробка інструментального забезпечення для дослідження часових рядів навантаження мережевої системи, охорона праці та безпека в надзвичайних ситуаціях

5 Перелік демонстраційного матеріалу презентація, доповідь, демонстраційне відео

6. Консультанти (з назвами розділів):

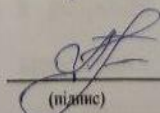
Розділ	Консультант	Підпис, дата	
		завдання видав	завдання прийняв
Техніко-економічні розрахунки	Микола ГНЕННИЙ	 18.10.21	 23.10.21
Охорона праці	Олег САБЛІН	 26.10.21	 24.11.21

КАЛЕНДАРНИЙ ПЛАН

№ пор.	Назва розділів дипломної роботи	Термін виконання розділів роботи	Примітка
1	Вступ	1.09.21	
2	Аналіз сучасного стану дослідження проблеми за науковими літературними джерелами	02.09.21 – 14.09.21	від 70 джерел
3	Аналіз сучасного стану програмно-апаратного забезпечення, яке потребує вдосконалення для вирішення проблем дослідження	16.09.21 – 10.10.21	
4	Постановка задачі, технічне завдання	11.10.21 – 17.10.21	30%
5	Техніко-економічні показники	18.10.21 – 19.10.21	
6	Розробка інструментальних засобів дослідження	20.10.21 – 8.11.21	
7	Виконання досліджень	08.11.21 – 14.11.21	60%
8	Оформлення тез доповідей	08.11.21 – 14.11.21	
9	Оформлення статті у фаховий журнал	19.11.21 – 22.11.21	
10	Оформлення пояснювальної записки	23.11.21 – 28.11.21	
11	Розробка демонстраційних матеріалів	29.11.21 – 05.12.21	100%

Дата видачі завдання «18» листопада 2020 р.

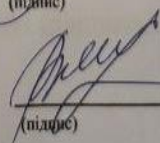
Керівник дипломної роботи


(підпис)

О.В. Горбова

(підп.)

Завдання прийняв до виконання


(підпис)

К.В. Медведєва

(підп.)

РЕФЕРАТ

Темою магістерської роботи є «Дослідження часових рядів навантаження мережевої системи».

Дипломний проект: пояснювальна записка містить 78 сторінки, 29 рисунків, 27 літературних джерел, 6 таблиць та 6 додатків.

Об'єктом дослідження є програмне забезпечення для аналізу навантаженості мережевих систем.

Мета й завдання дослідження. Мета поставленої задачі полягає у дослідженні методів прогнозування часових рядів та виявлення найбільш придатних з них для обробки даних, у розробці часових рядів для прогнозування перевантаження в мережевих системах.

Для досягнення поставленої мети необхідно:

- аналіз існуючих методів прогнозування часових рядів;
- побудова динамічної моделі рядів та аналіз прогнозування майбутніх значень;
- проведення аналізу методів ковзних середніх та найменших квадратів;
- перевірка метода прогнозування та аналіз отриманих результатів;
- дослідження методів прогнозування часових рядів та виявлення найбільш придатного для обробки даних, отриманих в результаті експериментів;
- оцінка придатності метода прогнозування часових рядів навантаженості мережевих систем;
- розробка метода прогнозування навантажень в комп'ютерних системах на основі аналізу часових рядів.

Дослідження роботи полягає в розробці методів моніторингу та прогнозування перенавантажень в мережевих системах. Розроблений додаток може бути використаний у мережах підприємств з ціллю підвищення рівня безпеки передачі даних.

ЗМІСТ

ВСТУП	8
1 Аналіз сучасного стану дослідження часових рядів навантаження мережевих систем	11
1.1 Аналіз предметної області	11
1.1.1 Опис проблеми. Актуальність дослідження	11
1.1.2 Огляд останніх досліджень і публікацій	14
1.2 Огляд програмних аналогів	17
1.3 Постановка задачі.....	20
Висновки до розділу 1.....	20
2 Огляд методів та моделей прогнозування	21
2.1 Опис моделей.....	21
2.1.1 Авторегресійна модель змінного середнього	21
2.1.2 Класифікаційно-регресійні дерева	22
2.1.3 Метод ковзної середньої	24
2.1.4 Метод найменших квадратів	25
2.2 Переваги і недоліки моделей	25
2.3 Методи оцінки точності прогнозу	26
Висновки до розділу 2.....	28
3 Проектування й розробка забезпечення для дослідження.....	29
3.1 Формалізація задачі.....	29
3.2 Базова архітектура системи	29
3.3 Внутрішнє проектування	32
3.3.1 Вибір мови програмування	32
3.3.2 Технологічна платформа.....	33
3.3.3 Ієрархія та взаємодія класів системи.....	34
3.3.4 Використанні принципи проектування.....	34
3.4 Розробка інтерфейсу користувача	36
3.4.1 Розробка структури інтерфейсу системи	36

3.4.2 Реалізація інтерфейсу користувача	37
3.5 Тестування та налагодження програми.....	44
3.5.1 Аналіз методів тестування та відлагодження	44
3.5.2 Тестування методом покриття операторів	46
3.5.3 Тестування методом припущення помилок.....	47
Висновки до розділу 3	47
4 Дослідження часових рядів навантаження мережевої системи	48
4.1 Підготовка до експерименту.....	48
4.1.1 Експеримент 1	48
4.1.2 Експеримент 2	49
4.1.3 Експеримент 3	51
4.1.4 Експеримент 4	53
4.1.5 Експеримент 5	54
4.2 Проведення експерименту	55
Висновки до розділу 4	56
5 Охорона праці та безпека в надзвичайних ситуаціях	57
5.1 Вимоги безпеки праці під час виконання робіт.....	57
5.1.1 Вимоги під час експлуатації ПК.....	57
5.1.2 Аналіз шкідливих та небезпечних виробничих факторів	58
5.1.3 Вимоги до приміщень	59
5.1.4 Вимоги до виробничого середовища приміщень з моніторами, ПК і ноутбуками	62
2. Дії працівників в аварійних ситуаціях	66
Висновки до розділу 5	70
Загальні висновки за роботою	71
Список використаних джерел	72
Додатки.....	75

ПЕРЕЛІК УМОВНИХ ПОЗНАК, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ПЗ – програмне забезпечення;

ПП – програмний продукт;

ЕОМ – електронна обчислювальна машина;

ПК – персональний комп'ютер;

НПАОП – нормативно-правові акти з охорони праці;

ДСанПіН – державні санітарні правила та норми.

ВСТУП

Актуальність теми. Останнім часом з'являється безліч провайдерів, які пропонують високошвидкісне підключення до мережі Інтернет. Це обумовлюється тим, що дані послуги актуальні і затребувані. З кожним днем кількість абонентів, підключених до Інтернету, росте, відповідно зростає і навантаження на мережі та сервери.

Від якості прогнозу завантаженості мереж залежить багато чинників. На даний момент існує величезна кількість алгоритмів прогнозування, результат яких безпосередньо залежить від вхідних даних, тобто вибірок, сформованих з часових рядів. Для підвищення точності прогнозів перенавантаження важливо розуміти, що різні моделі прогнозування підходять для різних часових рядів з різними характеристиками.

Завдання прогнозування часового ряду вирішується на основі створення моделі прогнозування, адекватно описує досліджуваний процес.

Аналіз часових рядів – сукупність математико-статистичних методів аналізу, призначених для виявлення структури часових рядів і для їх прогнозування. Сюди відносяться, зокрема, методи регресійного аналізу. Виявлення структури часового ряду необхідно для того, щоб побудувати математичну модель того явища, яке є джерелом аналізованого часового ряду. Прогноз майбутніх значень часового ряду використовується для ефективного прийняття рішень.

Розробка механізму прогнозування навантаженості мережевих систем є актуальною, важливою задачею та частиною мір по забезпеченню безпечної експлуатації комп'ютерної мережі. Прогнозування інтенсивності трафіку дає можливість заздалегідь вживати необхідних заходів попередження наслідків. Досліджуються причини виникнення перевантажень в мережі, наслідком яких є зниження рівня безпеки передачі даних.

Об'єктом дослідження є програмне забезпечення для аналізу навантаженості мережевих систем.

Предметом дослідження є оцінка алгоритмів прогнозування навантаженості мережевих систем.

Мета й завдання дослідження. Мета поставленої задачі полягає у дослідженні методів прогнозування часових рядів та виявлення найбільш придатних з них для обробки даних, у розробці часових рядів для прогнозування перевантаження в мережевих системах.

Для досягнення поставленої мети необхідно:

- аналіз існуючих методів прогнозування часових рядів;
- побудова динамічної моделі рядів та аналіз прогнозування майбутніх значень;
- проведення аналізу методів ковзних середніх та найменших квадратів;
- перевірка метода прогнозування та аналіз отриманих результатів;
- дослідження методів прогнозування часових рядів та виявлення найбільш придатного для обробки даних, отриманих в результаті експериментів;
- оцінка придатності метода прогнозування часових рядів навантаженості мережевих систем;
- розробка метода прогнозування навантажень в комп'ютерних системах на основі аналізу часових рядів.

Методи дослідження. Використанні методи дослідження базуються на теорії математичної статистики та теорії обчислювальних систем.

Наукова новизна. В даній роботі наукова новизна полягає у:

- дослідженні методів прогнозування часових рядів та виявлення серед них найбільш придатних методів для обробки даних;
- побудові динамічної моделі часових рядів та прогнозування майбутніх значень на вказаний період;
- застосуванні методів прогнозування до вивчення та прогнозування пікових перевантажень мережевих систем.

Практичне значення. Дослідження роботи складається в розробці методів моніторингу та прогнозування перенавантажень в мережевих системах. Розроблений додаток може бути використаний у мережах підприємств з ціллю підвищення рівня безпеки передачі даних.

Апробація результатів дослідження та публікації. Основні положення магістерської роботи доповідалися та були схвалені на XIV міжнародній науково-практичній конференції «Сучасні інформаційні та комунікаційні технології на транспорті, в промисловості та освіті», яка відбулася в Дніпровському національному університеті залізничного транспорту імені академіка В. Лазаряна (ДНУЗТ) 15 - 16 грудня 2020 року, див. додаток В.

На основні положень магістерської роботи було створено тези доповіді, котрі були схвалені на 81 міжнародній науково-практичній конференції "Проблеми та перспективи розвитку залізничного транспорту", яка відбулася в Дніпровському національному університеті залізничного транспорту імені академіка В. Лазаряна (ДНУЗТ) 22 - 23 квітня 2021 року, див. додаток Г, на 81 всеукраїнській науково-технічній конференції молодих учених, магістрантів та студентів «Наука і сталий розвиток транспорту», яка відбулася в Дніпровському національному університеті залізничного транспорту імені академіка В. Лазаряна (ДНУЗТ) 28 жовтня 2021 року, див. додаток Г.

Створено проект статті «АНАЛІЗ ЧАСОВИХ РЯДІВ НАВАНТАЖЕНОСТІ МЕРЕЖЕВИХ СИСТЕМ» до журналу «Вісник Дніпропетровського національного університету залізничного транспорту», див. додаток Е.

1 АНАЛІЗ СУЧАСНОГО СТАНУ ДОСЛІДЖЕННЯ ЧАСОВИХ РЯДІВ НАВАНТАЖЕННЯ МЕРЕЖЕВИХ СИСТЕМ

1.1 Аналіз предметної області

1.1.1 Опис проблеми. Актуальність дослідження

Останнім часом з'являється безліч провайдерів, які пропонують високошвидкісне підключення до мережі Інтернет. Це обумовлюється тим, що дані послуги актуальні і затребувані. З кожним днем кількість абонентів, підключених до Інтернету, росте, відповідно зростає і навантаження на сервери [19].

Від якості прогнозу завантаженості мереж залежить багато чинників. На даний момент існує величезна кількість алгоритмів прогнозування, результат яких безпосередньо залежить від вхідних даних, тобто вибірок, сформованих з часових рядів. Для підвищення точності прогнозів перенавантаження важливо розуміти, що різні моделі прогнозування підходять для різних часових рядів з різними характеристиками [19].

Перевантаження мережі в комп'ютерних мережах і теорії черг - це зниження якості обслуговування, яке виникає, коли мережевий вузол або лінія зв'язку переносить більше даних, ніж може обробляти. Типові ефекти включають затримку в черзі, втрату пакетів або блокування нових з'єднань. Наслідком перевантаження є те, що поступове збільшення пропонованої навантаження призводить або до невеликого збільшення, або навіть до зниження пропускної здатності мережі [19].

Мережеві протоколи, які використовують агресивну ретрансляцію для компенсації втрати пакетів через перевантаження, можуть збільшити перевантаження навіть після того, як початкова навантаження була зменшена до рівня, який зазвичай не викликав перевантаження мережі. Такі мережі демонструють два стабільних стани при однаковому рівні навантаження. Стабільний стан з низькою пропускною здатністю відомо як колапс перевантаження [19].

Мережі використовують методи відстеження перевантажень і запобігання перевантажень, щоб уникнути колапсу. До цій техніці відносяться: експоненціальна витримка в таких протоколах як CSMA/CA в 802.11 аналогічному CSMA/CD в оригінальному Ethernet, скорочення вікна в TCP і справедлива чергу в таких пристроях, як маршрутизатори і мережеві комутатори. Інші техніки, пов'язані з перевантаженням, включають схеми пріоритетів, які передають деякі пакети з більш високим пріоритетом ніж інші і явне виділення мережевих ресурсів на конкретні потоки за допомогою використання управління допуском [20].

Затримка мережі є важливою конструктивною і експлуатаційною характеристикою комп'ютерної мережі або телекомунікаційної мережі. Затримка мережі визначає інформаційне перевантаження відноситься до надлишку входить в нашу свідомість інформації [20].

Проблема перевантаження мережі – це стара проблема мереж яка існувала ще з перших років їх існування. Перевантаження мережі викликає погіршення управління обміном по мережі спеціально розраховані на кільце. У цих методах право на наступну передачу або, як ще кажуть, на захоплення мережі переходить послідовно контроль зменшення перевантаження - реалізація TCP з оптимізованим алгоритмом контролю перевантаження мережі для високошвидкісних мереж з великою затримкою, TCP передбачає що це пов'язано з перевантаженням мережі і вживає заходів щодо скорочення навантаження на мережу [20].

Мережеві протоколи, які використовують агресивні повторні передачі, щоб дати компенсацію за втрату пакету, мають тенденцію тримати системи в стані перевантаження мережі, навіть після того, як початковий вантаж був зменшений до рівня, який не буде зазвичай викликати перевантаження мережі. Таким чином мережі використовуючи ці протоколи можуть показати два стійких стану під тим же самим рівнем вантажу. Сталий стан з низькою пропускнуою здатністю відомо як тісний крах [20].

Тісний крах (або крах перевантаженості) є умовою, якого може досягти комп'ютерна мережа з пакетною комутацією, коли мінімальна корисна комунікація відбувається через перевантаженість. Крах перевантаженості зазвичай відбувається в «вузькому горлі» в мережі, де повне надходить рух до вузла перевищує комунікабельність смугу пропускання. Точки контакту між локальною мережею і глобальною мережею – найбільш ймовірне вузьке горло [20].

Коли мережа знаходиться в такому умови, вона влаштувалася (при перевантаженні) в стійкий стан, де транспортний вимога високо, але мало корисної пропускної здатності є, і є високі рівні затримки пакета і втрати (викликаний маршрутизаторами, що відмовляються від пакетів, тому що їх черзі продукції занадто повні), і загальну якість обслуговування надзвичайно погано [20].

Коли надіслано більше пакетів, ніж могло бути оброблено, проміжні маршрутизатори відмовилися від багатьох пакетів, чекаючи, що кінцеві точки мережі повторно передадуть інформацію. Однак у ранніх впровадженнях ТСП було дуже не якісна поведінка повторної передачі. Коли втрата пакета сталася, кінцеві точки надіслали додаткові пакети, які повторили втрачену інформацію, подвоївши надіслану швидкість передачі даних, точно протилежно тому, що повинно бути зроблено під час перевантаженості. При цьому мережа опинилась у стані «крах перевантаженості», де більшість пакетів було втрачено, з чого можливо зробити висновок що пропускна здатність була незначна [20].

Є багато способів класифікувати алгоритми управління перевантаженнями за:

- типом і сумою зворотного зв'язку, отриманої від мережі: втрата, затримка, єдиний біт або мультимбіт, явні сигнали;
- зростаючим розгортанням в потоковому Інтернеті: тільки відправнику потрібна модифікація, відправнику і керуючому потрібна

модифікація, тільки маршрутизаторам потрібна модифікація, відправнику, керуючому і маршрутизаторам потрібна модифікація [20].

Алгоритм запобігання перевантаженості ТСП – основна підстава для управління перевантаженнями в Інтернеті.

Проблеми відбуваються, коли багато паралельних потоків ТСП відчують зниження хвоста буфера черги порту. Тоді автоматичне запобігання перевантаженості ТСП недостатньо [6].

Запобігання перевантаженості ТСП/IP:

- активне управління чергою (AQM) - повторне замовлення або зниження мережових пакетів в передачі буферу, який пов'язаний з диспетчером мережового інтерфейсу (NIC);

- випадкове раннє виявлення - одне рішення полягає в тому, щоб використовувати випадкове раннє виявлення (RED) на буфері черги порту мережового обладнання. На портах мережового обладнання більше ніж з одним буфером черзі зважене випадкове раннє виявлення (WRED) могло використовуватися при наявності;

- чіпке випадкове раннє виявлення (RRED) - алгоритм міцного випадкового раннього виявлення (RRED) був запропонований, щоб поліпшити пропускну здатність ТСП проти нападів відмови в обслуговуванні (DoS), особливо нападів відмови в обслуговуванні з низькою ставкою (LDoS).

- Flowbased-RED / WRED – деяке мережове обладнання має порти, які виміряти кожен потік (flowbased-RED / WRED) та в поточному стані сигналізувати про занадто великий потік смуги пропускання згідно деякої політики QoS.

1.1.2 Огляд останніх досліджень і публікацій

В роботі [2] запропонований метод попередження перевантаження в вузлах комп'ютерної мережі. Основою методу є прогнозування довжини черги, що виникає в буфері вузла, при проходженні через нього

самоподібного трафіку. Значення параметрів навантаження, при яких чергу може перевищити розмір буфера, розраховуються за допомогою імітаційного моделювання. Визначаючи параметри вхідного трафіку, шляхом моніторингу можна прогнозувати переповнення буфера і не допускати перевантаження мережі. Численні дослідження процесів в мережі Інтернет показали, що статистичні характеристики трафіку мають властивість тимчасової масштабної [2].

Причина такого ефекту – в особливості розподілу файлів по серверам, їх розмірах, а також в типову поведінку користувачів. Виявилося, що з самого початку не проявляють властивостей самоподібності потоки даних, пройшовши обробку на вузлових серверах і активних мережевих елементах, починають подавати яскраво виражені ознаки самоподібності [2].

Самоподібний трафік має особливу структуру, що зберігається на багатьох масштабах – в реалізації завжди присутня деяка кількість дуже великих викидів при відносно невеликому середньому рівні трафіку. ці викиди викликають значні затримки і втрати пакетів, навіть коли сумарна потреба всіх потоків далека від максимально допустимих значень. У класичному випадку для пуассоновського вхідного потоку нам буде достатньо буферів помірного розміру: чергу може утворитися в короткостроковій перспективі, але за довгий період часу буфери очистяться. Однак при самоподібній навантаженні утворюються черги набагато більшої довжини [2].

Традиційний аналіз черг, в основі якого лежить припущення про пуассоновському потоці, що не може передбачати продуктивність системи для самоподібного трафіку, тому основним інструментом дослідження і прогнозування стає імітаційне моделювання. Варіюючи розміром буферів і розподілом потоків даних по каналах, можна знизити втрати і час простою системи. Для проведення статистичного моделювання, яке дозволяє розрахувати найбільш оптимальні параметри мережі та необхідний обсяг

буферної пам'яті необхідна модель вхідного самоподібного навантаження [19].

У [6] докладно показано нерівномірне зростання швидкостей каналів, по яких передача даних неминуче призводить до виникнення «вузьких» місць в телекомунікаційній мережі та відповідно до виникнення перевантажень, особливо при підключенні мереж доступу до транспортної мережі. Традиційні протоколи управління чергами та запобігання перевантажень керуванням трафіком зі складною динамікою і нелінійністю зміни навантаження, що призводить до виникнення перевантажень і появи глобальної синхронізації TCP потоків. Це зменшує ефективну швидкість передачі даних і погіршує параметри якості, такі як відсоток втрачених пакетів, затримки і варіації затримок.

У даній роботі проводиться дослідження мережі TCP/IP із застосуванням різних сучасних алгоритмів управління чергами таких як: RED (Random Early Detection), PI-controller (ProportionalIntegral controller), REM (Random Exponential Marking). Результат обробки сигналів, що відбуваються з різних напрямках сектору, де формуються однакові за структурою дискретно кодовані псевдо-випадкові сигнали, що відрізняються один від одного тільки часовим зсувом [6].

Основні завдання алгоритмів управління чергами – мінімізація середньої довжини черги при одночасному забезпеченні високого коефіцієнта використання каналу, а також справедливий розподіл буферного простору між різними потоками даних. Схеми управління чергами розрізняються, в основному, критерієм, за яким відкидаються пакети, і місцем у черзі, звідки проводиться відкидання пакетів (початок або кінець черги) [6].

Найбільш простим критерієм для відкидання пакетів є досягнення чергою певного порогу, званого максимальною довжиною черги. Прикладом алгоритмів пасивного управління чергами (PQM) є алгоритм «відкидання хвоста», де трафік не розділяється по типам пакетів і ймовірність відкидання

кожного пакету однакова. Коли черга заповнюється до деякого заданого максимального значення, все знову надходять пакети відкидаються, поки черга не звільниться для надходження вхідного трафіку. Є альтернативні алгоритми відкидання пакетів: Random Drop (відкидання випадково обраного) і Drop from Front (Відкидання першого пакету в черзі). Алгоритмам PQM притаманний ряд недоліків, для усунення яких використовуються алгоритми активного управління чергами (AQM), забезпечують завчасне виявлення перевантаження [6].

Основними цілями алгоритмів AQM є:

- мінімізація змикання затримки пакетів шляхом контролю середнього розміру черги;
- запобігання ефекту глобальної синхронізації TCP-трафіку;
- забезпечення неупередженого обслуговування трафіку, що характеризується короткочасними сплесками;
- суворе обмеження максимального середнього розміру черги.

1.2 Огляд програмних аналогів

Програмні засоби:

1. CommView – програма дозволяє наочно бачити повну картину трафіку, що проходить через комп'ютер або сегмент локальної мережі; настроюється система сигналізації дозволяє попереджати про наявність в трафіку підозрілих пакетів, появи в мережі вузлів з позаштатними адресами або підвищенні мережевого навантаження;

2. EtherApe – є програмним забезпеченням з відкритим вихідним кодом і розповсюджується безкоштовно. Однак те, чим він дійсно виділяється на тлі інших рішень - це орієнтація на графіку. Відображається за допомогою просунутого графічного інтерфейсу, де кожна вершина графа являє собою окремий хост, розміри вершин і ребер вказують на розмір мережевого трафіку, а кольором відзначаються різні протоколи;

3. TMeter – комплексна програма, призначена для управління і відстеження трафіку. Програма повністю аналізує мережевий трафік на підставі особистого хоста, графічно відображає об'єкт переданої інформації, автоматично видає звіт по заданому періоду і перевіряє дані свого облікового запису користувачів термінального сервера. TMeter постійно відстежує процеси, підключені до інтернету. Додаткова функція – шейпінг – встановлює обмеження в швидкості при переході критичного порога переданої інформації. Серед інструментів також є блокування з'єднання з певним процесом або локальним вузлом і підключення декількох машин до інтернету за єдиним IP-адресою (NAT-сервер);

4. NetWorx – невелика програма для обліку трафіку на ПК. NetWorx фіксує всі вхідні і вихідні дані для складання звітів. Посилаючись на отриману інформацію, NetWorx може відобразити потік певного користувача, скласти погодинний звіт або за певну кількість часу (від дня до місяця). Вмонтований інструмент блокування трафіку після перевитрати - програма спочатку повідомляє, а потім і зовсім забороняє доступ. Серед додаткових функцій можна виділити можливість трасування маршруту;

5. NetLimiter – програма NetLimiter призначена для контролю і обмеження мережевого трафіку. За допомогою неї можна контролювати підключення та регулювати кількість уживаного трафіку на комп'ютері. При необхідності можна блокувати доступ виділених процесів і будувати графік використання передачі даних за певний період. Є можливість поділу звіту для кожного адаптера. Наприклад, можна відобразити список процесів для локальної мережі або інтернет. Візуалізувати графік, виходить виявити процеси, перевантажують мережу. В окремому вікні відображається повна інформація про швидкість мережі, аналізатор маршруту і підрахунок показника передачі даних [22].

Представлений огляд базових програм моніторингу в таблиці 1.4.1.

1.3 Постановка задачі

Дослідити методи прогнозування часових рядів та виявити найбільш придатний з них для прогнозування перевантаження в мережевих системах.

Як інструмент для виконання дослідження, розробився програмне забезпечення з аналізу та прогнозування часових рядів, у якому буде реалізовано побудова графіку та прогнозування майбутнього навантаження комп'ютерних систем та мереж.

Виконати дослідження на базі отриманих експериментальних даних, оцінити придатності метода прогнозування часових рядів завантаженості мережевих систем.

Висновки до розділу 1.

Проведений аналіз аналогів показав, що жодна з програм не має чіткої відповіді на питання завантаженості мереж в залежності від часу її використання та не має функції прогнозування.

Встановлено, що до теперішнього часу розроблено безліч моделей для вирішення задачі прогнозування часового ряду.

Встановлено, що розробка механізму прогнозування навантаженості мережевих систем є актуальною, важливою задачею та частиною мір із забезпечення безпечної експлуатації комп'ютерної мережі.

2 ОГЛЯД МЕТОДІВ ТА МОДЕЛЕЙ ПРОГНОЗУВАННЯ

Завдання прогнозування значень тимчасового ряду не нова, але дослідження в цій галузі не втрачають актуальності та постійно виникають нові алгоритми та методики досліджень. На даний момент робіт з прогнозування часових рядів навантаження мережевих систем не багато, вони рідко спираються на дані за завантаженістю реальної мережі. Варіацій алгоритмів прогнозування часових рядів досить багато, а дослідження кожного з них досить трудомісткий процес, тому завдання короткострокових прогнозів мережевої системи залишається актуальним.

2.1 Опис моделей

2.1.1 Авторегресійна модель змінного середнього

В області аналізу часових рядів модель авторегресії (autoregressive, AR) і модель змінного середнього (moving average, MA) є однією з найбільш використовуваних [5].

Модель авторегресії є виключно корисної для опису деяких зустрічаються на практиці тимчасових рядів. У цій моделі поточне значення процесу виражається як кінцева лінійна сукупність попередніх значень процесу і імпульсу, який називається «білим шумом», [5]

$$Z_{(t)} = C + \varphi_1 Z_{(t-1)} + \varphi_2 Z_{(t-2)} + \dots + \varphi_p Z_{(t-p)} + \varepsilon_1. \quad (1)$$

Формула (1) описує процес авторегресії порядку p , який в літературі часто позначається $AR(p)$, тут C – речова константа, $\varphi_1, \dots, \varphi_p$ - коефіцієнти, ε_1 - помилка моделі. Для визначення φ_1 і C використовують метод найменших квадратів або метод максимальної правдоподібності. Інший тип моделі має велике значення в описі тимчасових рядів і часто використовується спільно з авторегресії називається моделлю змінного середнього порядку q і описується рівнянням [5]

$$Z_{(t)} = \frac{1}{q} (Z_{(t-1)} + Z_{(t-2)} + \dots + Z_{(t-q)}) + \varepsilon_1. \quad (2)$$

У літературі процес часто позначається $MA(q)$; тут q - порядок змінного середнього, E_t - помилка прогнозування. модель змінного середнього є по суті справи фільтром низьких частот. Потрібно відзначити, що існують прості, зважені, кумулятивні, експоненціальні моделі змінного середнього [5].

Для досягнення більшої гнучкості в підгонці моделі часто доцільно об'єднати в одній моделі авторегресії і ковзне середнє. Загальна модель позначається $ARMA(p, q)$ з'єднує в собі фільтр у вигляді змінного середнього порядку q і авторегресії фільтрованих значень процесу порядку p [5].

Якщо в якості вхідних даних використовуються не самі значення часового ряду, а їх різниця d -того порядку (на практиці d необхідно визначати, проте в більшості випадків $d \leq 2$), то модель носить назву авторегресії проінтегрувати змінного середнього. В літературі дану модель називають $ARIMA(p, d, q)$ (autoregression integrated moving average). Розвитком моделі $ARIMA(p, d, q)$ є модель $ARIMAX(p, d, q)$, яка описується рівнянням [5]

$$Z(t) = AR(p) + \alpha_1 X_1(t) + \dots + \alpha_s X_s(t). \quad (3)$$

Тут $\alpha_1, \dots, \alpha_s$, - коефіцієнти зовнішніх факторів $X_1(t), \dots, X_s(t)$. У даній моделі найчастіше процес $Z(t)$ є результатом моделі $MA(q)$, тобто відфільтрованими значеннями вихідного процесу. Далі для прогнозування $Z(t)$ використовується модель авторегресії, в якій введені додаткові регресорів зовнішніх факторів $X_1(t), \dots, X_s(t)$ [5].

2.1.2 Класифікаційно-регресійні дерева

Структурні моделі CART розроблені для моделювання процесів, на які впливають як безперервні зовнішні фактори, так і категоріальні. якщо зовнішні фактори, що впливають на процес $Z(t)$, неперервні, то використовуються регресійні дерева; якщо чинники категоріальні, то - класифікаційні дерева. У разі, якщо необхідно враховувати чинники обох типів, то використовуються змішані класифікаційно-регресійні дерева [6].

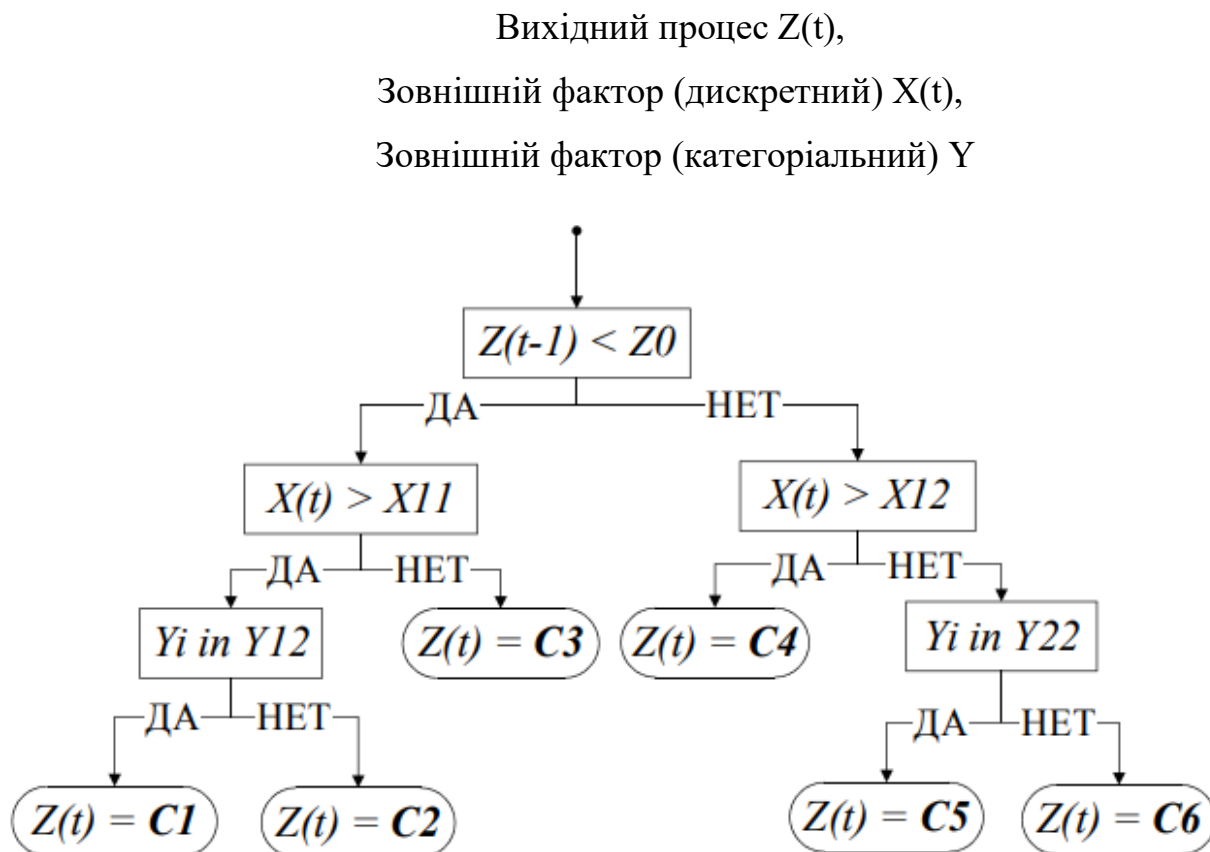


Рисунок 1.3.1 – Бінарне класифікаційно-регресивне дерево

Згідно з моделлю CART, прогнозне значення часового ряду залежить від попередніх значень, а також деяких незалежних змінних. На наведеному на рисунку 1.3.1 прикладі спочатку попереднє значення процесу порівнюється з константою Z_0 . Якщо значення $Z_{(t-1)}$ менше Z_0 , то виконується наступна перевірка: $X_{(t)} > X_{11}$. Якщо нерівність не виконується, то $Z_{(t)} = C_3$, інакше перевірки тривають до того моменту, поки не буде знайдений лист дерева, в якому відбувається визначення майбутнього значення процесу $Z_{(t)}$. Важливо, що при визначенні значення в розрахунок приймаються як безперервні змінні, наприклад, $Z_{(t)}$, так і категоріальні Y , для яких виконується перевірка присутності значення в одному з попередньо визначених підмножин. значення порогових констант, наприклад, Z_0 , X_{11} , а також підмножин Y_{12} , Y_{22} виконується на етапі навчання дерева [6].

Таким чином, CART моделює залежність майбутньої величини процесу $Z_{(t)}$ за допомогою структури дерева, а також граничних констант і підмножин [6].

2.1.3 Метод ковзної середньої

Виявлення основної тенденції ряду динаміки може бути здійснене також методом ковзної середньої. Для визначення ковзної середньої формують укрупнені інтервали, що складаються з однакового числа рівнів. При цьому кожен подальший ущільнений інтервал отримують шляхом поступового зрушення від початкового рівня ряду динаміки на один його рівень. Ущільнений інтервал згладжування як би ковзає по динамічному ряду з кроком, рівним одиниці. За сформованими укрупненими інтервалами визначають суму значень рівнів, на основі яких розраховуються ковзні середні. Отримані середні належать до середин ущільнених інтервалів. Тому при згладжуванні ковзної середньої технічно зручніше укрупнений інтервал складати з непарного числа рівнів ряду динаміки [17].

При використанні методу ковзної середньої велике значення має вибір періоду або інтервалу ковзання. Він повинен відповідати періоду коливань у даному динамічному ряду [17].

Метод ковзної середньої застосовується в статистиці досить часто й практично є найпоширенішим методом виявлення тренда. Зовні він є всього лише суто емпіричним прийомом попереднього аналізу, в якому переважають арифметичні дії з рівнями динамічного ряду, але осмислення характеру розвитку явища відбувається при визначенні періоду ковзання, оскільки середня, отримана методом ковзання, відображає не чисто арифметичну операцію, а перехід до укрупнених інтервалів часу [17].

2.1.4 Метод найменших квадратів

Суть методу найменших квадратів полягає в відшукуванні параметрів моделі тренда, яка краще всього описує тенденцію розвитку якого-небудь випадкового явища в часі або в просторі (тренд – це лінія, яка й характеризує тенденцію цього розвитку) [17].

Завдання методу найменших квадратів (МНК) зводиться до знаходження не просто якоїсь моделі тренда, а до знаходження кращої або оптимальної моделі. Ця модель буде оптимальною, якщо сума квадратичних відхилень між спостережуваними фактичними величинами й відповідними їм розрахунковими величинами тренда буде мінімальною (найменшою) [17].

Інструментарій МНК зводиться до наступних процедур:

Перша процедура. З'ясовується, чи існує взагалі яка-небудь тенденція зміни результативної ознаки при зміні вибраного чинника-аргументу, або іншими словами, чи є зв'язок між «у» і «х».

Друга процедура. Визначається, яка лінія (траєкторія) здатна краще всього описати або охарактеризувати цю тенденцію.

Третя процедура. Розраховуються параметри регресійного рівняння, що характеризує дану лінію, або іншими словами, визначається аналітична формула, що описує кращу модель тренда [17].

2.2 Переваги і недоліки моделей

Регресивні моделі і методи. До переваг даних моделей відносять простоту, гнучкість, а також однаковість їх аналізу і проектування. При використанні лінійних регресійних моделей результат прогнозування може бути отриманий швидше, ніж при використанні інших моделей. Крім того, перевагою є прозорість моделювання, доступність для аналізу всіх проміжних обчислень [1].

Основним недоліком нелінійних регресійних моделей є складність визначення виду функціональної залежності, а також трудомісткість визначення параметрів моделі. недоліками лінійних регресійних моделей є

низька адаптивність і відсутність здатності моделювання нелінійних процесів [2].

Моделі на базі класифікаційно-регресійних дерев. Перевагами даного класу моделей є: масштабованість, за рахунок якої можлива швидка обробка надвеликих обсягів даних; швидкість і однозначність процесу навчання дерева, а також можливість використовувати категоріальні зовнішні чинники [6].

Недоліками даних моделей є неоднозначність алгоритму побудови структури дерева; складність питання зупинки т. е. питання про те, коли варто припинити подальші розгалуження; відсутність однаковості їх аналізу та проектування [6].

2.3 Методи оцінки точності прогнозу

Для визначення ступеня адекватності моделі прогнозування необхідно використати ряд методик з оцінки точності прогнозу, зокрема – помилки прогнозування. Помилка прогнозування e_t – це різниця між реалізацією ряду та прогнозним значенням ряду:

$$e_t = X_{r+t} - X_{r+t} \quad . \quad (4)$$

де X_{r+t} - реальне значення ряду, X_{r+t} - прогнозне значення ряду.

Розрахунок середнього значення помилки прогнозування (Mean Forecast Error) MFE – найпростіший спосіб оцінити точність прогнозування:

$$MFE = \frac{1}{n} \sum_{t=1}^n e_t. \quad (5)$$

Дозволяє оцінити середнє відхилення прогнозних значень від реалізації ряду, а також знак або напрямок помилки. При цьому абсолютне значення MFE ще означає високої точності прогнозування. До того ж MFE залежить від тимчасового масштабу ряду та слабо враховує екстремальні значення помилки прогнозу [18].

Абсолютна середня помилка прогнозування MAE (MeanAbsoluteError):

$$MAE = \frac{1}{n} \sum_{t=1}^n |e_t|. \quad (6)$$

Дозволяє розрахувати середнє абсолютне відхилення прогнозу від реальних значень низки оцінити сумарну помилку прогнозу. На відміну від MFE, негативні та позитивні за значенням помилки прогнозування не компенсують один одного, при цьому неможливо оцінити напрямок помилки. Також залежить від тимчасового масштабу вибірки та слабо враховує екстремальні значення помилок [19].

Середня абсолютна помилка у відсотках MAPE (Mean Absolute Percentage Error):

$$MAPE = \frac{1}{n} \sum_{t=1}^n \left| \frac{e_t}{X_t} \right| * 100. \quad (7)$$

Оцінює середній відсоток помилки прогнозування без урахування знаку. MAPE не відображає екстремальних значень помилок, при цьому протилежні знак помилки не компенсують один одного [19].

Знакова середньоквадратична помилка SMSE (Signed Mean Squared)Error):

$$SMSE = \frac{1}{n} \sum_{t=1}^n \left(\frac{e_t}{|e_t|} \right) e_t^2. \quad (8)$$

Показує середньоквадратичне відхилення помилки прогнозу з урахуванням знак.

Нормальна середньоквадратична помилка (Normalized Mean Squared Error) NMSE:

$$NMSE = \frac{1}{\sigma^2 n} \sum_{t=1}^n e_t^2. \quad (9)$$

$$\sigma^2 = \frac{1}{n-1} \sum_{t=1}^n (X_t - \bar{X})^2. \quad (10)$$

$$\bar{X} = \frac{1}{n} \sum_{t=1}^n X_t. \quad (11)$$

Досить ефективний спосіб оцінки точності прогнозування, при якому чим нижче значення NMSE, то точніше модель [19].

Як правило, при оцінці точності методу прогнозування тимчасового ряду, використовується ряд метрик, що дозволяють розрахувати абсолютне значення помилки та її спрямованість [19].

Розглянуті алгоритми прогнозування процесів є загальновідомими та у завданнях дослідження тимчасових рядів застосовуються досить давно. Однак для прогнозування реального процесу, в силу його характеру, деякі моделі підходять краще за інших. При цьому підбір відповідної моделі, що описує досліджуваний процес, є нетривіальним завданням, вирішення якого необхідно базувати на точних статистичних та динамічних даних про природу самого процесу [19].

Висновки до розділу 2.

Для вибору відповідного методу прогнозування часових рядів були розглянуті найбільш відомі методи.

Було розглянуто методи оцінки точності прогнозу.

На основі огляду методів вибрано метод – для прогнозування часових рядів.

3 ПРОЕКТУВАННЯ Й РОЗРОБКА ЗАБЕЗПЕЧЕННЯ ДЛЯ ДОСЛІДЖЕННЯ

3.1 Формалізація задачі

Формалізація задачі на рівні зовнішнього проектування показана у вигляді діаграми варіантів використання. Користувач програми представлений у вигляді актора, що взаємодіє з системою.

Діаграма варіантів використання системи представлена на рис. 3.1.

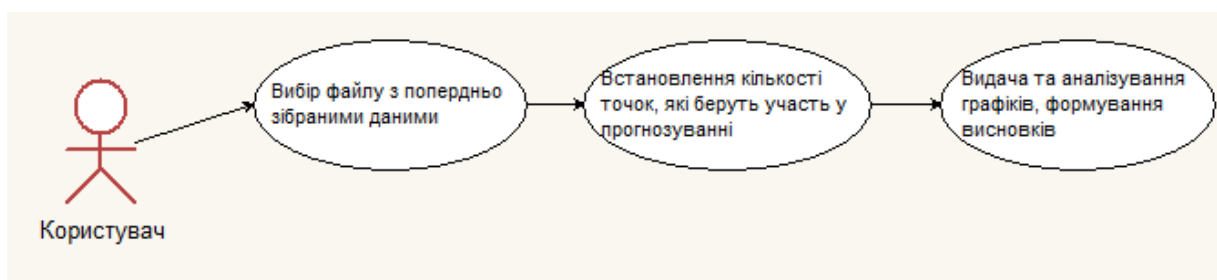
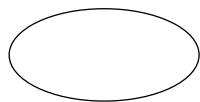


Рисунок 3.1 – Діаграма варіантів використання

На рисунку 3.1 показано:



– образне представлення користувача системи;



– можливості, що може виконувати користувач в системі.

3.2 Базова архітектура системи

Проектування інструментального засобу було розпочато з урахуванням розробки програмного продукту під 3-рівневу MVC-модель. Основна парадигма продукту – об'єктно-орієнтована, функціональна, на рівні контролера присутній зв'язок з модулем, який працює на об'єктно-орієнтованій мові C#.

Об'єктно-орієнтоване програмування є новим підходом до створення комп'ютерних додатків, який покликаний усунути багато проблем, що існують у традиційних методиках програмування.

Об'єктно-орієнтований підхід є ефективним засобом зниження складності, тривалості і трудоемкості процесів проектування і

супроводження великих програмних систем шляхом використання загальних механізмів, що забезпечують необхідну економію виразних засобів

Об'єктно–орієнтоване програмування створює умови для багаторазового використання програмних компонентів і об'єднання даних з методами їх обробки, що у свою чергу надає можливість управління елементами будь-якої складності. Об'єктна модель здатна однаково добре описати як елементи графічного інтерфейсу, так і об'єкти реального світу [7, 8].

Патерн проектування MVC включає кілька дрібніших шаблонів. При використанні MVC на три окремих компоненти розділені модель даних програми, інтерфейс користувача і логіка взаємодії користувача з системою, завдяки чому модифікація одного з цих компонентів надає мінімальний вплив на інші або не надає його зовсім [9].

Основна мета застосування MVC полягає у розподілі даних та бізнес-логіки від візуалізації. За рахунок такого розподілу підвищується можливість повторного використання програмного коду.

Концепція MVC поділяє дані, представлення та обробку дій користувача на компоненти:

Модель (Model) – являє собою об'єктну модель предметної області, включає дані та методи роботи з цими даними, реагує на запити з контролера, повертаючи дані та/або змінюючи свій стан. При цьому модель не містить інформації про способи візуалізації даних або форматах їх подання, а також не взаємодіє з користувачем безпосередньо.

Подання (View) – відповідає за відображення інформації. Одні й самі дані представляються різними способами й у різних форматах. Відповідає за отримання необхідних даних з моделі та надсилає їх користувачу. Подання не обробляє введені дані користувача

Контролер (Controller) – забезпечує зв'язок між користувачем та системою, використовує модель та подання для реалізації необхідної реакції на дії користувача. Як правило, на рівні контролера здійснюється фільтрація

отриманих даних та авторизація – перевіряються права користувача на виконання дій або отримання інформації.[9]

Схема взаємодії системи з участю MVC-моделі зображена на рис. 3.2.

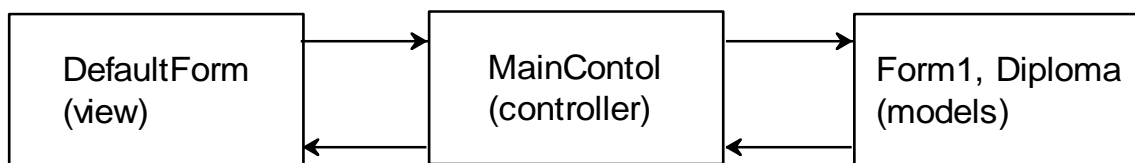


Рисунок 3.2 – Схема взаємодії частин моделі MVC

Детальна діаграма взаємодії компонентів системи представлена на рисунку 3.3.

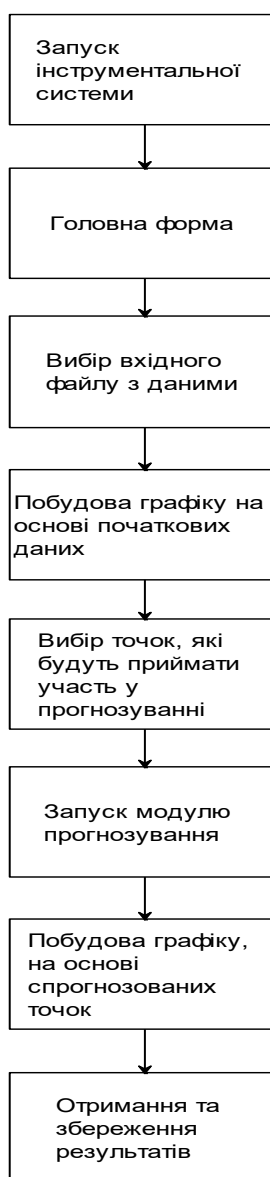


Рисунок 3.3 – Діаграма взаємодії компонентів системи

3.3 Внутрішнє проектування

3.3.1 Вибір мови програмування

Для розробки програмного продукту у якості мови для написання додатку було обрано C #.

C # – це мова програмування, що поєднує об'єктно-орієнтовані і контекстно-орієнтовані концепції. Розроблено в 1998-2001 роках групою інженерів під керівництвом Андерса Хейлсберга в компанії Microsoft як основна мова розробки додатків для платформи Microsoft .NET. Компілятор з C # входить в стандартну установку самої .NET, тому програми на ньому можна створювати і компілювати навіть без інструментальних засобів на кшталт Visual Studio [7].

Синтаксис C # дуже виразний, але простий у вивченні. Синтаксис C# робить простіше те, що було складно в C ++, і забезпечує потужні можливості, такі як типи значень Nullable, перерахування і прямий доступ до пам'яті. C# підтримує універсальні методи і типи, забезпечуючи більш високий рівень безпеки і продуктивності. Як об'єктно-орієнтована мова, C # підтримує поняття інкапсуляції, успадкування та поліморфізму. Клас може успадковувати безпосередньо з одного родового класу, але може реалізовувати будь-яке число інтерфейсів. Для методів, які скасовують віртуальні методи в батьківському класі, необхідно ключове слово `override`, щоб виключити випадковий повторне визначення. У мові C # структура схожа на полегшений клас: це тип, що розподіляється по стопках, який реалізує інтерфейси, але не підтримує спадкування [8].

Особливості C# полягають у тому, що присутня повна підтримка класів та об'єктно-орієнтованого програмування, включаючи успадкування інтерфейсів та реалізацій, віртуальних функцій та перевантаження операторів є повний та добре визначений набір основних типів, вбудована підтримка автоматичної генерації XML-документації, автоматичне звільнення динамічно пам'яті, має можливість позначки класів та методів атрибутами,

що визначаються користувачем. Це може бути корисно при документуванні та здатне впливати на процес компіляції [23].

Має покажчики і прямий доступ до пам'яті, якщо вони необхідні, проте мова розроблена таким чином, що практично завжди можна обійтися і без цього. Проста зміна ключів компіляції. Дозволяє отримувати файли або бібліотеки компонентів .NET, що виконуються, які можуть бути викликані іншим кодом [24].

3.3.2 Технологічна платформа

Серед сучасних продуктів для розробки програмного продукту займає місце Visual Studio, тому саме його було обрано в якості середовища для створення програмного продукту.

Microsoft Visual Studio – серія продуктів фірми Майкрософт, які включають інтегроване середовище розробки програмного забезпечення та ряд інших інструментальних засобів. Ці продукти дозволяють розробляти як консольні програми, так і програми з графічним інтерфейсом, в тому числі з підтримкою технології Windows Forms, а також веб-сайти, веб-застосунки, веб-служби як в рідному [10].

Функціональність Visual Studio охоплює всі етапи розробки програмного забезпечення, надаючи сучасні інструменти для написання коду, проектування графічних інтерфейсів, складання, налагодження та тестування програм. Можливості Visual Studio можуть бути доповнені шляхом підключення потрібних розширень.

Редактор коду Visual Studio підтримує підсвічування синтаксису, вставку фрагментів коду, відображення структури та пов'язаних функцій. Істотно прискорити роботу допомагає технологія IntelliSense – автозавершення коду у міру введення.

Вбудований відладчик Visual Studio використовується для пошуку та виправлення помилок у вихідному коді, у тому числі на низькому апаратному

рівні. Інструменти діагностики дозволяють оцінити якість коду з погляду продуктивності та використання пам'яті.

Дизайнер форм Visual Studio незамінний при розробці програм з графічним інтерфейсом, допомагаючи спроектувати зовнішній вигляд майбутньої програми та роботу кожного елемента інтерфейсу [10].

3.3.3 Ієрархія та взаємодія класів системи

Діаграма класів визначає типи класів системи та різного роду статичні зв'язки, які існують між ними. На діаграмах класів зображуються також атрибути класів, операції класів та обмеження, що накладаються на зв'язки між класами.

Результат проектування та розробки діаграми класів зображено на рис. 3.4.

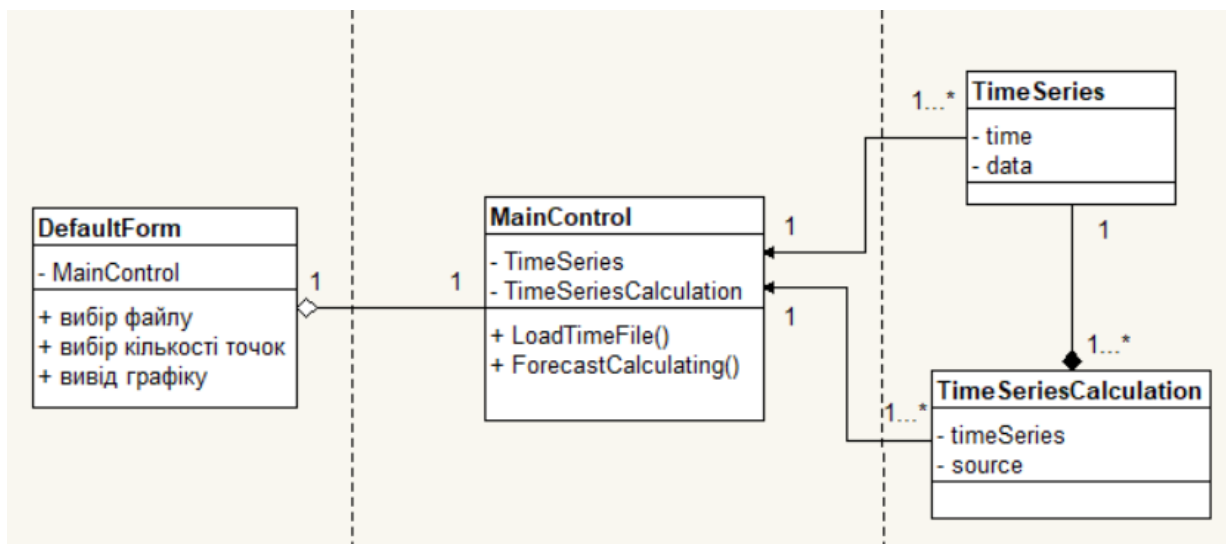


Рисунок 3.4 – Діаграма класів

3.3.4 Використанні принципи проектування

Об'єктно-орієнтоване програмування (ООП) – методологія програмування, заснована на представленні програми у вигляді сукупності об'єктів, кожен з яких є екземпляром певного класу, а класи утворюють ієрархію спадкування. ООП виникло в результаті розвитку ідеології процедурного програмування, де дані і підпрограми (процедури, функції) їх обробки формально не пов'язані. Для подальшого розвитку об'єктно-

орієнтованого програмування часто велике значення мають поняття події (так зване подієво-орієнтоване програмування) і компонента (компонентне програмування). Першою мовою програмування, в якому було запропоновано основні поняття, які згодом склалися в парадигму, була Симула, але термін «об'єктна орієнтованість» не використовувався в контексті використання цієї мови. У момент його появи в 1967 році в ньому було запропоновано революційні ідеї: об'єкти, класи, віртуальні методи і ін [24].

В ході створення програмного продукту системи було використано принципи об'єктно-орієнтованого програмування:

- абстракція – абстрагування означає виділення значущої інформації і виключення з розгляду незначною. У ООП розглядають лише абстракцію даних (нерідко називаючи її просто «абстракцією»), маючи на увазі набір значущих характеристик об'єкта, доступний решті програми;

- інкапсуляція – властивість системи, що дозволяє об'єднати дані і методи, що працюють з ними, в класі. Деякі мови (наприклад, C++) ототожнюють інкапсуляцію з приховуванням, але більшість (Smalltalk, Eiffel, OCaml) розрізняють ці поняття;

- спадкування – властивість системи, що дозволяє описати новий клас на основі вже існуючого з частково або повністю позичас функціональністю. Клас, від якого виробляється спадкування, називається базовим, батьківським або суперкласом. Новий клас – нащадком, спадкоємцем, дочірнім або похідним класом;

- поліморфізм підтипів (в ООП званий просто «поліморфізмом») — властивість системи, що дозволяє використовувати об'єкти з однаковим інтерфейсом без інформації про тип і внутрішню структуру об'єкта. Інший вид поліморфізму – параметричний – в ООП називають узагальненим програмуванням [26].

3.4 Розробка інтерфейсу користувача

3.4.1 Розробка структури інтерфейсу системи

На рис. 3.5 зображено прототип головної сторінки, з якою буде працювати користувач.

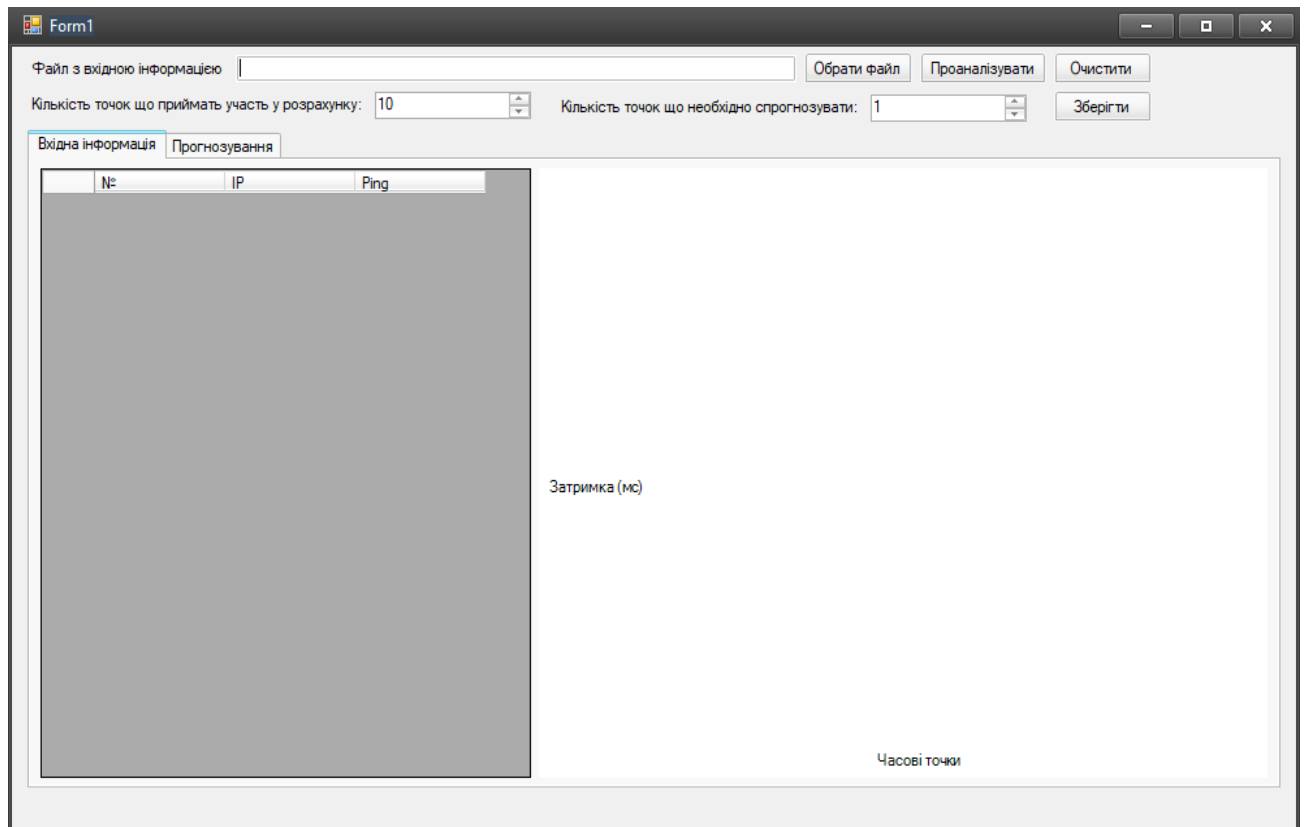


Рисунок 3.5 – Прототип головної сторінки

Об'єкти, які були використанні при розробці інтерфейсу програмного продукту та можливі дії над ними зображені у табл. 3.1.

Таблиця 3.1 – Використанні об'єкти

Назва об'єкту	Тип	Дії над об'єктом
1	2	3
label1	контейнер	Вивід підписів
label2	контейнер	Вивід підписів
textBox1	контейнер	Вивід тексту
numericUpDown1	засіб керування	Задання числових значень
numericUpDown2	засіб керування	Задання числових значень

Продовження таблиці 3.1

1	2	3
button1	засіб керування	Взаємодія з програмою
button2	засіб керування	Взаємодія з програмою
button3	засіб керування	Взаємодія з програмою
button4	засіб керування	Взаємодія з програмою
tabControl1	засіб керування	Перехід по вкладкам
dataGridView1	засіб керування	Відображення таблиці
dataGridView2	засіб керування	Відображення таблиці
chart2	засіб керування	Відображення графіку

3.4.2 Реалізація інтерфейсу користувача

Головна форма програми, що включає в себе усі необхідні для роботи з програмним засобом, такі як, поля вводу посилання на файл з вхідними даними, поле для вибору кількості точок, що будуть приймати участі у прогнозуванні, вікно з отриманим графіком, екрана форма з представленими даними у табличному вигляді зображено на рис. 3.6.

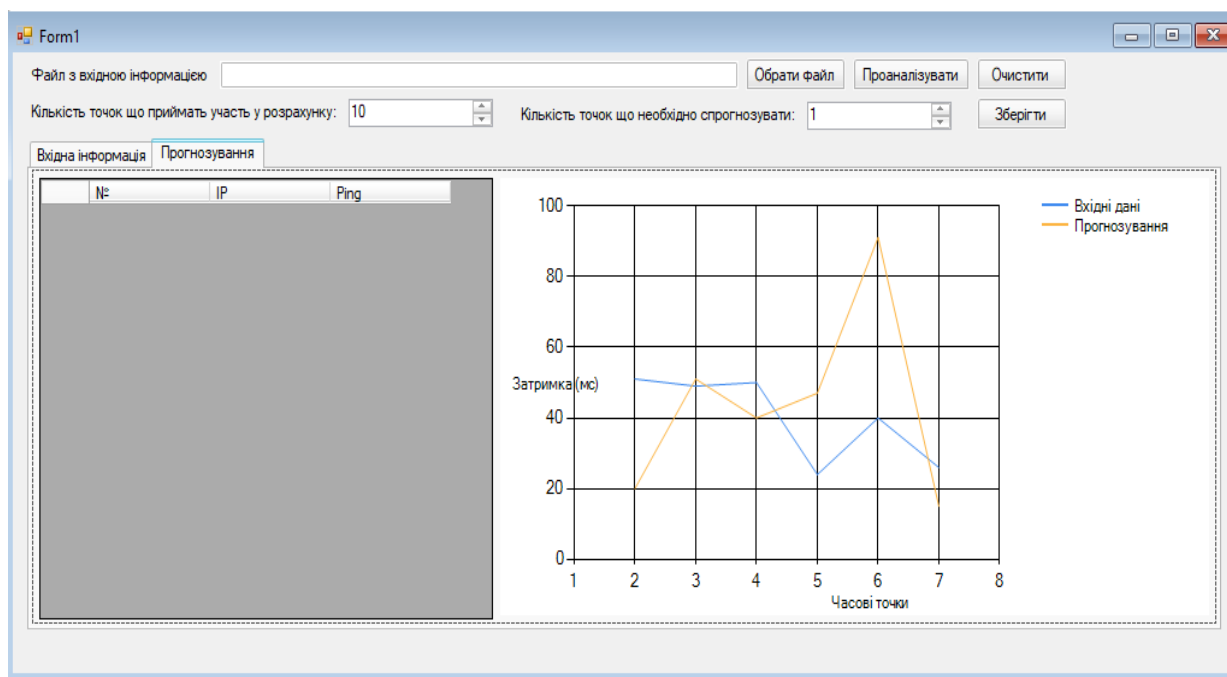


Рисунок 3.6 – Головна екрана форма

На рис. 3.7 зображена екрана форма після натискання кнопки «Обрати файл», при якій відкривається вікно вибору файлу для подальшої роботи програмного продукту.

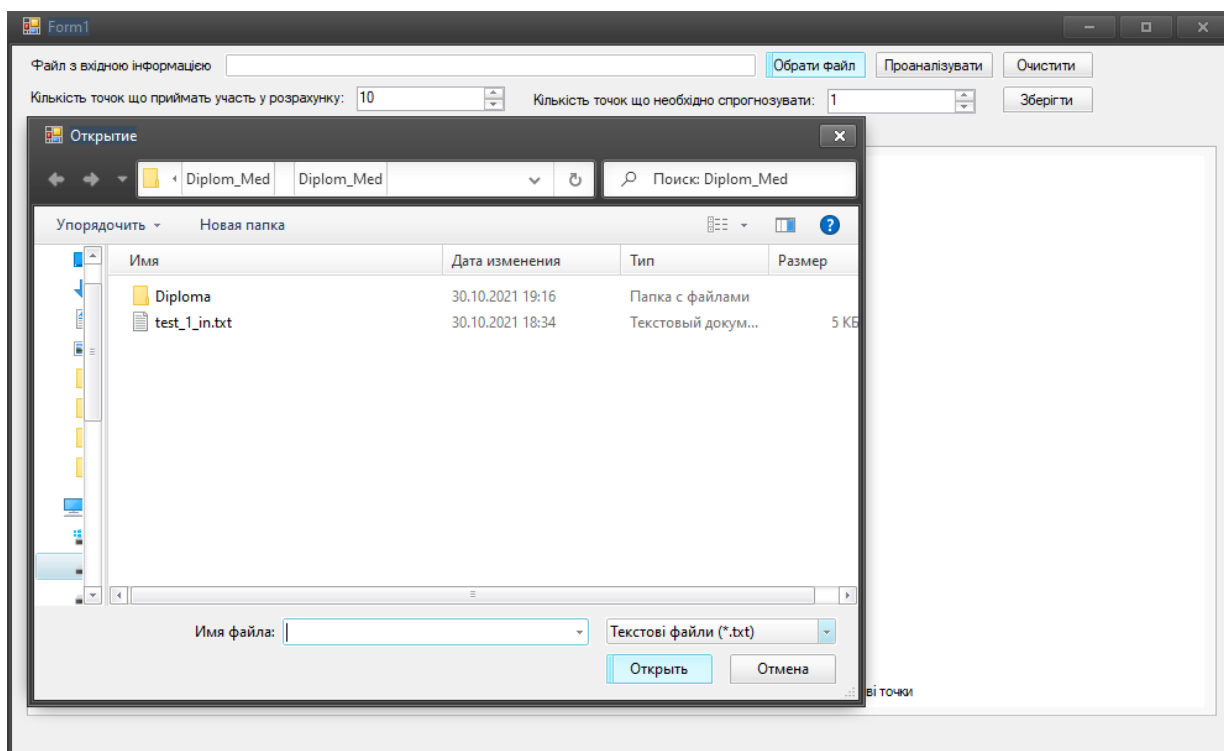


Рисунок 3.7 –Екрана форма вибору файлу з даними

На рис. 3.8 зображена екрана форма, після вибору файлу. Якє відображує табличні вхідні дані з файлу та побудований за даними графік.

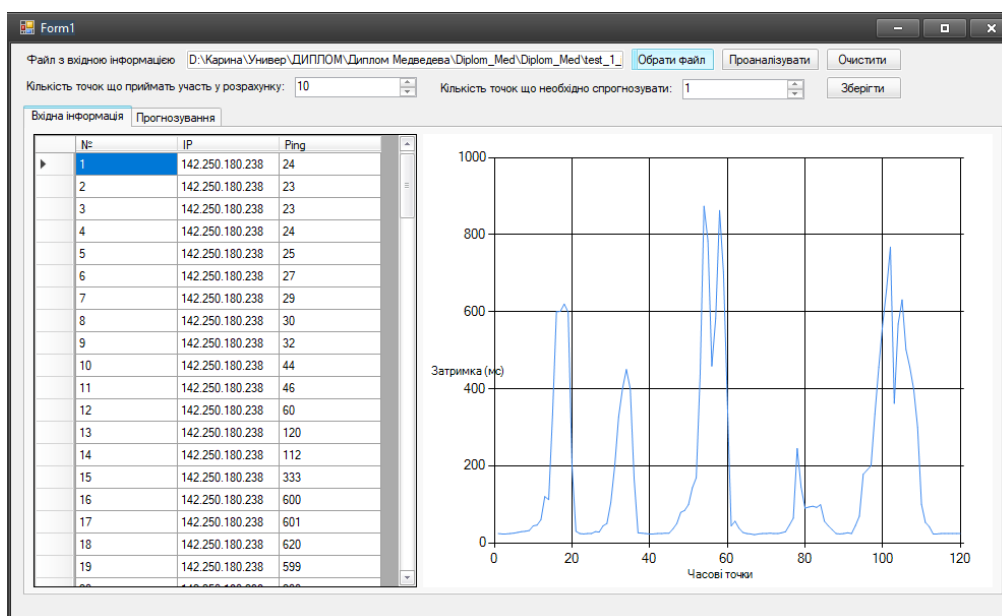


Рисунок 3.8 – Екрана форма головної форми з вхідними даними

На рис. 3.9 зображена екрана форма, у якому вибрано точки, які приймають участь у розрахунку.

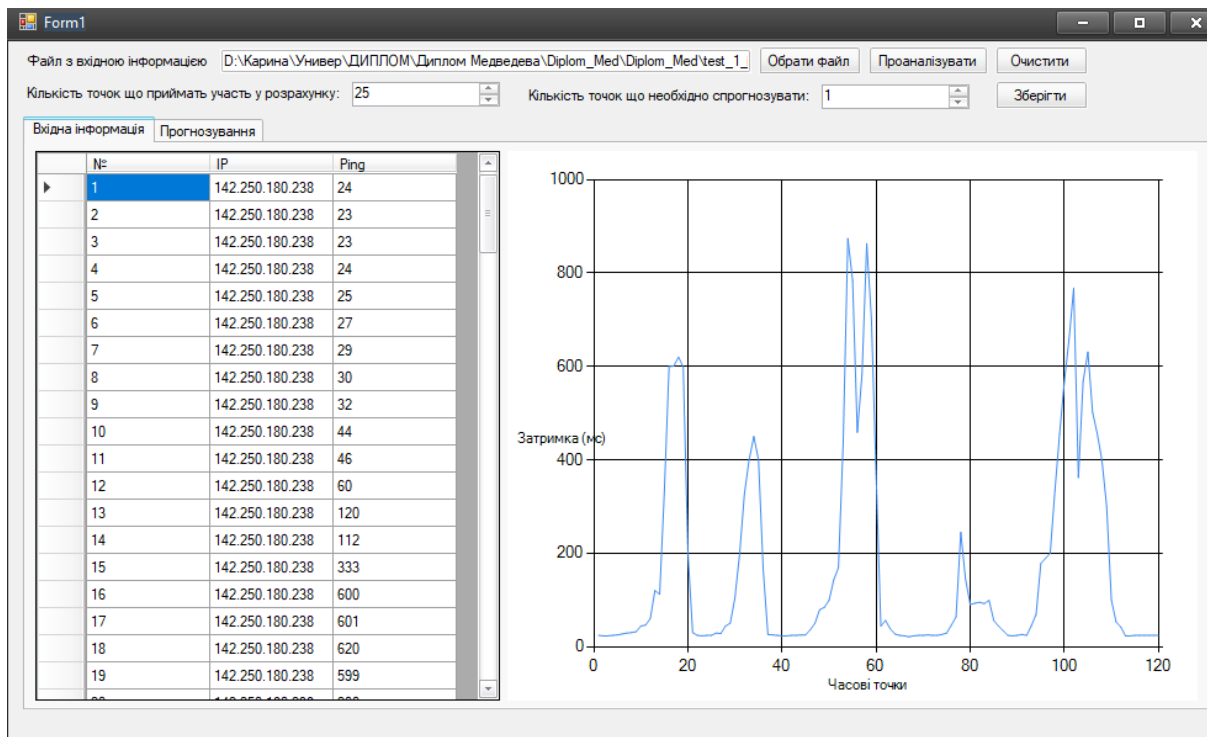


Рисунок 3.9 – Екрана форма з точками, які приймають участь у розрахунку

На рис. 3.10 зображена екрана форма програмного продукту, після того як вибрали кількість точок, які будуть приймати участь у прогнозуванні.

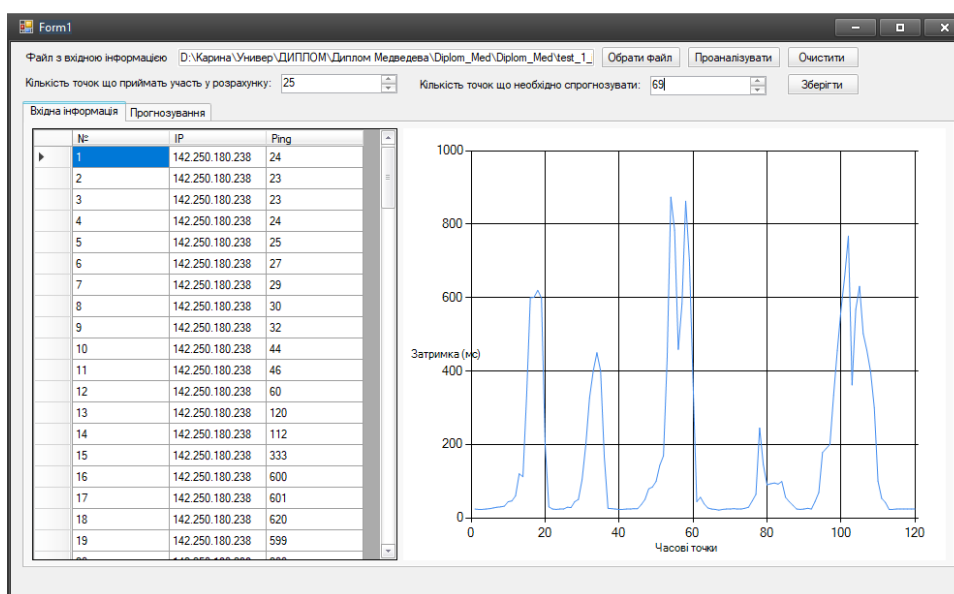


Рисунок 3.10 – Екрана форма з точками, які будуть приймати участь у прогнозуванні

На рис. 3.11 зображена екрана форма після натискання кнопки «Проаналізувати», на якій видно як змінився графік прогнозування навантаженості мережевої системи.

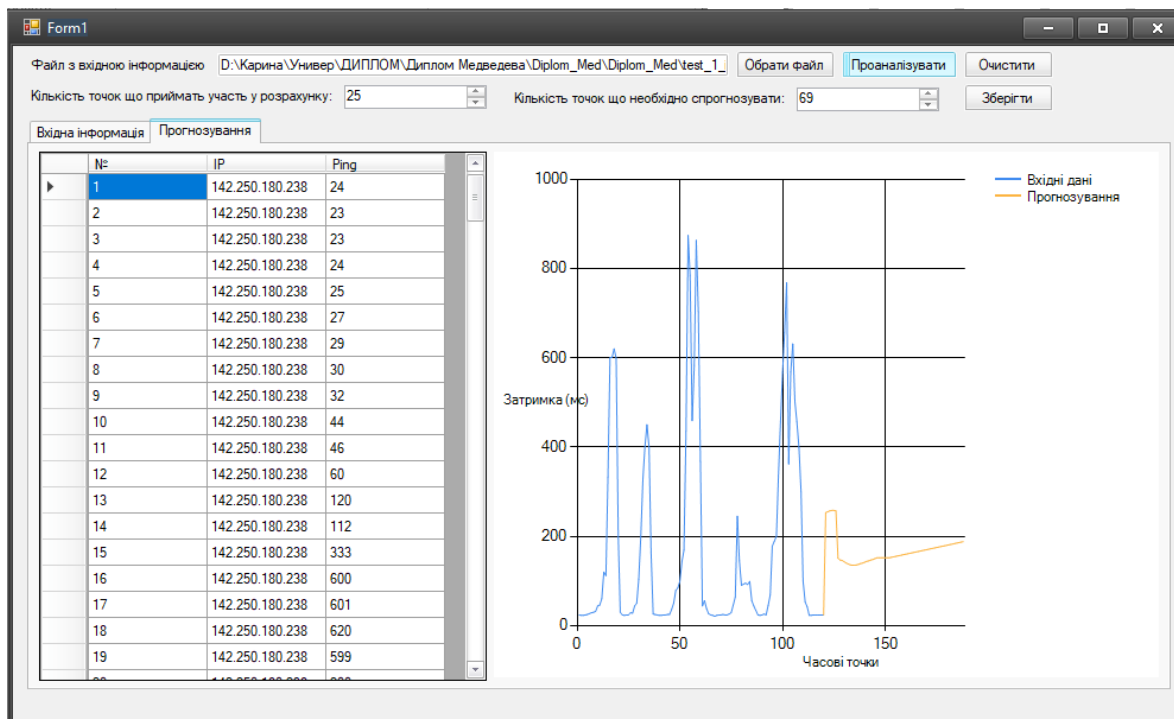


Рисунок 3.11 – Екрана форма після натискання кнопки «Проаналізувати»

На рис. 3.12 зображено таблицю головної форми, після проаналізованих даних з точками, які брали участь у прогнозуванні.

Вхідна інформація			Прогнозування
№	IP	Ping	
116	142.250.180.238	24	
117	142.250.180.238	24	
118	142.250.180.238	24	
119	142.250.180.238	24	
120	142.250.180.238	24	
121	Прогноз	253	
122	Прогноз	255	
123	Прогноз	257	
124	Прогноз	258	
125	Прогноз	258	
126	Прогноз	257	
127	Прогноз	151	
128	Прогноз	147	
129	Прогноз	146	
130	Прогноз	143	
131	Прогноз	140	
132	Прогноз	138	
133	Прогноз	136	
134	Прогноз	135	

Рисунок 3.12 – Екрана форма з таблицею прогнозованих точок

На рис. 3.13 зображено, як змінився графік після натискання кнопки «Проаналізувати», на якому видно як впливають спрогнозовані точки.

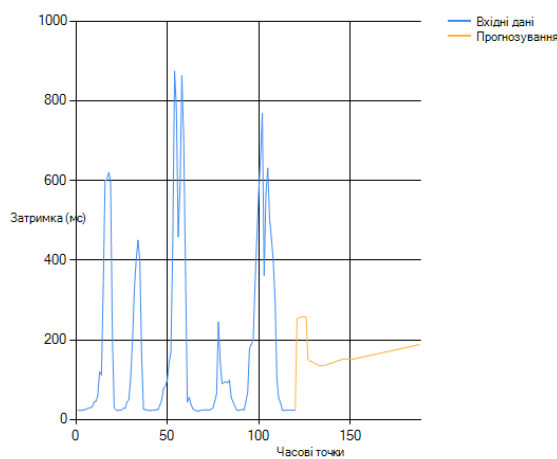


Рисунок 3.13 – Екрана форма після натискання кнопки «Проаналізувати»

На рис. 3.14 зображена екрана форма програмного засобу, після натискання кнопки «Зберегти».

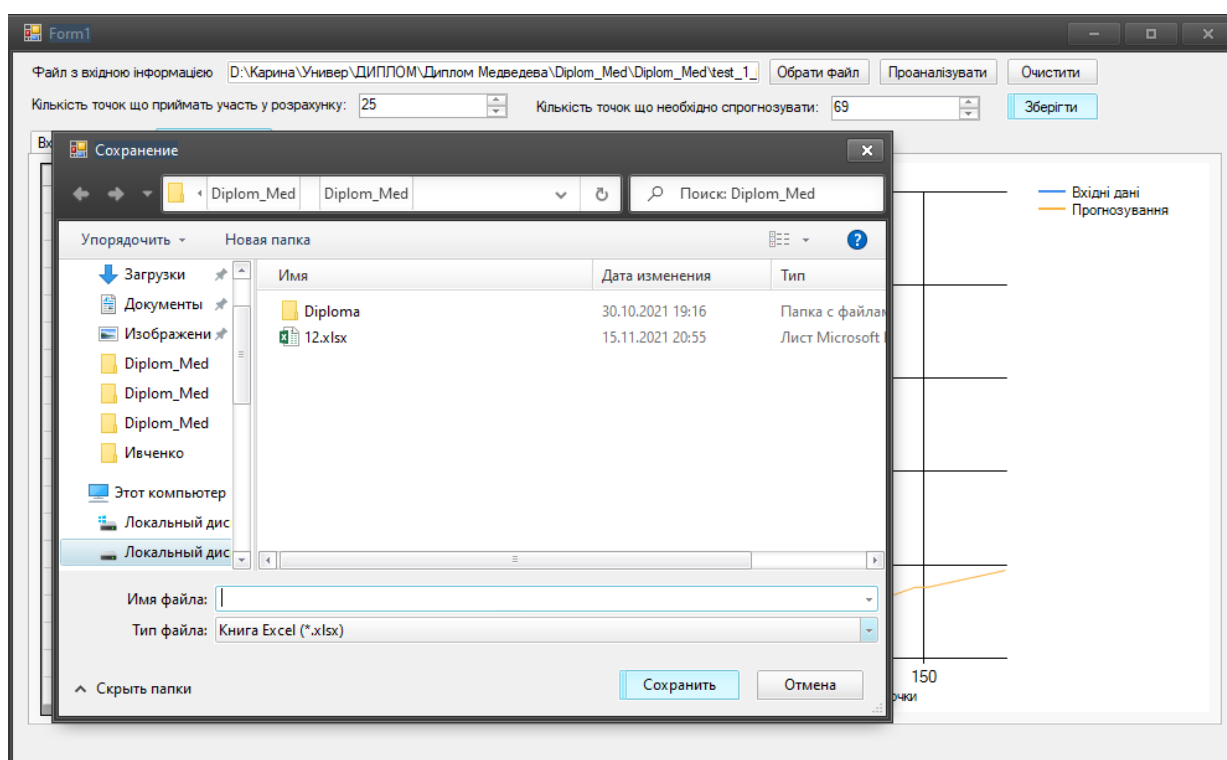


Рисунок 3.14 – Екрана форма вибору зберігання файлу

На рис. 3.15 зображена екрана форма успішного збереження файлу.

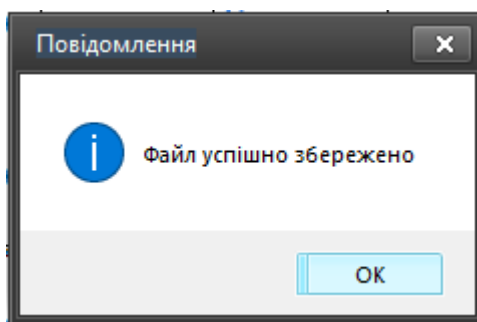


Рисунок 3.15 – Екрана форма успішного збереження файлу

На рис. 3.16 зображена екрана форма після натискання кнопки «Очистити».

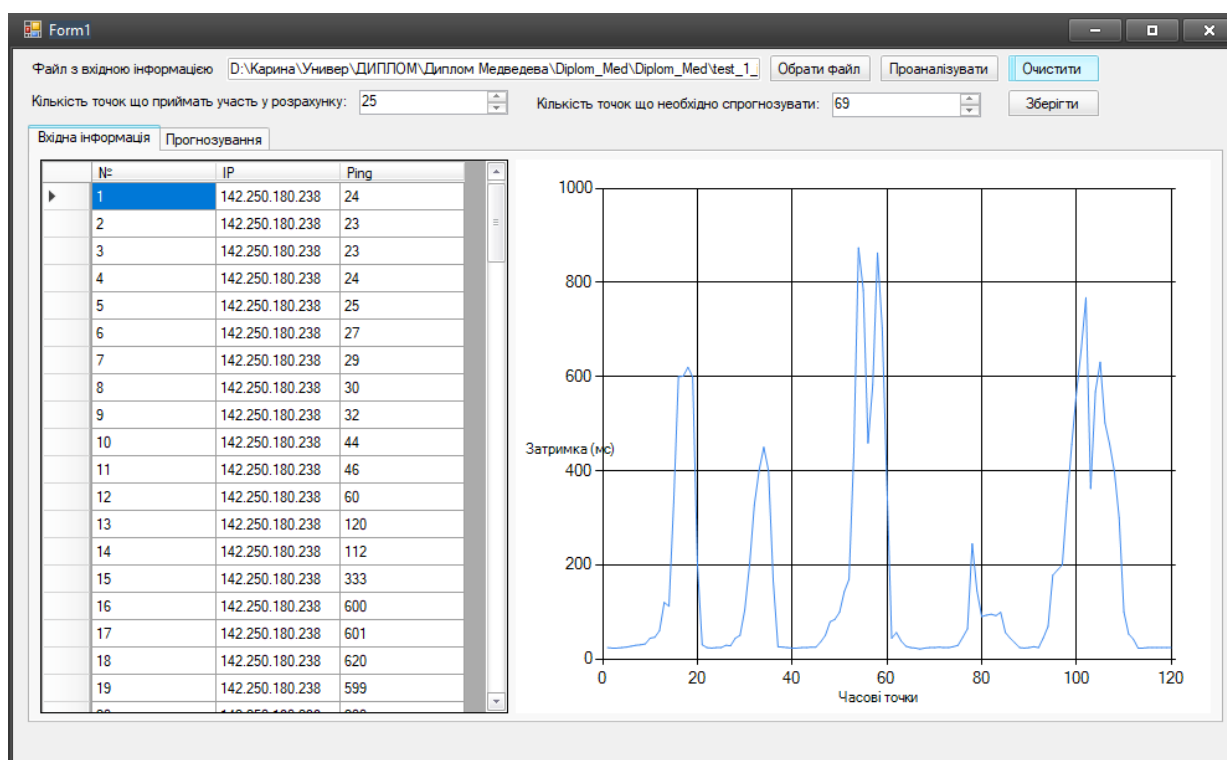


Рисунок 3.16 – Екрана форма, після натискання кнопки «Очистити»

На рис. 3.17 зображено, як змінилось вкладки «Прогнозування» після натискання кнопки «Очистити».

На рис. 3.18 зображено, як змінився графік після натискання кнопки «Очистити».

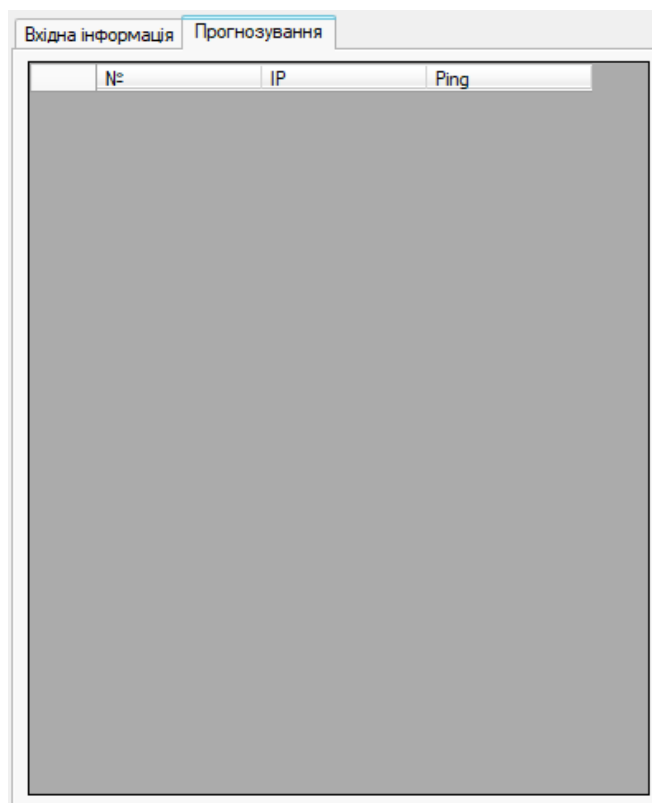


Рисунок 3.17 – Екрана форма вкладці «Прогнозування»

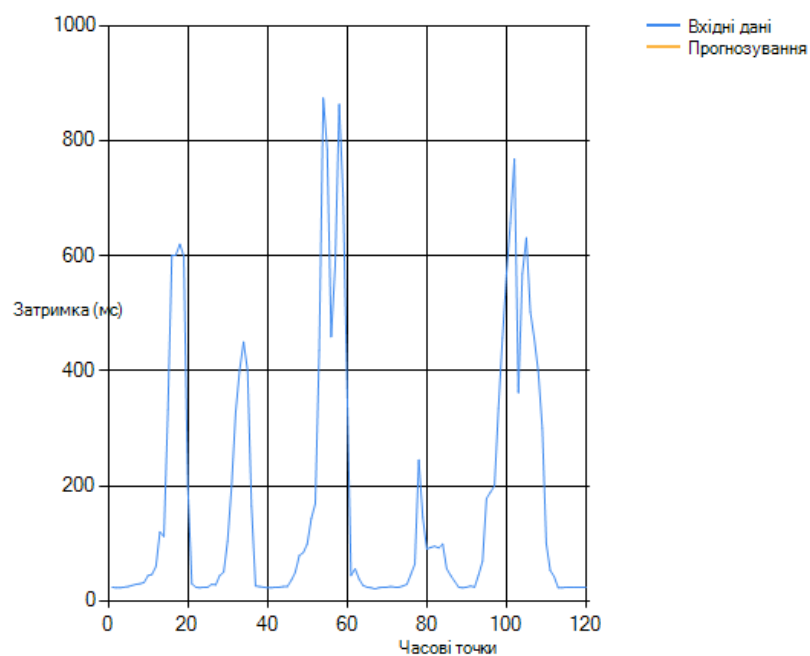


Рисунок 3.18 – Графік, після натискання кнопки «Очистити»

3.5 Тестування та налагодження програми

3.5.1 Аналіз методів тестування та відлагодження

Перші програмні системи розроблялися у межах програм наукових досліджень чи програм потреб міністерств оборони. Тестування таких продуктів проводилося строго формалізовано із записом усіх тестових процедур, тестових даних, отриманих результатів. Тестування виділялося в окремий процес, який починався після завершення кодування, але при цьому зазвичай виконувалося тим же персоналом.

У 1960-х багато уваги приділялося «вичерпному» тестуванню, яке має проводитися з використанням усіх шляхів у коді чи всіх можливих вхідних даних. Було зазначено, що в цих умовах повне тестування програмного забезпечення неможливе, тому що, по-перше, кількість можливих вхідних даних дуже велика, по-друге, існує безліч шляхів, по-третє, складно знайти проблеми в архітектурі та специфікаціях. З цих причин «вичерпне» тестування було відхилено та визнано теоретично неможливим.

На початку 1970-х років тестування програмного забезпечення позначалося як «процес, спрямований демонстрацію коректності продукту» чи як «діяльність з підтвердження правильності роботи програмного забезпечення». У програмній інженерії, що зароджувалася, верифікація ПЗ значилася як «доказ правильності». Хоча концепція була теоретично перспективною, на практиці вона вимагала багато часу і була недостатньо всеосяжною. Вирішили, що доказ правильності – неефективний метод тестування програмного забезпечення. Однак, у деяких випадках демонстрація правильної роботи використовується і в наші дні, наприклад, приймально-здатні випробування. У другій половині 1970-х тестування представлялося як виконання програми з наміром знайти помилки, а чи не довести, що вона працює. Успішний тест це тест, який виявляє раніше невідомі проблеми. Цей підхід прямо протилежний попередньому. Зазначені два визначення є «парадоксом тестування», в основі якого лежать два

протилежні твердження: з одного боку, тестування дозволяє переконатися, що продукт працює добре, а з іншого – виявляє помилки в програмах, показуючи, що продукт не працює. Друга мета тестування є більш продуктивною з точки зору покращення якості, оскільки не дозволяє ігнорувати недоліки програмного забезпечення.

У 1980-х роках тестування розширилося таким поняттям, як попередження дефектів. Проектування тестів – найефективніший із відомих методів попередження помилок. У цей час стали висловлюватися думки, що необхідна методологія тестування, зокрема, що тестування має включати перевірки протягом циклу розробки, і це має бути керований процес. У ході тестування треба перевірити не лише зібрану програму, а й вимоги, код, архітектуру, тести. «Традиційне» тестування, яке існувало до початку 1980-х, стосувалося тільки скомпільованої, готової системи (зараз це зазвичай називається системне тестування), але надалі тестувальники почали залучатися до всіх аспектів життєвого циклу розробки. Це дозволяло раніше знаходити проблеми у вимогах та архітектурі і тим самим скорочувати терміни та бюджет розробки. У 1980-х з'явилися перші інструменти для автоматизованого тестування. Передбачалося, що комп'ютер зможе виконати більше тестів, ніж людина, і зробить це надійніше. Спочатку ці інструменти були вкрай простими і не мали можливості написання сценаріїв скриптовими мовами.

На початку 1990-х років у поняття «тестування» стали включати планування, проектування, створення, підтримку та виконання тестів та тестових оточень, і це означало перехід від тестування до забезпечення якості, що охоплює весь цикл розробки програмного забезпечення. У цей час починають з'являтися різні програмні інструменти для підтримки процесу тестування: більш просунуті середовища для автоматизації з можливістю створення скриптів та генерації звітів, системи керування тестами, програмне забезпечення для навантаження тестування. У середині 1990-х років з

розвитком Інтернету та розробкою великої кількості веб-додатків особливу популярність стало набувати «гнучке тестування» [11].

При розробці програмного забезпечення одним з ключових моментів являться тестування та відлагодження програмного засобу. Основною ціллю даного етапу є знаходження та відлагодження помилок, що могли біти допущенні при написанні.

Ще при тестуванні можливо виявити глобальні помилки, які могли бути допущенні при проектуванні програмного продукту. У разі виявлення таких помилок виправлення роботи програмного продукту може витратити багато часу, тому рекомендують проводити тестування продукту на протязі.

Після проходження тестування неможливо бути в повній мірі впевненим, що усі помилки що містяться у програмі були виправлені чи виявленні при новому етапі тестуванні також були виправленні.

Для ефективного тестування рекомендується використовувати методи, які відносяться як до «чорної скриньки» так і до «білої скриньки».

3.5.2 Тестування методом покриття операторів

Для тестування програмного продукту методом покриття операторів та методом покриття переходів було складено тест-кейс, у який входять деяка кількість тестів, що дозволили виконати повне покриття усіх переходів та операторів.

Таким чином кожний тест представляє собою одноразовий запуск відповідної частини програмного продукту з вхідними даними. З цього виходить, що маючи набір з різними вхідними даними декілька файлів це забезпечує повне покриття усіх переходів та операторів, які існують у системі.

У ході тестування програмного продукту методами покриття операторів та переходів було виявлено декілька незначних помилок, які відносились до частини, що відповідає за зовнішній вигляд продукту, ці

помилки були одразу виправлені, але вони мали незначний вплив на роботу програми.

3.5.3 Тестування методом припущення помилок

Для тестування програмного продукту методом припущення помилок було використано декілька Unit-тестів для перевірки правильності роботи алгоритмів всередині методів. Цей метод добре підходить поставленим цілям, тому що має меншу трудомісткість та добре підходить для тестування простих лінійних функції та зв'язків між методами.

У ході тестування програмного продукту методом припущення помилок були виявлені незначні помилки, які мали незначний вплив на роботу програми, але всі вони були швидко виправлені.

Висновки до розділу 3

У розділі було виконано проектування інструментального засобу для дослідження часових рядів навантаження мережевої системи. В процесі проектування системи були використанні принципи ООП.

Проведено аналіз на основі котрого було вибрано технологічну платформу Microsoft Visual Studio та мову C# для розробки програмного продукту.

Розроблено діаграми взаємодії частин програмного продукту з участю MVC-моделі, також була розроблена діаграма класів.

Розроблений розділ з описом проектування інтерфейсу користувача, де використані стандартні елементи, що роблять інтерфейс дружнім та зрозумілим користувачу.

Виконано тестування програмного продукту за допомогою методів покриття операторів, покриття переходів, припущення про помилку та Unit-тестів.

4 ДОСЛІДЖЕННЯ ЧАСОВИХ РЯДІВ НАВАНТАЖЕННЯ МЕРЕЖЕВОЇ СИСТЕМИ

4.1 Підготовка до експерименту

Для проведення експерименту підготовлені набори вхідних даних, а саме файли з вхідними даними зібраними у певний період часу.

Для демонстрації роботи зібрано дані у святкові дні та день після свят, тому що саме у ці дні набагато краще видно різницю зміну навантаженості мережевої системи, що дуже підходить для проведення експерименту.

Файл з вхідною інформацією повинен бути у вигляді «.txt», приклад файлу наведений у додатку А.

4.1.1 Експеримент 1

Маємо статистику навантаженості мережевої системи у звичайний робочий день, дані приведено на рис. 4.1.1

Файл	Правка	Формат	Вид	Справка
142.250.180.238:	число	байт=32	время=24	
142.250.180.238:	число	байт=32	время=23	
142.250.180.238:	число	байт=32	время=23	
142.250.180.238:	число	байт=32	время=24	
142.250.180.238:	число	байт=32	время=25	
142.250.180.238:	число	байт=32	время=27	
142.250.180.238:	число	байт=32	время=29	
142.250.180.238:	число	байт=32	время=30	
142.250.180.238:	число	байт=32	время=32	
142.250.180.238:	число	байт=32	время=44	
142.250.180.238:	число	байт=32	время=46	
142.250.180.238:	число	байт=32	время=60	
142.250.180.238:	число	байт=32	время=120	
142.250.180.238:	число	байт=32	время=112	
142.250.180.238:	число	байт=32	время=333	
142.250.180.238:	число	байт=32	время=600	
142.250.180.238:	число	байт=32	время=601	
142.250.180.238:	число	байт=32	время=620	
142.250.180.238:	число	байт=32	время=599	
142.250.180.238:	число	байт=32	время=200	
142.250.180.238:	число	байт=32	время=30	
142.250.180.238:	число	байт=32	время=24	
142.250.180.238:	число	байт=32	время=23	
142.250.180.238:	число	байт=32	время=24	
142.250.180.238:	число	байт=32	время=24	
142.250.180.238:	число	байт=32	время=29	
142.250.180.238:	число	байт=32	время=28	
142.250.180.238:	число	байт=32	время=44	
142.250.180.238:	число	байт=32	время=50	
142.250.180.238:	число	байт=32	время=104	

Рисунок 4.1.1 – Вхідна інформація до першого експерименту

На рис 4.1.1 наведено:

- 1 – маска підмережі;
- 2 – розмір тестового пакету;
- 3 – час, за який передається дані.

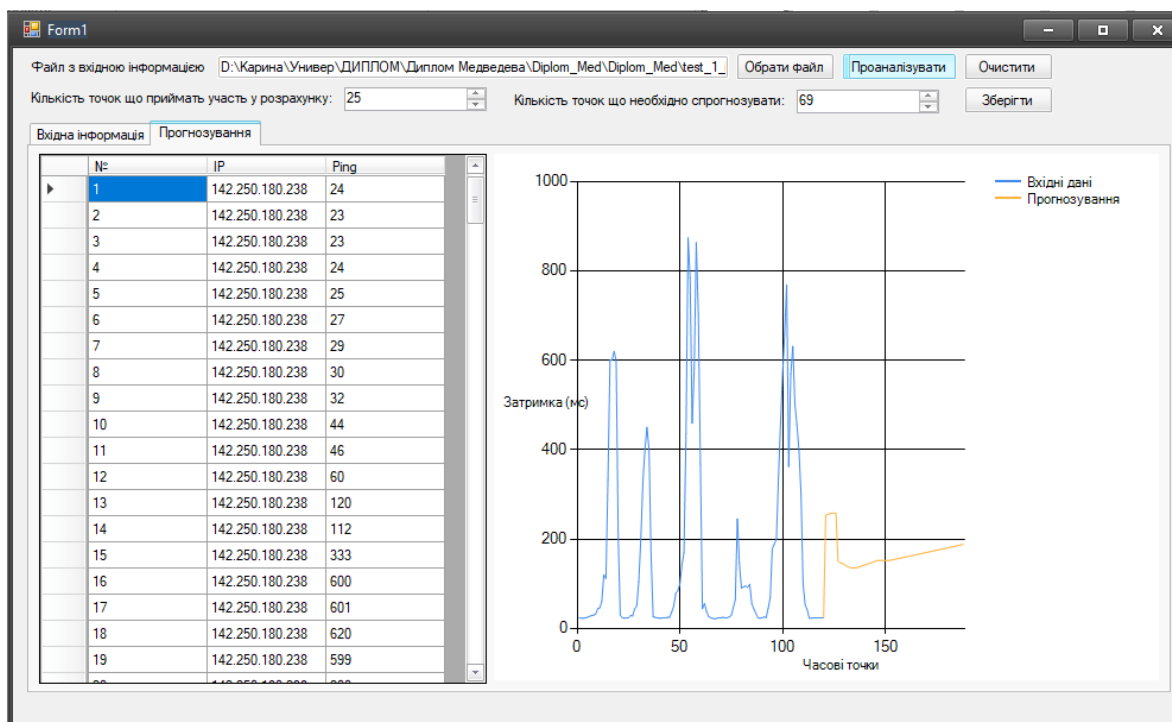


Рисунок 4.1.1 – Результат першого експерименту

Результат експерименту: при розробці графіку можна побачити зростання, падіння тренду, які є коректними. Зростання та падіння зумовлено тим, що є часи у котрі навантаження на мережу значно більше, через робочий день.

4.1.2 Експеримент 2

У цьому експерименті зібрані вхідні дані у новорічну ніч, тому що саме в цей час можливо побачити великий стрибок навантаженості в усьому світі. Вхідні дані наведено на рис. 4.1.2.

test_2_ночь НГ.txt – Блокнот				
Файл	Правка	Формат	Вид	Справка
142.250.180.238:	число	байт=32	время=635	
142.250.180.238:	число	байт=32	время=596	
142.250.180.238:	число	байт=32	время=456	
142.250.180.238:	число	байт=32	время=245	
142.250.180.238:	число	байт=32	время=356	
142.250.180.238:	число	байт=32	время=869	
142.250.180.238:	число	байт=32	время=456	
142.250.180.238:	число	байт=32	время=368	
142.250.180.238:	число	байт=32	время=153	
142.250.180.238:	число	байт=32	время=765	
142.250.180.238:	число	байт=32	время=453	
142.250.180.238:	число	байт=32	время=646	
142.250.180.238:	число	байт=32	время=345	
142.250.180.238:	число	байт=32	время=112	
142.250.180.238:	число	байт=32	время=333	
142.250.180.238:	число	байт=32	время=600	
142.250.180.238:	число	байт=32	время=601	
142.250.180.238:	число	байт=32	время=620	
142.250.180.238:	число	байт=32	время=599	
142.250.180.238:	число	байт=32	время=200	
142.250.180.238:	число	байт=32	время=456	
142.250.180.238:	число	байт=32	время=956	
142.250.180.238:	число	байт=32	время=200	
142.250.180.238:	число	байт=32	время=24	
142.250.180.238:	число	байт=32	время=104	
142.250.180.238:	число	байт=32	время=200	
142.250.180.238:	число	байт=32	время=327	
142.250.180.238:	число	байт=32	время=400	
142.250.180.238:	число	байт=32	время=450	
142.250.180.238:	число	байт=32	время=400	

Рисунок 4.1.2 – Вхідна інформація до другого експерименту

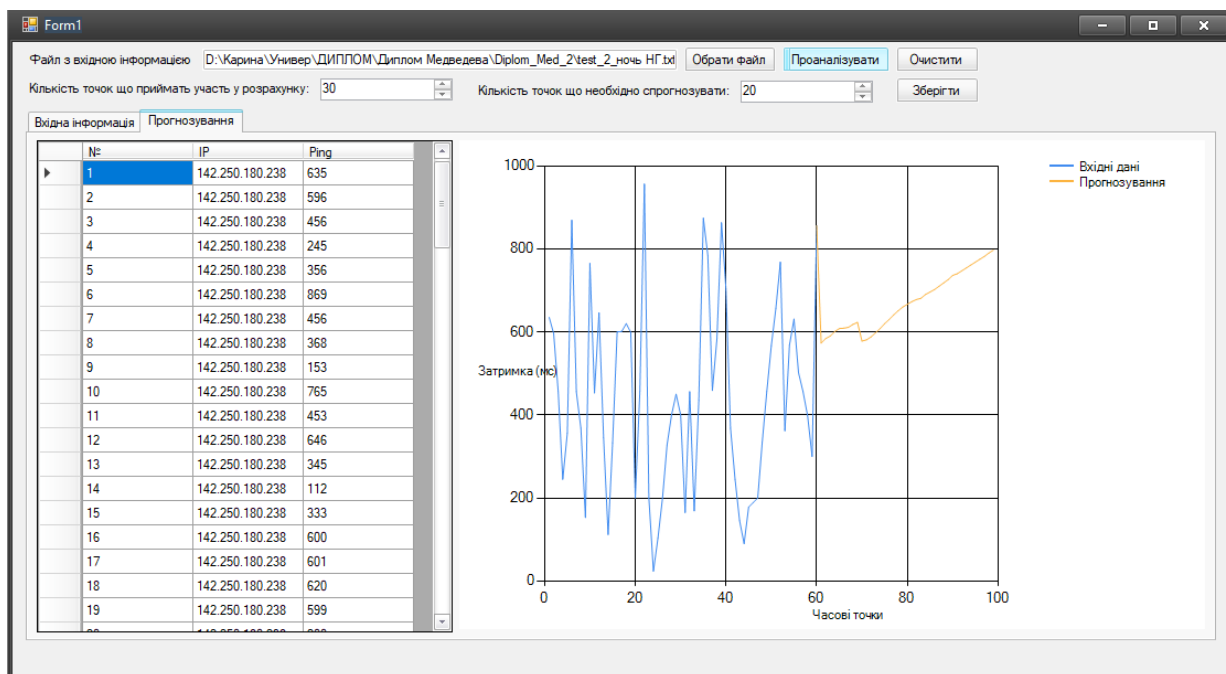


Рисунок 4.1.2 – Результат другого експерименту

Результат експерименту: при розробці графіку можна побачити зростання, падіння тренду, які є коректними. Зростання та падіння зумовлено тим, що у новорічну ніч велике навантаження на мережу через те що всі люди спілкуються по усьому світу.

4.1.3 Експеримент 3

Третій набір вхідної інформації зібраний після новорічної ночі для порівняння з експериментом 2, наведено на рис.4.1.3.

test_3_после НГ.txt – Блокнот

Файл Правка Формат Вид Справка

```

142.250.180.238: число байт=32 время=24
142.250.180.238: число байт=32 время=23
142.250.180.238: число байт=32 время=23
142.250.180.238: число байт=32 время=24
142.250.180.238: число байт=32 время=25
142.250.180.238: число байт=32 время=27
142.250.180.238: число байт=32 время=29
142.250.180.238: число байт=32 время=30
142.250.180.238: число байт=32 время=32
142.250.180.238: число байт=32 время=44
142.250.180.238: число байт=32 время=46
142.250.180.238: число байт=32 время=60
142.250.180.238: число байт=32 время=30
142.250.180.238: число байт=32 время=24
142.250.180.238: число байт=32 время=23
142.250.180.238: число байт=32 время=24
142.250.180.238: число байт=32 время=24
142.250.180.238: число байт=32 время=29
142.250.180.238: число байт=32 время=28
142.250.180.238: число байт=32 время=44
142.250.180.238: число байт=32 время=50
142.250.180.238: число байт=32 время=104
142.250.180.238: число байт=32 время=165
142.250.180.238: число байт=32 время=26
142.250.180.238: число байт=32 время=25
142.250.180.238: число байт=32 время=24
142.250.180.238: число байт=32 время=23
142.250.180.238: число байт=32 время=23
142.250.180.238: число байт=32 время=24
142.250.180.238: число байт=32 время=24

```

Рисунок 4.1.3 – Вхідна інформація до третього експерименту

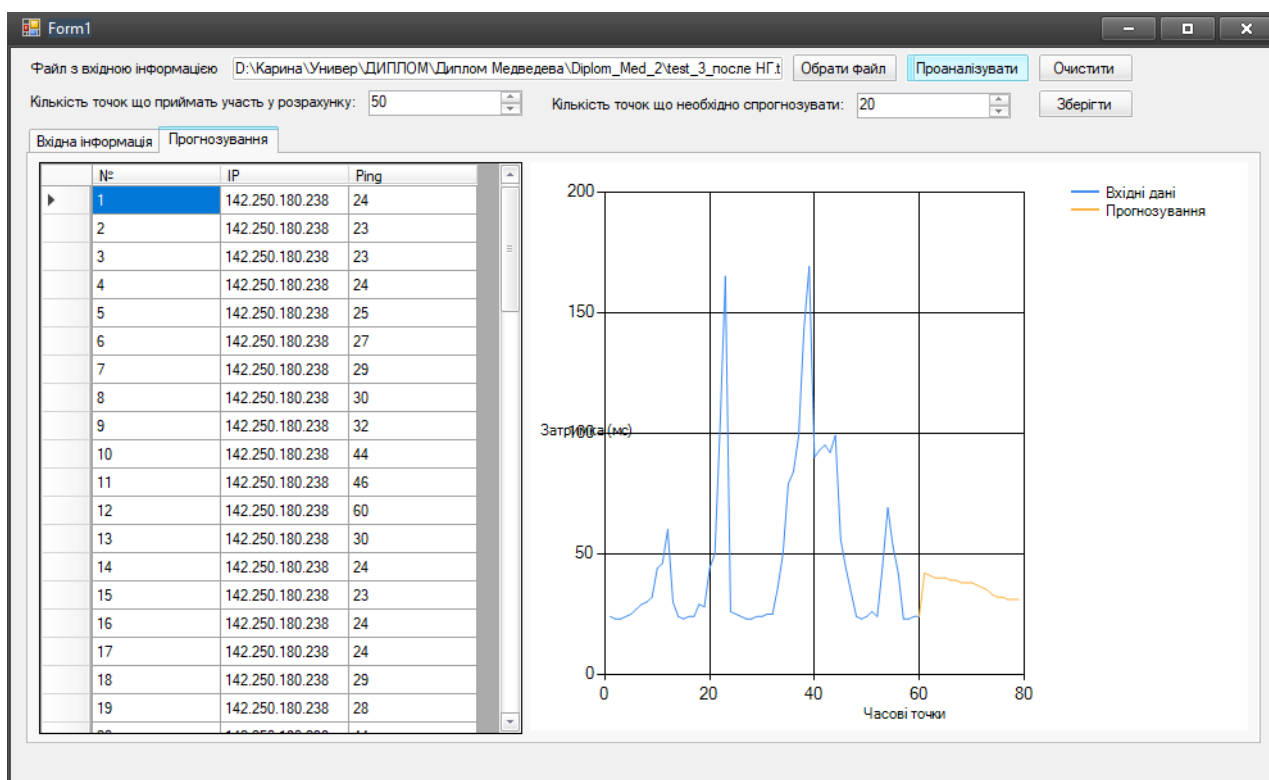


Рисунок 4.1.3 – Результат третього експерименту

Результат експерименту: при розробці графіку можна побачити зростання, падіння тренду, які є коректними. Зростання та падіння зумовлено тим, що у день після свята усі люди відпочивають та святкують.

4.1.4 Експеримент 4

Четвертий набір вхідної інформації, котрий збирався у святковий день, наведено на рис. 4.1.4.

test_4_день незалежності.txt – Блокнот				
Файл	Правка	Формат	Вид	Справка
142.250.180.238:	число	байт=32	время=90	
142.250.180.238:	число	байт=32	время=356	
142.250.180.238:	число	байт=32	время=96	
142.250.180.238:	число	байт=32	время=156	
142.250.180.238:	число	байт=32	время=368	
142.250.180.238:	число	байт=32	время=153	
142.250.180.238:	число	байт=32	время=106	
142.250.180.238:	число	байт=32	время=146	
142.250.180.238:	число	байт=32	время=203	
142.250.180.238:	число	байт=32	время=345	
142.250.180.238:	число	байт=32	время=112	
142.250.180.238:	число	байт=32	время=89	
142.250.180.238:	число	байт=32	время=132	
142.250.180.238:	число	байт=32	время=201	
142.250.180.238:	число	байт=32	время=120	
142.250.180.238:	число	байт=32	время=199	
142.250.180.238:	число	байт=32	время=100	
142.250.180.238:	число	байт=32	время=256	
142.250.180.238:	число	байт=32	время=156	
142.250.180.238:	число	байт=32	время=200	
142.250.180.238:	число	байт=32	время=24	
142.250.180.238:	число	байт=32	время=104	
142.250.180.238:	число	байт=32	время=200	
142.250.180.238:	число	байт=32	время=327	
142.250.180.238:	число	байт=32	время=200	
142.250.180.238:	число	байт=32	время=150	
142.250.180.238:	число	байт=32	время=100	
142.250.180.238:	число	байт=32	время=165	
142.250.180.238:	число	байт=32	время=256	
142.250.180.238:	число	байт=32	время=169	

Рисунок 4.1.4 – Вхідна інформація до четвертого експерименту

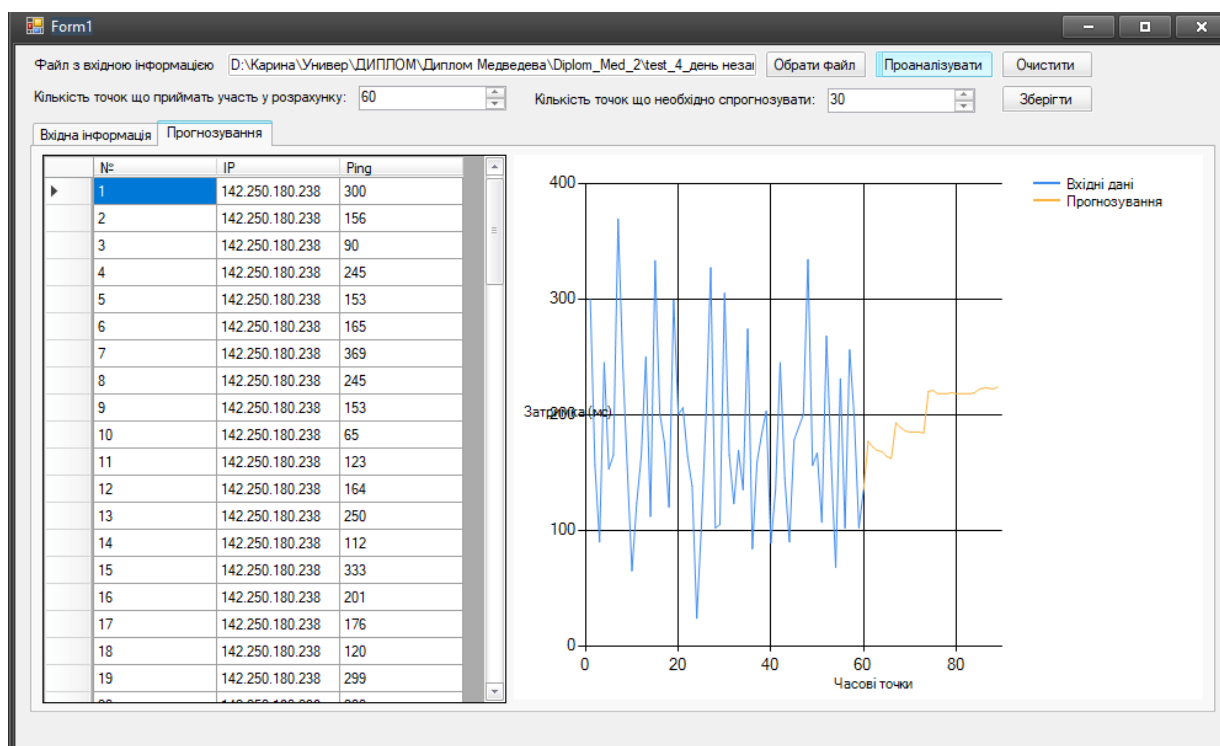


Рисунок 4.1.4 – Результат четвертого експерименту

Результат експерименту: при розробці графіку можна побачити зростання, падіння тренду, які є коректними. Зростання та падіння зумовлено тим, що цей день святковий, але тільки в Україні, тому навантаження було не таке велике як у новорічну ніч.

4.1.5 Експеримент 5

П'ятий набір вхідної інформації збирався на день знань та показано на рис. 4.1.5.

```
test_4_день незалежності.txt – Блокнот
Файл  Правка  Формат  Вид  Справка
142.250.180.238: число байт=32 время=90
142.250.180.238: число байт=32 время=356
142.250.180.238: число байт=32 время=96
142.250.180.238: число байт=32 время=156
142.250.180.238: число байт=32 время=368
142.250.180.238: число байт=32 время=153
142.250.180.238: число байт=32 время=106
142.250.180.238: число байт=32 время=146
142.250.180.238: число байт=32 время=203
142.250.180.238: число байт=32 время=345
142.250.180.238: число байт=32 время=112
142.250.180.238: число байт=32 время=89
142.250.180.238: число байт=32 время=132
142.250.180.238: число байт=32 время=201
142.250.180.238: число байт=32 время=120
142.250.180.238: число байт=32 время=199
142.250.180.238: число байт=32 время=100
```

Рисунок 4.1.5 – Вхідна інформація до п'ятого експерименту

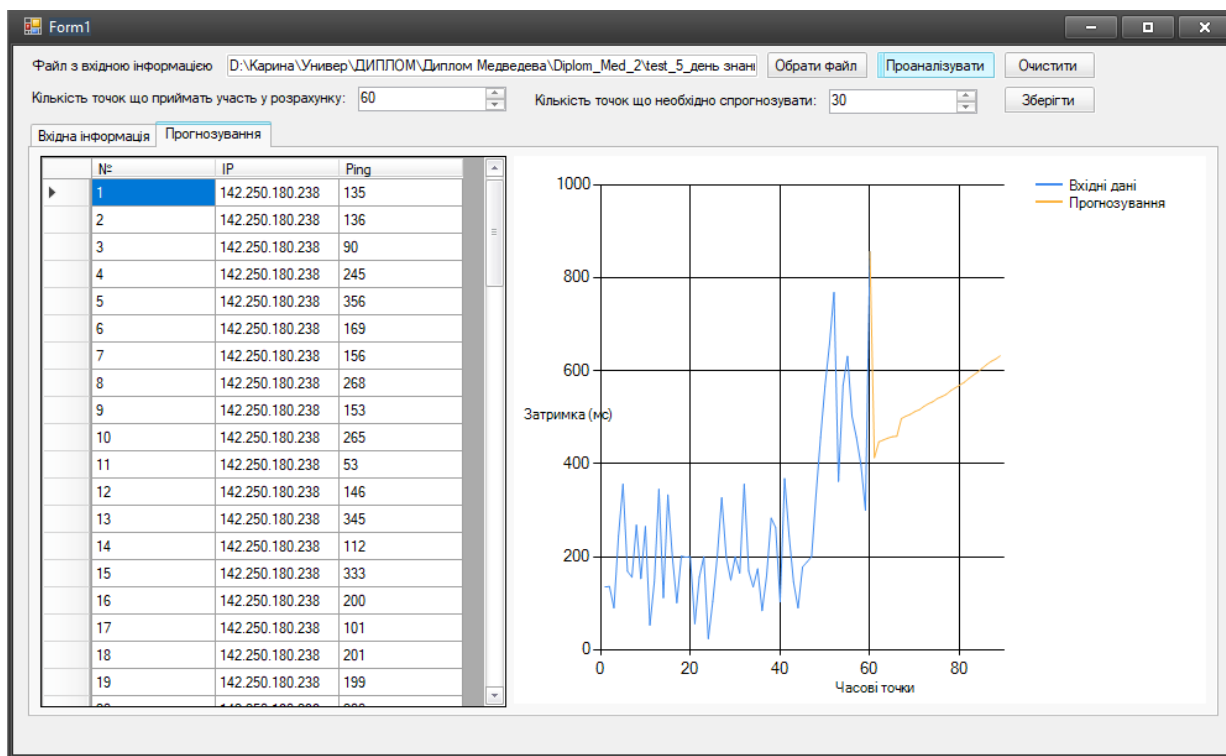


Рисунок 4.1.5 – Результат п'ятого експерименту

Результат експерименту: при розробці графіку можна побачити зростання, падіння тренду, які є коректними. Зростання та падіння зумовлено тим, що день є робочим, ще й частково святковий для певної частини населення світу.

4.2 Проведення експерименту

Проведення експерименту полягає в процесі запуску програмного продукту та проведенні прогнозування на вхідних даних.

Для проведення експерименту необхідно:

1. Запустити програмний продукт.
2. Вибрати необхідний файл, в котрому знаходиться інформація для прогнозування майбутніх значень.
3. Вказати кількість точок, що приймають участь у розрахунку.
4. Вказати кількість точок, що приймають участь у прогнозуванні.
5. Запустити прогнозування.
6. Отримати результати та спрогнозований графік.
7. Зберегти отриману інформацію.

8. Проаналізувати отриману інформацію та зробити висновки.

Висновки до розділу 4

У розділі описано процес дослідження та результати аналізу часових рядів навантаженості мережевої системи.

Для демонстрації роботи зібрано дані у святкові дні та день після свят, тому що саме у ці дні набагато краще видно різницю зміну навантаженості мережевої системи, що дуже підходить для проведення експерименту.

Експерименти показали, що метод прогнозування часових рядів навантаженості мережевої системи має право на існування та їм можна користуватися для прогнозування зростання та падіння навантаженості.

5 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

5.1 Вимоги безпеки праці під час виконання робіт

Охорона праці – це система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних та лікувально-профілактичних заходів і засобів, спрямованих на збереження здоров'я та працездатності людини в процесі праці.

Відповідно до статті 18 Закону України "Про охорону праці", працівник зобов'язаний «знати і виконувати вимоги нормативних актів про охорону праці, правила поводження з машинами, механізмами, устаткуванням та іншими засобами виробництва, користуватися засобами колективного та індивідуального захисту, проходити у встановленому порядку попередні та періодичні медичні огляди» [12].

5.1.1 Вимоги під час експлуатації ПК

Впровадження ЕОМ у різні галузі виробництва позитивно вплинуло на умови праці, її якість та продуктивність. Разом з тим робота за монітором, електронною обчислювальною машиною(ЕОМ), ноутбуком та персональним комп'ютером (ПК) має низку чинників, які у разі порушення правил експлуатації можуть негативно впливати на стан здоров'я користувачів.

Основними проявами їх впливу є зоровий дискомфорт. Він проявляється як біль в очах, почервоніння повік і очних яблук, відчуття піску в очах, головний біль, швидка втома. Кістково-м'язовий дискомфорт проявляється у вигляді болю різної сили у суглобах та м'язах, оніміння тощо. Частіше кістково-м'язовий дискомфорт проявляється у людей старшої вікової категорії. Також монітори та ЕОМ зумовлюють захворювання шкіри. Вони проявляються у вигляді висипу, лущення, деяких видів дерматитів. Стресові чинники, до яких належать несприятливі умови та режим праці, зміст праці, здібності працівника, звички, умови життя, можуть стати причиною

виникнення фізіологічних і психологічних змін, погіршення здоров'я та змін у поведінці людини.

Користувачі повинні знати потенційно шкідливі та небезпечні виробничі чинники під час роботи за монітором, ПК, ноутбуком та виконувати вимоги НПАОП 0.00-1.28-10 «Правила охорони праці під час експлуатації електронно-обчислювальних машин» і ДСанПіН 3.3.2.007-98 «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами ЕОМ» та інших нормативних документів щодо виключення або зменшення їх негативного впливу [14].

5.1.2 Аналіз шкідливих та небезпечних виробничих факторів

Аналіз шкідливих та небезпечних виробничих факторів проводять згідно з ГОСТ 12.1.003-74. Небезпечними та шкідливими називають фактори, що призводять до раптового погіршення здоров'я людини, або навіть смерті, під час її трудової діяльності.

Згідно з класифікацією ГОСТ 12.1.003-74 до числа небезпечних та шкідливих факторів, що виникають при роботі з ПК слід віднести:

- нервово-психічні перевантаження (розумове перевантаження, монотонність праці, емоційні перевантаження);
- фізичне перевантаження;
- невідповідність параметрів мікроклімату робочої зони санітарним нормам;
- недостатня або надмірна освітленість робочої зони;
- несприятливе забарвлення стін та підлоги, віддзеркалення;
- робота з комп'ютером [13].

Робоче місце призначено для оператора, що працює з програмною системою обліку студентів коледжу, які навчаються за контрактом. Кількість робочих місць – 1. Робоче місце оснащено ПК, принтером, монітором та іншими периферійними пристроями. Розміщення принтера або інших пристроїв введення-виведення інформації на робочому місці має

забезпечувати добру видимість екрану, зручність ручного керування пристроєм введення-виведення інформації в зоні досяжності моторного поля: по висоті від 900 до 1300 мм, по глибині від 400 до 500 мм. Під матричні принтери потрібно підкладати вібраційні килимки для гасіння вібрації та шуму.

Площа, виділена для одного робочого місця ПК повинна складати не менше 6 м², а об'єм – не менше 20 м³. Робочі місця з моніторами відносно світлових прорізів повинні розміщуватися так, щоб природне світло падало збоку, переважно зліва.

Режим праці та відпочинку.

При організації праці, пов'язаної з використанням ПК, для збереження здоров'я працівника передбачаються:

- перерви для відпочинку і вживання їжі;
- перерви для відпочинку й особистих потреб;
- додаткові перерви.

5.1.3 Вимоги до приміщень

Об'ємно-планувальні рішення будівель і приміщень для роботи з моніторами, ПК, ноутбуками мають відповідати вимогам чинних нормативних актів, зокрема СНиП 2.09.04-87 «Административные и бытовые здания», ДСанПіН 3.3.2.007-98, НПАОП 0.00-1.28 10 та іншим і мати ступінь вогнестійкості не нижчу II.

Заборонено розміщувати робочі місця з моніторами, ПК та ноутбуками у підвальних приміщеннях, на цокольних поверхах, поряд з приміщеннями, в яких рівні шуму та вібрації перевищують допустимі значення (поряд з механічними цехами, майстернями тощо), з мокрими виробництвами, з вибухопожежонебезпечними приміщеннями категорій А і Б, а також над такими приміщеннями або під ними [12].

Площу приміщень визначають із розрахунку, що на одне робоче місце вона має становити не менше ніж 6 м^2 , а об'єм не менше ніж 20 м^3 з урахуванням максимальної кількості осіб, які одночасно працюють у зміні.

Приміщення мають бути оснащені природним і штучним освітленням відповідно до ДБН В.2.5-28-2006. Приміщення мають бути обладнані системами водяного опалення, кондиціонування або припливно-витяжною вентиляцією відповідно до ДБН 8.2.5-67-2013. Заземлені конструкції приміщення (батареї опалення, водопровідні труби, кабелі з заземленим відкритим екраном тощо) надійно захищені діелектричними щитками або сітками від випадкового дотику [12].

Для всіх споруд і приміщень, у яких експлуатують монітори, ПК і ноутбуки визначають категорію з вибухопожежної та пожежної безпеки відповідно до НАПБ Б.03.002-2007 «Определение категории помещений и зданий по взрывопожарной и пожарной опасности» і НАПБ В.01.053-2000/520 «Правила пожежної безпеки в галузі зв'язку» та клас вибухонебезпечних зон відповідно до НПАОП 0.00-1.32-01 «Правила будови електроустановок. Електрообладнання спеціальних установок». Відповідні позначення наносять на входні двері приміщення. Крім того, приміщення з моніторами, ПК і ноутбуками оснащують системою пожежної сигналізації з димовими пожежними сповіщувачами та переносними вуглекислотними вогнегасниками з розрахунку 2 од. на кожні 20 м^2 площі приміщення. Підходи до засобів пожежогасіння мають бути вільними.

Робочі місця з моніторами, щодо світлових прорізів розміщують так, щоб природне світло падало збоку, переважно зліва.

Екран монітора і клавіатура мають розміщуватися на оптимальній відстані від очей користувача, але не ближче 600 мм з урахуванням розміру алфавітно-цифрових знаків і символів. Відстань від екрана до ока працівника залежить від діагоналі екрана та наведена у табл. 5.1.

Таблиця 5.1 – Відстань від екрана до ока працівника

Діагональ екрана	Відстань від екрана
35/38 см (14"/15")	від 600 до 700 мм
43 см (17")	від 700 до 800 мм
48 см (19")	від 800 до 900 мм
53 см (21")	від 900 до 1000 мм

Розміщення екрана монітору має забезпечувати зручність зорового спостереження у вертикальній площині під кутом $\pm 30^\circ$ від лінії зору працівника.

Обладнання й організація робочих місць з моніторами, ПК і ноутбуками мають забезпечувати відповідність усіх елементів робочого місця та їх розміщення ергономічним вимогам ГОСТ 12.2.032 78 «ССБТ. Рабочее место при выполнении работ сидя. Общие эргономические требования», характеру й особливостям діяльності. Конструкція робочого місця має забезпечити підтримання оптимальної робочої пози й оптимальне розміщення на робочій поверхні використовуваного обладнання (екрану, клавіатури, принтера) та документів [14].

Робочий стіл, як правило, обладнують підставкою для ніг ширшою не менше 300 мм, глибиною не менше 400 мм з можливістю регулювання по висоті в границях 150 мм та кутом нахилу опорної поверхні в границях 20° . Підставка має бути з рифленою поверхнею і бортиком по передньому краю заввишки 10 мм.

Робоче сидіння (стілець, крісло) має складатися з таких основних елементів: сидіння, спинки, стаціонарних або змінних підлокітників. Стілець має бути підйомно-поворотним, таким, що регулюється за висотою, кутом нахилу сидіння і спинки, за відстанню спинки до переднього краю сидіння, висотою підлокітників. Крок регулювання для лінійних розмірів — від 15 до 20 мм, для кутових – від 2° до 5° . Зусилля регулювання – не більше 20 Н.

Клавіатуру слід розташовувати на поверхні столу або на спеціальній регульованій за висотою робочій поверхні окремій віл столу на відстані від 100 до 300 мм від краю, ближчого до працівника. Кут нахилу клавіатури має бути у границях від 5 до 15°.

Розміщення принтера або іншого пристрою вводу-виводу інформації на робочому місці має забезпечувати гарну видимість екрана монітора і зручність ручного керування в зоні досяжності моторного поля.

Робочі місця слід оснащувати пюпітром (тримачем) для документів, що легко перемішувати. Його встановлюють вертикально (або з нахилом) на тому ж рівні та відстані від очей користувачів, що і монітор. Схема вимог до робочого місця показано на рисунку 5.1.

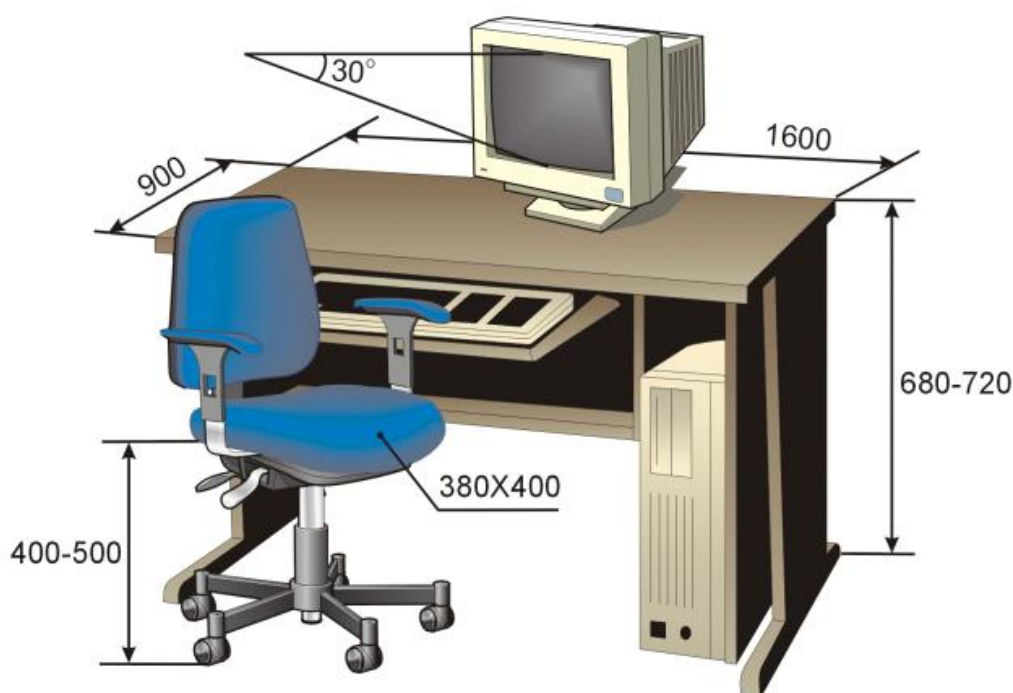


Рисунок 5.1 – Схема вимог до робочого місця

5.1.4 Вимоги до виробничого середовища приміщень з моніторами, ПК і ноутбуками

Згідно з Гігієнічною класифікацією праці за показниками шкідливості та небезпечності чинників виробничого середовища, тяжкості та напруженості трудового процесу умови праці користувачів ПК і ноутбуків

мають відповідати I класу (оптимальним) або II класу (допустимим) умовам праці.

Приміщення з моніторами, ПК і ноутбуками мають бути забезпечені природним і штучним освітленням. Коефіцієнт природного освітлення (КПО) має бути не нижчим 1,5%. Розраховують площу світлових прорізів, яка забезпечує нормоване значення КПО в робочій зоні користувачів комп'ютерів, відповідно до ДБН В.2.5-28-2006.

Штучне освітлення має бути загальним, робочим і рівномірним. У випадку, коли робота переважно з документами, допускається додатково використовувати місцеве освітлення. Рівень освітленості в зоні розташування документів має бути в границях від 300 до 500 лк.

Систему загального освітлення має бути виконано у вигляді суцільних або переривчатих ліній світильників, що розмішують збоку від виробничих місць (переважно зліва), паралельно лінії зору працівників. Допускається застосовувати світильники таких класів світлорозподілу: світильники прямого світла – П; переважно прямого світла – Н; переважно відбитого світла – В.

У разі розташування моніторів, ПК по периметру приміщення лінії світильників мають бути розміщені локально над робочими місцями [14].

Уміст шкідливих речовин у повітрі робочої зони не має перевищувати ГДК відповідно до ГОСТ 12.1.005-88 уміст озону не більше 0,1 мг/м³, вміст оксидів азоту – не більше 5 мг/м³. уміст пилу – не більше 4 мг/м³. Оптимальні величини температури, відносної вологості та швидкості руху повітря для приміщень з моніторами, ПК і ноутбуками наведені у табл. 5.2.

Рівні іонізації повітря приміщень під час роботи з моніторами, ПК і ноутбуками наведено у табл. 5.3.

Допустимі рівні звуку, еквівалентні рівні звуку, рівні звукового тиску в октавних смугах частот представлений в табл. 5.4.

Таблиця 5.2 – Оптимальні величини температури, відносної вологості та швидкості руху повітря для приміщень з моніторами, ПК і ноутбуками

Період року	Категорія робіт	Температура повітря, °С	Відносна вологість повітря, %	Швидкість руху повітря, м/с
холодний	Легка-1 а	від 22 до 24	від 40 до 60	0,1
	Легка-1б	від 21 до 23	від 40 до 60	0,1
теплий	Легка-1а	від 23 до 25	від 40 до 60	0,1
	Легка-1б	від 22 до 24	від 40 до 60	0,2

Таблиця 5.3 – Рівні іонізації повітря приміщень під час роботи з моніторами, ПК і ноутбуками

Рівні іонізації повітря	Кількість іонів у 1 см ³ повітря	
	n ⁺	n ⁻
Мінімально необхідні	400	600
Оптимальні	від 1500 до 3000	від 3000 до 5000
Максимально допустимі	50000	50000

Для підтримання оптимальних значень параметрів повітря робочої зони потрібно застосовувати вентиляцію приміщень, кондиціонування повітря, використовувати установки або прилади зволоження та штучної іонізації.

У приміщеннях з ПК рівні звукового тиску, рівні звуку та еквівалентні рівні звуку мають відповідати вимогам ДСН 3.3.6.037-99 та ДСанПіН 3.3.2.007 98. Устаткування, яке є джерелом шуму, слід розташовувати поза приміщеннями з ПК і ноутбуками.

Значення напруженості електромагнітних полів на робочих місцях із моніторами мають відповідати нормативним значенням ГОСТ 12.1.006-84 і ДСанПіН 3.3.2.007-98 [9].

Таблиця 5.4 – Допустимі рівні звуку, еквівалентні рівні звуку, рівні звукового тиску в октавних смугах частот

Види трудової діяльності	Рівні звукового тиску, дБ, в октавних смугах із середньо геометричними частотами, Гц									Рівні звуку, еквівалентні рівні звуку, дБА/дБАекв
	31.5	63	125	250	500	1000	2000	000	000	
Програмісти ПК	86	71	61	54	49	45	42	40	38	50
Оператори в залах оброблення інформації на ПК та оператори комп'ютерного набору	96	83	74	68	63	60	57	55	54	65

Допустимі рівні електромагнітних випромінювань радіочастотного діапазону наведено у таблиці 5.5.

Таблиця 5.5 – Допустимі рівні електромагнітних випромінювань радіочастотного діапазону

Діапазон частот, МГц	Допустимі рівні ЕМП	
	Електрична складова E , В/м	Магнітна складова H , А/м
від 0.06 до 3,0	50	5
від 3.0 до 30,0	20	-
від 30.0 до 50.0	10	0.3
від 50,0 до 300,0	5	-

2. Дії працівників в аварійних ситуаціях

Інструкція визначає порядок дій керівництва, працівників підприємств у випадку виникнення надзвичайної або аварійної ситуації (аварії) і є обов'язковою для суворого виконання.

У питаннях, не врегульованих нею, слід керуватися Кодексом цивільного захисту України, законодавчими та нормативно-правовими актами у сфері цивільного захисту, вимогами інструкцій (правил) з охорони праці.

Аварією слід вважати небезпечну подію техногенного характеру, що спричинила ураження, травмування населення або створює на окремій території чи території суб'єкта господарювання загрозу життю або здоров'ю населення та призводить до руйнування будівель, споруд, обладнання і транспортних засобів, порушення виробничого або транспортного процесу чи спричиняє наднормативні, аварійні викиди забруднюючих речовин та інший шкідливий вплив на навколишнє природне середовище.

Аварійною ситуацією являється порушення меж та/або умов безпечної експлуатації об'єкта (устаткування, обладнання), яке не перейшло в аварію, за якого всі несприятливі впливи джерел небезпеки на працівників та навколишнє середовище утримуються в прийнятних межах за допомогою відповідних технічних засобів.

Джерелами небезпечних і шкідливих виробничих чинників можуть бути:

- нерегламентовані режими роботи технологічного устаткування, транспортні засоби, вантажопідіймальне устаткування, механізми, обладнання, деталі та вироби, що рухаються;
- устаткування, що працює під тиском;
- електромережі, електрифіковане устаткування та інструменти;
- інженерні комунікації;
- роботи, що призводять до психофізіологічних перевантажень;

- токсичні речовини;
- помилкові дії працівників;
- аварії.

У разі виникнення надзвичайної або аварійної ситуації чи аварії працівники зобов'язані діяти спокійно, не панікувати, точно й оперативно слідувати вказівкам керівництва, осіб, відповідальних за стан цивільного захисту (цивільної оборони) та техногенної безпеки, за протипожежну безпеку, охорону праці, а також представників аварійно-рятувальних, пожежних, медичних підрозділів.

Працівник, який першим виявив загрозу виникнення аварійної ситуації, повинен:

- негайно припинити роботу та подати команду "СТОП!". Команда "СТОП!" подана будь-яким працівником, має негайно бути виконаною всіма працівниками, котрі її почули;
- якнайшвидше сповістити про надзвичайну або аварійну ситуацію особу, відповідальну за стан цивільного захисту;
- викликати аварійно-рятувальні служби (пожежну, медичну тощо).

У випадку виникнення надзвичайної або аварійної ситуацій кожний працівник повинен:

- припинити роботу;
- приступити до ліквідації (локалізації) аварії наявними засобами;
- за необхідності долучитись до виклику інших аварійно-рятувальних служб (пожежну, медичну тощо).

Керівники, у разі виникнення надзвичайної або аварійної ситуації (аварії), зобов'язані:

- оцінити умови ситуації, з'ясувати кількість і місцезнаходження людей, захоплених аварією, за потреби вжити заходів щодо оповіщення працівників про аварію;

- у разі наявності загрози життю людей негайно організувати їх евакуацію (рятування), використовуючи для цього наявні сили й засоби;
- вжити негайних заходів щодо локалізації та ліквідації аварії;
- вжити заходів щодо оточення району аварії (небезпечної зони);
- забезпечити виведення з небезпечної зони працівників, які не беруть безпосередньої участі в ліквідації аварії;
- контролювати правильність дій працівників щодо локалізації і ліквідації аварії;
- у разі необхідності виконати відключення електроенергії та вжити інших заходів, що сприяють ліквідації (локалізації) аварії (пожежі);
- сприяти діям аварійно-рятувальних, пожежних, медичних підрозділів щодо рятування людей, локалізації, ліквідації аварії у приміщенні;
- організувати надання екстреної медичної допомоги потерпілим під час ліквідації аварії;
- інформувати безпосереднє керівництво про хід і характер аварії, кількість потерпілих внаслідок виконання рятувальних робіт.

Особливості дій працівників при деяких надзвичайних ситуаціях:

- при загрозі хімічного ураження оповіщаються всі працівники та відвідувачі, які знаходяться в приміщеннях;
- вентиляційні установки та кондиціонери терміново виключаються, закриваються вікна, двері, квартирки, приміщення герметизуються. Вихід із будівлі й вхід до неї припиняється до особливого розпорядження керівництва;
- при виявленні у приміщенні, де укриваються працівники, хімічно небезпечної речовини, працівники повинні вийти з приміщення та з дозволу керівництва залишити зону забруднення. Виходити із зони необхідно тільки у засобах індивідуального захисту та рухатися в напрямку, перпендикулярному напрямку вітру;

- при виникненні пожежі в приміщеннях всі працівники зобов'язані діяти згідно з Інструкцією з пожежної безпеки для працівників, евакуацію проводити згідно з Планом евакуації;

- при радіоактивному забрудненні або при загрозі забруднення всі працівники повинні уважно слідкувати за мовним повідомленням управління з питань надзвичайних ситуацій, яке передається по радіо та телебаченню, за інформацією інших засобів масової інформації про обстановку в місті та суворо виконувати рекомендації із захисту від радіоактивного зараження;

- при загрозі або виникненні катастрофічних стихійних лих працівники за розпорядженням керівництва повинні зупинити роботу, виконати необхідні протипожежні заходи, відключити від електромережі електрообладнання, підготуватися до евакуації та вивезення до безпечного місця найбільш цінних матеріальних засобів;

- якщо з'явилися потерпілі, їм надається перша медична допомога або вживаються заходи щодо їх госпіталізації до медичних закладів;

- при надходженні анонімної інформації про загрозу в приміщеннях або на території поблизу них терористичного акту працівник, який прийняв її, повинен терміново доповісти керівництву та повідомити правоохоронні органи і діяти згідно з розпорядженнями та рекомендаціями.

У разі нещасного випадку:

- працівник, який його виявив, або сам потерпілий повинен терміново повідомити безпосереднього керівника та/або особу, відповідальну за стан цивільного захисту (цивільної оборони) та техногенної безпеки про випадок, масштаби загрози та кількість потерпілих;

- особа, відповідальна за стан цивільного захисту (цивільної оборони) та техногенної безпеки, працівники терміново організовують надання екстреної медичної допомоги потерпілим, а за необхідності - супроводження до лікувального закладу.

– працівники повинні зберегти до прибуття комісії з розслідування обстановку на робочому місці та обладнання в такому стані, у якому вони були на момент події (якщо це не загрожує життю й здоров'ю інших працівників і не призведе до більш тяжких наслідків).

– працівники НСПП вживають заходів щодо ліквідації ситуації, що склалася, та недопущення в подальшому подібних випадків.

У випадку припинення подачі електроенергії працівник повинен від'єднати електричні прилади від мережі.

Виявивши обрив електропроводів, пошкодження їхньої ізоляції, не варто торкатися їх. Необхідно повідомити про це безпосереднього керівника та/або осіб, відповідальних за протипожежну безпеку, охорону праці, стан цивільного захисту (цивільної оборони) та техногенної безпеки.

Посадові особи, на яких чинними нормативно-правовими актами покладаються обов'язки щодо локалізації (ліквідації) аварійної ситуації (аварії), несуть відповідальність згідно із законодавством України.

Особи, винні в порушенні цієї Інструкції, несуть відповідальність згідно із законодавством України [15].

Висновки до розділу 5

У цьому розділі було приведено та проаналізовано тема охорони праці та типові інструкції при виникненні надзвичайних ситуацій.

Описані умови праці під час роботи у приміщенні, вимоги під час експлуатації ПК, проведено аналіз шкідливих та небезпечних виробничих факторів, вимоги до виробничого середовища приміщень з моніторами, ПК і ноутбуками. Приведено типові інструкції дій працівників при аварійних ситуаціях.

ЗАГАЛЬНІ ВИСНОВКИ ЗА РОБОТОЮ

Дослідження роботи заключається в розробці методів моніторингу та прогнозування перенавантажень в мережевих системах. Розроблений додаток може бути використаний у мережах підприємств з ціллю підвищення рівня безпеки передачі даних.

У першому розділі проведено аналіз аналогів який показав, що жодна з програм не має чіткої відповіді на питання завантаженості мереж в залежності від часу її використання та не має функції прогнозування. Встановлено, що розробка механізму прогнозування навантаженості мережевих систем є актуальною, важливою задачею та частиною мір із забезпечення безпечної експлуатації комп'ютерної мережі.

У другому розділі було розглянуто методи прогнозування часових рядів, також було розглянуто методи оцінки точності прогнозу.

Третій розділ присвячений виконанню проектування інструментального засобу для дослідження часових рядів навантаження мережевої системи. В процесі проектування системи були використанні принципи ООП. Розроблено діаграми взаємодії частин програмного продукту з участю MVC-моделі, також була розроблена діаграма класів. Виконано тестування програмного продукту за допомогою методів покриття операторів, покриття переходів, припущення про помилку та Unit-тестів.

У четвертому розділі були проведені експерименти, які довели коректність роботи програмного продукту. Для демонстрації роботи зібрано дані у святкові дні та день після свят. Експерименти показали, що метод прогнозування часових рядів навантаженості мережевої системи має право на існування та їм можна користуватися для прогнозування зростання та падіння навантаженості.

Практична цінність програмного продукту полягає в розробці методу прогнозування навантаження. Продукт може бути використаний у локальних мережах підприємств з ціллю покращення рівня безпеки передачі даних.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Нагорних Д. Ю. 1. «Прогнозування часових рядів за допомогою гібридної нейронної мережі» / Д. Ю. Нагорних. 2012.
2. Кондратьєва Т. М. Прогнозування тенденції фінансових часових рядів за допомогою нейронної мережі LSTM», / Кондратьєва Т. М. – ФГБОУ ВО «Донський державний технічний університет.
3. Прогнозирование временных рядов с помощью гибридной нейронной сети. [Електронний ресурс] / URL: http://www.ict.nsc.ru/jspui/bitstream/ICT/4694/25/DICR-2019-V3_p135-138.pdf. (дата звернення 15.09.2021)
4. Бенгус Б. В. Прогнозирование тенденции временного ряда с помощью искусственной нейронной сети / Б. В. Бенгус // Обзорение прикладной и промышленной математики. Том 21 / Б. В. Бенгус., 2014.
5. Кириченко Л. О. 5. «Алгоритм попередження перенавантаження комп'ютерної мережі шляхом прогнозування середньої довжини черги» / Кириченко Л. О. УДК, 2015.
6. Дослідження мережі TCP/IP з використанням основних алгоритмів активного управління чергою / Гостев В. И. УДК, 2016
7. Троелсен Э. Язык программирования C# 5.0 и платформа .NET 4.5 / Эндрю Троелсен., 2012. – 756 с.
8. Нейгел К. C# 5.0 и платформа .NET 4.5 для профессионалов. / К. Нейгел, Б. Ивсен.. – 1440 с.
9. MVC — модель-представление-контроллер. [Електронний ресурс] / URL: <https://web-creator.ru/articles/mvc>. (дата звернення 15.10.2021)
10. Microsoft Visual Studio. [Електронний ресурс] / URL: <https://open-file.ru/programs/microsoft-visual-studio>. (дата звернення 15.10.2021)
11. Тестирование программного обеспечения - основные понятия и определения. [Електронний ресурс] / URL: <http://www.protesting.ru/testing/>. (дата звернення 13.11.2021)

12. Охорона праці. [Електронний ресурс] / URL: https://uk.wikipedia.org/wiki/Охорона_праці.
13. Природне і штучне освітлення. ДБН В.2.5-28-2006. — ДБН В.2.5-28-2006. — [Чинний від 2006—05—15]. — К. : Державні будівельні норми України, 2006.
14. Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин. ДСанПІН 3.3.2.007-98. — ДСанПІН 3.3.2.007-98. — [Чинний від 1998—12—10]. — К. : Міністерство Охорони Здоров'я України, 2004.
15. Інструкція щодо дій працівників Національної служби посередництва і примирення при загрозі/виникненні надзвичайних або аварійних ситуацій (аварій). [Електронний ресурс] / URL: <https://nspp.gov.ua/normativno-pravova-baza-nspp/instruktsiji/5006-instruktsiia-shchodo-dii-pratsivnykiv-natsionalnoi-sluzhby-poserednytstva-i-prymyrennia-pry-zahrozi-vynyknenni-nadzvychainykh-abo-avariinykh-sytuatsii-avarii>.
16. Ціна на електроенергію. [Електронний ресурс] / URL: http://mpe.kmu.gov.ua/minugol/control/publish/article?art_id=245583544.
17. Метод прогнозирования перегрузок в компьютерных сетях на основе анализа временных рядов: Автореферат / А.С. Соколов. 2012.
18. Руководство по схемам компьютерных сетей. [Електронний ресурс] / URL: <https://www.lucidchart.com/pages/ru/>. (дата звернення 13.09.2021)
19. Перегрузка сети. [Електронний ресурс] / URL: <https://www.google-info.org/7315853/1/peregruzka-seti.html>. (дата звернення 13.09.2021)
20. Диагностика локальных сетей и Интернет. [Електронний ресурс] / URL: http://book.itер.ru/5/dia_5.htm. (дата звернення 10.09.2021)
21. Сетевая составляющая. [Електронний ресурс] / URL: <http://ru.knowledgr.com/00275196/ПерегрузкаСети>. (дата звернення 13.10.2021)
22. Программ для анализа сетевого трафика. [Електронний ресурс] / URL: <https://networkguru.ru/8-luchshikh-programm-dlia-analiza-setevogo->

trafika/. (дата звернення 20.09.2021)

23. Подбельский В.В. Язык С#. Базовый курс. 2-е изд. – БХВ-Петербург, 2010. – 408 с.
24. Котов, М. Язык С#. Краткое описание и введение в технологии программирования / М. Котов – КУДИЦ-Образ, 2014. – 208 с.
25. Албахарі, Дж. С# 5.0. Справочник. Полное описание языка / Дж. Албахарі – O'Reilly, 2014. – 1008 с.
26. Гуннерсон, Э. / Введение в С#. / Гуннерсон, Э. – СПб.: Питер, 2001.
27. Івченко, Ю.М. Основи стандартизації програмних систем [Текст]: методичні вказівки до дипломного проектування та лабораторних робіт / уклад.: Ю. М. Івченко, В. І. Шинкаренко, В. Г. Івченко; Дніпропетр. нац. ун-т залізн. трансп. ім. акад. В. Лазаряна. – Д.: Вид-во Дніпропетр. нац. ун-ту залізн. трансп. ім. акад. В. Лазаряна, 2009. – 38 с.

ДОДАТКИ

МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ

ДОДАТОК А

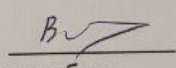
ЗАТВЕРДЖУЮ


Проректор Дніпровського
національного університету
залізничного транспорту
імені академіка В. Лазаряна
Борис БОДНАР

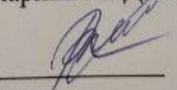
РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ «ПРОГНОЗУВАННЯ
ЧАСОВИХ РЯДІВ НАВАНТАЖЕННЯ МЕРЕЖЕВОЇ СИСТЕМИ»

Технічне завдання
ЛИСТ ЗАТВЕРДЖЕННЯ
1116130.01204-01 ЛЗ

Завідувача кафедри КІТ
Вадим ГОРЯЧКІН


Керівник розробки
Олександра ГОРБОВА


Виконавець
Карина МЕДВЕДСЬВА


Нормоконтролер
Олена КУРОП'ЯТНИК

АНОТАЦІЯ

Документ 1116130.01204-01 «Дослідження часових рядів навантаженості мережевих систем. Технічне завдання» є керівництвом для проектування та розробки програмного додатку «Дослідження часових рядів навантаженості мережевих систем», та входить до складу документації на проект.

Розробка механізму прогнозування навантаженості мережевих систем є актуальною, важливою задачею та частиною мір по забезпеченню безпечної експлуатації комп'ютерної мережі. Прогнозування інтенсивності трафіку дає можливість заздалегідь вживати необхідних заходів попередження наслідків. Досліджуються причини виникнення перевантажень в мережі, наслідком яких є зниження рівня безпеки передачі даних.

В документі містяться основні вимоги до розробки програмного додатку та його функціональні можливості.

1 Підстава для розробки.....	6
2 Призначення розробки	7
3 Вимоги до програми.....	8
3.1. Вимоги до функціональних характеристик	8
3.2. Вимоги до надійності.....	9
3.3. Умови експлуатації	9
3.4. Вимоги до складу та параметрів технічних засобів	9
3.5. Вимоги до інформаційної і програмної сумісності	10
3.6. Вимоги до маркування та упаковки	10
3.7. Вимоги по транспортуванню та зберіганню	10
4. Вимоги до програмної документації.....	12
5. Техніко–економічне обґрунтування проекту розробки програмного продукту	13
6. Стадії та етапи розробки	21
7. Порядок контролю та прийому	23
Бібліографічний список	24

ВСТУП

Розробка програмного продукту «Дослідження часових рядів навантаженості мережевих систем» підвищить ефективність відстеження навантаженості мережі у певних станах.

Основні терміни:

- IP – адреса вузла;
- Ping – час, за який пакет проходить з одного комп'ютеру до іншого та вертається назад.

У мережі передачі даних і теорії організації черг, відбувається перевантаження мережі, коли зв'язок або вузол несуть так багато даних, що його якість обслуговування погіршується. Типові ефекти включають затримку організації черг, втрату пакета або блокування нових зв'язків. Наслідок останніх двох ефектів - те, що зростаюче збільшення запропонованого навантаження призводить або тільки до маленького збільшення мережевої пропускної здатності, або до її фактичного скорочення [19].

Перевантаження в комп'ютерній мережі і теорія масового обслуговування - це зниження якості обслуговування, що відбувається, коли мережевий вузол або лінія зв'язку несе більше інформації, ніж можна передати. Типові проблеми, що можуть виникнути у мережі включають затримку в черзі, втрати пакетів або блокування нових з'єднань. Наслідком перевантаження є поступове збільшення, невеликого зростання або зниження пропускної здатності мережі. Мережеві протоколи, які використовують повторні передачі, щоб компенсувати втрати пакетів через перевантаження, можна збільшити після початкового завантаження, або скоротити до рівня, який зазвичай не викликає перевантаження мережі [19].

Розробка механізму прогнозування навантаженості мережевих систем є актуальною, важливою задачею та частиною заходів по забезпеченню експлуатації комп'ютерної мережі.

Розроблений додаток може бути використаний у мережах підприємств з ціллю підвищення рівня безпеки передачі даних або на серверах з великою кількістю запитів.

1 ПІДСТАВА ДЛЯ РОЗРОБКИ

Основою для розробки є наказ № 690 ст. від 18.11.2020 ректора Дніпровського національного університету залізничного транспорту імені академіка В. Лазаряна «Про затвердження керівників та затвердження тем магістерських робіт».

Тема проекту: «Дослідження часових рядів навантаженості мережевих систем».

Керівник дипломного проекту: доцент кафедри КІТ О. В. Горбова.

2 ПРИЗНАЧЕННЯ РОЗРОБКИ**2.1 Функціональне призначення**

Основним функціональним призначенням програмного продукту є відстеження навантаженості мережевого трафіку за певний час та прогнозування майбутнього навантаження за попередніми даними.

2.2 Експлуатаційне призначення

Головне призначення дослідження полягає в розробці та програмної реалізації методів моніторингу і прогнозування перевантажень комп'ютерних мереж. Розроблений програмний додаток може бути використано в компаніях спеціалізованих на роботі з серверним обладнанням або в компаніях, які мають власне серверне обладнання та мають необхідність оптимізації мережевого трафіку при критичних навантаженнях з метою підвищення рівня безпеки передачі даних.

3 ВИМОГИ ДО ПРОГРАМИ

3.1. Вимоги до функціональних характеристик

Вимоги до функціональних характеристик наступні:

– додавання, редагування, перегляд даних (кількості точок прогнозування);

– перегляд проведеного аналізу про мережу;

– перегляд графіків завантаженості мережі.

В ході розробки програмного продукту було висунуто наступні вимоги:

– вибір текстового документу з інформацією про запити до серверу;

– введення кількості запитів, що будуть приймати участь у прогнозуванні майбутнього навантаження ;

– введення кількості точок, які необхідно спрогнозувати;

– перегляд графіків завантаженості мережі.

Передбачити можливість введення некоректних даних та забезпечити реагування програми на них.

Вхідні дані

Метод організації вхідних даних полягає в наданні користувачу можливості введені вхідної інформації у вигляді файлу формату «.txt». Файл повинен мати у своїй структурі наступні дані:

– ір адреси вузла;

– розмір пакету;

– час відправки пакету.

Вихідні дані

Вихідними даними являється таблиця з інформацією про введений проміжок часу з навантаження мережі, а також з результатами прогнозування мережі, графік навантаження мережі. Відображення повинно бути коректним та зрозумілим непідготовленому користувачеві.

3.2. Вимоги до надійності

Програмний продукт має забезпечити стійку роботу, коректне виконання своїх основних функцій та цілісність і збереженість даних. Повинні виконуватися наступні вимоги:

- захисти персональних даних;
- завантаження текстового файлу;
- відображення даних;
- наявність архівної копії тексту програми на зовнішньому носії.

3.3. Умови експлуатації

Для нормального функціонування програмного продукту необхідно виконання наступних вимог:

- з метою полегшення надійного функціонування програми періодично проводити резервне копіювання даних на зовнішні носії;
- стан технічних пристроїв повинен задовольняти відповідні норми і вимоги;
- обчислювальна техніка, на якій буде експлуатуватися програма, повинна знаходитися в приміщенні з температурою повітря від 21° С до 25° С, вологістю повітря 40 – 60%, швидкістю руху повітря не більш 0,2 м/с;
- працювати з програмою може людина, яка володіє навичками роботи з комп'ютером на рівні користувача та ознайоmlена з керівництвом користувача.

3.4. Вимоги до складу та параметрів технічних засобів

Мінімальна конфігурація, що забезпечить нормальну експлуатацію програмного продукту:

- тип процесора Intel Pentium 1400 MHz або краще;
- обсяг оперативної пам'яті 128 Мб;
- обсяг вільного місця на жорсткому диску 475 Мб і більше;
- присутність монітора комп'ютера - пристрої виведення інформації про роботу, проведену програмним забезпеченням;

- наявність USB-портів;
- наявність підключення до Інтернету;
- присутність комп'ютерної миші і клавіатури - пристроїв введення інформації під час роботи з програмою.

3.5. Вимоги до інформаційної і програмної сумісності

Програма повинна робити прогнозування навантаженості мережевих систем на основі часових рядів. Програма повинна працювати під операційними системами Windows: Windows 7, Windows 8, Windows 10. Система повинна розроблятися в середовищі візуального програмування (наприклад Visual Studio).

3.6. Вимоги до маркування та упаковки

Програма може зберігатись на жорсткому диску або на знімних носіях (флеш-носії, CD диски). На упаковці повинно бути вказано назву продукту, його серійний номер, версія, вимоги до складу та параметрів технічних засобів, вимоги до інформаційної та програмної сумісності.

Приклад маркування наведено на рис. 3.1.

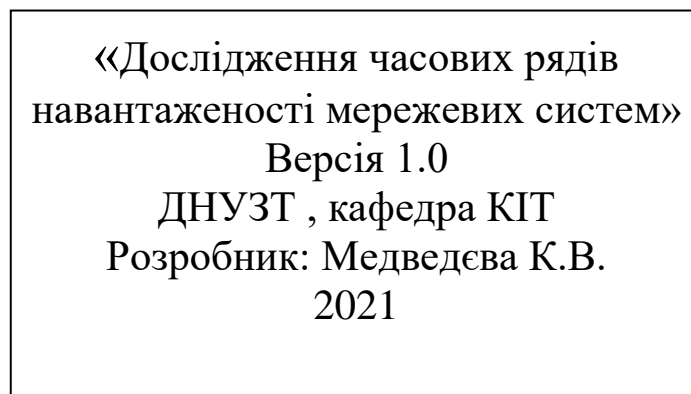


Рисунок 3.1 – Приклад маркування

3.7. Вимоги по транспортуванню та зберіганню

Транспортування програмного продукту може виконуватися шляхом його переносу на змінних інформаційних носіях чи по інформаційних каналах зв'язку через мережу Інтернет. Не допускається механічний вплив на носії. Інформаційні носії повинні зберігатися в сухому теплом місці, не допускати

попадання прямих сонячних променів, температура повітря 18-40⁰С, при вологості 50-70%, не допускати попадання пилу.

4 ВИМОГИ ДО ПРОГРАМНОЇ ДОКУМЕНТАЦІЇ

Програмна документація на даний програмний продукт повинна включати наступні документи:

- специфікації програми;
- текст програми;
- опис програми;
- керівництво користувача. Керівництво адміністратора сервера.

5 ТЕХНІКО–ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ ПРОЕКТУ РОЗРОБКИ ПРОГРАМНОГО ПРОДУКТУ

Техніко–економічне обґрунтування (ТЕО) – це обов'язкова складова частина будь–якого інвестиційного проекту, тобто проекту, що потребує певних фінансових витрат. Основна мета розробки ТЕО – дати фінансову оцінку передбачуваних витрат та одержуваного корисного результату, а також оцінити прибутковість проекту і, в кінцевому підсумку, економічну доцільність його розробки та впровадження.

Нова техніка, технологія, засоби автоматизації, що розробляються і впроваджуються у виробництво, повинні приносити певний корисний результат – ефект. Ефект може проявлятися у поліпшенні умов праці працюючих (соціальний), в зниженні шкідливого впливу виробництва на навколишнє середовище (екологічний), у підвищенні безпеки держави (оборонний), та, врешті, в економії витрат підприємства на виробництво продукції та збільшенні його прибутку (економічний).

Абсолютна величина економічного ефекту без співставлення його з витратами підприємства не дозволяє однозначно оцінити, наскільки вдалим виявився відповідний інноваційний проект. Таку оцінку дають показники економічної ефективності (прибутковості) проекту.

При впровадженні інвестиційного проекту підприємство несе разові витрати, пов'язані з розробкою проекту, а також з придбанням і налагодженням необхідного обладнання, засобів програмного забезпечення і таке інше.

Такі разові витрати називають капітальними витратами або інвестиціями. При використанні інновацій підприємство отримує певний ефект, що зазвичай виражається приростом прибутку. При розрахунках ефективності необхідно врахувати додаткові річні витрати підприємства, пов'язані з експлуатацією нового обладнання. Величина щорічного прибутку, додатково одержуваного підприємством за рахунок впровадження інвестиційного проекту, повинна бути достатньо високою у порівнянні з

капітальними витратами підприємства та у порівнянні з іншими можливими варіантами вкладення коштів у розвиток виробництва.

Початковим етапом розрахунку величини трудових витрат розробників є оцінка розміру програмного забезпечення. Основні відмінності методик, що застосовуються в оцінці трудовитрат, полягають у використовуваному типі критерію оцінки якості (кількісний або якісний).

Згідно моделі COCOMO, розмір проекту S вимірюється в рядках коду LOC (KLOC), а трудовитрати в людино-місяцях.

$$E = a \cdot S^b \cdot EAF. \quad (1)$$

де E – витрати праці на проект (в людино-місяцях);

S^b – розмір коду (в KLOC);

EAF – фактор уточнення витрат (effort adjustment factor).

Для простих систем, $a = 2,4$; $b = 1,05$

Припустимо, що розмір програмного коду програмного засобу – 900 рядків:

$$E = 2,4 \cdot 0,9^{1,05} \cdot 1 = 2,14. \quad (2)$$

Отже, згідно моделі COCOMO, орієнтовні трудовитрати на проект складуть приблизно 2,14 людино-місяці.

Розрахуємо вартість розробки програмного забезпечення для моделювання залежності попиту від ціни за допомогою функції Перла-Ріда
Основними статтями витрат прийняті:

- основна заробітна плата;
- відрахування на соціальні потреби;
- накладні витрати;
- витрати на персональний комп'ютер і ліцензійні базові програмні засоби.

Основна заробітна плата (ОЗП) оцінює працю інженера–програміста зі створення програмного продукту і визначається виходячи з кількості розробників, часу виконання розробки (годин), а також заробітної плати в розрахунку на одну годину. Рекомендована кількість виконавців – 3 чол;

тривалість розробки – 2,14 місяців. Розрахунок зарплати проводиться по формі табл. 5.1.

Таблиця 5.1 – Фонд місячної заробітної плати

№ п/п	Посада виконавця	Оклад, <i>грн/міс</i>	Кількість		Сума зарплати, <i>грн</i>
			<i>чол</i>	місяців	
1	інженер- програміст	16560	1	2,14	41400

Інформація взята з сайту Work.ua.

Описаний в проекті програмний продукт розроблений одним програмістом в період з 01.09.21 до 19.11.21, що складає 56 днів або 10 робочих тижнів. Витрати робочого часу приймемо 40 часів у тиждень. Погодинна ставка кваліфікованого інженера–програміста складає 103,5 *грн/год*. Таким чином, витрачено робочого часу:

$$t_{\text{розробки}} = N_{\text{чол}} * N_{\text{тиж}} * N_{\text{год}}, \quad (3)$$

де $N_{\text{чол}}$ – кількість виконавців, *чол*;

$N_{\text{тиж}}$ – тривалість розробки;

$N_{\text{год}}$ – витрати робочого часу, *год*;

$$t_{\text{розробки}} = 1 \cdot 10 \cdot 40 = 400 \text{ чол/год}$$

ОЗП визначається за формулою:

$$\text{ОЗП} = t_{\text{розробки}} \cdot N \cdot K_{\text{КВ}}, \quad (4)$$

де $t_{\text{розробки}}$ – витрати праці у *чол/год*;

N – погодинна ставка;

$K_{\text{КВ}}$ – коефіцієнт кваліфікації програміста, приймаємо 0.75.

$$\text{ОЗП} = 400 \cdot 103,5 \cdot 0,75 = 31\,050 \text{ грн.}$$

Відрахування на соціальні потреби встановлюються у відсотках від суми заробітної плати:

$$C_{\text{соц}} = \frac{1116130.01204 \cdot 0.22}{100}, \quad (4)$$

$$C_{\text{соц}} = \frac{31050 \cdot 22\%}{100\%} = 6831 \text{ грн.}$$

Отримані результати за (5.2) – (5.3) підсумовуються. Вони складають 111 011,82 *грн.* та визначають основні прямі витрати.

Накладні витрати враховують загально-господарчі витрати по забезпеченню проведення роботи: витрати на опалення, електроенергію, амортизація будівель, зарплату адміністративного персоналу та інше. Вони визначаються в процентах (30–40 %) від суми прямих витрат:

$$C_{\text{накл}} = \frac{(OЗП + C_{\text{соц}}) \cdot 35\%}{100\%}, \quad (6)$$

$$C_{\text{накл}} = \frac{(31050 + 6831) \cdot 35\%}{100\%} = 13258,35 \text{ грн.}$$

Протягом усього терміну використання нової техніки підприємство щорічно витрачає певні кошти, пов'язані з її експлуатацією.

Експлуатаційні витрати на персональний комп'ютер визначаються протягом терміну розробки програмного засобу в залежності від вартості комп'ютеру. В експлуатаційні витрати входять:

- витрати на електроенергію;
- вартість витратних матеріалів;
- витрати на ремонт;
- заробітна плата ремонтника;
- додаткові витрати – прибирання приміщення, охорона, оренда, комунальні послуги;
- амортизаційні витрати на персональний комп'ютер і програмне забезпечення.

Витрати на електроенергію ($C_{\text{ел}}$) визначаються за формулою, взято з сайту yasno.ua:

$$C_{\text{ел}} = P \cdot B \cdot T_{\text{розр}}, \quad (7)$$

де P – потужність комп'ютера та допоміжних споживачів електричної енергії, приймаємо $0,35 \text{ кВт/год}$;

B – вартість 1 кВт/год складає $1,68 \text{ грн}$;

$T_{\text{розр}}$ – час роботи з ЕВМ, прийнято рівним робочому часу.

$$C_{\text{ел}} = 0,35 \cdot 1,68 \cdot 400 = 235,5 \text{ грн.}$$

Витрати на витратні матеріали ($C_{\text{вм}}$) протягом всього терміну експлуатації приблизно 10% від вартості комп'ютеру. Вартість комп'ютеру приймаємо 25000 грн , термін експлуатації – 3 роки, див. рис. 5.1.


Робочий ПК #46711				✉ ✎ 🗑		
	• Корпус GAMEMAX ET-209-NP	652 грн	1 шт.	652	грн	
	• Клавіатура LOGITECH K120 UA OEM (920-002643)	349 грн	1 шт.	349	грн	
	• Миша LOGITECH M90 Dark Gray (910-001794)	249 грн	1 шт.	249	грн	
	• Процесор INTEL Core i7-10700K 3.8GHz s1200 (BX8070110700K)	10 659 грн	1 шт.	10 659	грн	
	• Відеокарта AFOX GeForce GT 730 LP (V6) (AF730-2048D3L6)	1 986 грн	1 шт.	1 986	грн	
	• Материнська плата MSI H510M Pro-E	2 090 грн	1 шт.	2 090	грн	
	• Модуль пам'яті G.SKILL Ripjaws V Classic Black DDR4 3200MHz 16GB (F4-3200C16S-16GVK)	1 646 грн	1 шт.	1 646	грн	
	• Блок живлення 500W GAMEMAX Eco GM-500B	745 грн	1 шт.	745	грн	
	• Кулер для процесора DEEPCOOL Gammaxx 300 (DP-MCH3-GMX300)	585 грн	1 шт.	585	грн	
	• SSD KINGSTON A400 240GB 2.5" SATA (SA400S37/240G)	939 грн	1 шт.	939	грн	
	• Монітор PHILIPS 243V7QDAB/00	5 099 грн	1 шт.	5 099	грн	
Збірці 11 товарів				Покращити	КУПИТИ ЗБІРКУ	Загалом: 24 999 грн

Рисунок 5.1 – Вартість комп'ютеру

Отже, можна визначити ці витрати за період створення програмного засобу:

$$C_{\text{вм}} = B \cdot \frac{N_{\text{д}}}{N_{\text{екс}} \cdot 365} \cdot \frac{10\%}{100\%}, \quad (8)$$

де $B_{\text{ком}}$ – вартість персонального комп'ютеру;

$N_{\text{д}}$ – кількість днів розробки програмного продукту;

$N_{\text{експ}}$ – термін експлуатації персонального комп'ютеру.

$$C_{\text{вм}} = 25000 \cdot \frac{56}{2,14 \cdot 365} \cdot \frac{10}{100} = 179,2 \text{ грн}$$

Заробітна плата ремонтника ($C_{\text{рем}}$) визначена наступним чином: на ремонт 50 комп'ютерів потрібен один інженер–системотехнік. Його середньомісячна заробітна плата приймається 10500 *грн*. Тоді в перерахунку на один комп'ютер його заробітна плата складає:

$$C_{\text{рем}} = \frac{C'_{\text{рем}}}{N_{\text{ком}}}, \quad (9)$$

де $C'_{\text{рем}}$ – середньомісячна заробітна плата;

$N_{\text{ком}}$ – кількість комп'ютерів на одного ремонтника.

$$C_{\text{рем}} = \frac{10500}{50} = 210 \text{ грн.}$$

За статистикою витрати на комплектуючі вироби ($C_{\text{ком}}$) для ремонту персонального комп'ютера складає 10 % від його вартості за термін його експлуатації, тобто рівні витратам на витратні матеріали.

$$C_{\text{ком}} = C_{\text{вм}} = 179,2 \text{ грн}, \quad (10)$$

Амортизаційні відрахування на персональний комп'ютер (АПК) визначені з положення, що амортизаційний період в даний час дорівнює терміну морального старіння обчислювальної техніки і складає 3 роки. Отже, за 3 роки амортизаційні відрахування на персональний комп'ютер дорівнюють вартості комп'ютера. За період проектування амортизаційні відрахування складуть:

$$\text{АПК} = B_{\text{ком}} \cdot \frac{N_{\text{д}}}{N_{\text{експ}} \cdot 365}, \quad (11)$$

$$\text{АПК} = 25000 \cdot \frac{56}{2,14 \cdot 365} = 1792,34 \text{ грн.}$$

Амортизаційні відрахування на програмне забезпечення (АПЗ) залежать від його циклу заміни. Якщо прийняти термін морального старіння таким же, як у персонального комп'ютера, то амортизаційні відрахування на програмне забезпечення за 3 роки дорівнюють його вартості. Для функціонування персонального комп'ютера використовувалася операційна система Windows 7 Professional, для написання експертної системи оболонка програми. Розрахунок амортизаційних відрахувань на програмне забезпечення зведений в табл. 5.2.

Таблиця 5.2 – Використовуване програмне забезпечення

Найменування програмного забезпечення	Вартість програмного забезпечення, грн	Джерело придбання	Амортизаційні відрахування, грн
Windows 10 Professional	7800,1	Розетка	926,03
MS Office-14	18700	Розетка	2220,09
Всього:			3146,12

Додаткові витрати ($C_{\text{дод}}$): прибирання приміщень, охорона, оренда, комунальні послуги важко оцінити точно і прийняти рівними 50 % заробітної плати інженера–системотехніка, тобто 3500 грн, з якої доля оренди складає 2000 грн., – вартість оренди 10 м², дані взяти з сайту <https://dom.ria.com>, де приведено приклад приміщення розміром 30 м², отже вартість одного м² складає 200 грн.

Сумарні експлуатаційні витрати на один персональний комп'ютер складають:

$$C_{\text{експ}} = C_{\text{ел}} + C_{\text{вм}} + C_{\text{рем}} + C_{\text{ком}} + \text{АПК} + \text{АПО} + C_{\text{дод}}, \quad (12)$$

$$C_{\text{експ}} = 235,5 + 179,2 + 210 + 179,2 + 1792,34 + 3500 + 3146,12$$

$$= 9242,36 \text{ грн.}$$

Результати розрахунків зводимо у табл. 5.3.

Таким чином, витрати на створення програмного продукту складають:

$$C_{\text{розробки}} = \text{ОЗП} + C_{\text{соц}} + C_{\text{накл}} + C_{\text{експ}}, \quad (13)$$

$$C_{\text{розробки}} = 31050 + 6831 + 13258,35 + 9242,36 = 60\,381,71 \text{ грн.}$$

Таблиця 5.3 – Експлуатаційні витрати на ПК і ПО

Найменування витрат	Витрати, грн
Витрати на електроенергію	235,5
Вартість витратних матеріалів	179,2
Витрати на ремонт	179,2
Заробітна плата інженера системотехніка	210
Амортизація персонального комп'ютера	1792,34
Амортизація програмного забезпечення	3146,12
Додаткові витрати	3500
Всього	9242,36

Таким чином, витрати на створення програмного продукту складають:

$$C_{\text{розробки}} = \text{ОЗП} + C_{\text{соц}} + C_{\text{накл}} + C_{\text{експ}}, \quad (13)$$

$$C_{\text{розробки}} = 31050 + 6831 + 13258,35 + 9242,36 = 60\,381,71 \text{ грн.}$$

Розрахунок витрат зводимо у табл. 5.4.

Таблиця 5.4 – Кошторис витрат на розробку програмного засобу

Найменування витрат	Витрати, грн
Основна заробітна плата	31050
Відрахування на соціальні потреби	11646,85
Накладні витрати	14943,89
Експлуатаційні витрати	9242,36
Всього	60 381,71

За отриманими значеннями техніко–економічних показників проекту складено кошторис витрат на розробку сучасного програмного забезпечення для обробки результатів фінансової діяльності підприємства, призначеного для безперервного контролю фінансової діяльності на підприємстві, потоку грошових коштів в режимі постійного моніторингу.

6 СТАДІЇ ТА ЕТАПИ РОЗРОБКИ

Стадії та етапи розробки програмного продукту «Дослідження часових рядів навантаженості мережевих систем» представлені в табл. 6.1.

Таблиця 6.1 – Стадії та етапи розробки

№ пор.	Назва розділів дипломної роботи	Термін виконання розділів роботи	При - мітка
1	Вступ	1.09.21	
2	Аналіз сучасного стану дослідження проблеми за науковими літературними джерелами	2.09.21 – 14.09.21	від 70 джерел
2.1	Дослідження існуючих рішень та інших аспектів дослідження	3.09.21 – 10.09.21	
3	Аналіз сучасного стану програмно-апаратного забезпечення, яке потребує вдосконалення для вирішення проблем дослідження	16.09.21 – 10.10.21	
4	Постановка задачі, технічне завдання	11.10.21 – 17.10.21	30%
4.1	Розробка постановки задачі	11.10.21 – 13.10.21	
4.2	Розробка технічного завдання	14.10.21 – 16.10.21	
4.3	Затвердження постановки задачі та технічного завдання	17.10.21	
5	Техніко-економічні показники	18.10.21 – 19.10.21	
5.1	Розробка техніко-економічного обґрунтування розробки проекту	18.10.21	
5.2	Затвердження техніко-економічного обґрунтування розробки проекту	19.10.21	
6	Розробка інструментальних засобів дослідження	20.10.21 – 8.11.21	
6.1	Проектування інструментального засобу	20.10.21 – 25.10.21	
6.2	Розробка алгоритмів та програмування інструментального засобу	26.10.21 – 3.11.21	

Продовження таблиці 6.1

6.3	Розробка та програмування інтерфейсу користувача	4.11.21 – 6.11.21	
6.4	Тестування та відлагодження інструментального засобу	7.11.21	
7	Виконання досліджень	8.11.21 – 14.11.21	60%
8	Оформлення тез доповідей	15.11.21 – 18.11.21	
9	Оформлення статті у фаховий журнал	19.11.21 – 22.11.21	
10	Оформлення пояснювальної записки	23.11.21 – 28.11.21	
11	Розробка демонстраційних матеріалів	29.11.21 – 5.12.21	100%

7 ПОРЯДОК КОНТРОЛЮ ТА ПРИЙОМУ

Контроль здійснюється за допомогою виконання набору тестів з метою виявлення помилок в програмному продукті та його специфікаціях. Контроль за виконанням роботи здійснює головний керівник розробки – О.В. Горбова.

БІБЛІОГРАФІЧНИЙ СПИСОК

1. Івченко, Ю.М. Основи стандартизації програмних систем [Текст]: методичні вказівки до дипломного проектування та лабораторних робіт / уклад.: Ю. М. Івченко, В. І. Шинкаренко, В. Г. Івченко; Дніпропетр. нац. ун-т залізн. трансп. ім. акад. В. Лазаряна. – Д.: Вид-во Дніпропетр. нац. ун-ту залізн. трансп. ім. акад. В. Лазаряна, 2009. – 38 с.

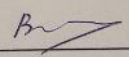
ДОДАТОК Б
МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ

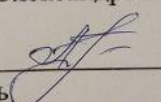
ЗАТВЕРДЖУЮ
Проректор Дніпровського
національного університету
залізничного транспорту
імені академіка В. Лазаряна
Борис БОДНАР

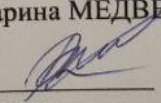
РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ «НАВАНТАЖЕННЯ
МЕРЕЖЕВИХ СИСТЕМ»

Робочий проект
ЛИСТ ЗАТВЕРДЖЕННЯ
1116130.01204 ЛЗ

Завідувача кафедри КІТ
Вадим ГОРЯЧКІН


Керівник розробки
Олександра ГОРБОВА


Виконавець
Карина МЕДВЕДЄВА


Нормоконтролер
Олена КУРОП'ЯТНИК

ЗАТВЕРДЖЕНО

1116130.01204-01-ЛЗ

ДОСЛІДЖЕННЯ ЧАСОВИХ РЯДІВ НАВАНТАЖЕНОСТІ
МЕРЕЖЕВИХ СИСТЕМ

Специфікація

1116130.01204-01

Аркушів 2

Позначення	Найменування	Примітка
1116130.01204-01-ЛЗ	Лист затвердження	
1116130.01204-01 12 01-ЛЗ	Лист затвердження	
1116130.01204-01 12 01	Текст програми	
1116130.01204-01 13 01-ЛЗ	Лист затвердження	
1116130.01204-01 13 01	Опис програми	
1116130.01204-01 13 01-ЛЗ	Лист затвердження	
1116130.01204-01 13 01	Керівництво користувача. Керівництво адміністратора серверу	

ЗАТВЕРДЖЕНО

1116130.01204–01 12 01 –ЛЗ

ДОСЛІДЖЕННЯ ЧАСОВИХ РЯДІВ НАВАНТАЖЕНОСТІ МЕРЕЖЕВИХ
СИСТЕМ

Текст програми

1116130.01204-01 12 01-ЛЗ

Листів 10

АНОТАЦІЯ

Документ 1116130.01204-01 12 01 «Дослідження часових рядів навантаженості мережевих систем. Текст програми» входить до складу документації на проект.

В даному документі представлено текст програми. Програма реалізована на мові високого рівня C# у програмному середовищі Visual Studio. Об'єм пам'яті, що займають програми комплексу складає 4 Мб. Конфігурація ком'ютера стандартна. Комплекс функціонує в середовищі MS Windows.

ЗМІСТ

1. Схеми взаємодії модулів програми.....	4
2. Текст програми	6
2.1. Form1.cs	6
2.2. MainControl.cs	8
2.3 Diploma.cs.....	10

1 СХЕМА ВЗАЄМОДІЇ МОДУЛІВ ПРОГРАМИ

На рис. 1.1 приведено схема взаємодії модулів програми.

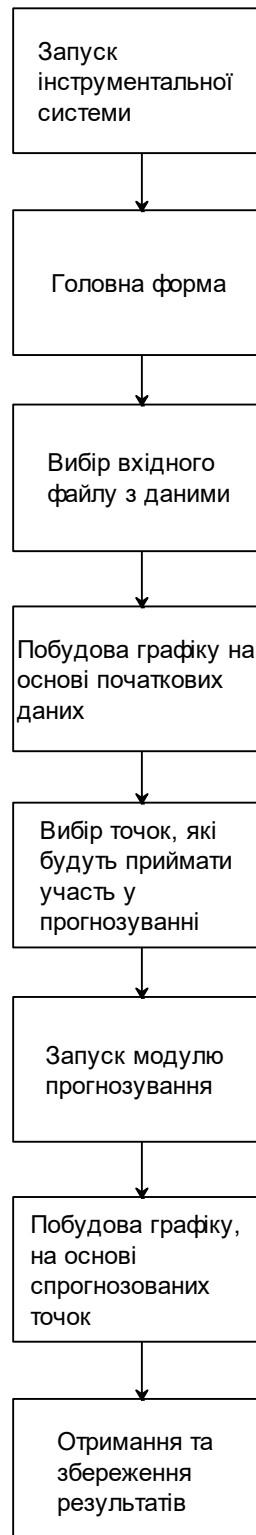


Рисунок 1.1 – схема взаємодії модулів програми

Програма складається з трьох модулів. Опишемо призначення кожного модуля програми:

Form1.cs – модуль з прописом прототипів функцій інтерфейсної частини програмного комплексу.

MainControl.cs – модуль з прописом прототипів функцій частини програмного комплексу.

Diploma.cs – виконавчий модуль запуску програми.

2 ТЕКСТ ПРОГРАМИ

2.1. Form1.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Diploma
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        MainControl mainControl = new MainControl();

        private void button1_Click(object sender, EventArgs e)
        {
            try
            {
                openFileDialog1.Filter = "Текстові файли (*.txt)|*.txt";
                openFileDialog1.ShowDialog();
                textBox1.Text = openFileDialog1.FileName;
                if (textBox1.Text != "" && textBox1.Text != "openFileDialog1")
                {
                    var infoIn = mainControl.LoadTimeFile(openFileDialog1.FileName);
                    for (int i = 0; i < infoIn.Item1.Count; i++)
                    {
                        dataGridView1.Rows.Add(i+1,infoIn.Item1[i],infoIn.Item2[i]);
                        dataGridView2.Rows.Add(i + 1, infoIn.Item1[i], infoIn.Item2[i]);
                        chart1.Series[0].Points.AddXY(i + 1, infoIn.Item2[i]);
                        chart2.Series[0].Points.AddXY(i + 1, infoIn.Item2[i]);
                    }
                    numericUpDown1.Maximum = infoIn.Item2.Count;
                }
                else { textBox1.Text = ""; }
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message, "Помилка" , MessageBoxButtons.OK,
                MessageBoxIcon.Error);
            }
        }
    }
}
```



```
}

private void button2_Click(object sender, EventArgs e)
{
    try
    {
        var result = mainControl.ForecastCalculation((int)numericUpDown1.Value,
(int)numericUpDown2.Value);
        if (chart2.Series[1].Points.Count == 0)
            chart2.Series[1].Points.AddXY((int)chart2.Series[0].Points.Last().XValue,
(int)chart2.Series[0].Points.Last().YValues.Last());
        for (int i = result.Keys.First(); i < result.Keys.Last(); i++)
        {
            dataGridView2.Rows.Add(i, "Прогноз", result[i]);
            chart2.Series[1].Points.AddXY(i, result[i]);
            chart2.Update();
        }
        MessageBox.Show("Прогнозування успішно завершено!", "Повідомлення",
MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
    catch(Exception ex)
    {
        MessageBox.Show(ex.Message, "Помилка", MessageBoxButtons.OK,
MessageBoxIcon.Error);
    }
}

private void button3_Click(object sender, EventArgs e)
{
    dataGridView2.Rows.Clear();
    chart2.Series[1].Points.Clear();
    mainControl.Clear();
}

private void button4_Click(object sender, EventArgs e)
{
    saveFileDialog1.Filter = "Книга Excel (*.xlsx)|*.xlsx";
    saveFileDialog1.ShowDialog();
    var path = saveFileDialog1.FileName;
    try
    {
        Microsoft.Office.Interop.Excel.Application excelapp = new
Microsoft.Office.Interop.Excel.Application();
        Microsoft.Office.Interop.Excel.Workbook workbook = excelapp.Workbooks.Add();
        Microsoft.Office.Interop.Excel.Worksheet worksheet = workbook.ActiveSheet;

        for (int i = 1; i < dataGridView2.RowCount + 1; i++)
        {
            for (int j = 1; j < dataGridView2.ColumnCount + 1; j++)
            {
```

2.2. MainControl.cs

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Diploma
{
    class MainControl
    {
        Dictionary<int, int> timeSeries = new Dictionary<int, int>();
        Dictionary<int, int> timeSeriesCalculation = new Dictionary<int, int>();
        public (List<string>, List<int>) LoadTimeFile(string path)
        {
            try
            {
                string f = "";
                using (StreamReader sr = new StreamReader(path))
                {
                    f = sr.ReadToEnd();
                }
                List<string> ip = new List<string>();
                List<int> ping = new List<int>();
                string[] lines = f.Split('\n');
                int i = 1;
                foreach (string line in lines)
                {
                    string[] s = line.Split(':');
                    ip.Add(s[0]);
                    string[] parts = s[1].Split('=');
```

```

        ping.Add(Convert.ToInt32(parts[2]));
        timeSeries.Add(i, Convert.ToInt32(parts[2]));
        i++;
    }
    foreach(var item in timeSeries)
        timeSeriesCalculation.Add(item.Key,item.Value);
    return (ip, ping);
}
catch (Exception e)
{
    throw e;
}
}
public Dictionary<int, int> ForecastCalculation(int countPointsCalculation, int
countPointNeed)
{
    Dictionary<int, int> timeSeriesForecast = new Dictionary<int, int>();

    for (int j = 0; j < countPointNeed; j++)
    {
        int sxy = 0;
        int sx = 0;
        int sy = 0;
        int sx2 = 0;
        int n = timeSeriesCalculation.Count;

        for (int i = n + 1 - countPointsCalculation; i < n + 1; i++)
        {
            sxy += i * timeSeriesCalculation[i];
            sx += i;
            sy += timeSeriesCalculation[i];
            sx2 += i * i;
        }
        int a = (n * sxy - sx * sy) / (n * sx2 - sx * sx);
        int b = (sy - a * sx) / n;
        int y = a * (n + 1) + b;
        timeSeriesCalculation.Add(n + 1, y);
        timeSeriesForecast.Add(n + 1, y);
    }
    return timeSeriesForecast;
}

public void Clear()
{
    timeSeriesCalculation.Clear();
    foreach (var item in timeSeries)
        timeSeriesCalculation.Add(item.Key, item.Value);
}
}
}

```

Diploma.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Diploma
{
    static class Program
    {
        /// <summary>
        /// Главная точка входа для приложения.
        /// </summary>
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new Form1());
        }
    }
}
```

ЗАТВЕРДЖЕНИЙ

1116130.01204-01 13 01-ЛЗ

ДОСЛІДЖЕННЯ ЧАСОВИХ РЯДІВ НАВАНТАЖЕНОСТІ МЕРЕЖЕВИХ
СИСТЕМ

Опис програми

1116130.01204-01 13 01

Листів 21

АНОТАЦІЯ

Документ 1116130.01204-01 13 01 «Дослідження часових рядів навантаженості мережевих систем. Опис програми» є складовою частиною документації на програму та входить до складу документації на проект.

В документі наведені функціональне призначення програми, опис логічної структури, технічні засоби, формат вхідних та вихідних даних. Програма реалізована на мові високого рівня C# у програмному середовищі Visual Studio. Об'єм пам'яті що займає програмний комплекс складає 4Мб. Конфігурація комп'ютера стандартна. Програма функціонує в середовищі MS Windows 7.

ЗМІСТ

1. Загальні відомості	4
2. Функціональне призначення.....	5
3. Опис логічної структури	6
3.1 Алгоритми роботи програми	6
3.2 Використані методи	7
3.3 Структура програми з описом функції.....	8
4. Технічні засоби, що використовуються	9
5. Виклик та завантаження	10
6. Вхідні дані	11
7. Вихідні дані	12
8. Опис інтерфейсу користувача	13
9. Порядок роботи з програмою	21
10. Повідомлення користувачу.....	22

1 ЗАГАЛЬНІ ВІДОМОСТІ

Розроблена програма має назву «Прогнозування навантаженості мережевих систем».

Програма розрахована на функціонування в операційних системах MS Windows 7/8/8.1/10.

Програма реалізована на мові C# у програмному середовищі Visual Studio. Об'єм пам'яті що займає програмний комплекс складає 4Мб. Конфігурація комп'ютера стандартна.

2 ФУНКЦІОНАЛЬНЕ ПРИЗНАЧЕННЯ

Функціональним призначенням програмного продукту є надання можливостей користувачу спрогнозувати та визначити навантаженість мережевої системи у певному сегменті мережі. Програмний комплекс дозволить користувачам спрогнозувати навантаженість мережевої системи, також допоможе системним адміністраторам покращити роботу сервісів та забезпечити безперебійну роботу мережі без втрати даних.

3 ОПИС ЛОГІЧНОЇ СТРУКТУРИ

3.1 Алгоритми роботи програми

Для роботи з програмним продукту необхідно підготувати вхідні файли.

Формат вхідних файлів описано у розділі 6.

Після загрузки файлу відбувається аналіз та розбиття інформації на необхідні групи.

Інформація розбивається на наступні групи:

- IP адреса вузлу, на котрий призводиться аналіз навантаження;
- розмір тестового пакету;
- час за котрий відбувається один цикл обігу пакету.

Після розбиття інформації на групи відбувається побудова графіку навантаження за введений проміжок часу.

Для прогнозування також потрібно обрати проміжок часу, за котрим буде відбуватися прогнозування майбутнього навантаження, а також проміжок часу на котрий буде виконано прогнозування.

Прогнозування відбувається з використанням методів найменших квадратів.

На основі спрогнозованого навантаження на головній формі інструментального засобу будується графік, що показує тренд поведінки навантаження мережевої системи.

Після завершення першого циклу прогнозування можливо повторний на основі вже спрогнозованого навантаження на попередньому циклу.

Структурна схема взаємодії складових програми наведена на рис. 3.1.

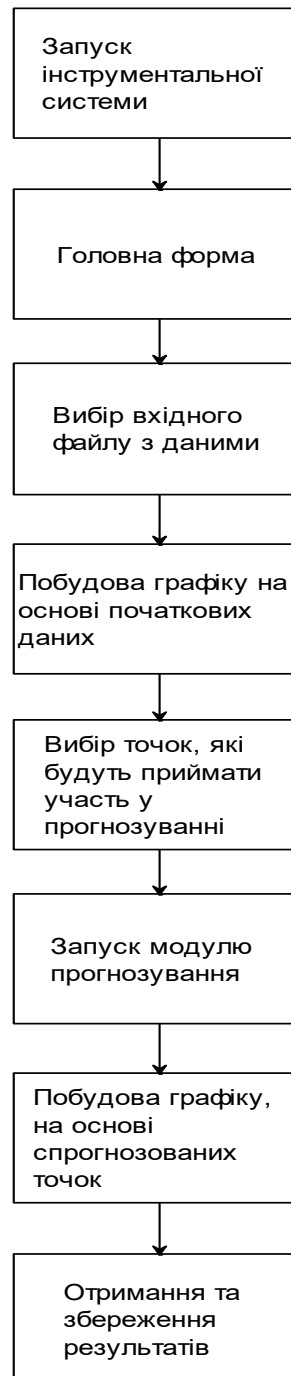


Рисунок 3.1 – Структурна схема взаємодії складових програми

3.2 Використані методи

В ході розробки програми були використані методи :

1. Для локального (тимчасового) зберігання інформації зі сторони клієнтської частини, програмного продукту, використовується ієрархічна система класів, кожен клас містить в собі відповідну інформацію з таблиці що знаходиться на сервері.

2. Програмний продукт розроблено згідно стандарту «MVC».

3.3 Структура програми з описом функції

Програма складається з трьох модулів. Опишемо призначення кожного модуля програми:

Form1.cs – модуль з прописом прототипів функцій інтерфейсної частини програмного комплексу.

MainControl.cs – модуль з прописом прототипів функцій частини програмного комплексу.

Diploma.cs – виконавчий модуль запуску програми.

4 ТЕХНІЧНІ ЗАСОБИ, ЩО ВИКОРИСТОВУЮТЬСЯ

Мінімальна конфігурація, що забезпечить нормальну експлуатацію програмного продукту:

- процесор: Intel Core 2 Duo 2.4 GHz або краще;
- оперативна Пам'ять: 2 GB DDR2 або краще;
- операційна система: Windows 7 або краще;
- вільний дисковий простір 50 Mb;
- наявність CD/DVD приводу або USB роз'ємну для встановлення необхідного ПЗ;
- підключення до локальної або глобальної мережі: 10mb/s або краще;
- монітор з роздільною здатністю екрану 1024x768;
- клавіатура;
- маніпулятор «миша».

5 ВИКЛИК ТА ЗАВАНТАЖЕННЯ

Для початку роботи програми на сервері необхідно встановити, налаштувати та заповнити файли з вхідними даними.

Для початку роботи програми на робочій станції необхідно виконати файл «Diploma\bin\Debug\Dip_Med.exe» Якщо програма знаходиться на зовнішньому носії даних (CD, DVD, Flash Memory), то її можна завантажити з них або скопіювавши на жорсткий диск.

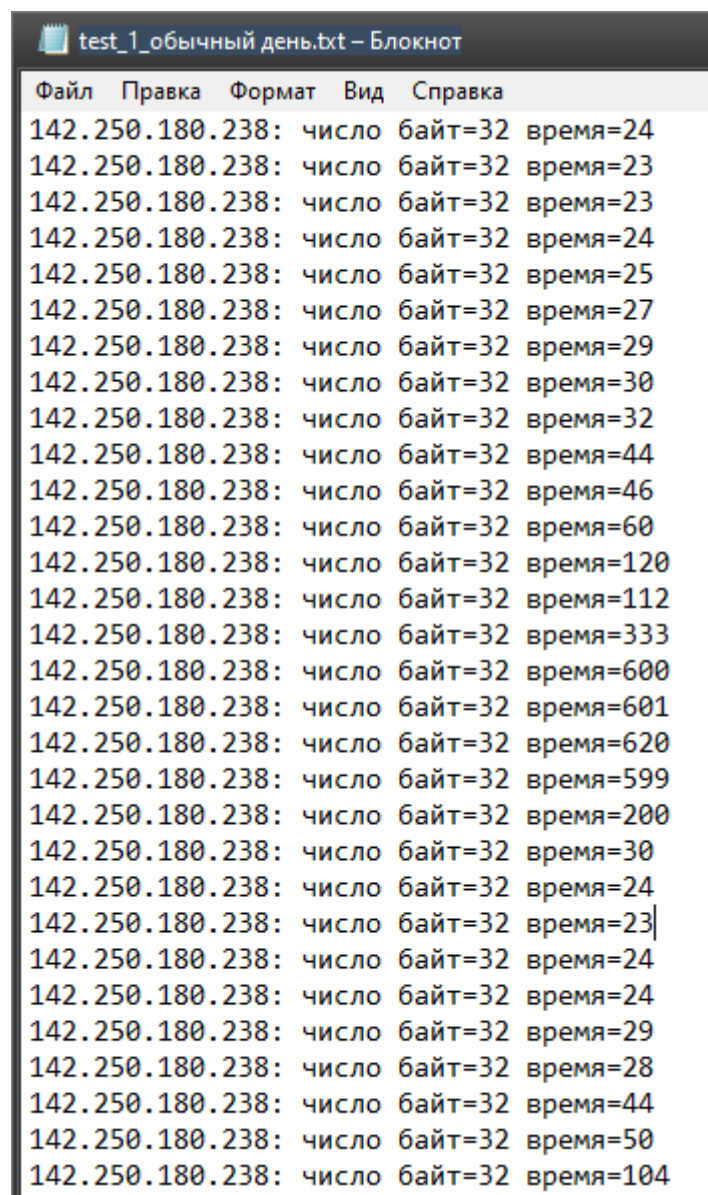
Об'єм програми складає 4 Мбайт.

6 ВХІДНІ ДАНІ

Файли що подаються на вхід в якості методичних матеріалів мають мати формат файлу – «.txt».

Вхідні дані представлені текстовим файлом, приклад наведено на рис. 6.1, який повинен мати у своїй структурі всі необхідні поля:

- IP – адреса вузла мережі;
- Число байт, які передаються;
- Ping – проміжок часу.



Файл	Правка	Формат	Вид	Справка
142.250.180.238:	число	байт=32	время=24	
142.250.180.238:	число	байт=32	время=23	
142.250.180.238:	число	байт=32	время=23	
142.250.180.238:	число	байт=32	время=24	
142.250.180.238:	число	байт=32	время=25	
142.250.180.238:	число	байт=32	время=27	
142.250.180.238:	число	байт=32	время=29	
142.250.180.238:	число	байт=32	время=30	
142.250.180.238:	число	байт=32	время=32	
142.250.180.238:	число	байт=32	время=44	
142.250.180.238:	число	байт=32	время=46	
142.250.180.238:	число	байт=32	время=60	
142.250.180.238:	число	байт=32	время=120	
142.250.180.238:	число	байт=32	время=112	
142.250.180.238:	число	байт=32	время=333	
142.250.180.238:	число	байт=32	время=600	
142.250.180.238:	число	байт=32	время=601	
142.250.180.238:	число	байт=32	время=620	
142.250.180.238:	число	байт=32	время=599	
142.250.180.238:	число	байт=32	время=200	
142.250.180.238:	число	байт=32	время=30	
142.250.180.238:	число	байт=32	время=24	
142.250.180.238:	число	байт=32	время=23	
142.250.180.238:	число	байт=32	время=24	
142.250.180.238:	число	байт=32	время=24	
142.250.180.238:	число	байт=32	время=29	
142.250.180.238:	число	байт=32	время=28	
142.250.180.238:	число	байт=32	время=44	
142.250.180.238:	число	байт=32	время=50	
142.250.180.238:	число	байт=32	время=104	

Рисунок 6.1 – приклад файлу з вхідною інформацією

7 ВИХІДНІ ДАНІ

До вихідних даних відносяться:

- файл формату «.xls»;
- сформований графік.

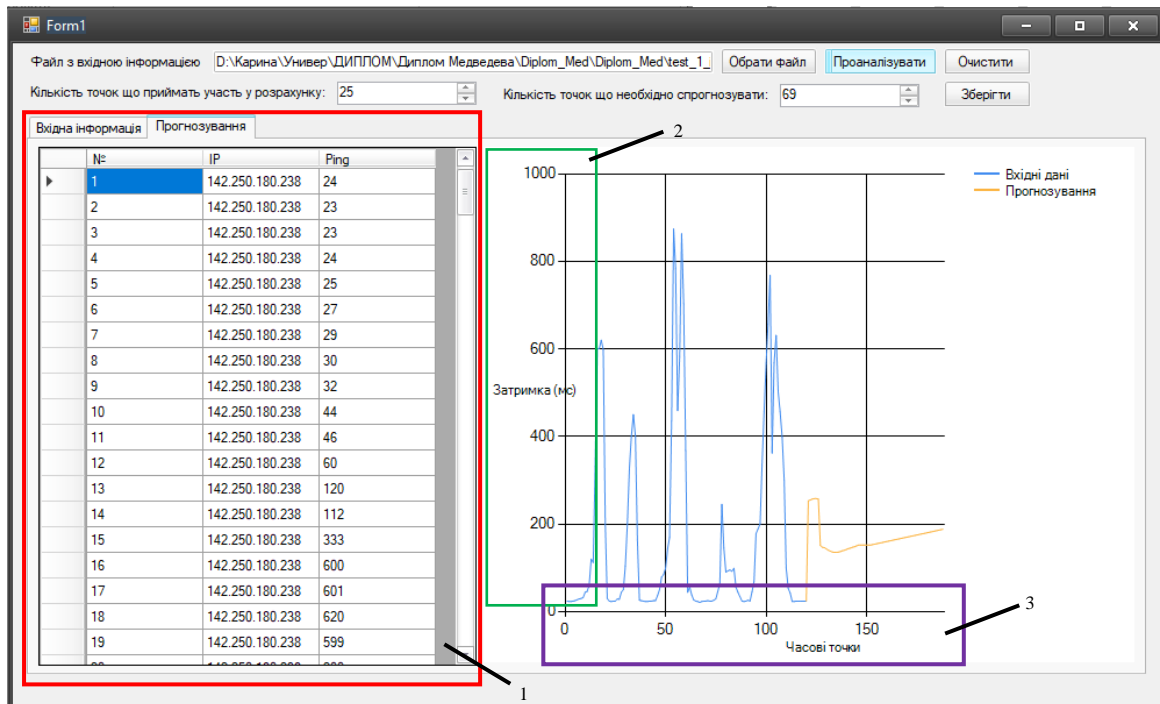


Рисунок 7.1 – приклад вихідних даних

На рис. 7.1 наведено:

- 1 – таблиця з прогнозованими точками;
- 2 – шкала затримки передачі даних;
- 3 – часові точки прогнозування.

8 ОПИС ІНТЕРФЕЙСУ КОРИСТУВАЧА

На рис. 8.1 зображено прототип головної сторінки, з якою буде працювати користувач.

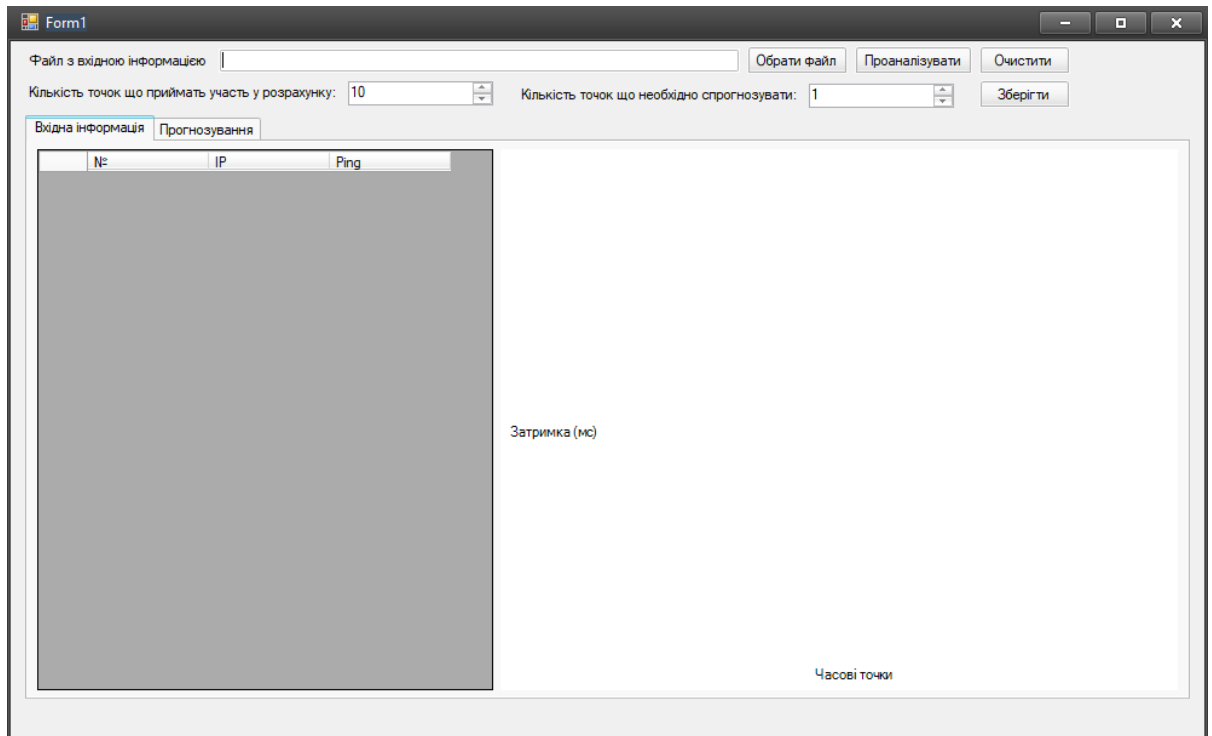


Рисунок 8.1 – Прототип головної сторінки

Об'єкти, які були використанні при розробці інтерфейсу програмного продукту та можливі дії над ними зображені у таблиці 8.1.

Таблиця 8.1

Назва об'єкту	Тип	Дії над об'єктом
label1	контейнер	Вивід підписів
label2	контейнер	Вивід підписів
textBox1	контейнер	Вивід тексту
numericUpDown1	засіб керування	Задання числових значень
numericUpDown2	засіб керування	Задання числових значень
button1	засіб керування	Взаємодія з програмою
button2	засіб керування	Взаємодія з програмою
button3	засіб керування	Взаємодія з програмою
button4	засіб керування	Взаємодія з програмою
tabControl1	засіб керування	Перехід по вкладкам
dataGridView1	засіб керування	Відображення таблиці
dataGridView2	засіб керування	Відображення таблиці
chart2	засіб керування	Відображення графіку

Головна форма програми, що включає в себе усі необхідні для роботи з програмним засобом, такі як, поля вводу посилання на файл з вхідними даними, поле для вибору кількості точок, що будуть приймати участі у прогнозуванні, вікно з отриманим графіком, вікна з представленими даними у табличному вигляді зображено на рис. 8.2

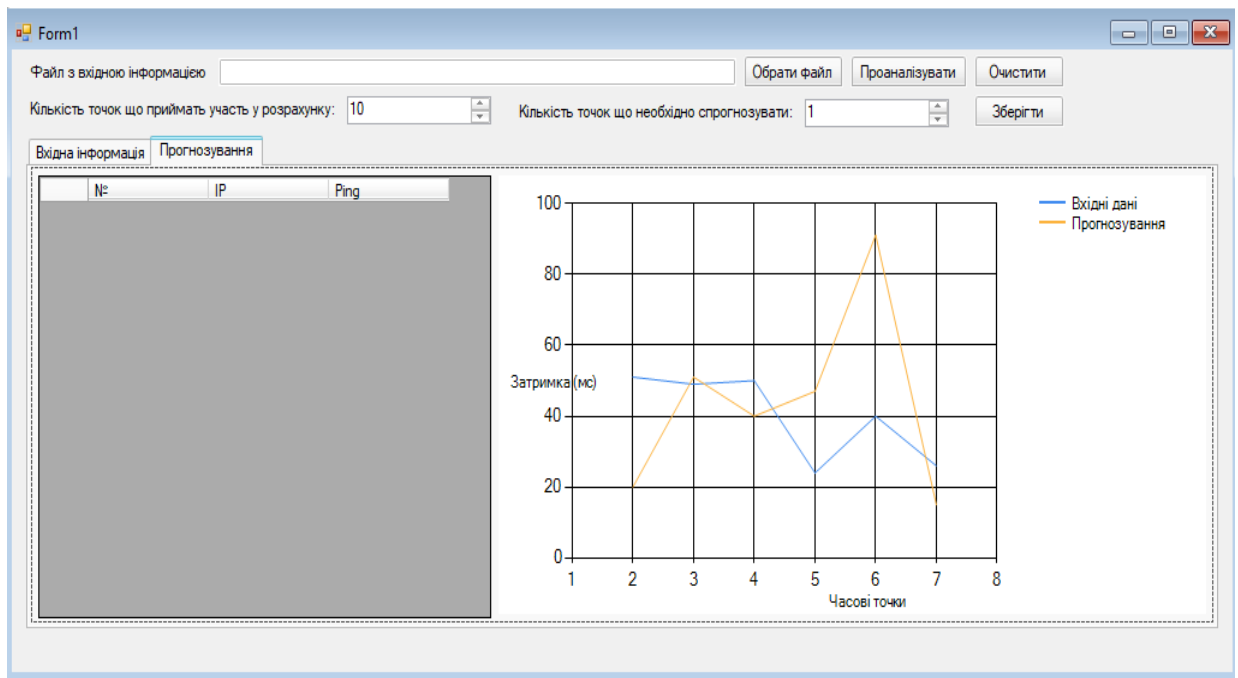


Рисунок 8.2 – Екранна форма екрана

На рис. 8.3 зображена головна форма після натискання кнопки «Обрати файл», при якій відкривається вікно вибору файлу для подальшої роботи програмного продукту.

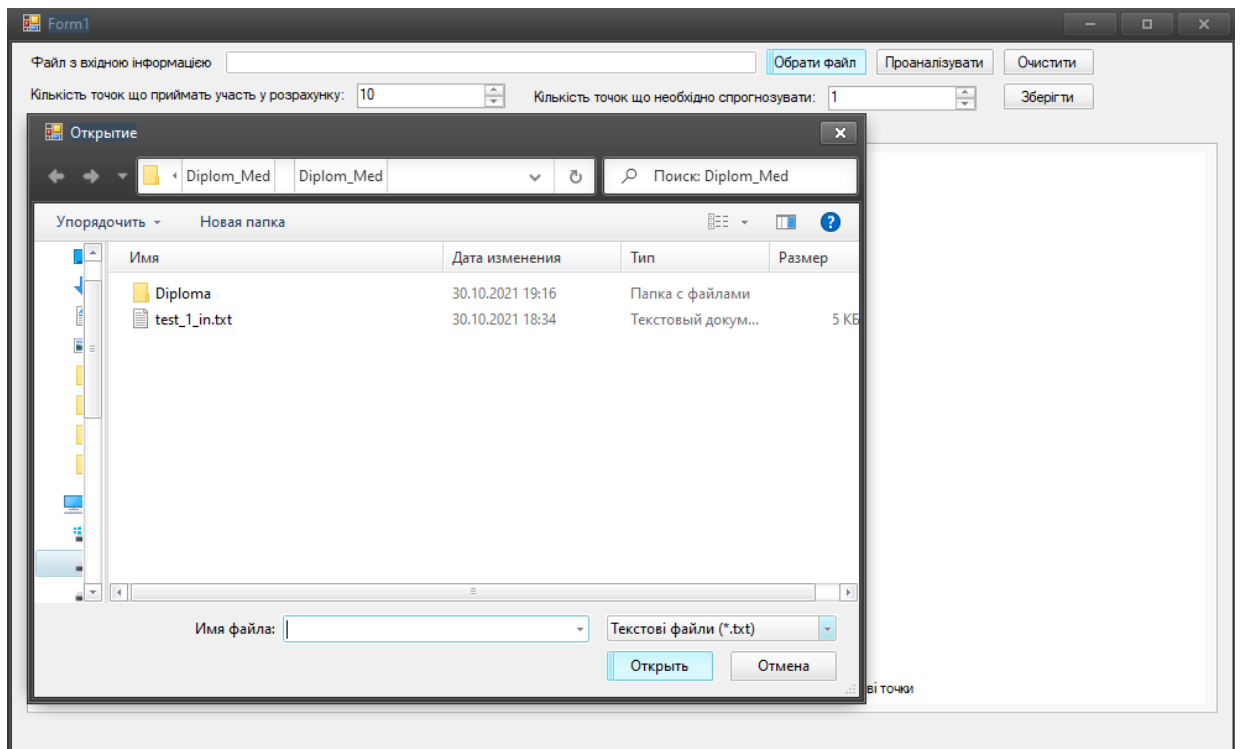


Рисунок 8.3 – Екранна форма вибору файлу з даними

На рис. 8.4 зображено вікно головної форми, після вибору файлу. Яке відображує табличні вхідні дані з файлу та побудований за даними графік.

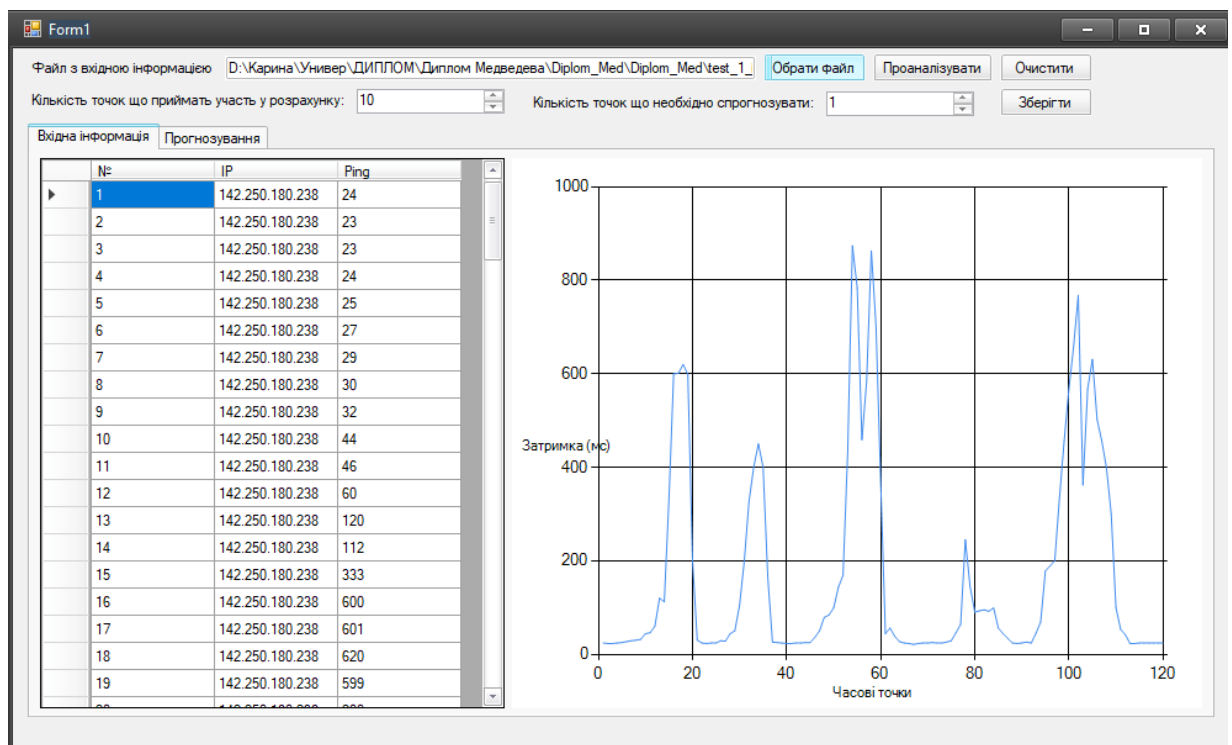


Рисунок 8.4 – Екранна форма головної форми з вхідними даними

На рис. 8.5 зображено вікно головної форми, у якому вибрано точки, які приймають участь у розрахунку.

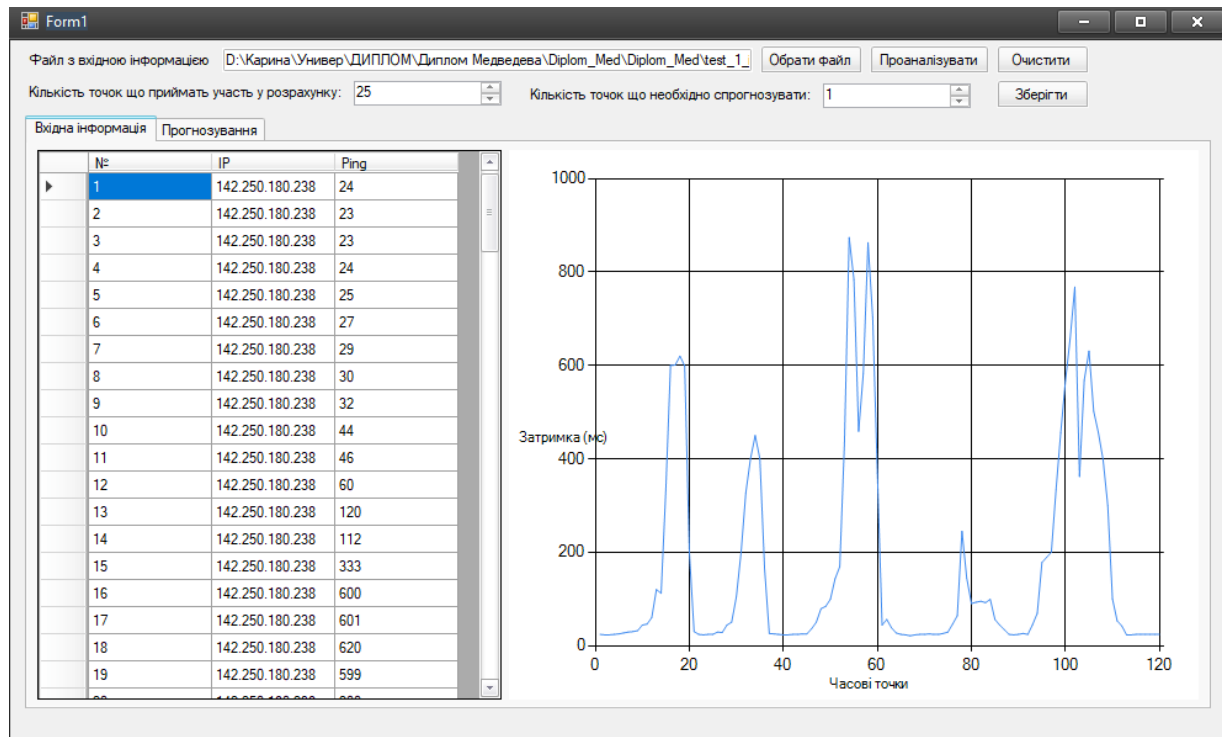


Рисунок 8.5 – Екранна форма з точками, які приймають участь у розрахунку

На рис. 8.6 зображено вікно головної форми програмного продукту, після того як вибрали кількість точок, які будуть приймати участь у прогнозуванні.

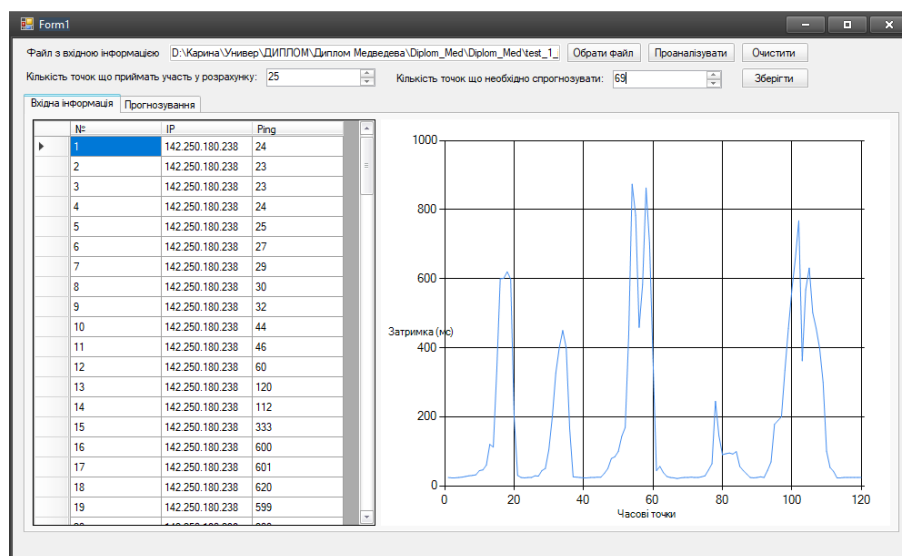


Рисунок 8.6 – Екранна форма з точками, які будуть приймати участь у прогнозуванні

На рис. 8.7 зображено вікно головної форми після натискання кнопки «Проаналізувати», на якій видно як змінився графік прогнозування навантаженості мережевої системи.

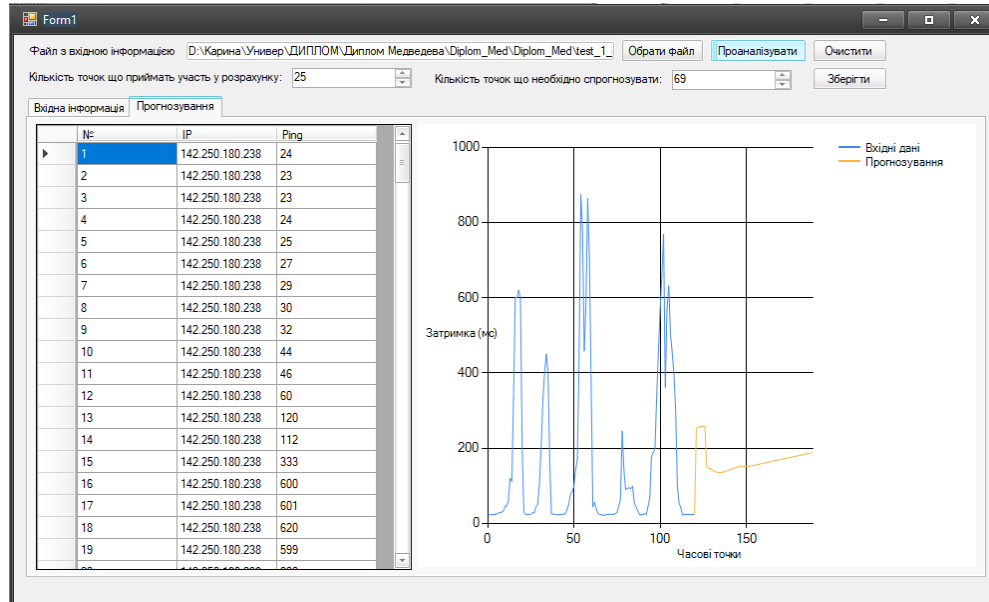


Рисунок 8.7 – Екранна форма після натискання кнопки «Проаналізувати»

На рисунку 8.9 зображено таблицю головної форми, після проаналізованих даних з точками, які брали участь у прогнозуванні.

Вхідна інформація		Прогнозування	
№	IP	Ping	
116	142.250.180.238	24	
117	142.250.180.238	24	
118	142.250.180.238	24	
119	142.250.180.238	24	
120	142.250.180.238	24	
121	Прогноз	253	
122	Прогноз	255	
123	Прогноз	257	
124	Прогноз	258	
125	Прогноз	258	
126	Прогноз	257	
127	Прогноз	151	
128	Прогноз	147	
129	Прогноз	146	
130	Прогноз	143	
131	Прогноз	140	
132	Прогноз	138	
133	Прогноз	136	
134	Прогноз	135	

Рисунок 8.9 – Екранна форма з таблицею прогнозованих точок

На рис. 8.10 зображено, як змінився графік після натискання кнопки «Проаналізувати», на якому видно як впливають спрогнозовані точки.

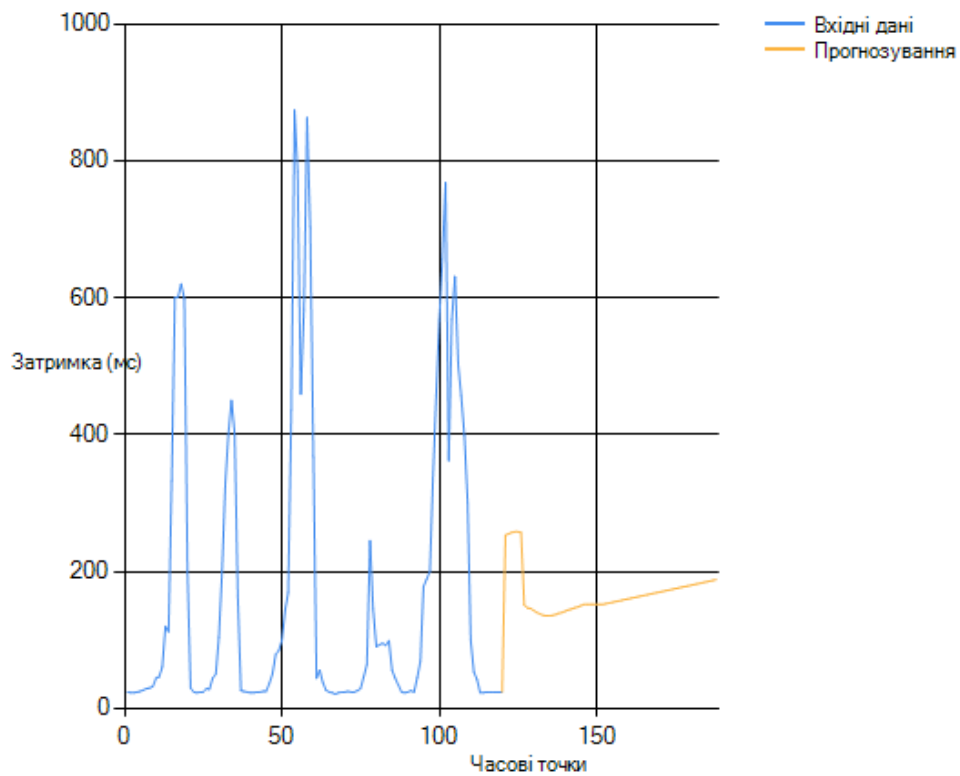


Рисунок 8.10 – Графік після натискання кнопки «Проаналізувати»

На рис. 8.11 зображено головну форму програмного засобу, після натискання кнопки «Зберегти».

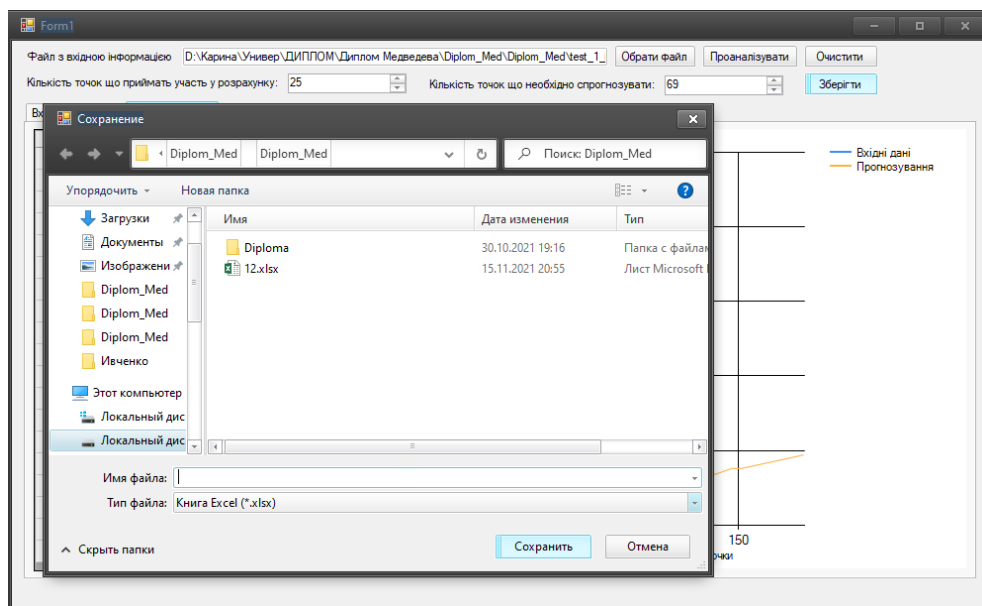


Рисунок 8.11 – Екранна форма вибору зберігання файлу

На рис. 8.12 зображено вікно успішного збереження файлу.

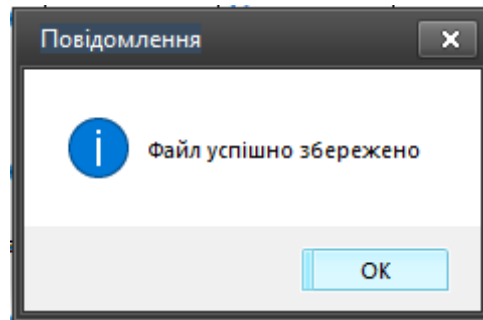


Рисунок 8.12 – Екранна форма успішного збереження файлу

На рис. 8.13 зображено вікно головної форми після натискання кнопки «Очистити».

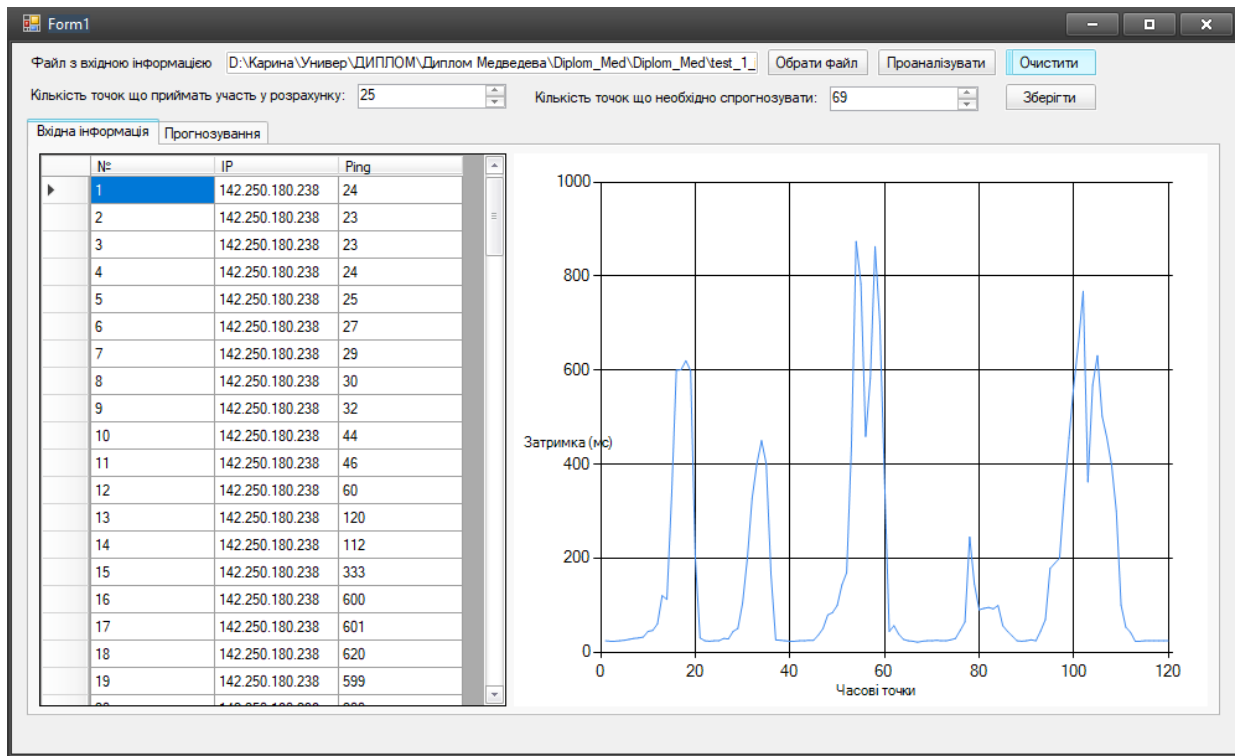


Рисунок 8.13 – Головна форма, після натискання кнопки «Очистити»

На рис. 8.14 зображено, як змінилось вкладка «Прогнозування» після натискання кнопки «Очистити».

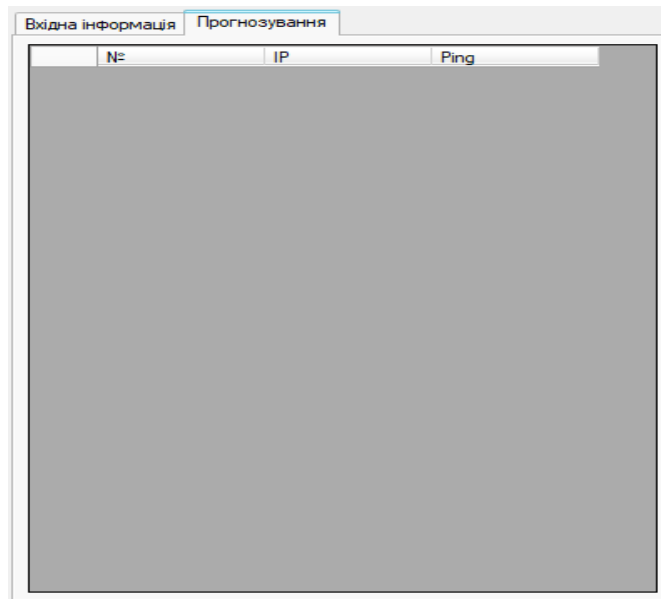


Рисунок 8.14 – Екранна форма вкладці «Прогнозування»

На рис. 8.15 зображено, як змінився графік після натискання кнопки «Очистити».

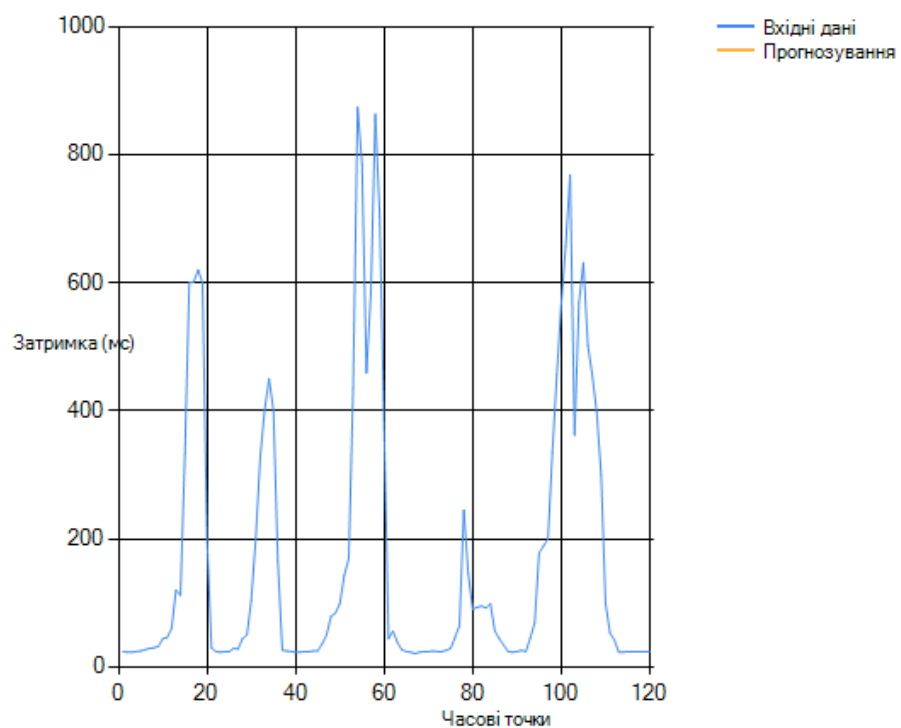


Рисунок 8.15 – Графік, після натискання кнопки «Очистити»

9 ПОРЯДОК РАБОТИ З ПРОГРАМОЮ

Технологію роботи з програмою і порядок виконання самої програми:

1. Додавання у систему файлу з вхідними даними (2-5 хв.).
2. Зчитування файлу (1 хв.).
3. Формування графіку на основі даних (1-2 хв.).
4. Прогнозування даних, на основі вхідного файлу (5-10 хв.).
5. Формування графіку, на основі прогнозованих даних (2-3 хв.).
6. Збереження файлу з прогнозованими даними (5-6 хв.).

10 ПОВІДОМЛЕННЯ КОРИСТУВАЧУ

Контролю коректності даних, що вводяться користувачем, необхідно приділяти чималу увагу, оскільки необроблені помилки, що виникають при неправильному введенні даних, приводять до помилок в роботі програми.

При проектуванні програми були усунені деякі помилки, повідомлення про які приведено на рисунках 10.1 – 10.2.

За відсутності назви документу, який необхідно зберегти.

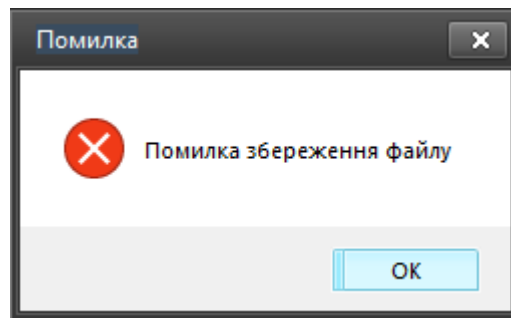


Рисунок 10.1 – Перевірка на назву файлу

Ця помилка з'являється при спробі відкрити файл, який вже був відкритий раніше.

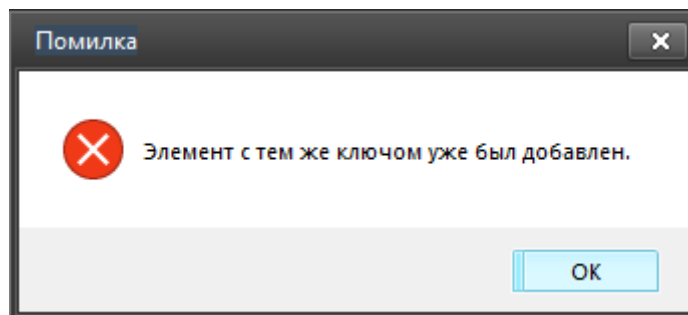


Рисунок 10.2 – Перевірка повторно відкриття файлу

ЗАТВЕРДЖЕНИЙ

1116130.01204-01 ІЗ 01-ЛЗ

ДОСЛІДЖЕННЯ ЧАСОВИХ РЯДІВ НАВАНТАЖЕНОСТІ МЕРЕЖЕВИХ
СИСТЕМ

Керівництво користувача.

Керівництво адміністратора мережі

1116130.01204-01 ІЗ 01

Листів 15

АНОТАЦІЯ

Документ 1116130.01204-01 ІЗ 01 «Дослідження часових рядів навантаженості мережевих систем. Керівництво користувача, Керівництво адміністратора мережі» є складовою частиною документації на програму, що призначена для роботи користувача програмним додатком «Дослідження часових рядів навантаженості мережевих систем», та входить до складу документації на проект.

В документі наведені функціональне призначення програми, опис логічної структури, технічні засоби, інструкція до продукту та повідомлення користувачеві. Програма реалізована на мові високого рівня C# у програмному середовищі Visual Studio. Об'єм пам'яті що займає програмний комплекс складає 4Мб. Конфігурація комп'ютера стандартна. Програма функціонує в середовищі MS Windows 7/8/8.1/10.

ЗМІСТ

1. Вступ.....	4
2. Функціональне призначення та умови застосування	5
3. Опис логічної структури	6
3.1 Використані методи	6
3.2 Структура програми з описом функції.....	6
4. Технічні засоби, що використовуються	7
5. Керівництво користувача.....	Ошибка! Закладка не определена.
6. Повідомлення користувачеві	15

ВСТУП

Розроблена програма має назву «Прогнозування часових рядів навантаженості мережевих систем».

Програма розрахована на функціонування в операційних системах MS Windows 7/8/8.1/10.

Програма реалізована на мові C# у програмному середовищі Visual Studio. Об'єм пам'яті що займає програмний комплекс складає 4Мб. Конфігурація комп'ютера стандартна.

2 ФУНКЦІОНАЛЬНЕ ПРИЗНАЧЕННЯ ТА УМОВИ ЗАСТОСУВАННЯ

Функціональним призначенням програмного продукту є надання можливостей користувачу спрогнозувати та визначити навантаженість мережевої системи у певному сегменті мережі.

Програмний комплекс дозволить звільнити користувачів від рутинної роботи налагодження системи, якщо вони вже зможуть його спрогнозувати та зможуть застосувати засоби до вирішення проблеми.

3 ОПИС ЛОГІЧНОЇ СТРУКТУРИ

3.1 Використані методи

В ході розробки програми були використані наступні методи та методології:

1. Для локального (тимчасового) зберігання інформації зі сторони клієнтської частини, програмного продукту, використовується ієрархічна система класів, кожен клас містить в собі відповідну інформацію з таблиці що знаходиться на сервері.

2. Програмний продукт розроблено згідно стандарту «MVC».

3.2 Структура програми з описом функції

Програма складається з трьох модулів. Опишемо призначення кожного модуля програми:

Form1.cs – модуль з прописом прототипів функцій інтерфейсної частини програмного комплексу.

MainControl.cs – модуль з прописом прототипів функцій частини програмного комплексу.

Diploma.cs – виконавчий модуль запуску програми.

4 ТЕХНІЧНІ ЗАСОБИ, ЩО ВИКОРИСТОВУЮТЬСЯ

Мінімальна конфігурація, що забезпечить нормальну експлуатацію програмного продукту:

- процесор: Intel Core 2 Duo 2.4 GHz або краще;
- оперативна Пам'ять: 2 GB DDR2 або краще;
- операційна система: Windows 7 або краще;
- вільний дисковий простір 50 Mb;
- наявність CD/DVD приводу або USB роз'ємну для встановлення необхідного ПЗ;
- підключення до локальної або глобальної мережі: 10mb/s або краще;
- монітор з роздільною здатністю екрану 1024x768;
- клавіатура;
- маніпулятор «миша».

5 КЕРІВНИЦТВО КОРИСТУВАЧА

На рисунку 5.1 зображено ярлик програми, на який потрібно клацнути два рази лівою кнопкою миші.

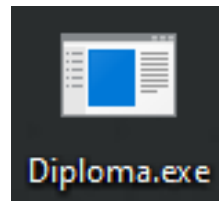


Рисунок 5.1 – Ярлик програми

Після відкриття з'являється головна форма програми, що включає в себе усе необхідні для роботи з програмним засобом, такі як, поля вводу посилання на файл з вхідними даними, поле для вибору кількості точок, що будуть приймати участі у прогнозуванні, вікно з отриманим графіком, вікна з представленими даними у табличному вигляді зображено на рис. 5.2.

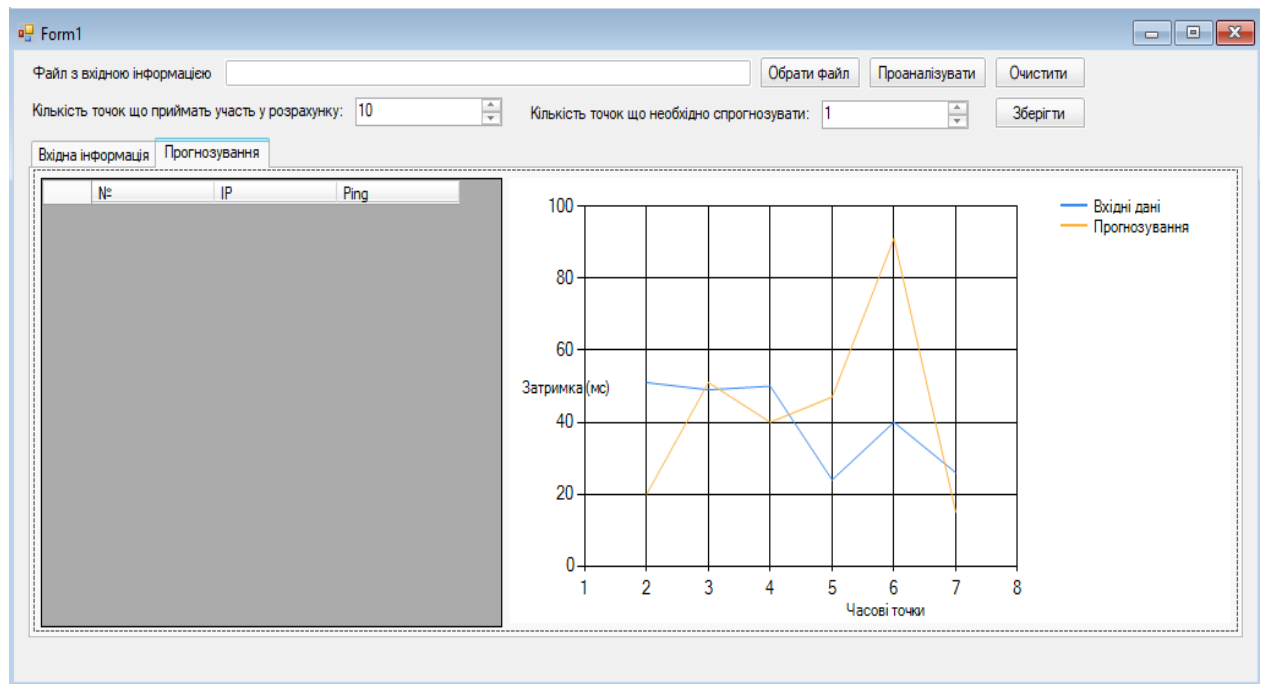


Рисунок 5.2 – Головна форма

Після відкриття головної форми потрібно вибрати файл з вхідними даними, що зображено на рис. 5.3.

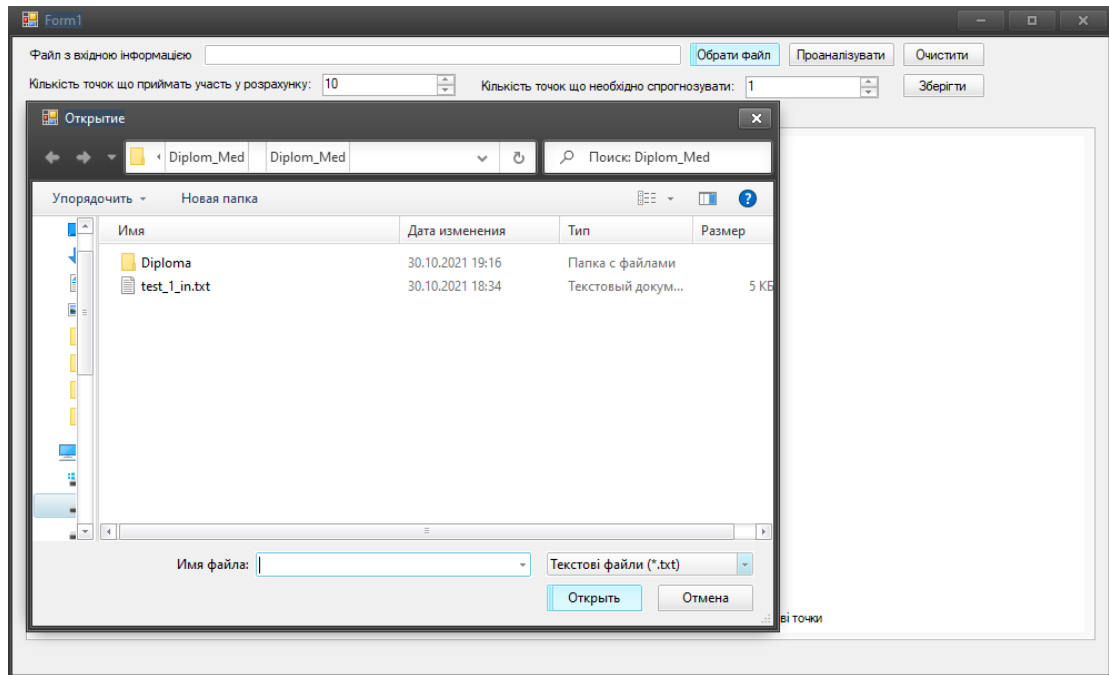


Рисунок 5.3 – Екранна форма вибору файлу з даними

На рис. 5.4 зображено вікно головної форми, після вибору файлу. Якє відображує табличні вхідні дані з файлу та побудований за даними графік.

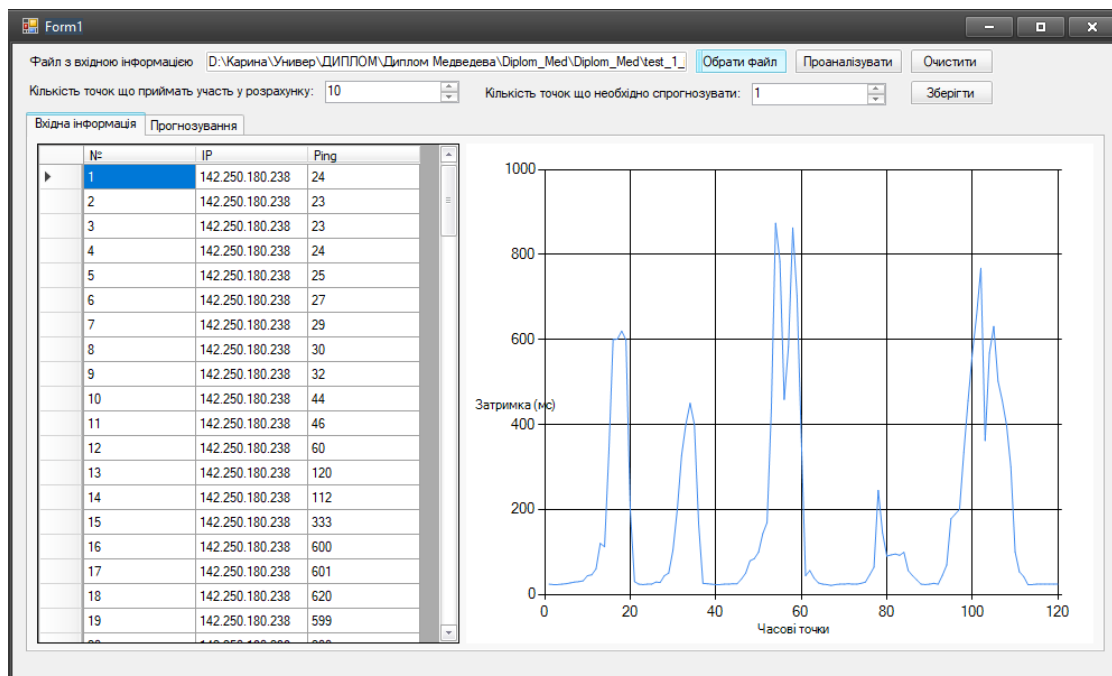


Рисунок 5.4 – Екранна форма головної форми з вхідними даними

Після вибору вхідного файлу потрібно вказати точки, які приймають участь у розрахунку, що зображено на рис. 5.5

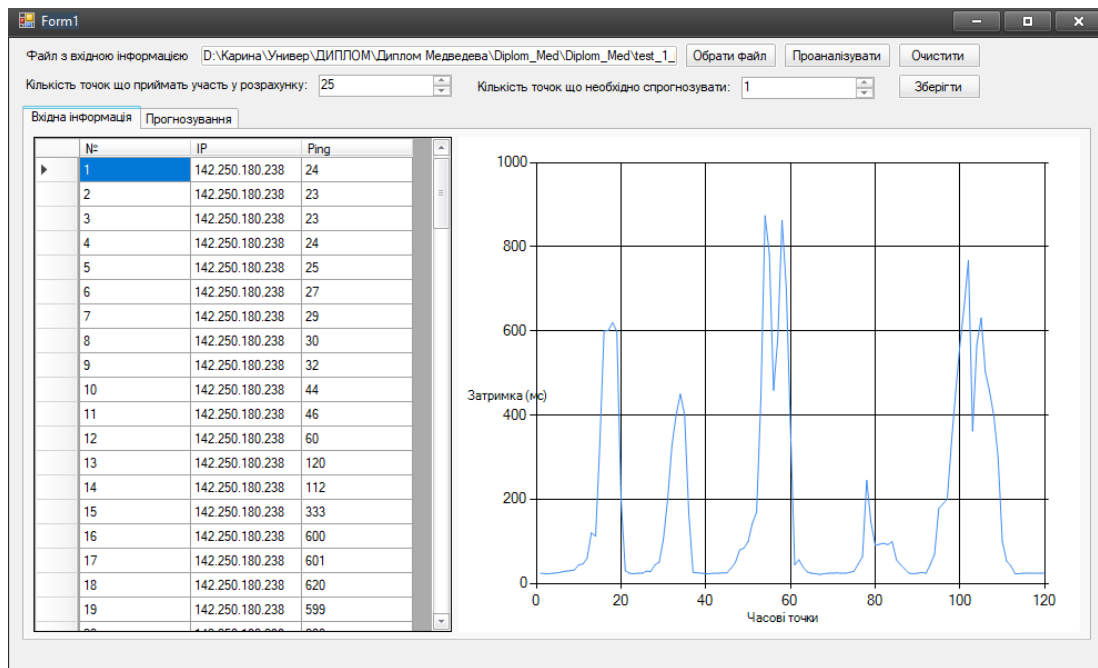


Рисунок 5.5 – Екранна форма з точками, які приймають участь у розрахунку

На рис. 5.6 зображено вікно головної форми програмного продукту, після того як вибрали кількість точок, які будуть приймати участь у прогнозуванні.

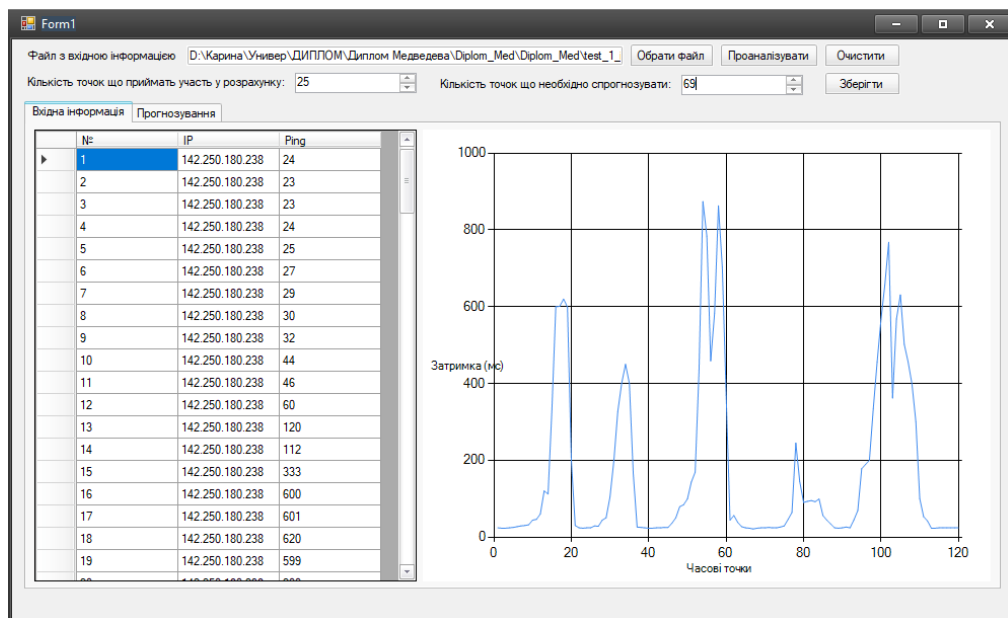


Рисунок 5.6 – Екранна форма з точками, які будуть приймати участь у прогнозуванні

Після вибору точок потрібно натиснути кнопку «Проаналізувати», як на наведено на рис. 5.7.

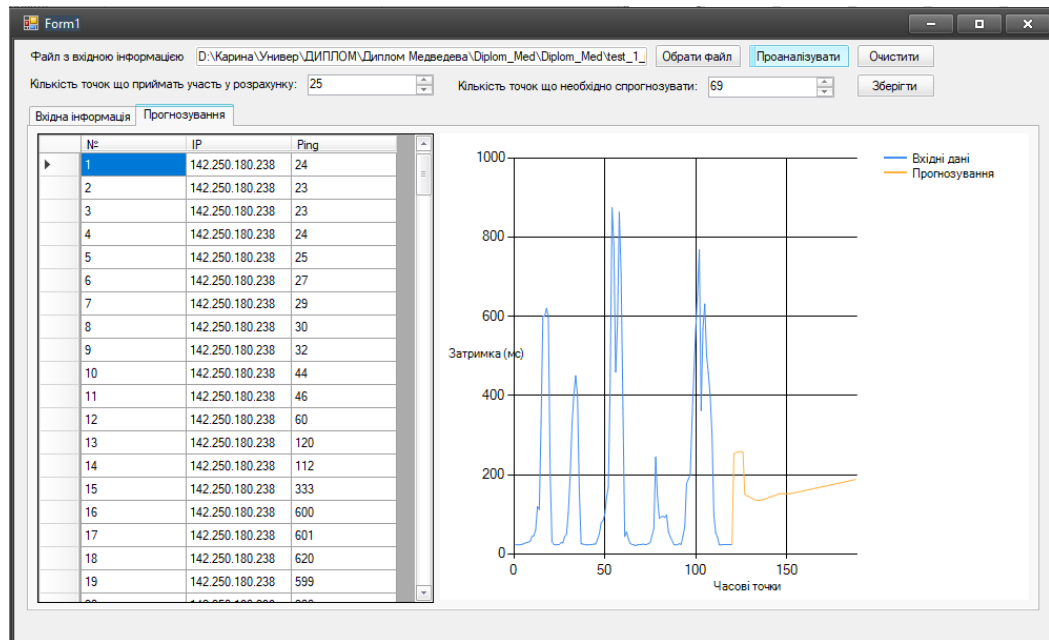


Рисунок 5.7 – Екранна форма після натискання кнопки «Проаналізувати»

На рис. 5.8 зображено таблицю головної форми, після проаналізованих даних з точками, які брали участь у прогнозуванні.

Вхідна інформація			Прогнозування
№	IP	Ping	
116	142.250.180.238	24	
117	142.250.180.238	24	
118	142.250.180.238	24	
119	142.250.180.238	24	
120	142.250.180.238	24	
121	Прогноз	253	
122	Прогноз	255	
123	Прогноз	257	
124	Прогноз	258	
125	Прогноз	258	
126	Прогноз	257	
127	Прогноз	151	
128	Прогноз	147	
129	Прогноз	146	
130	Прогноз	143	
131	Прогноз	140	
132	Прогноз	138	
133	Прогноз	136	
134	Прогноз	135	

Рисунок 5.8 – Екранна форма з таблицею прогнозованих точок

Після На рис. 5.9 зображено, як змінився графік після натискання кнопки «Проаналізувати», на якому видно як впливають спрогнозовані точки.

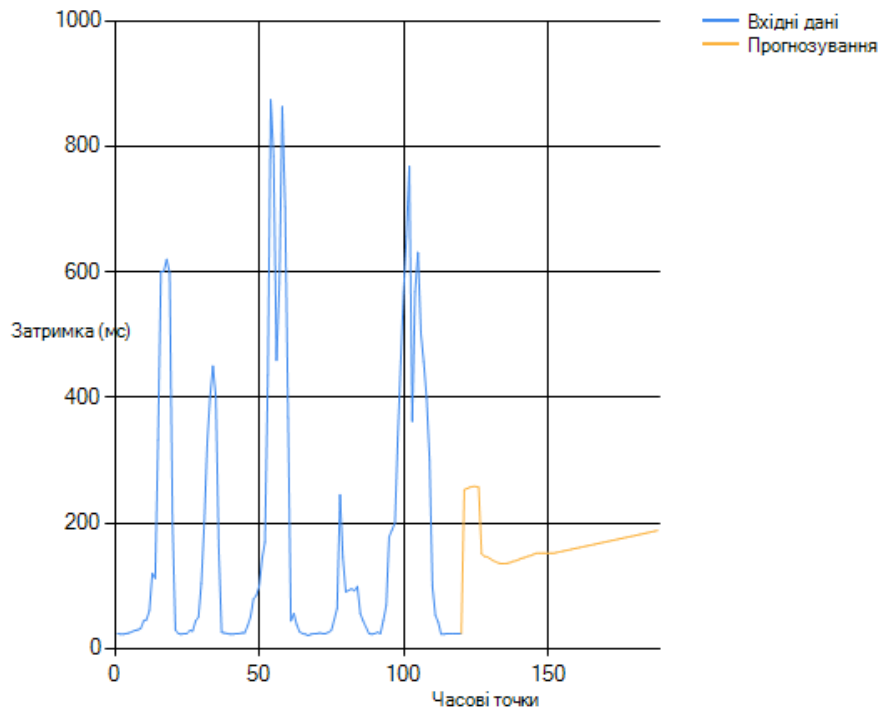


Рисунок 5.9 – Екранна форма після натискання кнопки «Проаналізувати»

Після перегляду графіку після аналізування, потрібно натиснути кнопку «Зберегти», як зображено на рис. 5.10 .

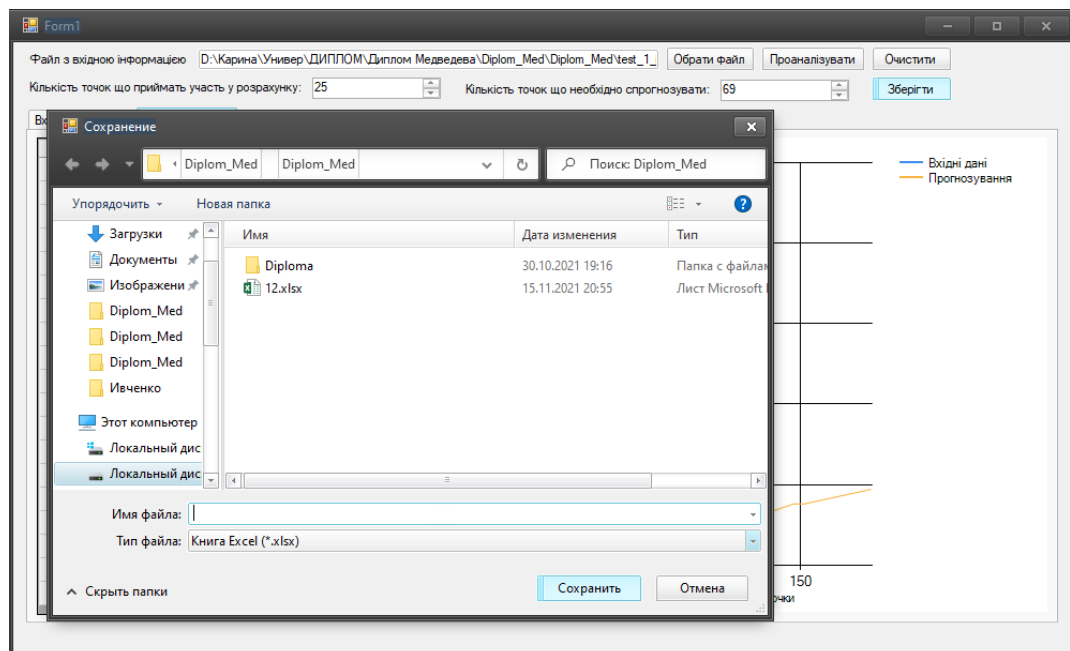


Рисунок 5.10 – Екранна форма вибору зберігання файлу

На рис. 5.11 зображено вікно успішного збереження файлу.

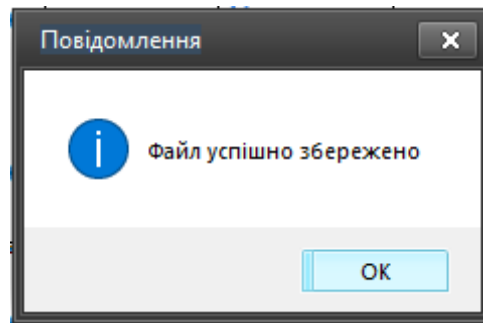


Рисунок 5.11 – Екранна форма успішного збереження файлу

На рис. 5.12 зображено вікно головної форми після натискання кнопки «Очистити».

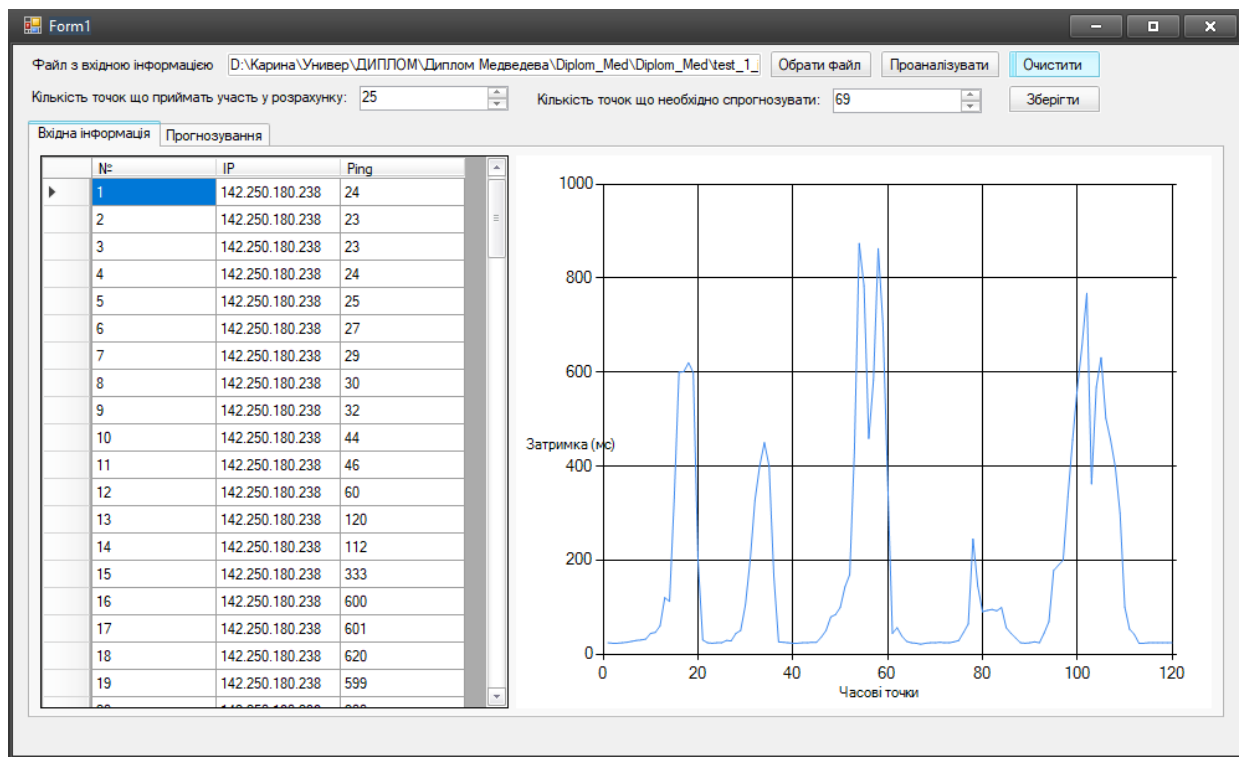


Рисунок 5.12 – Головна форма, після натискання кнопки «Очистити»

На рис. 5.13 зображено, як змінилось вкладка «Прогнозування» після натискання кнопки «Очистити».

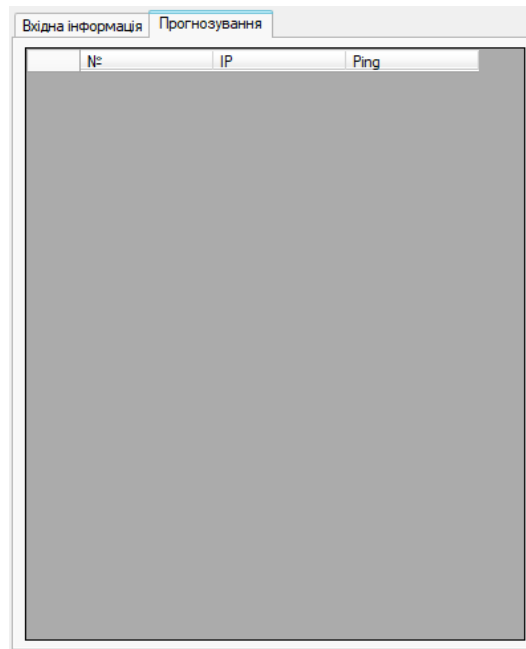


Рисунок 5.13 – Вкладка «Прогнозування»

На рис. 5.14 зображено, як змінився графік після натискання кнопки «Очистити».

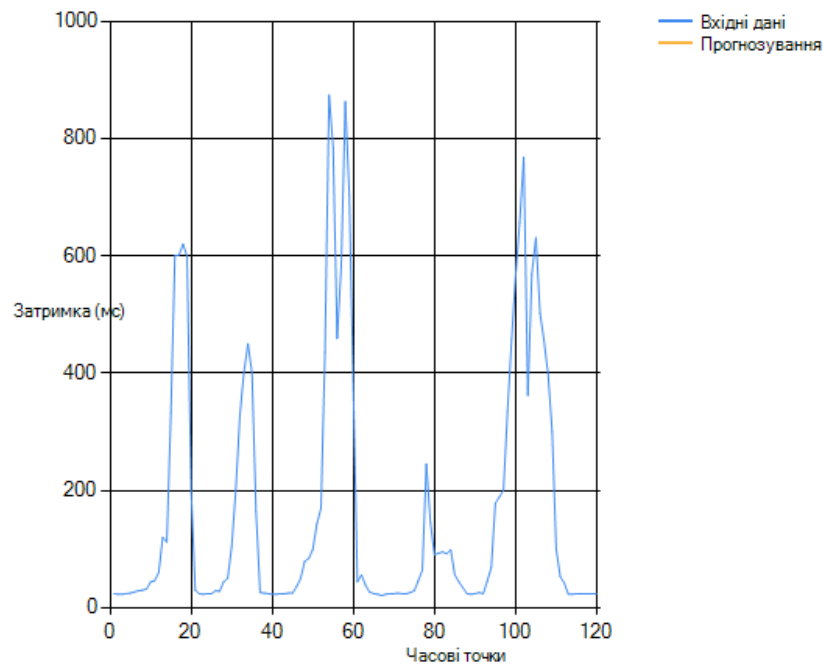


Рисунок 5.14 – Графік, після натискання кнопки «Очистити»

Після збереження файлу потрібно натиснути хрестик у правому верхньому куті для завершення програми.

6 ПОВІДОМЛЕННЯ КОРИСТУВАЧЕВІ

Контролю коректності даних, що вводяться користувачем, необхідно приділяти чималу увагу, оскільки необроблені помилки, що виникають при неправильному введенні даних, приводять до помилок в роботі програми.

При проектуванні програми були усунені деякі помилки, повідомлення про які приведено на рис. 6.1 – 6.2.

За відсутності назви документу, який необхідно зберегти.

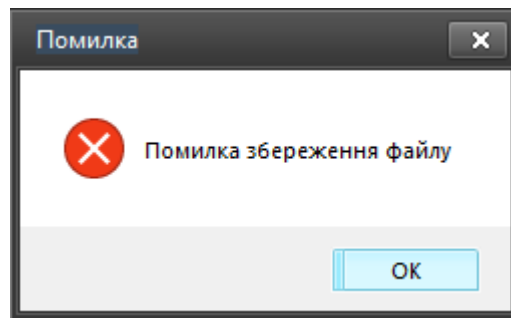


Рисунок 6.1 – Перевірка на назву файлу

Ця помилка з'являється при спробі відкрити файл, який вже був відкритий раніше.

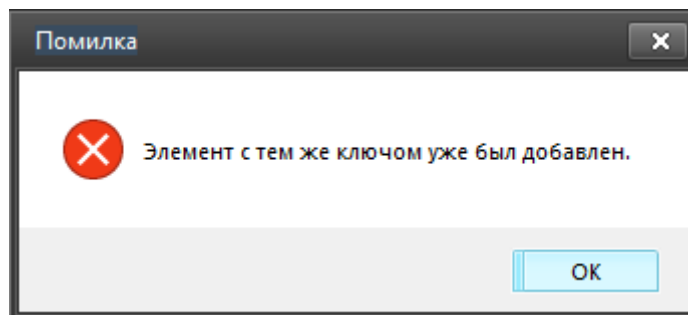


Рисунок 6.2 – Перевірка відкриття файлу

Міністерство освіти і науки України

Дніпровський національний університет
залізничного транспорту ім. акад. В. Лазаряна

Східний науковий центр транспортної академії наук



ПКТБ
ІТ



TEMPUS: CITISET & SEREIN & CRENG

ТЕЗИ

**XIV Міжнародної науково-практичної конференції
«СУЧАСНІ ІНФОРМАЦІЙНІ ТА КОМУНІКАЦІЙНІ
ТЕХНОЛОГІЇ НА ТРАНСПОРТІ, В ПРОМИСЛОВОСТІ
ТА ОСВІТІ»**

ABSTRACTS

**of the XIV International Conference
«MODERN INFORMATION AND COMMUNICATION
TECHNOLOGIES ON A TRANSPORT, IN INDUSTRY
AND EDUCATION»**

ТЕЗИСЫ

**XIV Международной научно-практической конференции
«СОВРЕМЕННЫЕ ИНФОРМАЦИОННЫЕ И
КОМУНИКАЦИОННЫЕ ТЕХНОЛОГИИ
НА ТРАНСПОРТЕ, В ПРОМЫШЛЕННОСТИ
И ОБРАЗОВАНИИ»**

15.12.2020 – 16.12.2020

**Дніпро
2020**

Вибір параметрів номінального режиму електрорухомого складу з асинхронним тяговим приводом.....	43
Гетьман Г. К., Васильєв В.Є., Дніпровський національний університет залізничного транспорту ім. акад. В. Лазаряна, Україна	
Дослідження та розробка мікропроцесорної системи управління штучною екосистемою. Апаратна частина та програма керування.....	44
Гирька А. О., Дзюба В.В., Дніпровський національний університет залізничного транспорту імені академіка В.Лазаряна, Україна	
Загальні підходи до алгоритмів оцифрування піксельної графіки у криві.....	45
Горбова О.В., Борець Р.С., Дніпровський національний університет залізничного транспорту імені академіка В.Лазаряна, Україна	
Дослідження і оптимізація автомобільних потоків засобами імітаційного моделювання.....	46
Горбова О.В., Мерзлий О.Д., Дніпровський національний університет залізничного транспорту імені академіка В.Лазаряна, Україна	
Дослідження часових рядів навантаженості мережевих систем.....	47
Горбова О.В., Михайлова Т.Ф., Медведєва К.В., Дніпровський національний університет залізничного транспорту імені академіка В.Лазаряна, Україна	
Методи поетапного моделювання складних процесів.....	48
Горбова О.В., Муркович М.С., Дніпровський національний університет залізничного транспорту імені академіка В. Лазаряна, Україна	
Загальні підходи до дослідження наслідків використання патернів в побудові архітектури крос-платформних додатків під Android і IOS.....	49
Горбова О.В., Сирота О.А., Дніпровський національний університет залізничного транспорту імені академіка В.Лазаряна, Україна	
Аналіз та шляхи вдосконалення робочого місця машиніста локомотива.....	50
Горобченко О. М., Неведров О. В., Державний університет інфраструктури та технологій, Україна	
Уточнення моделі для оцінювання похибки вимірювання ходового опору руху вагонів коліями сортувальних гірок.....	51
Жуковицький І. В., Устенко А. Б., Дзюба В. В., Дніпровський національний університет залізничного транспорту ім. академіка В. Лазаряна, Україна	
Системы мониторинга безопасного потребления газа в жилых домах для умного дома и умного города.....	52
Иашвили Н.Г., Грузинский Технический Университет, Тбилиси, Грузия	
Дослідження та розробка комплексу генерації випадкових та псевдовипадкових чисел.....	53
Іванчак О. С., Остапєць Д. О., Дніпровський національний університет залізничного транспорту імені академіка В.Лазаряна, Україна	
Дослідження та розробка мікропроцесорної системи управління штучною екосистемою. Серверна та веб версії програмної частини.....	54
Кирпа Д.Р., Дзюба В.В., Дніпровський національний університет залізничного транспорту імені академіка В.Лазаряна, Україна	

Дослідження часових рядів навантаженості мережевих систем

Горбова О.В., Михайлова Т.Ф., Медведєва К.В.

Дніпровський національний університет залізничного
транспорту імені академіка В.Лазаряна,
(Україна)

Останнім часом з'являється безліч провайдерів, які пропонують високошвидкісне підключення до мережі Інтернет. Це обумовлюється тим, що дані послуги актуальні і затребувані. З кожним днем кількість абонентів, підключених до Інтернету, росте, відповідно зростає і навантаження на сервери. Вимоги та додатки користувачів можуть перевищувати трафік, це може порушити пропускну здатність каналів зв'язку.

Все це призводить до виникнення перевантажень на ділянках мережі, а отже до порушення цілісності, виникнення загроз втрати даних, помилок, відмов у обслуговуванні та сповільненою роботі в мережі всіх абонентів.

Від якості прогнозу завантаженості мереж залежить багато чинників. На даний момент існує величезна кількість алгоритмів прогнозування, результат яких безпосередньо залежить від вхідних даних, тобто вибірок, сформованих з часових рядів. Для підвищення точності прогнозів перенавантаження важливо розуміти, що різні моделі прогнозування підходять для різних часових рядів з різними характеристиками.

Часовий ряд – це зібраний в різні моменти часу статистичний матеріал про значення будь-яких параметрів досліджуваного процесу. Кожна одиниця статистичного матеріалу називається виміром або відліком, також допустимо називати його рівнем на вказаний з ним момент часу. Часові ряди мають величезне значення для виявлення і вивчення складних закономірностей в розвитку явищ життєдіяльності. Аналіз часових рядів - сукупність математико-статистичних методів аналізу, призначених для виявлення структури часових рядів і для їх прогнозування.

Розробка механізму прогнозування навантаженості мережевих систем є актуальною, важливою задачею та частиною мір по забезпеченню безпечної експлуатації комп'ютерної мережі. Прогнозування інтенсивності трафіку дає можливість заздалегідь вживати необхідних заходів по попередженню наслідків. Досліджуються причини виникнення перевантажень в мережі, наслідком яких є зниження рівня безпеки передачі даних.

Мета поставленої задачі полягає у дослідженні методів прогнозування часових рядів та виявлення найбільш придатних з них для обробки даних, у розробці часових рядів для прогнозування перевантаження в мережевих системах. Також буде проведено аналіз часових рядів, а саме: побудова та дослідження графіка, побудова динамічної моделі часових рядів та прогнозування майбутніх значень. Буде розглянуто та проаналізовано методи ковзних середніх та найменших квадратів, як підходящих для обробки даних.

Таким чином, після проведення аналізу існуючих методів моніторинга та обзору базових засобів моніторинга мережевих систем отримаємо алгоритм дослідження, механізм прогнозування навантаженості мережевих систем за допомогою часових рядів та розроблено метод прогнозування перенавантажень в мережевих систем на основі аналізу часових рядів. Також були проведенні аналіз методів прогнозування часових рядів та виявленні найбільш підходящі для обробки отриманих експериментальних даних.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

АТ «УКРАЇНСЬКА ЗАЛІЗНИЦЯ»

ДНІПРОВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ЗАЛІЗНИЧНОГО
ТРАНСПОРТУ ІМЕНІ АКАДЕМІКА В. ЛАЗАРЯНА

УКРАЇНСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ ЗАЛІЗНИЧНОГО ТРАНСПОРТУ

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФРАСТРУКТУРИ ТА ТЕХНОЛОГІЙ

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ АВТОМОБІЛЬНО-ДОРОЖНІЙ УНІВЕРСИТЕТ

**ІНФОРМАЦІЙНО-ТЕЛЕКОМУНІКАЦІЙНІ
ТЕХНОЛОГІЇ ТА КОМП'ЮТЕРНЕ
МОДЕЛЮВАННЯ**

ЗБІРНИК ТЕЗ ДОПОВІДЕЙ

**81 Всеукраїнської науково-технічної конференції
молодих учених, магістрантів та студентів**

**«НАУКА І СТАЛИЙ РОЗВИТОК
ТРАНСПОРТУ»**

28 жовтня 2021 року

**INFORMATION-TELECOMMUNICATION
TECHNOLOGY TA COMPUTER MODELING**

CONFERENCE PROCEEDINGS

**81th all Ukrainian Scientific and Technical Conference
of young scientists, masters and students**

**“SCIENCE AND SUSTAINABLE DEVELOPMENT
OF TRANSPORT”**

October 28, 2021

ЗМІСТ

Дослідження стохастичних та стохастико-детермінованих алгоритмів сортування.....	4
Кластеризація текстів за приналежністю до автора на основі словнику атрибутів.....	4
Процедури вибору операторів для упорядкування недетермінованих послідовностей замовлень на основі нейронних мереж.....	5
Конструктивні просторові перетворення двовимірних фракталів.....	6
Дослідження структурної схожості об'єктно-орієнтованих програм.....	7
Визначення відповідності тексту програми графічному представленню алгоритму.....	8
Дослідження характеристик ієрархічного краудсорсингу в розробці програм.....	9
Використання методів Монте-Карло для визначення очікуваної суми.....	10
Дослідження і моделювання автотранспортних потоків.....	11
Дослідження часових рядів навантаженості мережевих систем.....	12
Дослідження наслідків використання патернів в побудові архітектури крос-платформних додатків під Android і IOS.....	13
Методи поетапного моделювання складних процесів.....	14
Рефакторинг SQL запитів.....	15
Traction supply systems and their influence on the railway automatics devices.....	16
Electromagnetic influence on the digital communication devices of railway.....	17
Improving the operational parameters and characteristics of Bi directional power converters intended for railway transport application.....	18
Battery management systems in the electromagnetic influence of traction supply railway system.....	19
Дослідження та розробка засобів демонстрації стеганографічного захисту інформації та стегоаналізу.....	20
Стеганографічний захист інформації з використанням текстових контейнерів.....	20
Дослідження та розробка засобів генерації випадкових чисел.....	21
Стеганографічний захист інформації з використанням графічних контейнерів.....	22
Визначення категорії мережевих атак на комп'ютерну мережу з використанням нейронечіткої мережі.....	23
Створення самоорганізуючої карти для визначення класів мережевих атак категорії Probe.....	24
Дослідження та розробка засобів вивчення решіткового кодування.....	25
До застосування методів штучного інтелекту для управління швидкістю скочування відчепів на сортувальних гірках.....	26

Дослідження часових рядів навантаженості мережевих систем

Горбова О.В., Михайлова Т.Ф., Медведєва К.В.

Дніпровський національний університет залізничного
транспорту імені академіка В.Лазаряна,
(Україна)

У мережі передачі даних і теорії організації черг, відбувається перевантаження мережі, коли зв'язок або вузол несуть так багато даних, що його якість обслуговування погіршується. Типові ефекти включають затримку організації черг, втрату пакета або блокування нових зв'язків. Наслідок останніх двох ефектів - то, що зростає збільшення пропонованого вантажу призводить або тільки до маленького збільшення мережевий пропускної здатності, або до фактичного скорочення мережевий пропускної здатності.

Перевантаження мережі в комп'ютерні мережі і теорія масового обслуговування - зниження якості обслуговування, що відбувається, коли мережевий вузол або лінія зв'язку несе більше інформації, ніж він може впоратися. типові ефекти включають затримку в черзі, втрати пакетів або блокування нових з'єднань. наслідком перевантаження полягає в тому, що поступове збільшення в пропонованій навантаженні призводить або до невеликого зростання або навіть зниження пропускної здатності мережі. Мережеві протоколи, які використовують агресивні повторні передачі, щоб компенсувати втрати пакетів через перевантаження, може збільшити перевантаження навіть після початкового завантаження було скорочено до рівня, який зазвичай не викликає перевантаження мережі.

Завдання прогнозування часового ряду вирішується на основі створення моделі прогнозування, адекватно описує досліджуваний процес.

Аналіз часових рядів – сукупність математико-статистичних методів аналізу, призначених для виявлення структури часових рядів і для їх прогнозування. Сюди відносяться, зокрема, методи регресійного аналізу. Виявлення структури часового ряду необхідно для того, щоб побудувати математичну модель того явища, яке є джерелом аналізованого часового ряду. Прогноз майбутніх значень часового ряду використовується для ефективного прийняття рішень.

Розробка механізму прогнозування навантаженості мережевих систем є актуальною, важливою задачею та частиною мір по забезпеченню безпечної експлуатації комп'ютерної мережі. Прогнозування інтенсивності трафіку дає можливість заздалегідь вживати необхідних заходів попередження наслідків. Досліджуються причини виникнення перевантажень в мережі, наслідком яких є зниження рівня безпеки передачі даних.

Мета поставленої задачі полягає у дослідженні методів прогнозування часових рядів та виявлення найбільш придатних з них для обробки даних, у розробці часових рядів для прогнозування перевантаження в мережевих системах. Для досягнення мети потрібно виконати аналіз існуючих методів прогнозування часових рядів, побудувати динамічну модель часових рядів та провести аналіз прогнозування майбутніх значень на основі попередніх досліджень.

Після проведення аналізу існуючих методів моніторинга та обзору базових засобів моніторинга мережевих систем отримаємо алгоритм дослідження, механізм прогнозування навантаженості мережевих систем за допомогою часових рядів та розроблено метод прогнозування перенавантажень в мережевих систем на основі аналізу часових рядів. Також були проведенні аналіз методів прогнозування часових рядів та виявленні найбільш підходящі для обробки отриманих експериментальних даних.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДНІПРОВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ЗАЛІЗНИЧНОГО
ТРАНСПОРТУ ІМЕНІ АКАДЕМІКА В. ЛАЗАРЯНА

АТ «УКРАЇНСЬКА ЗАЛІЗНИЦЯ»

ПАТ «КРЮКІВСЬКИЙ ВАГОНБУДІВНИЙ ЗАВОД»

АТ «ДНІПРОПЕТРОВСЬКИЙ СТІЛОЧНИЙ ЗАВОД»

ТОВ «ЗАВОД РЕЙКОВИХ СКРІПЛЕНЬ»

INSTYTUT KOLEJNICTWA

КОРПОРАЦІЯ «ДЕТАЛЬ ВАГОН ГРУП»

**ПРОБЛЕМИ ТА ПЕРСПЕКТИВИ
РОЗВИТКУ ЗАЛІЗНИЧНОГО ТРАНСПОРТУ**

**МАТЕРІАЛИ
81 МІЖНАРОДНОЇ НАУКОВО-ПРАКТИЧНОЇ КОНФЕРЕНЦІЇ
22.04.2021–23.04.2021**

ПЕРСПЕКТИВИ ЦИФРОВІЗАЦІЇ СИСТЕМ КЕРУВАННЯ РУХОМ ПОЇЗДІВ НА ЗАЛІЗНИЧНОМУ ТРАНСПОРТІ УКРАЇНИ Мойсєєнко В. І., Каменєв О. Ю., Лапко А. О., Щєблїкіна О. В., Каменєва Н. В.	385
УПРАВЛІННЯ РЕЖИМАМИ РОБОТИ ЕЛЕКТРОТЯГОВИХ МЕРЕЖ МІСЬКОГО ЕЛЕКТРОТРАНСПОРТУ НА ОСНОВІ НЕЧІТКОГО ОПІСУ ЇХ СТАНУ Васенко В. О.	387
ДОСЛІДЖЕННЯ ЕЛЕКТРОМАГНІТНОЇ СУМІСНОСТІ РЕЙКОВИХ КЛІ З СИСТЕМОЮ ТЯГОВОГО ЕЛЕКТРОПОСТАЧАННЯ Сердюк Т. М., Сердюк К. М., Пушкарьов Є. О., Борякін А. О.	389
МОНІТОРИНГ ТЕМПЕРАТУРИ, ВОЛОГОСТІ ТА ВІБРАЦІЇ ЕЛЕМЕНТІВ СИЛОВИХ ПЕРЕТВОРЮВАЧІВ СИСТЕМ ЕЛЕКТРОТРАНСПОРТУ Горпинич О. В., Стружко І. С., Сердюк Т. М.	391
ДІАГНОСТУВАННЯ ПОТУЖНИХ ТРИФАЗНИХ АСИНХРОННИХ ДВИГУНІВ Сердюк Т. М., Сердюк К. М., Ботнаревская Р. В.	392
МЕТОД ДІАГНОСТУВАННЯ ДВИГУНІВ СТІЛОЧНИХ ПРИВОДІВ Сердюк Т. М., Сердюк К. М., Шозда І. В., Перельотов А. В., Карлюкова А. Ю.	394
ПРОГРАМНИЙ КОМПЛЕКС ДЛЯ АДАПТИВНОГО КЕРУВАННЯ СВІТЛОФОРНИМИ ОБ'ЄКТАМИ Панік Л. О., Фокша Л. В., Литвиненко К. В.	395
МОДЕЛЮВАННЯ СКЛАДНИХ ПРОЦЕСІВ ШЛЯХОМ ДЕКОМПОЗИЦІЇ ТА ПОЕТАПНОГО МОДЕЛЮВАННЯ ДЛЯ ДОСЛІДЖЕННЯ ЧАСОВОЇ ЕФЕКТИВНОСТІ Горбова О. В., Муркович М. С.	396
ДОСЛІДЖЕННЯ ЧАСОВИХ РЯДІВ НАВАНТАЖЕНОСТІ МЕРЕЖЕВИХ СИСТЕМ Горбова О. В., Медведєва К. В.	397
ІМІТАЦІЙНЕ МОДЕЛЮВАННЯ АВТОМОБІЛЬНИХ ПОТОКІВ Горбова О. В., Мерзлий О. Д.	398
PHISHING ATTACKS Domanska H., Ychorov O., Kulyk V., Troshin E.	399
ІМЕННИЙ ПОКАЖЧИК	401

Дослідження часових рядів навантаженості мережевих систем

Горбова О.В., Медведєва К.В.

Дніпровський національний університет залізничного
транспорту імені академіка В.Лазаряна,
(Україна)

Сьогодні великі і малі виробничі підприємства фактично не можуть обходитися без створення мереж передачі даних. Весь документообіг в будь-якій організації ведеться за допомогою комп'ютера. Величезні стелажі з папками документації замінені цифровими копіями документів, пересилання яких здійснюється по мережах передачі даних. Комп'ютери об'єднуються в локальні обчислювальні мережі, корпоративні інформаційно-обчислювальні мережі і найчастіше мають вихід також в глобальну мережу Інтернету.

Функціонування і поточний розвиток корпоративних обчислювальних мереж визначається практично повсюдно організаційною структурою підприємства і розширенням його організаційно-штатної структури. Такий «хаотичний» розвиток мережі призводить до неконтрольованого збільшення числа користувачів, обсягів циркулюючої інформації, інтенсивності трафіку, дисбалансу окремих ділянок мережі і пов'язаним з цими обставинами погіршення якості мережевих послуг

Все це призводить до виникнення перевантажень на ділянках мережі, а отже до порушення цілісності, виникнення загроз втрати даних, помилок, відмов у обслуговуванні та сповільненою роботі в мережі всіх абонентів.

Одна з найважливіших характеристик мережі - її завантаженість. Завантаженість мережі визначається навантаженням на її ділянки і схемою з'єднання мережі

Мережева топологія - це схема з'єднання комп'ютерів, кабельної системи і інших мережевих компонентів. Найбільш поширеними видами мережевих топологій є: лінійна, кільцева, деревоподібна, зіркоподібна і повно зв'язна.

При цьому дослідження топології корпоративних інформаційно-обчислювальних мереж доцільно здійснювати в рамках реальної концепції архітектури інформаційної мережі, яка може бути однією з наступних п'яти видів: архітектура термінал-головний комп'ютер, архітектура інтелектуальної мережі, архітектура клієнт-сервер, однорангова архітектура та архітектура комп'ютер-мережу.

Тому розробка методів прогнозування навантаження мережі є актуальною, важливою задачею та частиною мір по забезпеченню безпечної експлуатації комп'ютерної мережі. Прогнозування інтенсивності трафіку дає можливість заздалегідь вживати необхідних заходів по попередженню наслідків. Досліджуються причини виникнення перевантажень в мережі, наслідком яких є зниження рівня безпеки передачі даних.

Мета поставленої задачі полягає у дослідженні методів прогнозування часових рядів та виявлення найбільш придатних з них для обробки даних, у розробці методу для прогнозування перевантаження в мережевих системах. Також буде проведено аналіз часових рядів, а саме: побудова та дослідження графіка, побудова динамічної моделі часових рядів та прогнозування майбутніх значень. Буде розглянуто та проаналізовано методи ковзних середніх та найменших квадратів, як підходящих для обробки даних. Також буде проведені модельні експерименти з метою досягнення збалансованості навантажень в мережі.

Після проведення аналізу існуючих методів моніторингу та обзору базових засобів моніторингу мережевих систем отримаємо алгоритм дослідження, механізм прогнозування навантаженості мережевих систем за допомогою часових рядів та розроблено метод прогнозування перенавантажень в мережевих систем на основі аналізу часових рядів. Також були проведені аналіз методів прогнозування часових рядів та виявленні найбільш підходящі для обробки отриманих експериментальних даних.

ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНІ ТЕХНОЛОГІЇ ТА МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ

О. В. ГОРБОВА¹, К. В. МЕДВЕДЕВА²

¹Кафедра «Комп'ютерні інформаційні технології», Дніпровський національний університет залізничного транспорту імені академіка В.Лазаряна, вул. Лазаряна, 2, Дніпро, Україна, 49010, тел.. +38 (056) 373-15-35, ел. пошта alexandra.gorbova@gmail.com, ORCID - [0000-0002-5612-2715](https://orcid.org/0000-0002-5612-2715).

²Кафедра «Комп'ютерні інформаційні технології», Дніпровський національний університет залізничного транспорту імені академіка В.Лазаряна, вул. Лазаряна, 2, Дніпро, Україна, 49010, тел.. +38 (056) 373-15-35, ел. пошта karinamedvedeva15@gmail.com

АНАЛІЗ ЧАСОВИХ РЯДІВ НАВАНТАЖЕНОСТІ МЕРЕЖЕВИХ СИСТЕМ

Мета. Мета поставленої задачі полягає у дослідженні методів прогнозування часових рядів та виявлення найбільш придатних з них для обробки даних, у розробці часових рядів для прогнозування перевантаження в мережевих системах. Розробка механізму прогнозування навантаженості мережевих систем є актуальною, важливою задачею та частиною мір по забезпеченню безпечної експлуатації комп'ютерної мережі. Прогнозування інтенсивності трафіку дає можливість заздалегідь вживати необхідних заходів попередження наслідків. Досліджуються причини виникнення перевантажень в мережі, наслідком яких є зниження рівня безпеки передачі даних. **Методика.** Для рішення поставленої задачі необхідно виконати декілька умов що можливо умовно розділити на 3 етапи. На першому етапі збирається інформація за певний період часу для певного ресурсу або вузлу мережевої сітки, котрий залежить від того яка необхідна точність прогнозування майбутнього навантаження. Після збору інформації її необхідно інтерпретувати у необхідний формат та зберегти у вигляді файлу. На другому етапі відбувається подальша обробка, що виконується шляхом розбиття даних на елементи для використання кожного у відповідній частині методу. У результаті проходження даного етапу буде отримано графік навантаження мережі за даними введеного проміжку часу. На останньому етапі відбувається прогнозування майбутнього навантаження, також буде відображено у вигляді продовження графіку. **Результати.** Розроблений метод може бути використаний для аналізу, прогнозуванню та попередженню перевантаженості мережевих систем на підприємствах. Метод дозволяє прогнозувати майбутнє навантаження мережі у короткострокові періоди часу. Використання цього підходу може бути дуже ефективним у рамках мір забезпеченню безпечної експлуатації комп'ютерної мережі. Прогнозування інтенсивності трафіку дає можливість заздалегідь вживати необхідних заходів попередження наслідків. Досліджуються причини виникнення перевантажень в мережі, наслідком яких є зниження рівня безпеки передачі даних. **Наукова новизна.** У статті виконано дослідження методів прогнозування часових рядів та виявлення серед них найбільш придатних методів для обробки даних та застосовано методи прогнозування до вивчення пікових перенавантажень мережевих систем. **Практична значимість.** Запропонований метод призначен для прогнозування навантаження мережевої системи з побудовою графіку навантаження з відображенням навантаження на часовій осі.

Ключові слова: прогнозування; мережева система; часові ряди; навантаження; графік; перевантаження мережі; метод найменших квадратів.

Вступ

Останнім часом з'являється безліч провайдерів, які пропонують високошвидкісне підключення до мережі Інтернет. Це обумовлюється тим, що дані послуги актуальні і затребувані. З кожним днем кількість абонентів, підключених до Інтернету, росте, відповідно зростає і навантаження на мережі та сервери.

Завдання прогнозування часового ряду вирішується на основі створення моделі прогнозування, адекватно описує досліджуваний процес. Прогнозування інтенсивності трафіку дає можливість заздалегідь вживати необхідних заходів попередження наслідків. Досліджуються причини виникнення перевантажень в мережі, наслідком яких є зниження рівня безпеки передачі даних.

Перевантаження мережі в комп'ютерних мережах і теорії черг - це зниження якості обслуговування, яке виникає, коли мережевий вузол або лінія зв'язку переносить більше даних, ніж може обробляти. Типові ефекти включають затримку в черзі, втрату пакетів або блокування нових з'єднань. Наслідком перевантаження є те, що поступове збільшення пропонованої навантаження призводить або до невеликого збільшення, або навіть до зниження пропускної здатності мережі [1].

Мережеві протоколи, які використовують агресивну ретрансляцію для компенсації втрати пакетів через перевантаження, можуть збільшити перевантаження навіть після того, як початкове навантаження було зменшене до рівня, який зазвичай не викликав перевантаження мережі. Такі мережі демонструють два стабільних стани при однаковому рівні навантаження. Стабільний стан з низькою пропускною здатністю відомо як колапс перевантаження.

Мережі використовують методи відстеження перевантажень і запобігання перевантажень для уникнення колапсу. До цієї техніки відносяться: експоненціальна витримка в таких протоколах як CSMA/CA в 802.11 аналогічному CSMA/CD в оригінальному Ethernet, скорочення вікна в TCP і справедлива чергу в таких пристроях, як маршрутизатори і мережеві комутатори. Інші техніки, пов'язані з перевантаженням, включають схеми пріоритетів, які передають деякі пакети з більш

високим пріоритетом ніж інші і явне виділення мережевих ресурсів на конкретні потоки за допомогою використання управління допуском [2].

Затримка мережі є важливою конструктивною і експлуатаційною характеристикою комп'ютерної мережі або телекомунікаційної мережі. Затримка мережі визначає інформаційне перевантаження відноситься до надлишку входить в нашу свідомість інформації.

Мережеві протоколи, які використовують агресивні повторні передачі, щоб дати компенсацію за втрату пакету, мають тенденцію тримати системи в стані перевантаження мережі, навіть після того, як початковий вантаж був зменшений до рівня, який не буде зазвичай викликати перевантаження мережі.

Тісний крах (або крах перевантаженості) є умовою, за якою може досягати комп'ютерна мережа з пакетною комутацією, коли мінімальна корисна комунікація відбувається через перевантаженість. Крах перевантаженості зазвичай відбувається в «вузькому горлі» в мережі, де повне надходить рух до вузла перевищує комунікабельність смугу пропускання. Точки контакту між локальною мережею і глобальною мережею – найбільш ймовірне вузьке горло. Коли мережа знаходиться в таких умовах, вона налаштувалася на стійкий стан, де транспортна вимога є високою, але має корисну пропускну здатність, високі рівні затримки пакета і його втрати [2].

В роботі [3] запропонований метод попередження перевантаження в вузлах комп'ютерної мережі. Основою методу є прогнозування довжини черги, що виникає в буфері вузла, при проходженні через нього самоподібного трафіку. Значення параметрів навантаження, при яких чергу може перевищити розмір буфера, розраховуються за допомогою імітаційного моделювання. Визначаючи параметри вхідного трафіку, шляхом моніторингу можна прогнозувати переповнення буфера і не допускати перевантаження мережі. Численні дослідження процесів в мережі Інтернет показали, що статистичні характеристики трафіку мають властивість тимчасової масштабної.

Причина такого ефекту – в особливості розподілу файлів по серверам, їх розмірах, а також в типову поведінку користувачів.

Виявилося, що з самого початку не проявляють властивостей самоподібності потоки даних, пройшовши обробку на вузлових серверах і активних мережових елементах, починають подавати яскраво виражені ознаки самоподібності.

Самоподібний трафік має особливу структуру, що зберігається на багатьох масштабах – в реалізації завжди присутня деяка кількість дуже великих викидів при відносно невеликому середньому рівні трафіку. ці викиди викликають значні затримки і втрати пакетів, навіть коли сумарна потреба всіх потоків далека від максимально допустимих значень. У класичному випадку для пуассоновського вхідного потоку нам буде достатньо буферів помірного розміру: чергу може утворитися в короткостроковій перспективі, але за довгий період часу буфери очистяться. Однак при самоподібній навантаженні утворюються черги набагато більшої довжини.

У [7] докладно показано нерівномірне зростання швидкостей каналів, по яких передача даних неминуче призводить до виникнення «вузьких» місць в телекомунікаційній мережі та відповідно до виникнення перевантажень, особливо при підключенні мереж доступу до транспортної мережі. Традиційні протоколи управління чергами та запобігання перевантажень керуванням трафіком зі складною динамікою і нелінійністю зміни навантаження, що призводить до виникнення перевантажень і появи глобальної синхронізації TCP потоків. Це зменшує ефективну швидкість передачі даних і погіршує параметри якості, такі як відсоток втрачених пакетів, затримки і варіації затримок. У даній роботі проводиться дослідження мережі TCP/IP із застосуванням різних сучасних алгоритмів управління чергами таких як: RED (Random Early Detection), PI-controller (ProportionalIntegral controller), REM (Random Exponential Marking). Результат обробки сигналів, що відбуваються з різних напрямках сектору, де формуються однакові за структурою дискретно кодовані псевдо-випадкові сигнали, що відрізняються один від одного тільки часовим зсувом.

Основні завдання алгоритмів управління чергами – мінімізація середньої довжини черги при одночасному забезпеченні високого коефіцієнта використання каналу, а також справедливий розподіл буферного простору між різними потоками даних. Схеми управління чергами розрізняються, в основному, критерієм,

за яким відкидаються пакети, і місцем у черзі, звідки проводиться відкидання пакетів (початок або кінець черги).

Найбільш простим критерієм для відкидання пакетів є досягнення чергою певного порогу, званого максимальною довжиною черги. Прикладом алгоритмів пасивного управління чергами (PQM) є алгоритм «відкидання хвоста», де трафік не розділяється по типам пакетів і ймовірність відкидання кожного пакету однакова. Коли черга заповнюється до деякого заданого максимального значення, все знову надходять пакети відкидаються, поки черга не звільниться для надходження вхідного трафіку. Є альтернативні алгоритми відкидання пакетів: Random Drop (відкидання випадково обраного) і Drop from Front (Відкидання першого пакету в черзі). Алгоритмам PQM притаманний ряд недоліків, для усунення яких використовуються алгоритми активного управління чергами (AQM), забезпечують завчасне виявлення перевантаження.

На рис. 1 представлений огляд базових програм моніторингу.

Програма		Ping (Windows)	Tracer	Pathping	Ping (Linux)	Traceroute	MTR	PingT	PingPlotter	PingTool
Функції	Визначення доступності хоста	+	+	+	+	+	+	+	+	+
	Трасировка маршрута	По вхідним інтерфейсам пристроїв	-	+	-	+	+	-	+	-
		По вихідним інтерфейсам пристроїв	+	-	+	-	-	-	-	-
	Визначення втрат пакетів	На всьому маршруті	+	-	+	-	+	-	+	-
		На проміжних вузлах	-	-	+	-	+	-	+	-
	Визначення затримки пакетів	На всьому маршруті	+	+	+	+	+	+	+	+
		На проміжних вузлах	-	+	+	-	+	-	+	-
Використовувані протоколи	ICMP		+	+	+	+	+	+	+	+
	UDP		-	-	-	-	+	-	-	+
	TCP		-	-	-	-	+	-	+	-
Установка спец. параметра	Маршрутизація по заданим углам		+	+	+	+	+	-	+	+
	Розмір відправлюючого пакета		+	-	-	+	-	-	-	-
	Коп-во відправлених пакетів		+	-	+	+	-	+	-	+
	Інтервал між запитами		+	+	+	+	+	+	+	-
Графічний інтерфейс		-	-	-	-	-	+	+	+	+
Запис даних	Графік		-	-	-	-	-	-	+	+
	Текстовий файл		-	-	-	-	-	-	-	+

Рис. 1 – Програмні аналоги

Мета

Основною метою цієї статті є практичне застосування методів прогнозування часових рядів навантаженості мережових систем та обґрунтування або спростування методу при пікових навантаженнях на сервері за встановлених умов.

Методика

Для рішення поставленої задачі пропонується методика, що відповідає потребам, які виникають під час дослідження та є ефективним рішенням задачі.

У задачі «розробка алгоритму» у більшості випадків є використання відомих методик, їх комбінування та використання різних підходів до реалізації для отримання необхідного результату. Для розв'язку задачі використовується методологія з використанням прогнозування. Прогнозування передбачає 3 етапи.

Перший етап. На першому етапі даної реалізації проводиться збір необхідної

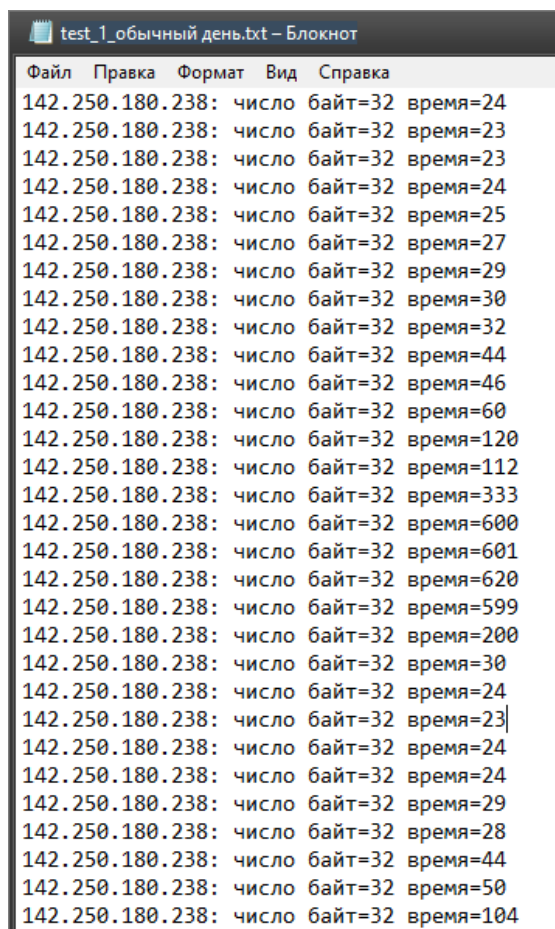


Рис. 2 – Вид вхідного файлу

Другий етап. На цьому етапі відбувається подальша обробка інформації що була надана на першому етапі.

Обробка виконується шляхом перевірки на відповідність вхідних даних та процесу з питанням, як повинен виконуватися цей процес. Алгоритм на відповідність вхідних даних та процесу виконується за допомогою програми, написаної на мові C#, наведено на рис. 3. Як

інформації для проведення експерименту. Ця інформація представляється у вигляді зібраної статистики за певний період часу для певного ресурсу або вузлу мережевої системи, котрий залежить від того яка необхідна точність прогнозування майбутнього навантаження.

Вимоги до представлення інформації:

- інформація про вузол, що досліджується, представляється у вигляді IP адреса;
- інформація про розмір тестового пакету, що досліджується представляється у вигляді байт, які передаються через мережеву систему;
- інформація про час, що досліджується, представляється у вигляді секунд, приклад вхідного файлу наведено на рис. 2;

результат роботи програми, будується графік навантаження мережі за даними введенного проміжку часу, наведено на рис. 4.

Вхідна інформація		Прогнозування	
№	IP	Ping	
1	142.250.180.238	24	
2	142.250.180.238	23	
3	142.250.180.238	23	
4	142.250.180.238	24	
5	142.250.180.238	25	
6	142.250.180.238	27	
7	142.250.180.238	29	
8	142.250.180.238	30	
9	142.250.180.238	32	
10	142.250.180.238	44	
11	142.250.180.238	46	
12	142.250.180.238	60	
13	142.250.180.238	120	
14	142.250.180.238	112	
15	142.250.180.238	333	
16	142.250.180.238	600	
17	142.250.180.238	601	
18	142.250.180.238	620	
19	142.250.180.238	599	

Рис. 3 – Таблично представлені вхідні дані

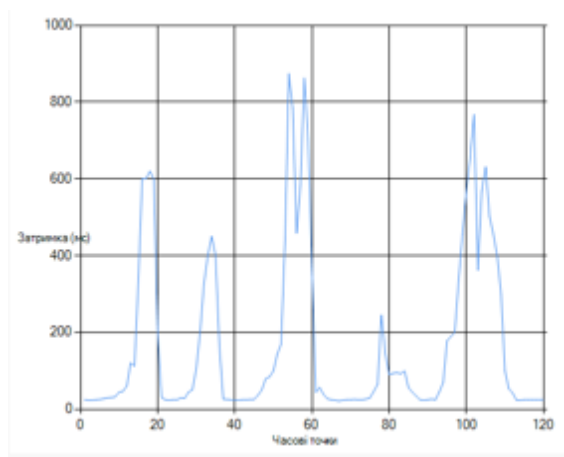


Рис. 4 – Графік, побудований за вхідними даними

Третій етап. На останньому етапі відбувається прогнозування майбутнього навантаження, який представлений графіком, на якому видно зростання та падіння навантаження, наведено на рис. 5.

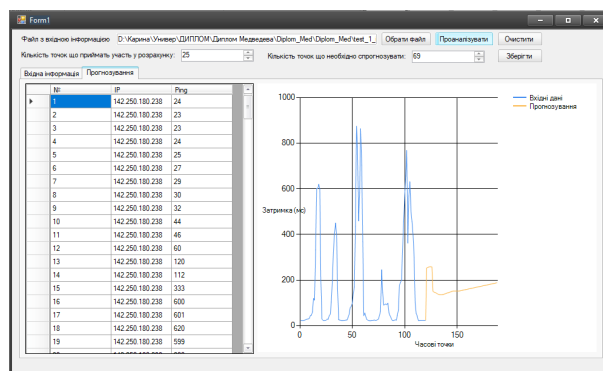


Рис. 5 – Результат прогнозування

На рис. 6 наведено пояснення до результату.

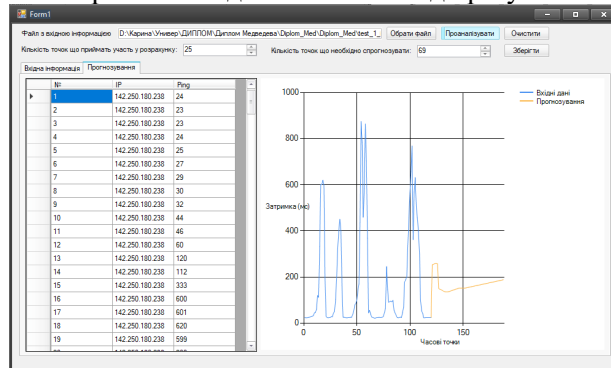


Рис. 6 – пояснення до рис.5

- 1 – таблиця з прогнозованими точками;
- 2 – шкала затримки передачі даних;
- 3 – часові точки прогнозування.

Результати

Метод дозволяє прогнозувати майбутнє навантаження мережі у короткострокові періоди часу. Використання цього підходу може бути дуже ефективним у рамках мір забезпеченню безпечної експлуатації комп'ютерної мережі. Прогнозування інтенсивності трафіку дає можливість заздалегідь вживати необхідних заходів попередження наслідків. Досліджуються причини виникнення перевантажень в мережі, наслідком яких є зниження рівня безпеки передачі даних.

Наукова новизна та практична значимість

Наукова новизна роботи полягає у дослідженні методів прогнозування часових рядів та виявлення серед них найбільш придатних методів для обробки даних та застосовано методи прогнозування до вивчення пікових перевантажень мережевих систем.

Методика може бути використаний для аналізу, прогнозуванню та попередженню перевантаженості мережевих систем на підприємствах. Метод дозволяє прогнозувати майбутнє навантаження мережі у короткострокові періоди часу. Використання цього підходу може бути дуже ефективним у рамках мір забезпеченню безпечної експлуатації комп'ютерної мережі. Прогнозування інтенсивності трафіку дає можливість заздалегідь вживати необхідних заходів попередження наслідків. Досліджуються причини виникнення перевантажень в мережі, наслідком яких є зниження рівня безпеки передачі даних.

Висновки

У цій статті запропоновано метод аналізу часових рядів навантаження мережевих систем та використано метод найменших квадратів для прогнозування перевантаження.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Перегрузка сети. [Электронный ресурс] / URL: <https://ww.google-info.org/7315853/1/peregruzka-seti.html>. (дата звернення 13.09.2021)
2. Диагностика локальных сетей и Интернет. [Электронный ресурс] / URL: http://book.itep.ru/5/dia_5.htm. (дата звернення 10.09.2021)
3. Кондратьева Т. М. Прогнозування тенденції фінансових часових рядів за допомогою нейронної мережі LSTM», / Кондратьева Т. М. – ФГБОУ ВО «Донський державний технічний університет.
4. Прогнозирование временных рядов с помощью гибридной нейронной сети. [Электронный ресурс] / URL: http://www.ict.nsc.ru/jspui/bitstream/ICT/4694/25/DICR-2019-V3_p135-138.pdf. (дата звернення 15.09.2021)
5. Бенгус Б. В. Прогнозирование тенденции временного ряда с помощью искусственной нейронной сети / Б. В. Бенгус // Обозрение прикладной и промышленной математики. Том 21 / Б. В. Бенгус., 2014.
6. Кириченко Л. О. 5. «Алгоритм попередження перенавантаження комп'ютерної мережі шляхом прогнозування середньої довжини черги» / Кириченко Л. О. УДК, 2015.
7. Дослідження мережі TCP/IP з використанням основних алгоритмів активного управління чергою / Гостев В. И. УДК, 2016
8. Метод прогнозирования перегрузок в компьютерных сетях на основе анализа временных рядов: Автореферат / А.С. Соколов. 2012.
9. Руководство по схемам компьютерных сетей. [Электронный ресурс] / URL: <https://www.lucidchart.com/pages/ru/>. (дата звернення 13.09.2021)
10. Перегрузка сети. [Электронный ресурс] / URL: <https://ww.google-info.org/7315853/1/peregruzka-seti.html>. (дата звернення 13.09.2021)
11. Диагностика локальных сетей и Интернет. [Электронный ресурс] / URL: http://book.itep.ru/5/dia_5.htm. (дата звернення 10.09.2021)
12. Сетевая составляющая. [Электронный ресурс] / URL: <http://ru.knowledgr.com/00275196/ПерегрузкаСети>. (дата звернення 13.10.2021)
13. Программ для анализа сетевого трафика. [Электронный ресурс] / URL: <https://networkguru.ru/8-luchshikh-programm-dlia-analiza-setevogo-trafika/>. (дата звернення 20.09.2021)