Міністерство освіти і науки України
Харківський національний економічний університет імені Семена Кузнеця

ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ТА СИСТЕМИ

Монографія
під ред. д.е.н., проф. Пономаренка В.С.

Харків, ХНЕУ ім. С. Кузнеця, 2020

УДК 004.891.2

Авторський колектив: к.ф.-м.н., доц. В. П. Бурдаєв – глава 1; д.т.н., проф. О. Г. Руденко, д.т.н., проф. О. О. Безсонов – глава 2; аспірант К. О. Олейник, аспірант О. С. Романюк – глава 3; д.т.н., проф. М. М. Корабльов, к.т.н., ст. викл. О. О. Фомічов – глава 4; к.т.н., доц. М. Ю. Лосєв – глава 5; д.е.н., проф. О. І. Пушкар, к.е.н., доц. Є. М. Грабовський – глава 6; к.т.н., доц. О. К. Пандорін – глава 7; д.т.н., проф. Н. Г.Аксак, к.т.н., доц. Н. М. Сердюк – глава 8; д.т.н., проф. С. Г. Удовенко, к.т.н., доц. Л. Е. Чала – глава 9; к.т.н., доц. О. Ю. Чередніченко, к.т.н., доц. О. В. Янголенко – глава 10.

Рецензенти:

Годлевський М. Д. – доктор технічних наук, професор, завідувач кафедри програмної інженерії та інформаційних технологій управління Національного технічного університету "Харківський політехнічний інститут";

Петров К. Е. – доктор технічних наук, професор, завідувач кафедри інформаційних управляючих систем Харківського національного університету радіоелектроніки;

Іохов О. Ю. – доктор технічних наук, доцент, начальник кафедри інформатики та прикладних інформаційних технологій Національної академії Національної гвардії України.

Рекомендовано до видання рішенням ученої ради Харківського національного економічного університету імені Семена Кузнеця (протокол № 7 від 22.04.2020 р.)

Інформаційні технології та системи: монографія / за заг. ред . В. С. Пономаренка. - Х. : Видавництво «Стиль-іздат», 2020. - 174 с .

В монографії розглянуті сучасний стан та перспективи розвитку інформаційних технологій та систем різних видів і різного прикладного характеру. Монографія представляє інтерес як для фахівців, сфера діяльності яких безпосередньо пов'язана з розробкою прикладних інформаційних технологійі систем, так і для більш широкого кола фахівців. Вона буде корисною викладачам, аспірантам і студентам, що спеціалізуються в області інформаційних технологій, і всім, хто серйозно цікавиться проблемами інформаційного суспільства.

За достовірність викладених фактів, цитат та інших відомостей відповідальність несе автор.

ЗМІСТ

ВСТУП

Сучасний стан інформаційних технологій та систем можна охарактеризувати наступними тенденціями:

наявність великої кількості баз даних великого обсягу;

створення технологій, що забезпечують інтерактивний хмарний доступ користувача до інформаційних ресурсів;

включення в інформаційні системи елементів інтелектуалізації інтерфейсу користувача, експертних систем, систем глибокого навчання і машинного перекладу.

Стосовно до бізнесу ці тенденції призводять до: здійснення розподілених персональних обчислень, створення розвинених систем комунікацій, усунення проміжних ланок в системі прийняття рішення.

Поява нових інформаційних технологій та систем, поряд з уже наявними, просунутими областями їх використання, здатна формувати такі напрямки, які будуть давати найбільший ефект і сприяти підвищенню технологічного рівня наукомістких виробництв.

У монографії представлені результати наукових досліджень в галузі інформаційних технологій та систем, що представлені на Міжнародній науково-практичній конференції «Інформаційні технології та системи», що відбулася 9-10 квітня 2020 року на базі Харківського національного економічного університету. У ній відображені найбільш цікаві результати за найсучаснішими науковими напрямками застосування, проектування та визначення проблем і перспектив інформаційних технологій в технічних системах тощо.

В главі 1 монографії представлені результати інтегрування чат-бота @es_economy_karkas_bot з прототипами експертних систем для організації консультування в режимі онлайн. Дано опис архітектури і реалізація інтеграції чат-бота месенджера телеграм і системи "КАРКАС". Розглянуто структуру взаємодії чат-бота і агентів експертної системи в онлайн режимі. Наведено приклад онлайн консультації експертної системи у фінансовій та економічній предметній області на прикладі вибору комерційного банку.

На основі використання методів теорії ідентифікації в главі 2 даної монографії досліджено властивості модифікованого регуляризованого алгоритму Качмажа. Визначено умови збіжності регуляризованого алгоритму Качмажа при оцінюванні настаціонарних параметрів при наявності завад вимірів. Отримано неасимптотичні та асимптотичні оцінки, як є досить загальними і залежать як від ступеня нестаціонарності об'єкту, так і від

статистичних характеристик завад. Якщо ж ці параметри не відомі, слід скористатися будь-якої рекурентною процедурою їх оцінювання і використовувати одержувані оцінки для уточнення параметрів, що входять в алгоритми.

В главі 3 проведено порівняльний аналіз градієнтних алгоритмів навчання згортальних нейронних мереж при вирішенні задачі візуального розпізнавання. Дослідження показують, що для цієї задачі досить ефективним є простий алгоритм градієнтного спуску. Використання імпульсів в розглянутих модифікаціях призвело до деякого поліпшення процесу розпізнавання, але також збільшило вартість обчислень. У розглянутих задачах найбільш ефективним алгоритмом є Adamax. У той же час, однак, проблема вибору оптимальних значень параметрів алгоритмів, що забезпечують максимальну швидкість навчання, залишається відкритою.

В главі 4 розглянуто рішення задач класифікації і кластеризації даних на основі гібридних моделей штучних імунних систем, які дозволяють змінювати як спосіб угруповання вихідних даних, так і структуру мережі імунних об'єктів. Гібридизація імунних методів і моделей інтелектуальної обробки інформації дозволяє також підвищити їх ефективність, що виражається в підвищенні швидкодії і точності їх роботи. Запропоновано гібридні імунні моделі класифікації даних на основі методу k-найближчих сусідів, нечіткого підходу, а також кластеризації даних на основі методу k-середніх з використанням моделей штучних імунних систем. Використання адаптивної імунної моделі дозволяє групувати дані або на основі навчальної вибірки, або в процесі імунної навчання. Проведено експериментальні дослідження, що підтверджують ефективність запропонованих гібридних моделей класифікації даних.

В главі 5 розглядаються питання оцінки ефективності управління розподіленими інформаційними системами. При цьому пропонується критерій, який дозволяє враховувати імовірнісно-часові характеристики, навантаження мережі, а також цінність інформації, що передається. Розглядається методика нечітко-множинного аналізу старіння інформації в процесі її передачі в розподілених інформаційних системах. Методика дозволяє визначати нечіткі імовірнісні характеристики старіння даних. Можливо використання методики для оцінювання ефективності управління в умовах невизначеності.

В главі 6 запропоновано методику просування інтернет-магазину в соціальних мережах. Визначено мету рекламної кампанії для інтернет-

магазину одягу «Футбоголік», проведено налаштування спліт-тесту рекламної кампанії, виконано налаштування та запуск рекламної кампанії для плейсменту (стрічка Facebook, стрічка Instagram, Instagram Stories).

Здійснено оцінку ефективності створеної рекламної кампанії та прогнозування поведінки інтернет-магазину після застосування рекламної кампанії.

В главі 7 наведено аналіз сучасних технологій зберігання, розповсюдження та опрацювання мультимедійного контенту систем електронного навчання. Розглянуті особливості застосування key - value архітектури у технологіях розповсюдження мультимедійного контенту. Систематизовано переваги та недоліки скрипкових мов. Окрему увагу наділено технології передачі струменевого відео у веб - браузер користувача.

В главі 8 розроблено систему для надання дистанційних послуг на виробництві для контролю стану здоров'я співробітника. Система реалізована на основі запропонованої трирівневої архітектури для сервіс-орієнтованих систем, що характеризується поєднанням розподілених методів та засобів збору, зберігання, обробки різнорідних даних, що дозволяє використовувати результати віддаленого моніторингу при прийнятті рішень для своєчасного реагування при настанні екстреної ситуації. Одним з ключових компонентів системи є застосування моделі Гаммерштейна, що дозволило кількісно оцінити зміну стану здоров’я працівника при виконанні професійної діяльності підприємства.

В главі 9 запропоновано та програмно реалізовано метод формування контекстних множин ключових слів та зв'язних словоформ, призначений для подальшого автоматизованого анотування електронних текстових документів. Метод базується на використанні моделі семантичного стиснення текстів та модифікованого алгоритму Гінзбурга, що дозволяє формувати матрицю зв'язків між словоформами анотації. Наведено результати тестування розробленого методу на прикладі анотування електронних методичних матеріалів. Отримана анотація може потребувати часткової корекції в разі наявності синтаксичних або семантичних помилок, але є досить точним описом основних положень сенсу аналізованого тексту.

Медичні тексти характеризуються високою складністю та вживанням великої кількості специфічних термінів. Читачами медичних текстів не завжди є лікарі та медичний персонал. Потреба прочитати та зрозуміти медичний текст також часто виникає у людей, що не мають необхідного рівня підготовки для адекватного сприйняття викладеного матеріалу.

Інструкції лікарських засобів, дані медичної карти, приписи лікарів, документація медичних приладів – всі ці документи найчастіше містять слова та словосполучення, які не обов'язково є зрозумілими для широкого загалу. Тому задача симпліфікації медичних текстів є достатньо актуальною, враховуючи кількість потенційних зацікавлених осіб та відсутність інструментів, які б її вирішували щодо текстів українською мовою.

Глава 10 аналізує підходи до оцінки складності текстів. В якості прикладу розглянуто тексти медичних протоколів України, проаналізовано термінологію, яка в них вживається. Зроблено висновок, що складність тексту залежить від частоти вживаних у ньому термінів. Разом з тим, аналіз текстів українською мовою вимагає розробки нових програмних інструментів, що мають враховувати мовні особливості.

ГЛАВА 1

ВИКОРИСТАННЯ ЧАТ-БОТА @ES_ECONOMY_KARKAS_BOT ДЛЯ ОНЛАЙН КОНСУЛЬТАЦІЇ В ЕКОНОМІКО-ФІНАНСОВІЙ СФЕРІ

Вступ і постановка задачі

В даний час онлайн спілкування відіграє величезну роль в житті людей. Тому багато компаній використовують текстові повідомлення як для спілкування між співробітниками (чати), так і для консультації з експертними системами в онлайн режимі за допомогою чат-ботів [1, 2].

У бізнес-середовищі корпоративним стандартом комунікацій став безкоштовний месенджер TELEGRAM. Це обумовлено наступними причинами: високим ступенем шифрування даних в ньому, стабільністю роботи, можливістю передачі великих обсягів інформації, відкритістю протоколу, кроссплатформенністю.

З іншого боку, що дуже важливо для інтегрування месенджера TELEGRAM в інших програмах це те, що розробники надає бібліотеку на основі API для роботи з чат-ботами.

Бот (чат-бот, віртуальний співрозмовник) – це програма, яка імітує людське спілкування на основі елементів штучного інтелекту. Сьогодні боти можуть спілкуватися і між собою для досягнення своїх цілей, інакше кажучи, вони можуть використовуватися як агенти в многоагентних системах.

Чат-боти у фінансовій сфері дозволяють переводити кошти, брати кредити, оплачувати послуги, дізнаватися про тарифи компаній і багато іншого.

Чат-бот PayLastic (месенджер TELEGRAM) надає сервіс оплати картками за товари без терміналів (мобільний еквайринг).

Чат-бот @avalrates (месенджер TELEGRAM) дозволяє дізнатися курс валют на поточну дату.

Чат-бот @GreenZbot (месенджер TELEGRAM) має наступний доступний функціонал: облік витрат, доходів і боргів; розрахунок особистого балансу на кожен день, синхронізацію з google-таблицею.

Відзначимо деякі переваги використання чат-ботів для компаній:

1. Збільшення прибутку. Оскільки чат-боти дозволяють оптимізувати обслуговування до максимального рівня. Наприклад, чат-боти позбавляють

компанії від необхідності тримати власні call-centers тому, що вони істотно збільшують цільову аудиторію за рахунок цілодобової роботи.

2. Зниження операційних витрат. Оскільки чат-боти успішно виконують завдання збору та обробки даних про клієнтів, організацію відгуків про продукт компанії, нагадують про зустрічі та відправлення повідомлень в месенджери.

Використовуючи чат-боти в месенджерах, невеликі компанії можуть відмовитися від витрат на створення власного сайту або мобільного додатку.

Зауважимо, що найуспішніші чат-боти у фінансовій та економічній сферах створюються на основі рішень штучного інтелекту і технологій машинного навчання. До них можна віднести віртуальних помічників для мобільних телефонів, корпоративних асистентів.

Однією з перших програм, що реалізують концепцію чат-бота, була програма ELIZA, що імітувала поведінку лікаря-психотерапевта при первинному опитуванні пацієнта. Ідея реалізації цієї програми полягала в знаходженні в тексті спілкування ключових слів або сполучень, для того щоб задати питання для підтримки діалогу зі співрозмовником. Якщо ключове слово знайдено в базі даних, то питання до співрозмовника задається відповідно до заздалегідь підготовленого шаблоном питання, інакше твердження співрозмовника перетворюється в питання. Якщо слово поєднання не знайдено, то програма ставить співрозмовнику питання загального вигляду, наприклад, "Чому Ви так вважаєте?".

Є кілька стратегій для реалізації такого діалога:

1. Питання співрозмовника вибирається зі списку питань, що відносяться до ключового слова, згідно з більш високою частотою використання питання в предметній області.

2. Бот збирає питання і фрази, які використовуються співрозмовниками, таким чином, навчаючись і збільшуючи свій контент предметної області.

3. Синтаксичний підхід, заснований на граматичному розборі фрази співрозмовника і забезпечений правилами виду "якщо – то".

Природно, недоліком такого спілкування стає не зовсім логічний діалог між ботом і співрозмовником. У співрозмовника з'являлася ілюзія, що бот розуміє його, хоча насправді це не так. Іншими словами, чат-боту не вистачає реалізації взаємодії з машиною висновку, як це широко застосовується в експертних і експертно-навчальних системах [3 – 5].

Експертні системи в економіко-фінансовій сфері здійснюють підтримку логічно складних бізнес правил, фінансового і оперативного управління виробництвом, постачання, збуту і планування ресурсів підприємства.

Експертна система Nikko Portfolio Consultation Management System допомагає керуючим фондами вибрати оптимальний портфель для своїх клієнтів [3].

Система підтримки прийняття рішень при управлінні портфелем PMIDSS здійснює вибір портфеля цінних паперів, довгострокове планування інвестицій [3].

Експертна система Intelligent Hedger формує рішення проблеми величезної кількості постійно зростаючих альтернатив страхування від ризиків [3].

Більшість експертних систем у фінансовій та економічній діяльності функціонують в офлайн режимі. Тому парадигма інтегрування чат-ботів для роботи з експертними системами в онлайн режимі зараз стає все більш актуальною [2].

За допомогою бібліотеки API TELEGRAM був створений бот @es_economy_karkas_bot для онлайн консультації користувача з інструментальним засобом для створення баз знань з системою "КАРКАС" [4, 5].

Основна частина

В останні десятиліття, з появою смартфона, зросла популярність концепції штучного інтелекту стосовно додатків для обміну повідомленнями. Світовий ринок чат-ботів буде рости в найближчі роки. Одне з головних переваг чат-ботів в обслуговуванні клієнтів полягає в тому, що співрозмовники можуть вільно задавати питання, які вони не задали б представнику служби підтримки або менеджеру компанії. Крім того, бот здатний миттєво відповідати на питання.

Чат-боти зазвичай інтегровані в діалогові системи, наприклад, віртуальних співрозмовників, що дає їм можливість природного спілкування з клієнтами компанії.

У більшості випадків чат-боти використовують програми обміну повідомленнями для спілкування з клієнтами. Людина може надрукувати або задати питання, і чат-бот відповість правильну інформацію. Залежно від

ситуації, багато чат-ботів можуть вчитися на тому, що говорить клієнт, щоб персоналізувати взаємодію і вибудувати попередню взаємодію.

Чат-бот можна розглядати як питально-відповідну систему (QA-система) з елементами машинного навчання, а саме з функціями розбору природної мови, машиною логічного висновку і модулем зв'язку із зовнішніми програмами. Актуальною проблемою для чат-ботів QA-систем є створення машини логічного висновку, що визначає релевантність знань до заданого питання.

Чат-боти месенджера TELEGRAM, як співрозмовники, при роботі з системою "КАРКАС" дають більше можливостей мобільно консультуватися з експертною системою через смартфон, що, наприклад, важливо для прийняття ефективних рішень в економіко-фінансовій сфері. Іншими словами, тепер можна відправити текстове повідомлення боту @es_economy_karkas_bot і отримати моментально необхідну інформацію, тобто здійснювати консультацію в режимі реального часу. Вміст команди чат-бота /help показано на рис. 1.

Рис. 1. **Вид команди /help чат-бота @es_economy_karkas_bot**

Бот @ es_economy_karkas_bot дозволяє провести онлайн консультацію з наступними прототипами експертних систем:

1. Команда /fa викликає прототип ЕС для аналізу фінансового стану підприємства, призначена для підвищення якості результату оцінки фінансового стану підприємства.

2. Команда /finsost викликає прототип ЕС для аналізу фінансового стану підприємства (критичне, нестабільне, стабільне, перспективне).

3. Команда /bank_commercial викликає прототип ЕС по підбору банку для фінансового обслуговування підприємства.

4. Команда /insurance_company викликає прототип ЕС для вибору страхової компанії.

5. Команда /credit_insurance викликає прототип ЕС для страхування комерційних кредитів.

6. Команда /creditworthiness викликає прототип ЕС для визначення класу кредитоспроможності позичальника.

7. Команда /enterprise_strategy викликає прототип ЕС для вибору стратегії підприємства.

8. Команда /product_suppliers викликає прототип ЕС для вибору постачальників продукції.

9. Команда /product_competitiveness викликає прототип ЕС для оцінки конкурентоспроможності продукції.

При виклику команди /bank_commercial викликається прототип ЕС по підбору банку для фінансового обслуговування підприємства.

Призначення прототипу ЕС – це консультування з підбору комерційного банку для фінансового обслуговування підприємства.

Сфера застосування прототипу ЕС – це різні підприємства, які потребують фінансового обслуговуванні банками.

Мета прототипу ЕС – підбір найбільш оптимального варіанту банку для фінансового обслуговування підприємства в залежності від його потреб в проведенні касово-розрахункових, кредитних, депозитних та трастових операцій.

Початкові дані:

для аналізу діяльності підприємства використовуються виробнича, збутова, закупівельна діяльності, наявність або відсутність вільних грошових коштів;

для визначення платоспроможності банку використовуються власні кошти, активи банку;

для визначення ліквідності банку використовуються кошти на розрахункових, поточних і депозитних рахунках.

Очікувані результати (список можливих значень мети консультації):

вимоги до фінансового обслуговування підприємства – це терміновість грошових платежів, форми грошових платежів (готівковий, безготівковий), депозитні, кредитні, касово-розрахункові або трастові операції;

вимоги до банків – платоспроможний або неплатоспроможний, ліквідний або неліквідний банк.

Ідентифікація предметної області. Обов'язковими для кожного комерційного банку є наступні економічні нормативи, що встановлюються Національним банком України і визначають надійність даного банку:

платоспроможність банку;

показники ліквідності балансу;

максимальний розмір ризику на одного позичальника;

розмір обов'язкових резервів, що розміщуються в Національному банку України.

При виборі комерційного банку підприємство, як правило, спирається на наступні показники:

надійність банку;

яка форма платежу підходить для підприємства: готівкова або безготівкова;

операції, які бажає здійснювати підприємство;

форма розрахунку, бажана підприємством.

Концептуальну модель предметної області оцінки комерційного банку можна представити у вигляді дерева рішень (рис. 2). Тут перетинає дугою позначена вершина типу "І", а відсутність її - "АБО".

У предметної області виділені класи, які представлені в табл. 1.

Таблиця 1

Класи БЗ

Клас	Кількість примірників класу	Рівень ієрархії класів
Банк	10	1
Конфіденційність	3	2
Надійність і стійкість	4	2
Комплексність обслуговування	3	2
Операції	9	2

Рис. 2. **Дерево рішень для задачі вибору комерційного банку**

На рис. 3 зображено логічні варіанти визначення надійності і стійкості банку.

Рис. 3. **Варіанти визначення надійності і устойчивости**

Аналогічним чином прототип ЕС будує варіанти для визначення конфіденційності угод, варіанти комплексності фінансового обслуговування.

Логічна модель визначення набору банківських послуг приведена на рис.4.

Рис. 4. **Логічна модель визначення набору банківських послуг**

Вибір комерційного банку для надання послуг юридичним особам буде здійснюватися на підставі чотирьох критеріїв:

надійність і стійкість;

комплексність (зручність) фінансового обслуговування;

набір банківських послуг (наданих операцій);

конфіденційність угод.

На рис. 5 наведено приклад логічної схеми визначення комерційного банку на підставі обраних критеріїв. Наведено приклад для трьох банків, але в системі реалізована можливість вибору з десяти комерційних банків України.

Рис.5. **Логічна схема визначення банку**

Система «КАРКАС" представляет собою інструментарій для розробки прототипів баз знань для експертних и експертно-навчальних систем як в офлайн, так і онлайн режимах на смартфонах. Подання знань ґрунтується на ієрархічній функціональній системі, яка генерується системою "КАРКАС" на базі правил продукцій и фреймів.

Машина висновку використовує ієрархічну функціональну систему під час проведення консультації з користувачем. Користувач може вибрати різні режими роботи машини висновку: використання прямого висновку, зворотнього висновку, непрямого висновку, формули Байєса, таблиці крітеріїв, коли консеквент продукції представляє собою список параметрів.

Вид ієрархічної функціональної системи для вибору комерційного банку наведено на рис. 6.

Рис. 6. **Вид ієрархічної функціональної системи для вибору комерційного банку**

Модуль онлайнової консультації (співрозмовник) дозволяє за допомогою месенджера TELEGRAM обмінюватися повідомленнями з базами знань системи "КАРКАС" через Інтернет, іншими словами здійснювати консультацію в режимі реального часу. Зауважимо, що формування бази знань і її налагодження здійснюється на локальному комп'ютері.

Месенджер TELEGRAM володіє мільйонами активних користувачів і є найшвидшим додатком для обміну повідомленнями. Він працює на всіх пристроях, на мобільних і настільних платформах.

Система "КАРКАС" – це інструментальний засіб для побудови моделей баз знань. Структура предметної області може бути різноманітною, наприклад, вибір рішення серед певного набору варіантів, використання ненадійних знань. Система "КАРКАС" дозволяє, як розробляти моделі баз знань, так і може бути використана для тестування і навчання студентів по локальній мережі. Система "КАРКАС" за допомогою чат-ботів: @Ribs_karkas_bot, @test_karkas_bot, @it_karkas_bot дозволяє проводити онлайн консультацію з користувачами і тестування знань студентів в різних

предметних областях: комп'ютерна графіка, технології баз даних, веб-аналітика, системи бізнес інтелекту.

Інтегрування чат-бота з модулями консультації та діалогу системи "КАРКАС" полягає в обміні інформацією між ними без участі користувача, а також передачею і прийомом запитів для роботи з серверами TELEGRAM з використанням TELEGRAM API і JSON з безпечного протоколу HTTPS рис. 7.

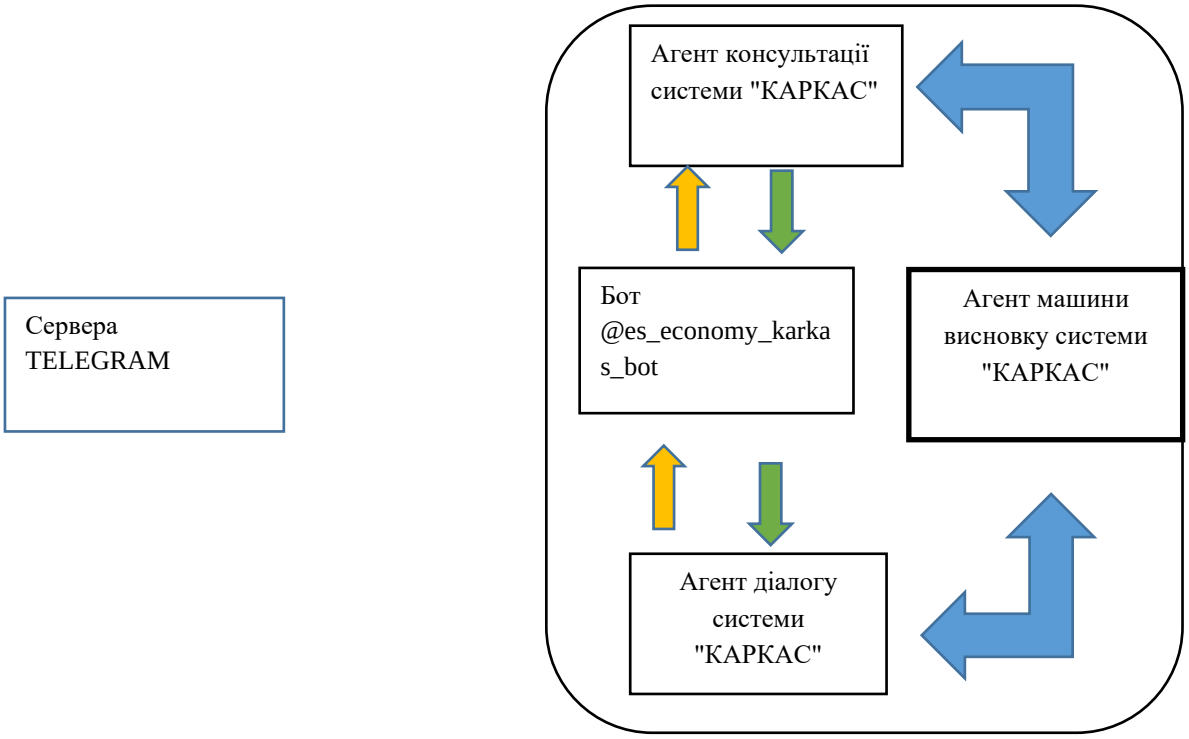


Рис. 7. **Архітектура інтеграції чат-бота з системою "КАРКАС"**

Для роботи із запитами до серверів TELEGRAM використані наступні компоненти:

1. Для парсинга JSON об'єктів використана бібліотека superobject.

2. Для здійснення роботи за допомогою протоколу https використані бібліотеки OpenSSL: libeay32.dll і ssleay32.dll.

3. Для відправки запитів http і завантаження баз знань по протоколу ftp з https://it-karkas.com.ua використана бібліотека Indy 10.

4. Для роботи з серверами TELEGRAM використана бібліотека TelegAPI.

Агенти консультації та діалогу обмінюються повідомленнями між собою для виконання наступних операцій:

1. Натискання: кнопок, чек боксів, радіо кнопок.

2. Передача і прийом повідомлень між візуальними об'єктами на формі.

Таким чином, зазначені вище модулі виконують функції агентів і в цьому сенсі імплементований чат-бот @es_economy_karkas_bot в систему "КАРКАС" можна розглядати, як багатоагентну систему.

Для інтеграції системи "КАРКАС" з чат-ботом @es_economy_karkas_bot використовується агент консультації (рис. 8) і агент діалогу.

Рис. 8. **Вид форми агента консультації системи "КАРКАС"**

Передача і прийом повідомлень агента консультації.

1. Активувати додаток (ribs_karkas_bot.exe), що запускає бот, можна на ресурсі, що має доступ до інтернету (хостинг, домашній комп'ютер). Потім в месенджері телеграм запустити його: @Ribs_karkas_bot. Набрати команди / help або / start бот і бот запропонує вибрати команди для запуску експертних систем, тестів (рис. 1).

2. Наприклад, при виборі команди / ribs виконуються наступні операції:

завантажується база знань ribs.knb з сайту https://it-karkas.com.ua;

виконується модуль консультації і запускається машина висновку експертної системи;

активізується модуль діалогу.

3. Результат консультації експертної системи передається боту по широкомовному протоколу.

Таким чином, алгоритм роботи чат-бота @es_economy_karkas_bot складається з наступних кроків:

Крок 1. Активувати чат-бот @es_economy_karkas_bot в месенджері TELEGRAM.

Крок 2. Вибрати команди: /help або /start, потім, наприклад, команда /bank_commercial викликає прототип ЕС по підбору банку для фінансового обслуговування підприємства.

Крок 3. Бот запускає агента консультації системи "КАРКАС".

Крок 4. Активізується машина висновку системи "КАРКАС".

Крок 5. Формується ієрархічна функціональна система для ведення діалогу з користувачем.

Крок 6. Активізується агент діалогу, котрий посилає боту повідомлення з текстом питання і відповідями. Бот приймає повідомлення у вигляді об'єкта JSON, виконує його парсинг, відображає повідомлення в чаті і чекає відповіді користувача.

Крок 7. Користувач в чат-боті вибирає або вводить відповідь. Бот відсилає відповідь машині висновку експертної системі.

Крок 8. Агент консультації експертної системи приймає повідомлення і передає його машині висновку, яка передає повідомлення агенту діалогу. Мета консультації уточнюється, на основі ієрархічної функціональної системи, під час діалогу з користувачем.

Крок 9. Ітеративний процес консультації триває поки машина висновку не отримає результат від експертної системи. Користувач може в будь-який момент припинити консультацію командою /quit.

Висновки

В роботі представлені результати інтегрування чат-бота @es_economy_karkas_bot з експертною системою для організації консультування в режимі онлайн. Розглянуто алгоритм взаємодії чат-бота і агентів експертної системи в онлайн режимі.

В результаті був створений повністю функціонуючий чат-бот @es_economy_karkas_bot, який інтегрований в систему "КАРКАС" і дозволяє в режимі онлайн проводити консультацію з прототипами експертних систем в економіко-фінансовій предметній області. Після розгортання програми планується значно розширити функціональність бота.

Система "КАРКАС" являє собою інструментарій для розробки прототипів баз знань для експертних і експертно-навчальних систем як в офлайн, так і онлайн режимах на смартфонах. Подання знань ґрунтується на ієрархічній функціональній системі, яка генерується системою "КАРКАС" на базі правил продукцій і фреймів.

За допомогою системи "КАРКАС" розроблений ряд прототипів ЕС в наступних предметних областях: медицина, економіка, мобільний зв'язок і кластерний аналіз багатовимірних даних.

Система "КАРКАС" за допомогою чат-ботів: @Ribs_karkas_bot, @test_karkas_bot, @it_karkas_bot дозволяє проводити онлайн консультацію з користувачами і тестування знань студентів в різних предметних областях: комп'ютерна графіка, технології баз даних, веб-аналітика, системи бізнес-інтелекту.

Література

1. Рассел С. Искусственный интеллект: современный поход / С. Рассел, П. Норвиг. – 2-е изд.; [Пер. с англ. – М.: Изд. дом "Вильямс", 2006. – 1408 с.

2. Бурдаєв В.П. Інтегрування месенджерів з системою "КАРКАС". // Тези доповідей: міжнарод. наук.-практ. конф. Проблеми і перспективи розвитку ІТ-індустрії, Харків, 2018, с.7.

3. Волкова В. Н. Информационные системы в экономике: учебник для академического бакалавриата / В. Н. Волкова, В. Н. Юрьев, С. В. Широкова, А. В. Логинова; под ред. В. Н. Волковой, В. Н. Юрьева. – М.: Юрайт, 2016. – 402 с.

4. Бурдаєв В. П. Моделі баз знань: моногр. / В. П. Бурдаєв – Харків: ХНЕУ, 2010. – 300 с.

5. Burdaev V. P. ″About one concept of constructing a temporal knowledge base″, in Proc. of the 1st International Congress Fundamental and Applied Studies in the Pacific and Atlantic Oceans Countries, Tokyo University Press, 2014, pp. 272–276.

ГЛАВА 2

ПРО ОДИН АЛГОРИТМ НАВЧАННЯ АДАЛІНИ В ЗАДАЧІ ОЦІНЮВАННЯ НЕСТАЦІОНАРНИХ ПАРАМЕТРІВ

Вступ

Багато задач управління, прогнозування, розпізнавання образів і т.д. пов'язані з побудовою моделі виду:

$$y(k+1) = \theta^T x(k+1) + \xi(k+1), \qquad (1)$$

де $y(k+1)$ - спостережуваний вихідний сигнал; $x(k+1) = (x_1(k+1), x_2(k+1),..x_N(k+1))^T$ - вектор вхідних сигналів $N \times 1; \theta^* = (\theta_1^*, \theta_2^*,..\theta_N^*)^T$ - вектор шуканих параметрів $N \times 1; \xi(k+1)$ - перешкода; k -дискретний час, і зводяться до мінімізації деякого наперед обраного функціоналу якості (критерію ідентифікації). Найбільш широко використовуваний на практиці квадратичний функціонал призводить до різних алгоритмах ідентифікації, що дозволяє отримати оцінки шуканого вектора θ^* при нормальних розподілах перешкоди, тобто $\xi(k) \sim N(0, \sigma_\xi^2)$.

Однак завдання ідентифікації істотно ускладнюється, якщо параметри θ змінюються (дрейфують) в часі, тобто $\theta^*(k) = \text{var}$.

Результати більшості робіт, присвячені цій проблемі, носять в основному рекомендаційний характер і слабо пов'язані з характером нестаціонарності досліджуваного об'єкта. Так для оцінки нестаціонарних параметрів зазвичай застосовуються модифіковані алгоритми МНК (використання ковзного вікна або експоненціального згладжування і т.д.), алгоритм Качмажа (і його динамічний варіант), алгоритми динамічної адаптації та ін. Відсутність досить загальних рекомендацій по вибору алгоритмів оцінювання нестаціонарних параметрів обумовлено , мабуть, труднощами теоретичного дослідження властивостей цих алгоритмів в нестаціонарних умовах.

Ефективність застосування того чи іншого алгоритму для оцінки дрейфуючих параметрів істотно залежить від обсягу апріорної інформації про характер дрейфу. Відповідно до цього адаптивні алгоритми ідентифікації нестаціонарних параметрів можна розділити на два класи:

1. Алгоритми оцінювання параметрів при відомому законі їх дрейфу.
2. Алгоритми оцінювання параметрів при невідомому законі дрейфу.

Перший напрямок бере початок з роботи Дупача [1]. У цій роботі запропоновано узагальнення методу стохастичною апроксимації на випадок,

коли дрейф кореня рівняння регресії близький до лінійного. У більшості наступних робіт розглядалося випадок, коли дрейф або в середньому близький до лінійного, або згасає. В [2] розглядався випадок, коли дрейф $\theta^*(k)$ параметризований, і завдання ідентифікації зводилася до оцінювання невідомого параметра багатовимірним методом стохастичної апроксимації.

Серед найбільш простих в обчислювальному відношенні однокрокових алгоритмів ідентифікації найбільш ефективними є алгоритми Качмажа і Нагумо-Ноди [3,4]. Однак їх швидкодія виявляється найчастіше недостатнім при оцінюванні нестаціонарних параметрів. Знання (або апроксимація) закон дрейфу дозволяє отримувати ефективні алгоритми відстеження нестаціонарних параметрів. В [5] розглянуто алгоритм, названий динамічним алгоритмом Качмажа, в якому використовується деяка апріорна модель модель дрейфу. Слід, однак, відзначити, що помилки в завданні закону зміни параметрів можуть привести до втрати властивостей збіжності алгоритму.

Відсутність інформації про характер дрейфу $\theta^*(k)$ вимагає розробки алгоритмів ідентифікації, що використовують мінімальний обсяг інформації про та зберігають працездатність в широкому діапазоні варіювання $\theta^*(k)$. Тому більш природним є вивчення динамічних властивостей конкретних алгоритмів і визначення та максимально досяжної точності стеження, що в кінцевому підсумку дозволить визначити області найбільш ефективного використання алгоритмів і розробити рекомендації щодо їх практичного застосування.

Природною платою за відсутність апріорної інформації про характер дрейфу параметрів є запізнювання в оцінці зміни параметрів, а, отже, менша точність спостереження, що обумовлено інерційністю алгоритмів.

Слід зазначити, що в ряді випадків досить ефективним при вирішенні даного завдання виявляється нейромережевий підхід, при якому математична модель об'єкта представляється деякою штучною нейронною мережею (ШНМ), утвореною нейронами. При цьому завдання ідентифікації зводиться до навчання мережі - оцінювання параметрів складових її нейронів, яке здійснюється шляхом мінімізації деякого функціоналу, що представляє собою деяку опуклу функцію від помилки навчання (ідентифікації).

Для навчання мережі застосовуються, як правило, методи, що вимагають обчислення градієнта функціоналу, що використовується: алгоритм зворотного поширення помилки (ЗП), метод сполучених градієнтів, алгоритм Гауса-Ньютона, Левенберга-Марквардта і т.д. Незважаючи на популярність цих методів не тільки при навчанні ШНМ, але і вирішенні інших завдань оптимізації, вони мають ряд суттєвих недоліків, наприклад, рішення, що

отримується, залежить від форми функції, яка мінімізується, вектора початкових умов; вони застрягають в локальних екстремумах, а при наявності декількох екстремумів не забезпечують знаходження глобального; їх застосування вимагає диференційованості функціоналів, які мінімізуються. Крім того, складним є визначення оптимальних параметрів алгоритмів, що забезпечують їх максимальну швидкість збіжності, точність, робастність і т.д.

У цих умовах доцільним видається використання найбільш простої ШНМ типу АДАЛІНА. Дана ШНМ, яка є першою лінійною нейронною мережею, була запропонована Б.Уідроу і М.Е.Хоффом в роботі [6] в 1960 р. Розроблена ними модель була названа адаптивним нейроном, а потім ADALINE - ADAptive LInear NEuron. З плином часу свою популярність в якості ШНМ вона втратила і тепер ADALINE це ADAptive LINear Element. АДАЛІНА мала структуру, представлену на рис. 1 [6,7].

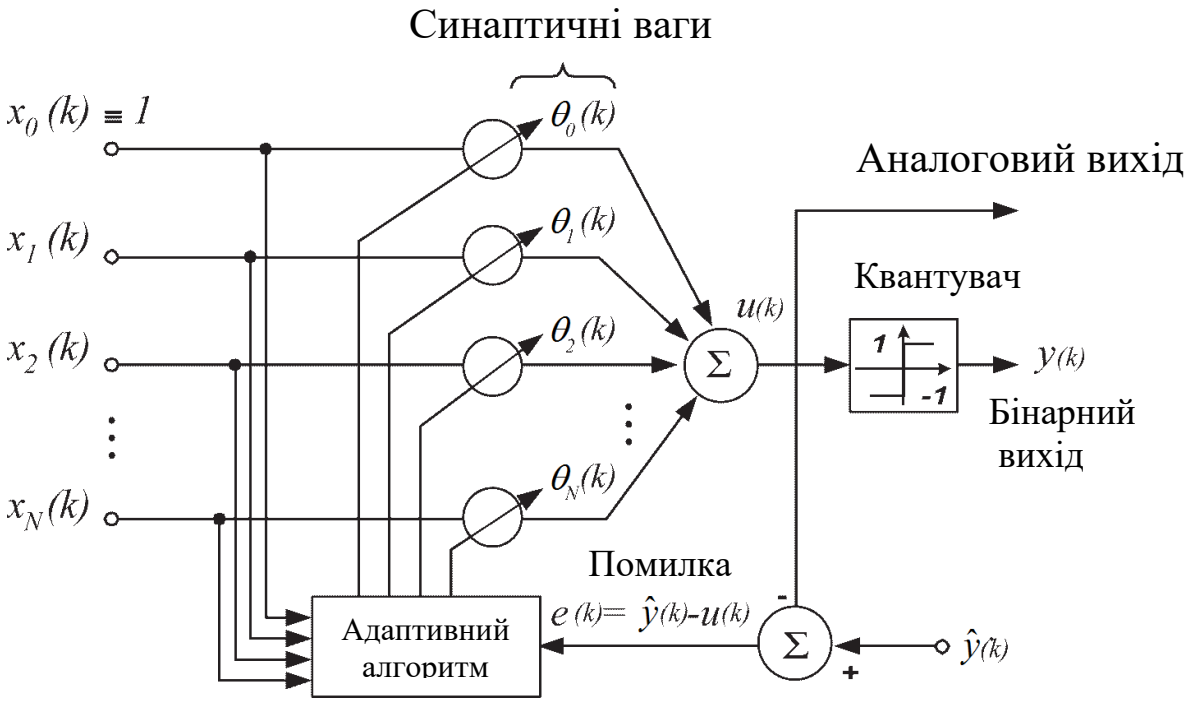


Рис.1 – АДАЛІНА

Основна відмінність АДАЛІНИ від персептрону полягає в використовуваному алгоритмі навчання. Хоча навчання АДАЛІНИ також складалося в корекції її вагових коефіцієнтів на підставі пред'явлених навчальних пар і порівняння реальних вихідних сигналів моделі з бажаними (необхідними), Уїдроу і Хофф, грунтуючись на роботах Н. Вінера, пов'язаних з дослідженням фільтрів, використовували для цього градієнтний алгоритм

$$\hat{\theta}(k+1) = \hat{\theta}(k) + \frac{e(k+1)x(k+1)}{\|x(k+1)\|^2}, \qquad (2)$$

де $e(k+1)=y(k+1)-\hat{y}(k+1)=y(k+1)-\hat{\theta}^T(k)x(k+1),\ \ \hat{y}(k+1)$ - вихідний сигнал нейромережевої моделі; $\hat{\theta}(k)=(\hat{\theta}_1(k),\hat{\theta}_2(k),..\hat{\theta}_N(k))^T$ - вектор вагових параметрів; - деякий параметр, що впливає на швидкість збіжності алгоритму (тривалість процесу навчання моделі); $\|\bullet\|$ - евклидова норма. Хоча даний алгоритм в літературі з ШНМ називається алгоритмом Уїдроу - Хоффа, значно раніше він був запропонований в роботі [3] для вирішення систем лінійних алгебраїчних рівнянь, а згодом, починаючи з [8,9], з успіхом почав застосовуватися для розв'язання задачі ідентифікації при побудові моделі типу (1). Оцінки швидкості збіжності даного алгоритму при ідентифікації стаціонарних об'єктів (1) вперше були отримані в [8-12].

Слід, однак, відзначити, що даний алгоритм застосовується не тільки в системах ідентифікації, а й набув широкого поширення при вирішенні задач фільтрації [13-15]. В іноземній літературі цей алгоритм більш відомий як нормалізований алгоритм найменших квадратів (NLMS - normilized least-mean-square).

Постановка задачі

Для підвищення обчислювальної стійкості (2) В.М. Чадеева [8, 9] була запропонована його модифікація - регуляризований алгоритм

$$\hat{\theta}(k+1)=\hat{\theta}(k)+\frac{e(k+1)x(k+1)}{\|x(k+1)\|^2+\delta}, \qquad (3)$$

де $\delta>0$ - параметр регуляризації.

Слід зазначити, що в згадуваних вище роботах були отримані оцінки швидкості збіжності алгоритму для стаціонарного випадку, тобто для випадку $\theta^*=const.$

В роботі досліджуються властивості модифікованого регуляризованого алгоритму

$$\hat{\theta}(k+1)=\hat{\theta}(k)+\gamma\frac{e(k+1)x(k+1)}{\|x(k+1)\|^2+\delta}, \qquad (4)$$

де $\gamma>0$ - деякий параметр, що впливає на швидкість збіжності алгоритму, в завданню оцінювання нестаціонарних параметрів, що представляють собою стаціонарні випадкові процеси, які можуть бути описані модифікованою Марківського моделлю першого порядку [12,16]

$$\theta(k+1)=\varepsilon\theta(k)+\sqrt{1-\varepsilon}S(k+1), \qquad (5)$$

де

$S(k+1)=(S_1(k+1),S_2(k+1)..,S_N(k+1))^T$ - вектор випадкової послідовності $N\times1$; $S_i\sim N(0,\sigma_s^2)$; $0<<\varepsilon<1$ - $0<<\varepsilon<1$ - параметр (при $\varepsilon=1$ маємо стаціонарну модель).

Дослідження збіжності алгоритму

Введемо в розгляд помилку оцінювання

$$\tilde{\theta}(k+1) = \theta^*(k) - \hat{\theta}(k) \tag{6}$$

З урахуванням (5) вираз для можна записати в такий спосіб:

$$\begin{aligned} e(k+1) &= \tilde{\theta}^T(k)x(k+1) + \xi(k+1) \\ &= \tilde{\theta}^T(k)x(k+1) + (\varepsilon - 1)\theta^T(k)x(k+1) + \sqrt{1-\varepsilon}S^T(k+1)x(k+1) + \xi(k+1). \end{aligned} \tag{7}$$

У припущенні нормальності перешкоди можна записати

$$\mathrm{M}\left\{e^2(k+1)\right\} = \sigma_\xi^2 + \sigma_x^2 \mathrm{M}\left\{\left\|\tilde{\theta}(k+1)\right\|^2\right\}, \tag{8}$$

де M- символ математичного очікування.

Беручи до уваги (5), уявімо помилку оцінювання (6) так:

$$\tilde{\theta}(k+1) = \tilde{\theta}(k) + (\varepsilon - 1)\theta(k) + \sqrt{1-\varepsilon^2}S(k+1) - \gamma \frac{x(k+1)}{\left\|x(k+1)\right\|^2 + \delta}\left[\xi(k+1) + \tilde{\theta}^T(k)x(k+1)\right] \tag{9}$$

Передбачається, що компоненти вектора помилки оцінювання підкоряються нормальному закону розподілу з і дисперсією [17], тобто компоненти все компоненти вектора оцінки розподілені по нормальному закону, з функцією щільності ймовірності

$$f\left(\left|\hat{\theta}_i(k)\right|\right) = \frac{1}{\sqrt{2\pi\sigma_i^2(k)}}\left[e^{-\frac{\left(\left|\hat{\theta}_i(k)\right| - m_i(k)\right)^2}{2\sigma_i^2(k)}} + e^{-\frac{\left(\left|\hat{\theta}_i(k)\right| + m_i(k)\right)^2}{2\sigma_i^2(k)}}\right] U\left(\hat{\theta}_i(k)\right), \tag{10}$$

де

$$m_i(k) = \theta_i(k) - \tilde{\tilde{\theta}}_i(k);$$

$$\sigma_i^2(k) = M\left\{\tilde{\theta}^2{}_i(k)\right\} - M^2\left\{\tilde{\theta}_i(k)\right\};$$

$$U\left(\hat{\theta}_i(k)\right) = \begin{cases} 0, \hat{\theta}_i(k) < 0, \\ 1, \hat{\theta}_i(k) \geq 0. \end{cases}$$

Середнє цього розподілу визначається формулою

$$M\left\{\left|\hat{\theta}(k)\right|\right\} = \int_{-\infty}^{\infty}\left|\hat{\theta}(k)\right| f\left(\left|\hat{\theta}(k)\right|\right) d\left|\hat{\theta}(k)\right| =$$

$$= m_i(k)erf\left(\frac{m_i(k)}{\sqrt{2\sigma_i^2(k)}}\right) + \sqrt{\frac{2}{\pi}}\sigma_i(k)e^{-\frac{m_i^2(k)}{2\sigma_i^2(k)}},$$

де $erf(x) = \frac{2}{\sqrt{\pi}}\int_0^x e^{-t^2}dt$ - інтеграл імовірності Гауса.

Обчислимо $\mathrm{M}\left\{\tilde{\theta}(k+1)\right\}$.

Враховуючи, що [10,18]

$$\mathrm{M}\left\{\frac{1}{\|x(k+1)\|^2+\delta}\right\}=\frac{1}{(N-2)\sigma_x^2+\delta}\left(1-\frac{2\delta\sigma_x^2}{\left((N-2)\sigma_x^2+\delta\right)^2}\right); \tag{11}$$

$$\mathrm{M}\left\{\frac{x(k+1)x^T(k+1)}{\|x(k+1)\|^2+\delta}\right\}=\frac{1}{N}I-\mathrm{M}\left\{\frac{\delta}{\|x(k+1)\|^2+\delta}\right\}I, \tag{12}$$

де I - одинична матриця $N\times N$, отримаємо

$$\begin{aligned}
&\mathrm{M}\{\tilde{\theta}(k+1)\}=\mathrm{M}\{\tilde{\theta}(k)\}+(\varepsilon-1)\mathrm{M}\{\theta(k)\}+\sqrt{1-\varepsilon^2}\,\mathrm{M}\{S(k+1)\}-\gamma\mathrm{M}\left\{\frac{x(k+1)\xi(k+1)}{\|x(k+1)\|^2+\delta}\right\}-\\
&-\gamma\frac{1}{N}\mathrm{M}\{\tilde{\theta}(k)\}+\gamma\mathrm{M}\left\{\frac{\delta}{\|x(k+1)\|^2+\delta}\right\}\mathrm{M}\{\tilde{\theta}(k)\}=\mathrm{M}\{\tilde{\theta}(k)\}+(\varepsilon-1)\mathrm{M}\{\theta(k)\}+\sqrt{1-\varepsilon^2}\,\mathrm{M}\{S(k+1)\}-\\
&-\gamma\frac{1}{N}\mathrm{M}\{\tilde{\theta}(k)\}+\frac{1}{(N-2)\sigma_x^2+\delta}\left(1-\frac{2\delta\sigma_x^2}{\left((N-2)\sigma_x^2+\delta\right)^2}\right)\mathrm{M}\{\tilde{\theta}(k)\}=\\
&=1-\gamma\frac{1}{N}\left(1-\frac{\delta N}{(N-2)\sigma_x^2+\delta}\left(1-\frac{2\delta\sigma_x^2}{\left((N-2)\sigma_x^2+\delta\right)^2}\right)\right)\mathrm{M}\{\tilde{\theta}(k)\}
\end{aligned} \tag{13}$$

З отриманого виразу (13) випливає, що для збіжності алгоритму (4) в середньому необхідно виконання нерівності

$$\left|1-\gamma\frac{1}{N}\left(1-\frac{\delta N}{(N-2)\sigma_x^2+\delta}\left(1-\frac{2\delta\sigma_x^2}{\left((N-2)\sigma_x^2+\delta\right)^2}\right)\right)\right|<1 \tag{14}$$

або

$$0<\gamma\frac{1}{N}\frac{\left((N-2)\sigma_x^2+\delta\right)^3-\delta N\left((N-2)\sigma_x^2+\delta\right)^2+2N\delta\sigma_x^2}{\left((N-2)\sigma_x^2+\delta\right)^3}<2,$$

тобто

$$0<\gamma<\frac{2N\left((N-2)\sigma_x^2+\delta\right)^3}{\left((N-2)\sigma_x^2+\delta\right)^3-\delta N\left((N-2)\sigma_x^2+\delta\right)^2+2N\delta\sigma_x^2}. \tag{15}$$

При виконанні (15) алгоритм (4) сходиться в середньому, тобто

$$\lim_{k\to\infty}\mathrm{M}\{\tilde{\theta}(k+1)\}=0.$$

З нерівності (14) для класичного алгоритму Качмажа ($\delta=0$) слід відоме умова $0<\gamma<2N$.

Для дослідження збіжності в середньоквадратичному помножимо (9) зліва на $\tilde{\theta}^T(k+1)$

$$\left\|\tilde{\theta}(k+1)\right\|^2 = \left\|\tilde{\theta}(k)\right\|^2 + 2\tilde{\theta}^T(k)\theta(k)(\varepsilon-1) + (\varepsilon-1)^2\left\|\tilde{\theta}(k)\right\|^2 +$$

$$+ 2\tilde{\theta}^T(k)S(k+1)\sqrt{1-\varepsilon^2} + 2(\varepsilon-1)\sqrt{1-\varepsilon}\theta^T(k)S(k) + (1-\varepsilon^2)\left\|S(k+1)\right\|^2 -$$

$$- 2\gamma\frac{\tilde{\theta}^T(k)x(k+1)}{\left\|x(k+1)\right\|^2+\delta}\left[\xi(k+1) + \tilde{\theta}^T(k)x(k+1)\right] -$$

$$- 2\gamma\frac{(\varepsilon-1)\theta^T(k)x(k+1)}{\left\|x(k+1)\right\|^2+\delta}\left[\xi(k+1) + \tilde{\theta}^T(k)x(k+1)\right] -$$

$$- 2\gamma\frac{\sqrt{1-\varepsilon^2}x^T(k+1)S(k)}{\left\|x(k+1)\right\|^2+\delta}\left[\xi(k+1) + \tilde{\theta}^T(k)x(k+1)\right] +$$

$$+ \gamma^2\frac{\left\|x(k+1)\right\|^2}{\left(\left\|x(k+1)\right\|^2+\delta\right)^2}\left[\xi(k+1) + \tilde{\theta}^T(k)x(k+1)\right]^2$$

Обчислимо $\mathrm{M}\left\{\left\|\tilde{\theta}(k+1)\right\|^2\right\}$.

З урахуванням статистичних властивостей сигналів і перешкод маємо

$$\mathrm{M}\left\{\left\|\tilde{\theta}(k+1)\right\|^2\right\} = \mathrm{M}\left\{\left\|\tilde{\theta}(k)\right\|^2\right\} + 2(1-\varepsilon)\sigma_S^2 - 2\gamma M\left\{\frac{(\tilde{\theta}^T(k)x(k+1))^2}{\left\|x(k+1)\right\|^2+\delta}\right\} +$$

$$+ \gamma^2 M\left\{\frac{\left\|x(k+1)\right\|^2}{(\left\|x(k+1)\right\|^2+\delta)^2}\xi^2(k+1)\right\} + \gamma^2 M\left\{\frac{(\tilde{\theta}^T(k)x(k+1))^2\left\|x(k+1)\right\|^2}{(\left\|x(k+1)\right\|^2+\delta)^2}\right\}. \qquad (16)$$

Скориставшись співвідношеннями (10), (11), отримуємо

$$\mathrm{M}\left\{\frac{(\tilde{\theta}^T(k)x(k+1))^2}{\left\|x(k=1)\right\|^2+\delta}\right\} = \left[\frac{1}{N}I - \frac{\delta}{(N-2)\sigma_x^2+\delta}\left(1 - \frac{2\sigma_x^2\delta}{\left((N-2)\sigma_x^2+\delta\right)^2}\right)\right]\mathrm{M}\left\{\left\|\tilde{\theta}(k)\right\|^2\right\}$$

$$\mathrm{M}\left\{\frac{\left\|x(k+1)\right\|^2}{(\left\|x(k+1)\right\|^2+\delta)^2}\xi^2(k+1)\right\} = \frac{(N-2)\sigma_x^2\sigma_\xi^2}{\left((N-2)\sigma_x^2+\delta\right)^2};$$

$$\mathrm{M}\left\{\frac{\left\|x(k+1)\right\|^4}{(\left\|x(k+1)\right\|^2+\delta)^2}\right\} = -2\delta\left(\frac{1}{(N-2)\sigma_x^2+\delta}\left(1 - \frac{2\sigma_x^2\delta}{\left((N-2)\sigma_x^2+\delta\right)^2}\right) - \frac{\delta^2}{\left((N-2)\sigma_x^2+\delta\right)^2}\right) =$$

$$= -2\delta\frac{\left((N-2)\sigma_x^2+\delta\right)^2 - 2\sigma_x^2\delta}{\left((N-2)\sigma_x^2+\delta\right)^3} + \frac{\delta^2}{\left((N-2)\sigma_x^2+\delta\right)^2} =$$

$$= \frac{\delta^2\left((N-2)\sigma_x^2+\delta\right) - 2\delta\left((N-2)\sigma_x^2+\delta\right)^2 + 4\delta^2\sigma_x^2}{\left((N-2)\sigma_x^2+\delta\right)^3}.$$

Після підстановки даних виразів в (16) і нескладних перетворень маємо

$$\mathrm{M}\left\{\left\|\tilde{\theta}(k+1)\right\|^2\right\} = \left\{1 - \frac{\gamma}{N}\left[(2-\gamma) - \frac{2(\gamma-1)\delta\left((N-2)\sigma_x^2+\delta\right)^2 + 2\sigma_x^2\delta}{\left((N-2)\sigma_x^2+\delta\right)^3} - \right.\right.$$

$$\left.\left. - \frac{\gamma\delta^2}{\left((N-2)\sigma_x^2+\delta\right)^2}\right]\right\}\mathrm{M}\left\{\left\|\tilde{\theta}(k)\right\|^2\right\} + \gamma^2\frac{(N-2)\sigma_x^2\sigma_\xi^2}{\left((N-2)\sigma_x^2+\delta\right)^2} + 2(1-\varepsilon)\sigma_S^2. \qquad (17)$$

Зі співвідношення (16) можна отримати вираз для асимптотичноїпомилки оцінювання

$$\mathrm{M}\left\{\left\|\tilde{\theta}(\infty)\right\|^2\right\}=\frac{\gamma^2 N(N-2)\sigma_x^2\sigma_\xi^2\left((N-2)\sigma_x^2+\delta\right)+2N(1-\varepsilon)\sigma_S^2\left((N-2)\sigma_x^2+\delta\right)^3}{\gamma[(2-\gamma)\left((N-2)\sigma_x^2+\delta\right)^3-2(\gamma-1)[\delta\left((N-2)\sigma_x^2+\delta\right)^2+2\sigma_x^2\delta^2-\gamma\delta^2\left((N-2)\sigma_x^2+\delta\right)}. \quad (18)$$

З огляду на (18), можна записати наступний вираз для асимптотичної помилки ідентифікації (8):

$$\mathrm{M}\left\{e^2(\infty)\right\}=\sigma_\xi^2+\frac{\gamma^2 N(N-2)\sigma_x^2\sigma_\xi^2\left((N-2)\sigma_x^2+\delta\right)+2N(1-\varepsilon)\sigma_S^2\left((N-2)\sigma_x^2+\delta\right)^3}{\gamma[(2-\gamma)\left((N-2)\sigma_x^2+\delta\right)^3-2(\gamma-1)[\delta\left((N-2)\sigma_x^2+\delta\right)^2+2\sigma_x^2\delta^2-\gamma\delta^2\left((N-2)\sigma_x^2+\delta\right)}.$$

З отриманих співвідношень слідують неасимптотичні і асимптотичні оцінки для алгоритму Качмажа (Уїдроу-Хоффа).

Так для стаціонарного випадку () отримуємо

$$\mathrm{M}\left\{\left\|\tilde{\theta}(k+1)\right\|^2\right\}=\left(1-\frac{1}{N}\right)\mathrm{M}\left\{\left\|\tilde{\theta}(k)\right\|^2\right\}+\frac{\sigma_\xi^2}{(N-2)\sigma_x^2};$$

$$\mathrm{M}\left\{\left\|\tilde{\theta}(\infty)\right\|^2\right\}=\frac{N\sigma_\xi^2}{(N-2)\sigma_x^2}. \quad (19)$$

Якщо ж ($\gamma\neq 1, \delta=0\ \varepsilon=1$), то

$$\mathrm{M}\left\{\left\|\tilde{\theta}(k+1)\right\|^2\right\}=\left(1-\gamma(2-\gamma)\frac{1}{N}\right)\mathrm{M}\left\{\left\|\tilde{\theta}(k)\right\|^2\right\}+\gamma^2\frac{\sigma_\xi^2}{(N-2)\sigma_x^2};$$

$$\mathrm{M}\left\{\left\|\tilde{\theta}(\infty)\right\|^2\right\}=\frac{\gamma N\sigma_\xi^2}{(2-\gamma)(N-2)\sigma_x^2}. \quad (20)$$

З наведених співвідношень випливає, що традиційний алгоритм Качмажа (Уїдроу-Хоффа) в стаціонарному випадку сходиться до області, яка визначається виразом (18), тобто співвідношенням дисперсії перешкод і корисного сигналу. При використанні ж модифікованого алгоритму Качмажа з спостерігатиметься збіжність в точку, якщо цей параметр задовольняє умовам Дворецького [19].

У нестаціонарному випадку при використанні модифікованої Марківського моделі () відповідні оцінки будуть мати вигляд

$$\mathrm{M}\left\{\left\|\tilde{\theta}(k+1)\right\|^2\right\}=\left(1-\frac{1}{N}\right)\mathrm{M}\left\{\left\|\tilde{\theta}(k)\right\|^2\right\}+\frac{\sigma_\xi^2}{(N-2)\sigma_x^2}+2(1-\varepsilon)\sigma_S^2;$$

$$\mathrm{M}\left\{\left\|\tilde{\theta}(\infty)\right\|^2\right\}=\frac{N\sigma_\xi^2}{(N-2)\sigma_x^2}+2(1-\varepsilon)\sigma_S^2. \quad (21)$$

тобто область збіжності визначається як ступенем нестаціонарності об'єкту, так і співвідношенням статистичних характеристик корисних сигналів і перешкод.

Для модифікованого алгоритму Качмажа і модифікованої Марківської моделі ($\gamma\neq 1, \delta=0\ \varepsilon\neq 1$) неасимптотичні і асимптотичні оцінки помилок оцінювання можуть бути записані таким чином:

$$\mathrm{M}\left\{\left\|\tilde{\theta}(k+1)\right\|^2\right\} = \left(1-\gamma(2-\gamma)\frac{1}{N}\right)\mathrm{M}\left\{\left\|\tilde{\theta}(k)\right\|^2\right\} + \frac{\gamma^2\sigma_\xi^2}{(N-2)\sigma_x^2} + 2(1-\varepsilon)\sigma_S^2;$$

$$\mathrm{M}\left\{\left\|\tilde{\theta}(\infty)\right\|^2\right\} = \frac{\gamma N\sigma_\xi^2}{(2-\gamma)(N-2)\sigma_x^2} + \frac{2(1-\varepsilon)\sigma_S^2}{\gamma(2-\gamma)}. \qquad (22)$$

З виразу (22) видно, що для компенсації помеховой складової (першого члена правій частині цього виразу) параметр γ повинен задовольняти умовам Дворецького. При цьому збільшується друга складова, яка характеризує нестаціонарність об'єкта. Вибір же будь-якого постійного $\gamma \in (0,1)$ відповідність в зону, розміри якої визначаються за формулою (22).

Деякі зауваження щодо вибору параметра регуляризації δ

Мета використання параметра регуляризації δ- підвищення обчислювальної стійкості алгоритму оцінювання. Однак при цьому виникає задача ефективного вибору цього параметра. Принципово в літературі це питання практично не досліджувалося. Це пояснюється, мабуть, тим, що для його вирішення потрібна досить велика кількість інформації про статистичні властивості корисних сигналів і перешкод, що впливають на властивості алгоритму. Деякі рекомендації були надані в роботах [21,22]. В роботі [14] запропоновано визначати на основі максимізації співвідношення сигнал / шум

$$ENR = \frac{\sigma_y^2}{\sigma_\xi^2} = \frac{\theta^T R_x \theta}{\sigma_\xi^2},$$

де $\sigma_y^2 = M\left\{y_n^2\right\}; \sigma_\xi^2 = M\left\{\xi_n^2\right\}; R_x = M\left\{x_n x_n^T\right\}.$

Пропонований в цій роботі вираз для визначення параметра регуляризації має вигляд

$$\delta = \frac{N(1+\sqrt{1+ENR})}{ENR}\sigma_x^2. \qquad (23)$$

Таким чином, значення параметра δ залежить від розмірності моделі N, дисперсії вхідного сигналу σ_x^2 і ENR. Якщо величина N відома, а σ_x^2 може бути визначена або оцінена, то величина ENR часто не відома.

В роботі [20] була розглянута можливість визначення оптимального значення параметра δ з умов максимізації швидкості збіжності алгоритму. Такий підхід дозволив отримати наступний вираз для δ_{n+1}^{opt}

$$\delta^{opt}(k+1) = \frac{N\sigma_\xi^2}{M\left\{\left\|\tilde{\theta}(k)\right\|^2\right\}}. \qquad (24)$$

Вираз (22) містить крім невідомої величини σ_ξ^2 і невідому величину $M\left\{\left\|\tilde{\theta}(k)\right\|^2\right\}$. Тому дані рекомендації являють скоріше теоретичний інтерес. Їх аналіз дозволяє зробити висновок про те, що величина $\delta^{opt}(k+1)$ повинна вибиратися змінною. При цьому, однак, при $ENR \to 0$ і $M\left\{\left\|\tilde{\theta}(k)\right\|^2\right\} \to 0$ $\delta^{opt}(k+1) \to \infty,$ тобто усувається вплив перешкоди ξ_n, проте алгоритм не буде відслідковувати нестаціонарність параметрів.

Моделювання.

Вирішувалося завдання ідентифікації нестаціонарного об'єкта, що описується рівнянням (1) з наступними параметрами: N = 10, причому з 10 параметрів 4 були нестаціонарними, представленими Марківською моделлю (5) з $\sigma_S^2 = 0,25$ та $\sigma_S^2 = 0,5$. Решта шість параметрів приймалися рівними $\theta_5 = 0,8; \theta_6 = 0,4; \theta_7 = -0,2; \theta_8 = 0,55; \theta_9 = -0,67; \theta_{10} = 0,15.$

В якості вхідного сигналу $x(k)$ і адитивної перешкоди $\xi(k)$ вибиралися послідовності нормально розподілених величин $x(k) \sim N(0;1)$, $\xi(k) \sim N(0;3).$

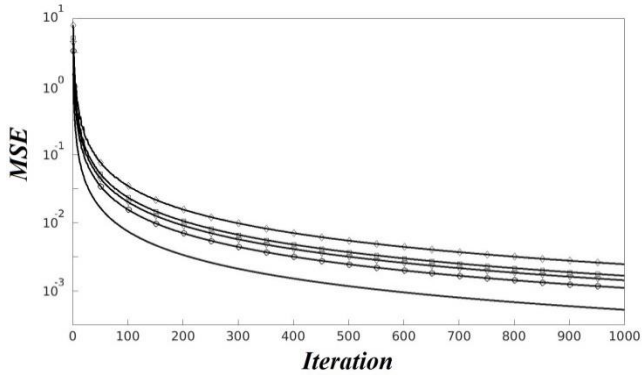
В якості критерію порівняння роботи алгоритмів використовувалася величина

$$MSE = M\left\{\left\|\tilde{\theta}(k+1)\right\|^2\right\}.$$

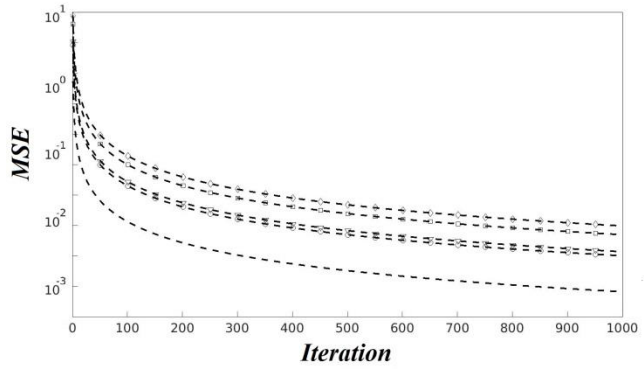
На рис.2-а, б наведені графіки зміни величини MSE для регуляризованого алгоритму Качмажа відповідно при різному виборі параметра релаксації γ. Так як при імітаційному моделюванні величина $\left\|\tilde{\theta}(k+1)\right\|^2$ відома, це робить можливим порівняння граничних (оптимальних) можливостей алгоритмів. Тому криві без маркерів на малюнках відповідають теоретично оптимального вибору параметрів $\gamma^{\text{опт}}$ відповідно до (20)), а решта - практичного вибору цих параметрів. Так криві з кружками відповідають вибору (завданням) $\gamma = 0,7$ і $\delta = 0$, криві трикутниками - вибору $\gamma = 0,7$ і $\delta = 0,01$, криві з квадратами - вибору $\gamma = 0,7$ і $\delta = 0,02$, криві з ромбами - вибору $\gamma = 0,6$ і $\delta = 0,02$.

На рис.2-в показані результати порівняно аналізу роботи досліджуваних алгоритмів при виборі відповідних $\gamma^{\text{опт}}$ (криві без маркерів) і завданні $\gamma = 0,7$ і $\delta = 0,01$ (криві з трикутниками).

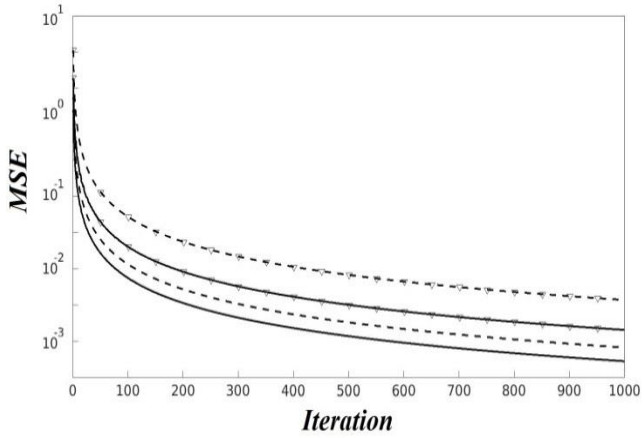
Аналізуючи результати моделювання, можна зробити висновок про те, що введення параметра регуляризації $\delta \neq 0$ приводить до уповільнення швидкості збіжності алгоритмів. При цьому, однак, не виникає проблеми розподілу на нуль, тобто стійкість алгоритмів підвищується.



а)



б)



в)

Рис. 2. Результати моделювання

Обговорення результатів дослідження

Дослідження, проведені в даній роботі, є продовженням і розвитком раніше проведених досліджень, описаних в [19]. Строго доведені в [19] результати, отримані при ідентифікації стаціонарних об'єктів, були застосовані для ідентифікації нестаціонарних об'єктів.

Отримані оцінки є досить загальними і залежать як від ступеня нестаціонарності об'єкту, так і від статистичних характеристик корисних сигналів і перешкод. Крім того, визначені вирази для оптимальних значень параметрів релаксації алгоритмів, що забезпечують їх максимальну швидкість збіжності. Так як дані вирази містять ряд невідомих параметрів (помилка оцінювання, ступінь нестаціонарності об'єкту), для їх практичного застосування слід скористатися оцінками цих параметрів. Так при ідентифікації в режимі on-line можна застосувати будь-яку рекуррентную процедуру їх оцінювання і використовувати одержувані оцінки для потактовой уточнення входять в алгоритми параметрів. При ідентифікації в режимі off-line слід здійснити корекцію отриманого після всіх обчислень результату.

Слід зазначити, що отримані в даній роботі оцінки є більш точними в порівнянні з існуючими. Тому при вирішенні практичних завдань дослідник може з більшою впевненістю попередньо оцінити можливості того чи іншого алгоритму і ефективність його застосування.

Висновок

Як показали результати досліджень, використання регулярізірующего добавки в алгоритмах ідентифікації, покращуючи стійкість алгоритмів, призводить до деякого уповільнення процесу побудови моделі. Визначено умови збіжності регуляризованого алгоритму Качмажа при оцінюванні нестаціонарних параметрів, описуваних модифікованою Марківською моделлю, при наявності перешкод виміру.

Отримані неасимптотичні і асимптотичні оцінки є досить загальними і залежать як від ступеня нестаціонарності об'єкту, так і від статистичних характеристик сигналів і перешкод. Якщо ж ці параметри не відомі, слід скористатися будь-якою рекурентною процедурою їх оцінювання і використовувати одержувані оцінки для уточнення параметрів, які входять в алгоритми.

Література

1. Dupac V. A dynamic stochastic approximation method / V .Dupac / Ann.Math.Statist.,-1965.-36.-P.1695-1702.

2. Цыпкин Я.З. Алгоритмы адаптации и обучения в нестационарных условиях / Я.З Цыпкин, А.И. Каплинский, К.А. Ларионов / Изв. АН СССР. Техн. Кибернетика.-1970.-5.-С.9-21

3. Kaczmarz S. Angenäherle Auflösung von Systemen linearer Gleichungen / S. Kaczmarz /Bull. Int. Acad. Polon. Sci. Lett., C 1, Sci. Math. Nat., Ser. A- 1937. – S. 355-357.

4. English translation: Kaczmarz S. Approximate solution of systems of linear equations / S. Kaczmarz / Int. J. of Control, 1993. – 57. – Pp. 1269-1271.

5. Nagumo I. A learning method for system identification / I. Nagumo, A. Noda / IEEE Trans. Autom. Control, 1967. – AC-12. - №3. – P.282-287.

6. Округ А.И. Динамический алгоритм Качмажа / А.И. Округ / Автоматика и телемеханика.-1981.-1.-С.74-79

7. Widrow B. Adaptive switching circuits. / B. Widrow, M. Hoff // IRE WESCON Convention Record. Part 4. New York: Institute of Radio Engineers.-1960. -pp. 96-104.

8. Widrow B. 30 years of adaptive neural networks: perceptron, madaline, and backpropagation/ B. Widrow, M. Lehr // Proc.IEE.-1990.-vol.78.- 9.- pp. 1415-1442.

9. Чадеев В.М. Определение динамических характеристик объектов в процессе их нормальной эксплуатации для целей самонастройки / В.М.Чадеев / Автоматика и телемеханика, 1964. – Т. 25. – 9. – С. 1302–1306.

10. Райбман Н.С. Адаптивные модели в системах управления / Н.С. Райбман, В.М. Чадеев. – М.: Сов. Радио. – 1966. – 156 с.

11. Руденко О.Г. Оценка скорости сходимости одношаговых устойчивых алгоритмов идентификации // Доклады АН УССР. – Сер. А. Физ-мат и техн. науки. – 1982. – №1. – С.64-66.

12. Ciochina S. An optimized NLMS algorithm for system identification./ S. Ciochină, C. Paleologu, J. Benesty / Signal Processing. – 2016. – 118. – P. 115-121.

13. Khong A. W. H. Selective-tap adaptive filtering with performance analysis for identification of timevarying systems, / A.W.H. Khong, P.A. Naylor / IEEE Trans. Audio, Speech, and Language Processing. –2007. – vol. 15. –P. 1681–1695.

14. Benesty J. On regularization in adaptive filtering / J. Benesty, C. Paleologu, S. Ciochină/IEEE Trans. Audio, Speech, Language Process. – 2011. – Vol. 19. – P. 1734-1742.

15. Paleologu C. An overview on optimized NLMS algorithms for acoustic echo cancellation/ C. Paleologu, S. Ciochină, J. Benesty, S.L. Grant // EURASIP J. Adv. Sig. Proc. – 2015: 97. – 19 p.

16. Bershad N. J. Performance comparison of RLS and LMS algorithms for tracking a first order Markov communications channel / N. J. Bershad, S. McLaughlin, C. F. N. Cowan // Proc. IEEE Int. Symposium on Circuits and Syst., 1990. –P. 266–270.

17. Wagner K. Towards analytical convergence analysis of proportion-type NLMS algorithms / K. Wagner, M. Doroslovacki // Proc. IEEE Int. Conf. Acoustics Speech Signal Processing, 2008.-P.3825–3828.

18. Либероль Б.Д. Исследование сходимости одношаговых адаптивных алгоритмов идентификации / Б.Д. Либероль, О.Г. Руденко, А.А. Бессонов / Проблемы управления и информатики.-2018.-5.-С.19-32.

19. Руденко О.Г., Бессонов А.А. Регуляризованный алгоритм обучения адалины в задаче оценивания нестационарных параметров / Управляющие системы и машины, 2019. –С. 22-30.

20. Вазан М. Стохастическая аппроксимация. / М. Вазан.- М.: Мир.1972.-289 с.

21. Тихонов А.Н. Методы решения некорректных задач. / А.Н. Тихонов, В.Я. Арсенин / М.: Наука, 1980.-223 с.

22. Демиденко Е.З. Линейная и нелинейная регрессии/ Е.З. Демиденко /– М.: Финансы и статистика, 1981.-302 с.

ГЛАВА 3
ГРАДІЄНТНІ АЛГОРИТМИ НАВЧАННЯ ЗГОРТАЛЬНИХ НЕЙРОННИХ МЕРЕЖ

Вступ

В даний час Deep Learning [1] демонструє майже сучасну продуктивність в різних завданнях в різних областях, таких як комп'ютерне зір і завдання обробки зображень, включаючи класифікацію, семантичну сегментацію і виявлення об'єктів, розпізнавання мови і обробку природної мови. Існує величезна кількість різних варіантів глибинних архітектур.

Після проривний результат в завданні класифікації ImageNet [2] були запропоновані різні види архітектури нейронних мереж, і продуктивність поліпшується з кожним роком. GoogLeNet [3], схоже, досяг вузького місця продуктивності в 2014 році завдяки традиційній структурі нейронної мережі з прямим зв'язком. Після цього були запропоновані нові структури нейронних мереж: ResNet [4], DenseNet [5] і CliqueNet [6], де пропущені з'єднання і щільні з'єднання, щоб полегшити навчання мережі і ще більше підштовхнути сучасний стан [7].

Штучна нейронна мережа (ШНМ) містить безліч вагових параметрів, пов'язаних на більш глибокому рівні топології мережі. Сигнал помилки вимірюється і передається назад в мережу за допомогою методів оптимізації. Оптимізація служить основою штучного навчання нейронної мережі. Існує дві категорії навчання, тобто алгоритми навчання похідних першого і другого порядку. Метод похідних першого порядку використовує інформацію градієнта для побудови наступної навчальної ітерації, тоді як похідні другого порядку використовують гессіан для обчислення ітерації на основі траєкторії оптимізації. Метод першого порядку спирається тільки на інформацію про градієнті. Потрібно серія точної настройки гіперпараметра для забезпечення оптимальної траєкторії на поверхні помилки. Численні поліпшення були зроблені з використанням адаптивного підходу до методу похідних першого порядку, щоб подолати потреби великої тонкої настройки [8].

Згідно з даними, що використовуються для обчислення градієнта цільової функції, методи градієнтного спуску можна розділити на три категорії: метод пакетного градієнтного спуску, метод міні-пакетного градієнта і метод стохастичного градієнта (SGD) [9].

Метод найшвидшого спуску - один з найпопулярніших алгоритмів для оптимізації і, безумовно, найпоширеніший спосіб оптимізації ШНМ. У той

же час кожна сучасна бібліотека глибокого навчання містить реалізації різних алгоритмів для оптимізації градієнтного спуску (наприклад, документація Lasagne 2, Caffe's 3 і Keras'4) [9].

Переважна методологія в навчанні глибокому навчанню захищає використання SGD. З огляду на великомасштабний набір даних і обмежену пам'ять обчислень, особливо на графічних процесорах, традиційний типовий метод пакетного градієнта і метод стохастичного градієнта не може проводити навчання ефективно і результативно.

Крім того, регулювання розміру кроку також грає важливу роль під час процедури навчання. Але дуже важко точно налаштувати розмір кроку в задачі, що вимагає великого набору даних і тривалого періоду навчання. Для вирішення цих проблем було запропоновано кілька варіантів типових градієнтних методів. У наступних розділах ми представимо основні рамки пов'язаних методів.

На сьогоднішній день одним з найбільш інтенсивно розвиваються наукових і технологічних напрямків є обробка і аналіз зображень. За останні роки було представлено ряд методів та алгоритмів, застосовуваних для вирішення цих завдань, серед яких одними з найбільш ефективних є штучні нейронні мережі (ШНМ). Використання персептрона, радіально-базисних мереж, автоенкодера, неокогнітрона, імовірнісних мереж виявилося досить ефективним при вирішенні широкого кола завдань. Однак новий поштовх до інтересу досліджень в даній області дало поява в 1998 р нового типу мереж - сверточних ІНС (СІнС).

Згорткова ШНМ (Convolution Neural Network, CNN) вперше була запропонована в [10] як розвиток моделі неокогнітрона, призначеного для ефективного розпізнавання зображень.

Згодом на основі CNN були побудовані мережі R-CNN (Regions With CNNs) для застосування CNN до задачі виявлення об'єктів. R-CNN створює обмежують рамки для кожного об'єкта на зображенні або пропозиції регіонів, використовуючи процес вибіркового пошуку. Fast R-CNN, збільшила продуктивність R-CNN, здійснюючи класифікацію об'єктів кожного регіону разом з більш жорсткими обмежуючими рамками. Наступна мережа Faster R-CNN поліпшила механізм генерації використовуваних в ній регіонів-кандидатів за рахунок обчислення регіонів не за початковим зображенням, а за картами ознак, отриманих з CNN. Для цього був доданий модуль під назвою Region Proposal Network (RPN). Нарешті, мережа Mask R-CNN розвиває архітектуру Faster R-CNN шляхом додавання ще однієї гілки, яка передбачає положення маски, що покриває знайдений об'єкт, і, таким чином вирішує вже завдання сегментації екземплярів.

При отриманні зображення мережа видає об'єкти (bbox), що обмежують рамки, класи (class) і маски (mask). Слід зазначити, що Mask R-CNN є найбільш швидкодіючої мережею на даний момент.

1. Структура згорткової нейронної мережі

Спочатку структура згорткової нейронної мережі створювалася з урахуванням особливостей будови деяких частин людського мозку, що відповідають за зір. В основу розробки таких мереж закладено три механізму:

- локальне сприйняття;
- - формування шарів у вигляді набору карт ознак (колективні ваги);
- - субдискретизація (підвибірка).

Під локальним сприйняттям мається на увазі, що на вхід нейрона надходить не все зображення, а тільки деяка його частина. Це дозволяє зберігати конфігурацію зображення при переході від шару до шару.

Ідея поділюваних ваг має на увазі, що до великої кількості зв'язків застосовується невеликий набір ваг, тобто кожна область зображення, на які воно розділене, буде оброблена одним і тим же набором ваг. При такому штучно створеному обмеженні ваг поліпшується властивість мережі до узагальнення.

СНС складається з шарів згортки, субдискретизації (підвибірки) і шарів повної нейронної мережі.

2. Шари згорткової нейронної мережі

CNN отримали свою назву від оператора «згортки». Основна мета згортки в разі CNN -вилучити елементи з вхідного зображення.

Згортка зберігає просторові відносини між пікселями, вивчаючи особливості зображення та використовуючи маленькі квадрати вхідних даних.

Кожен нейрон в площині згорткового шару отримує свої входи від деякої області попереднього шару (локальне рецептивне поле), тобто вхідне зображення попереднього шару сканується невеликим вікном і пропускається крізь набір ваг, а результат відображається на відповідний нейрон згорткового шару.

Підвибірковий шар зменшує масштаб площин шляхом локального усереднення значень виходів нейронів. Таким чином, досягається ієрархічна

організація. Наступні шари витягують найбільш загальні властивості, менше залежні від спотворень зображення.

За кожним згортковим шаром слід шар субдискретизації (підвибірки), або обчислювальний шар, який виробляє зменшення розмірності зображення шляхом локального усереднення значень виходів нейронів.

В архітектурі згорткової мережі прийнято вважати, що наявність ознаки важливіше інформації про його розташуванні. Тому з кількох сусідніх нейронів в карті ознак обирається максимальний і його значення вважається одним нейроном в карті ознак меншої розмірності.

Відмінність шару підвибірки від шару згортки полягає в тому, що в останньому області сусідніх нейронів перекриваються, чого не відбувається в шарі субдіскретізація.

Таким чином, CNN будується шляхом чергування шарів згортки і субдискретизації. На виході мережі зазвичай встановлюється декілька шарів повно нейронної мережі, на вхід яких подаються кінцеві карти ознак. Кожен нейрон цього шару є персептрон, який має нелінійну функцію активації.

3. Методи налаштування параметрів згорткової нейронної мережі

Для навчання згорткових нейронних мереж може застосовуватися як стандартний метод зворотного поширення помилки, так і його різні модифікації. В основі цього методу лежить алгоритм стохастичного градієнтного спуску (Stochastic Gradient Descent).

3.1 Навчання на основі стохастичного градієнта

Стохастичний градієнтний спуск (SGD) і його варіанти є найбільш широко вивченими алгоритмами для оптимізації в задачах машинного навчання і стохастичної апроксимації. В літературі по обробці сигналів SGD носить назву Least MeanSquares (LMS) алгоритм.

Звичайний градієнтний спуск описується співвідношеннями

$$\theta(k+1) = \theta(k) - \eta \nabla_\theta J(\theta(k)), \tag{1}$$

де θ - параметри мережі, $J(\theta(k))$ - цільова функція (функція втрат); η - параметр швидкості навчання (learning rate).

Збіжність алгоритму (1) в загальному випадку не гарантується, проте встановлено [11,12], що в разі опуклої функції $J(\theta(k))$ і при виконанні наступних умов:

$$\lim \eta(k) = 0; \quad \sum_{k=1}^{\infty} \eta(k) = \infty; \quad \sum_{k=1}^{\infty} \eta^2(k) < \infty. \tag{2}$$

процес градієнтного спуску буде збігатися.

Можливо два основні підходи до реалізації градієнтного спуску:

• Пакетний (batch), коли на кожній ітерації навчальна вибірка проглядається цілком, і тільки після цього змінюється. Це вимагає великих обчислювальних витрат.

• Стохастичний (stochastic / online), коли на кожній ітерації алгоритму з навчальної вибірки якимось (випадковим) чином обирається тільки один об'єкт. Таким чином, вектор θ налаштовується на кожний знову обираємий об'єкт.

Даному алгоритму притаманні такі недоліки:

• застрягання в локальних мінімумах або сідлових точкахмінімізіруемого функціоналу.

• Повільна збіжність внаслідок складного ландшафту цільової функції, коли плато чергуються з регіонами сильної нелінійності (похідна на плато практично дорівнює нулю, а раптовий обрив, навпаки, може занадто змінити оцінку параметра).

• Деякі параметри оновлюються значно рідше інших, особливо коли в даних зустрічаються інформативні, але рідкісні ознаки, що погано позначається на нюансах узагальнюючого правила мережі. З іншого боку, надання занадто великої значимості взагалі всім рідко зустрічається ознаками може привести до перенавчання.

• Занадто малі значення параметра η призводять до повільної збіжності і застрягання в локальних мінімумах, занадто великі -до «проскакування» вузьких глобальних мінімумів або зовсім розбіжність.

Використання методів другого порядку, розглянутих вище, вимагає обчислення гессіан - матриці похідних по кожній парі параметрів, а, для методу Ньютона-ще й зворотний до неї, тобто реалізація даних методів пов'язана зі значними обчислювальними витратами.

У зв'язку з цим на практиці широкого поширення набули методи, засновані на методі стохастичного градієнта і володіють в порівнянні з ним цілу низку переваг [13,14].

Розглянемо ці методи більш докладно.

3.2. Стохастичний середній градієнт (SAG) [14]

Метод стохастичного середнього градієнта (SAG) - це метод зменшення дисперсії, запропонований для поліпшення швидкості збіжності. Ітерації SAG приймають форму

$$\theta(k+1) = \theta(k) - \frac{\alpha(k)}{n}\sum_{i=1}^{n} y_i(k),$$
$$y_i = \begin{cases} \nabla_\theta J_i(\theta(k)) & if \quad i = i_k, \\ y_{i-1} & otherwise. \end{cases} \quad (3)$$

Де $\alpha(k)$-швидкість навчання, i_k - випадковий індекс, що обирається на кожній ітерації.

По суті, SAG зберігає в пам'яті градієнт по кожній функції в сумі функцій, що оптимізуються. На кожній ітерації значення градієнта тільки однієї такої функції обчислюється і оновлюється в пам'яті. Метод SAG оцінює загальний градієнт на кожній ітерації шляхом усереднення значень градієнта, що зберігаються в пам'яті. Як і методи SG, вартість кожної ітерації не залежить від кількості функцій в сумі.

Витрати на обчислення не відрізняються від SGD, але накладні витрати пам'яті набагато більше. Це типовий спосіб використання простору для економії часу. Було показано, що SAG є алгоритмом лінійної збіжності, який набагато швидше, ніж SGD, і має великі переваги в порівнянні з іншими алгоритмами стохастичного градієнта.

Однак метод SAG застосовують тільки в разі, коли функція втрат є гладкою, а цільова функція є опуклою, наприклад, проблеми опуклого лінійного прогнозування.

В цьому випадку SAG досягає більш високої швидкості збіжності, ніж SGD. Крім того, при деяких специфічних проблемах він може навіть забезпечити кращу збіжність, ніж при стандартному градієнтному спуску.

3.3. Стохастичний градієнт зменшення дисперсії (SVRG) [15]

Оскільки метод SAG застосовують тільки до гладких і опуклих функцій і він повинен зберігати градієнт кожного зразка, його незручно застосовувати в неопуклих нейронних мережах. Метод стохастичного зменшення градієнта дисперсії (SVRG) був запропонований для підвищення ефективності оптимізації в складних моделях. Алгоритм SVRG підтримує середній інтервальний градієнт, обчислюючи градієнти всіх вибірок на кожній ітерації:

$$\tilde{\mu} = \frac{1}{N}\sum_{i=1}^{N} g_i(\tilde{\theta}), \quad (4)$$

де $\tilde{\theta}$ -параметр поновлення інтервалу.

Параметр інтервалу $\tilde{\theta}$ містить середню пам'ять всіх градієнтів вибірки за минулий час для кожного інтервалу часу. SVRG вибирає форму випадковим чином і виконує поновлення градієнта для поточних параметрів:

$$\theta(k+1) = \theta(k) - \eta(g_i(k)(\theta(k) - g_i(k)(\tilde{\theta}) + \tilde{\mu}). \quad (5)$$

Градієнт розраховується до двох разів на кожному оновленні. Після $_w$ ітерацій виконується $\tilde{\theta} \leftarrow \theta_w$ і починаються такі ітерації. Завдяки цим оновленням $\theta(k+1)$ і інтервалу оновлення параметр $\tilde{\theta}$ буде збігатися до оптимального, а потім θ^* і $\tilde{\mu} \to 0$.

$$g_i(k)(\theta(k) - g_i(k)(\tilde{\theta}) + \tilde{\mu} \to g_i(k)(\theta(k) - g_i(k)(\theta^*) \to 0. \quad (6)$$

SVRG пропонує життєво важливу концепцію, яка називається зменшенням дисперсії. Ця концепція пов'язана з аналізом збіжності SGD, в якому необхідно передбачити, що існує постійна верхня межа для дисперсії градієнтів. Ця постійна верхня межа означає, що SGD не може досягти лінійної збіжності. Однак в SVRG верхня межа дисперсії може безперервно зменшуватися завдяки спеціальному елементу оновлення, завдяки чому досягається лінійна збіжність.

Стратегії SAG і SVRG пов'язані зі зменшенням дисперсії. У порівнянні з SAG SVRG не потрібно тримати все градієнти в пам'яті, що означає, що ресурси пам'яті зберігаються, і це може ефективно застосовуватися для вирішення складних завдань. Експерименти показалигарну продуктивність SVRG для неопуклої нейронної мережі [15]. Існує також багато варіантів таких алгоритмів стохастичної оптимізації лінійної збіжності, таких як, наприклад, алгоритм SAGA [16].

3.4 Momentum [17]

Замість того щоб залежати тільки від поточного градієнта для поновлення ваги, в градієнтному спуску з імпульсом поточний градієнт

замінюється на (означає швидкість), експонентну ковзаючу середню поточного і минулих градієнтів (тобто до часу

$$\theta(k+1) = \theta(k) - \eta v(k+1); \quad (7)$$

$$v(k+1) = \gamma v(k) + (1-\gamma)\nabla_\theta J(\theta(k), \quad (8)$$

де $\gamma = 0,9$.

Пізніше це оновлення імпульсу стає стандартним оновленням для компонента градієнта

3.5 NAG (Nesterov Accelerated Gradient) [18]

Даний алгоритм реалізує ідею накопичення імпульсу, використовуючи інформацію про зміну кожного параметра у вигляді експоненційного змінного середнього

$$v(k+1) = \gamma v(k) + (1-\gamma)x. \quad (9)$$

В алгоритмі накопичується градієнт цільової функції мережі

$$v(k+1) = \gamma v(k) + \eta\nabla_\theta J(\theta(k)), \quad (10)$$

$$\theta(k+1) = \theta(k) - v(k). \quad (11)$$

який і використовується при корекції параметрів

Більш точні співвідношення для алгоритму NAG мають вигляд

$$\theta(k+1) = \theta(k) - v(k+1). \quad (12$$

$$v(k+1) = \gamma v(k) + \eta\nabla_\theta J(\theta(k) - \gamma v(k)); \quad (13)$$

3.6 Adagrad [19]

Adagrad (adaptive gradient) враховує частоту активації нейронів шляхом зберігання для кожного параметра мережі суму квадратів його оновлень. При цьому використовується модифікована формула поновлення

$$M\{g^2(k+1)\} = \gamma M\{g^2(k)\} + (1-\gamma)g^2(k), \quad (14)$$

а корекція параметрів здійснюється за правилом

$$\theta(k+1) = \theta(k) - \frac{\eta}{\sqrt{G(k)+\varepsilon}} g(k), \quad (15)$$

де $G(k)$ - сума квадратів оновлень, а ε - згладжує параметр, необхідний для того, щоб уникнути поділу на 0. У часто обновлялись в минулому величина параметра $G(k)$ буде великою (великий знаменник в (15)), тобто параметр зміниться незначно. Рідко змінювався параметр зміниться суттєво. Параметр зазвичай вибирають порядку 10^{-6} - 10^{-8}.

3.7 RMSProp [20]

Недолік Adagrad в тому, що в (15) може збільшуватися скільки завгодно, в результаті чого через деякий час оновлень стають занадто маленькими, що призводить до паралічу алгоритму. RMSProp і Adadelta покликані виправити цей недолік. RMSprop - це неопублікований метод адаптивної швидкості навчання, запропонований Дж. Хінтоном в лекції 6 його курсу Coursera.

З цією метою Adagrad модифікується таким чином, щоб часто оновлювані ваги коректувалися рідше. Для цього замість повної суми оновлень використовується усереднений по історії квадрат градієнта, тобто ковзне середнє виду

$$M\{g^2(k+1)\} = \gamma M\{g^2(k)\} + (1-\gamma)g^2(k), \quad (16)$$

тоді замість (15) отримаємо

$$\theta(k+1) = \theta(k) - \frac{\eta}{\sqrt{M\{g^2(k)\}+\varepsilon}} g(k). \quad (17)$$

Так як знаменник являє собою корінь із середнього квадратів градієнтів

$$RMS\{g(k)\} = \sqrt{M\{g^2(k)\} + \varepsilon}, \quad (18)$$

алгоритм отримав назву RMSProp - root mean square propagation.

3.8 Adadelta [21]

Adadelta є продовженням Adagrad, метою якого є недопусканіе монотонного зменшення швидкості навчання. Замість того, щоб накопичувати все градієнти минулого квадрата, Adadelta обмежує вікно накопичених градієнтів минулого фіксованим раз мером.

Adadelta відрізняється від RMSProp тим, що ми додаємо в чисельник (15) стабілізуючий член пропорційний RMS від $\Delta\theta(k)$. На кроці $k+1$ значення ще не відомо, тому оновлення параметрів відбувається в три етапи, а не в два: спочатку накопичується квадрат градієнта, потім оновлюється θ, після чого оновлюється $RMS\{\Delta\theta(k)\}$, тобто

$$\theta(k+1) = \theta(k) - \frac{RMS\{\Delta\theta(k)\}}{RMS\{g(k)\}} g(k); \quad (19)$$

$$M\{\|\Delta\theta(k+1)\|^2\} = \gamma M\{\|\Delta\theta(k)\|^2\} + (1-\gamma)\|\Delta\theta(k)\|^2; \quad (20)$$

$$RMS\{\Delta\theta(k)\} = \sqrt{M\{\|\Delta\theta(k)\|^2\} + \varepsilon}. \quad (21)$$

Для RMSProp, Adadelta і для Adagrad не потрібно дуже точно підбирати швидкість навчання - досить його наближене значення. Зазвичай радять почати підгін η з 0,1-1, а γ так і залишити 0,9. Чим ближче γ до 1, тим довше RMSProp і Adadelta з великим $RMS\{\Delta\theta(k)\}$ будуть сильно оновлювати мало використовувані ваги. Якщо ж $\gamma \approx 1$ і $RMS\{\Delta\theta(k)\} = 0$, то Adadelta буде довго «з недовірою» ставитися до рідко використовуваних ваг, що може привести як до паралічу алгоритму, так і викликати навмисно «жадібну» поведінку, коли алгоритм спочатку оновлює нейрони, які кодують найкращі ознаки.

3.9 Adam [22]

Adam (adaptive moment estimation) - ще один оптимізаційний алгоритм. Він поєднує в собі і ідею накопичення руху і ідею слабшого поновлення ваг для типових ознак.

За аналогією з (9) маємо:

$$m(k+1) = \beta_1 m(k) + (1-\beta_1) g(k). \quad (22)$$

Від алгоритму Нестерова Adam відрізняється тим, що в ньому накопичується не $\Delta\theta$, а значення градієнта. Для отримання інформації про зміну градієнта в [9] запропоновано оцінювати ще й середню нецентрованого дисперсію:

$$v(k+1) = \beta_2 v(k) + (1-\beta_2) g^2(k). \quad (23)$$

Так як це, по суті, являє собою вираз (16), то даний алгоритм не відрізняється від RMSProp. Автори Adam пропонують в якості значень за замовчуванням $\beta_1 = 0,9, \beta_2 = 0,999, \varepsilon = 10^{-8}$ і стверджують, що алгоритм працює краще або приблизно так само, як і всі попередні алгоритми на широкому наборі датасета за рахунок початкового калібрування.

3.10 Adamax [23]

Алгоритм Adamax є модифікацією алгоритму Adam, в якому замість дисперсії використовується інерційний момент розподілу градієнтів довільного ступеня p. Хоча це може привести до нестабільності обчислень, на практиці випадок $p \to \infty$, працює на диво добре

$$v(k+1) = \beta_2^p v(k) + (1-\beta_2^p) \left| g(k) \right|^p. \quad (24)$$

3.11 Nadam [24]

Алгоритм Nadam (Nesterov-acceleratedAdaptiveMomentEstimation) є модифікацією алгоритму Нестерова з адаптацією параметрів імпульсу

$$\theta(k+1) = \theta(k) - \frac{\eta}{\sqrt{G(k)+\varepsilon}}\left(\beta_1 \hat{g}(k) + \frac{1-\beta_1}{1-\beta_2}\nabla_\theta J(\theta(k)\right), \quad (25)$$

$$\hat{g}(k) = \frac{g(k)}{1-\beta_1}; \quad (26)$$

$$\hat{G}(k) = \frac{G(k)}{1-\beta_2}; \quad (27)$$

$$g(k+1) = \beta_1 g(k) + (1-\beta_1)\nabla_\theta J(\theta(k); \quad (28)$$

$$G(k+1) = \beta_2 G(k) + (1-\beta_2)\left[\nabla_\theta J(\theta(k)\right]^2; \quad (29)$$

$\alpha = 0.002; \beta_1 = 0.9; \beta_2 = 0.999\ \varepsilon = 10^{-7}$.

3.12 AMSGrad [25]

Іншим варіантом Adam є AMSGrad. Цей варіант переглядає компонент адаптивної швидкості навчання в Adam і змінює його, щоб гарантувати, що поточне значення *G* завжди було б більше, ніж попередній часовий крок.

$$\theta(k+1) = \theta(k) - \frac{\eta}{\sqrt{\hat{G}(k)+\varepsilon}} g(k); \quad (30)$$

$$\hat{G}(k+1) = \max\left(\hat{G}(k), G(k+1)\right); \quad (31)$$

$$g(k+1) = \beta_1 g(k) + (1-\beta_1)\nabla_\theta J(\theta(k); \quad (32)$$

$$G(k+1) = \beta_2 G(k) + (1-\beta_2)\left[\nabla_\theta J(\theta(k)\right]^2; \quad (33)$$

$$G(k+1) = \beta_2 G(k) + (1-\beta_2)\left[\nabla_\theta J(\theta(k)\right]^2; \quad (34)$$

$\alpha = 0.001; \beta_1 = 0.9; \beta_2 = 0.999\ \varepsilon = 10^{-7}$.

3.13 WNGrad [26]

В алгоритмі WNGrad (weight normalization Grad) використовується метод динамічного оновлення швидкості навчання відповідно до отриманих досі градієнтів

$$\theta(k+1) = \theta(k) - \frac{1}{b(k)} \nabla_\theta J(\theta(k)), \quad (35)$$

$$b(k+1) = b(k) + \frac{1}{b(k)} \left\| \nabla_\theta J(\theta(k)) \right\|. \quad (36)$$

За результатами численних експериментів WNGrad є конкурентом простому стохастичному градієнтному спуску з точки зору стійкості і помилки узагальнення при навчанні нейронних мереж. В роботі [12] запропоновано модифікації WNGrad, в яких використовується імпульс (WN-Adam and WNGrad-Momentum)

Висновок

У роботі проведено порівняльній аналіз алгоритмів градієнтного навчання згортальніх нейронних мереж при вірішенні завдання візуального розпізнавання. Дослідження показують, що для цього завдання досить ефективним є простий алгоритм градієнтного спуску. Використання імпульсів в розглянутих модіфікаціях прізводить до деякого поліпшення процесу розпізнавання, але збільшіло вартість обчислення. У розглянутих завданнях найбільш ефективним алгоритмом є Adamax. В [4] рекомендується завжди починати з оптімізатора Adam, незалежно від архітектури нейронної мережі та проблемних областей, в якіх він використовується. На нашу думку, при вирішенні завдань розпізнавання слід використовувати алгоритм Adamax. Однак, проблема вибору оптимальних значень параметрів алгоритмів, що забезпечують максимальну швидкість навчання, залишається відкритою.

Література

1. Гудфеллоу Я. Глубокое обучение/ Я. Гудфеллоу, И. Бенджио, А. Курвилль / пер. с анг.. – 2-е изд., испр. – М.: ДМК Пресс, 2018. – 652 с

2. Николенко С. Глубокое обучение. / С. Николенко, А. Кадурин, Е. Архангельская – СПб.: Питер, 2018. – 480 с.

3. Бринк Х. Машинное обучение. / Х. Бринк, Д. Ричардс, М. Феверолф –СПб.: Питер, 2017. –336 с.

4. A. Krzhevsky, I. Sutshever, and G. Hinton. "ImageNet classification with deep convolutional neural networks". In Advances in neural information processing systems, pp. 1097–1105, 2012

5. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. "Going deeper with convolutions". In CVPR, 2015. CoRR,abs/1409.4842, 2014.

6. He, K., Zhang, X., Ren, S., and Sun, J. Delving "Deepinto Rectifiers: Surpassing Human-Level Performanceon ImageNet Classification". ArXiv e-prints, February2015

7. Gao Huang, Zhuang Liu, Kilian Q. Weinberger, and Laurens van der Maaten. "Densely connected convolutional networks". arXiv preprint arXiv:1608.06993, 2016

8. Yang, Y., Zhong, Z., Shen, T., Lin, Z.: "Convolutional neural networks with alternately updated clique". In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition. pp. 2413–2422, 2018

9 Huan Li, Yibo Yang, Dongmin Chen, Zhouchen Lin "Optimization Algorithm Inspired Deep Neural Network Structure Design". In: Proc. of Machine Learning Research, 95 pp. 614-629, 2018

10. LeCun Y., Bottou L., Bengio Y., Haffner P., Gradient Based Learning Applied to Document Recognition // Proc. of IEEE, 1998. V. 86, №11. —P. 2278-2324.

11. Robbins H., Monro S. A stochastic approximation method. //Ann. Math. Stat. 1951., vol. 22, Vol. 22.- No. 3. - P. 400-407

12. Вазан М. Стохастическая аппроксимация. – М.: Мир, 1972. – 295 с.

13. Sebastian Ruder, "An overview of gradient descent optimization algorithms". arXiv:1609.04747v2 [cs.LG] 15 Jun 2017

14.Oppermann A. Optimization Algorithms in Deep Learning https://www.deeplearning-academy.com/p/ai-wiki-optimization-algorithms

15. Nicolas Le Roux Mark Schmidtmark Francis Bach, A Stochastic Gradient Method with an Exponential Convergence Rate for Finite Training Sets arXiv:1202.6258v4 [math.OC] 11 Mar 2013

15. R. Johnson and T. Zhang, "Accelerating stochastic gradient descent using predictive variance reduction". In Advances in Neural Information Processing Systems, 2013, pp. 315–323.

16. Aaron Defazio, Francis Bach, Simon Lacoste-Julien, "SAGA: A Fast Incremental Gradient Method With Support for Non-Strongly Convex Composite Objectives" arXiv:1407.0202v3 [cs.LG] 16 Dec 2014

17. Поляк Б.Т. О некоторых способах ускорения сходимости итерационных методов // Журнал. вычисл. матем. и матем. физики. – 1964. – Т. 4. –5. – С. 791–803.

18. Нестеров Ю.Е. Метод минимизации выпуклых функций со скоростью сходимости // Докл. АН СССР. – 1983. – Т. 269.- Вып. 3. – С. 543–547.

19. Duchi, J., Hazan, E., & Singer, Y. (2011).Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. Journal of Machine Learning Research, 12, 2121–2159. Retrieved from http://jmlr.org/papers/v12/duchi11a.html

20. A. Tieleman, T.; Hinton, G. Lecture 6.5-"rmsprop: Divide the gradient by a running average of its recent magnitude". COURSERA: Neural Networks Mach. Learn. 2012, 4, 26–31

21. Zeiler M. D. (2012). ADADELTA: An Adaptive Learning Rate Method. Retrieved from http://arxiv.org/abs/1212.5701

22. Kingma, D. P., & Ba, J. L. (2015). Adam: a Method for Stochastic Optimization. International Conference on Learning Representations, 1–13.

23. Zhanhong Jiang, Aditya Balu, Sin Yong Tan,Young M Lee, Chinmay Soumik Sarkar "On Higher-order Moments in Adam" / arXiv:1910.06878v1 [cs.LG] 15 Oct 2019

24. Dozat, T. (2016). Incorporating Nesterov Momentum into Adam. ICLR Workshop, (1), 2013–2016.

25. Reddi S.J., Kale S., Kumar S., On the convergence of adam and beyond // Int. Conference on Learning Representations (ICLR), Vancouver, Canada, 2018.- 23 p.

26. Wu X., Ward R,. Bottou L. WNGrad: Learn the Learning Rate in Gradient Descent .// arXiv:1803.02865v1 [stat.ML] 7 Mar 2018

ГЛАВА 4
КЛАСИФІКАЦІЯ ДАНИХ НА ОСНОВІ ГІБРИДНИХ МОДЕЛЕЙ ШТУЧНИХ ІМУННИХ СИСТЕМ

Вступ і постановка задачі

В теперішній час існує велика кількість методів класифікації, реалізованих на основі різних принципів навчання і поділу об'єктів, що групуються. Основними підходами, що використовуються для цього, є [1, 2]: ієрархічний, ймовірнісно-статистичний та оптимізаційний підходи, а також підходи на основі використання теорії графів і методів моделювання систем штучного інтелекту, що функціонують на основі біологічних принципів організації обчислень. Такими системами є штучні нейронні мережі (ШНМ), еволюційні алгоритми, штучні імунні системи (ШІС), системи моделювання колективного розуму та ін. Їх основною перевагою є можливість самонавчання і гнучкість управління процесом угруповання об'єктів.

При вирішенні практичних завдань виникає необхідність розробки моделі класифікації, що дозволяє змінювати спосіб угруповання вихідних даних в процесі роботи. При цьому модель повинна бути адаптивною і поєднувати особливості різних способів навчання. Апарат ШІС [3] дозволяє проводити угруповання даних різними способами, а також змінювати не тільки структуру мережі імунних об'єктів, а й методи вирішення поставленого завдання. Використання адаптивної імунної моделі дозволяє групувати дані або на основі навчальної вибірки (НВ), або в процесі імунного навчання (ІМН). При класифікації даних в процесі ІМН об'єкти, що класифікуються, представляються популяцією антитіл, а об'єкти НВ – популяцією антигенів. В процесі ІМН антитіла клонуються, піддаються мутації і відбору, внаслідок чого досягають стану специфічності з антигенами НВ.

Класифікація даних на основі імунного підходу може бути виконана з контрольованим і неконтрольованим навчанням, а також шляхом автоматичної класифікації [4, 5]. Особливістю класифікації з контрольованим навчанням є використання НВ. Процес класифікації умовно розбивається на три основних етапи: підготовчий етап PRP, етап імунного навчання LRN і етап уточнення меж вихідних класів CLS. Підготовчий етап необхідний для визначення даних, які можуть бути класифіковані в процесі ІМН, і визначення даних, які класифікуються на завершальному етапі і не

використовуються в процесі ІМН.

Особливістю кластеризації на основі імунного підходу є відсутність НВ, яка складається з початкових антигенів, що призводить до необхідності формування популяції антигенів з набору антитіл. Процес кластеризації на основі імунного підходу також містить три основні етапи. При цьому велика увага приділяється підготовчому етапу, так як на даному етапі відбувається формування популяції антигенів для пошуку центрів кластерів, що формуються.

Завдання автоматичної класифікації – не тільки розподіл згрупованих даних між множиною вихідних класів, а й виділення кластерів для даних, які не можуть бути класифіковані. Автоматична класифікація поділяється на два етапи [4, 5]: 1) класифікація частини вихідних даних при використанні НВ без проведення ІМН; 2) кластеризація даних, що залишилися, в процесі ІМН.

Таким чином, рішення задачі класифікації на основі імунного підходу зводиться до вирішення задачі розділення даних між заздалегідь відомими групами, або формування нових груп з вихідної множини даних.

Найбільш поширеними моделями ШІС, які використовуються для класифікації даних, є моделі клонального відбору і штучної імунної мережі [3]. Модель клонального відбору є однією з найбільш простих в реалізації імунних моделей. Найбільш поширеними алгоритмами, що реалізують модель клонального відбору, є алгоритми CLONALG і BCA [3]. Основна відмінність між даними алгоритмами полягає в способі організації обробки антитіл в процесі ІМН, використанні оператора первинного відбору і особливостей клонування антитіл. В [4] розглянуто вирішення задач класифікації та кластеризації об’єктів з використанням імунної моделі клонального відбору.

Модель штучної імунної мережі є однією з найбільш поширених моделей, що використовуються при вирішенні різних практичних завдань. Це обумовлюється великими можливостями, вираженими у взаємодії антитіл, що класифікуються, не тільки з навчальними антигенами, а й між собою. Найбільш поширеними алгоритмами, що реалізують дану імунну модель, є алгоритми aiNET і opt-aiNET [3]. Алгоритм opt-aiNET є модифікацією алгоритму aiNET, призначеного для вирішення специфічних завдань оптимізації. В [5] для вирішення задач класифікації, кластеризації та автоматичної класифікації на основі моделі імунної мережі базовим був обраний алгоритм aiNET, який було модифіковано для підвищення ефективності класифікації.

Гібридизація імунних методів і алгоритмів інтелектуальної обробки інформації проводиться з метою підвищення їх ефективності, що виражається в підвищенні швидкості або точності роботи. Стосовно задачі класифікації на основі імунного підходу модифікація імунних моделей полягає в використанні різних не імунних підходів до вирішення завдання угруповання об'єктів. В [5] були запропоновані гібридні алгоритми класифікації, кластеризації та автоматичної класифікації даних, що функціонують на основі алгоритму CLONALG в поєднанні із методами k-найближчих сусідів (kNN), k-середніх (k-means) і нечіткою логікою. Однак можливості алгоритму CLONALG обмежені в порівнянні з іншими імунними моделями, які передбачають не тільки міжпопуляційні взаємодії, але і взаємодії всередині популяції антитіл або клонів. Тому для підвищення ефективності класифікації з метою формування імунних гібридів, в даній роботі за основу був обраний модифікований алгоритм імунної мережі aiNETm [4].

Метою даної роботи є розробка гібридних моделей на основі моделі імунної мережі для класифікації даних при використанні методу k найближчих сусідів (kNN), кластеризації даних при використанні методу k-середніх (k-means) і автоматичної класифікація даних на основі нечіткого походу.

1. Імунне навчання методів класифікації

Застосування імунних моделей і алгоритмів при вирішенні задачі класифікації характеризується рядом особливостей, пов'язаних з організацією навчання та угруповання об'єктів.

Залежно від постановки завдання класифікації, антитіла і антигени можуть представляти різні види об'єктів [4, 5]:

– при класифікації з контрольованим навчанням антигени представляють об'єкти вихідних класів, ознаки яких відтворюються антитілами, що представляють класифікуються об'єкти;

– при класифікації з неконтрольованим навчанням (кластеризації) в умовах відсутності НВ, кожне антитіло є антигеном для інших імунних об'єктів. При цьому антитіла за допомогою клонування і мутації відтворюють ознаки інших антитіл, що є по відношенню до них антигенами. Отримані групи специфічних антитіл є кластерами.

Для визначення ступеня взаємодії між імунними об'єктами (антитілами, клонами та антигенами) використовується спеціальний критерій

афінності між i-м антитілом та j-м антигеном [4, 5]:

$$aff_{ij} = \frac{1}{1 + d_{ij}}, \tag{1}$$

де d_{ij} – відстань між i-м антитілом та j-м антигеном.

При організації ІМН використовуються послідовна або популяційна обробка антитіл. При послідовній обробці антитіл на етапі ІМН виконання ІО проводиться по черзі для кожного антитіла. Відповідно до цього для кожного антитіла формується окрема популяція клонів, яка потім піддається мутації і редагуванню. На рівні ІО даний процес представляється в такий спосіб:

$$\text{Iteration}\ (AB, AG) = \left\langle \begin{array}{l} \text{Presentation}(AG, ab_i) \rightarrow \\ \rightarrow \text{Cloning}(ab_i, CL) \rightarrow \\ \rightarrow \text{Mutation}(CL) \rightarrow \\ \rightarrow \text{Presentation}(AG, CL) \rightarrow \\ \rightarrow \text{Editing}(CL, ab_i) \end{array} \right\rangle^{1}_{AB}, \tag{2}$$

де AB – популяція антитіл; AG – популяція антигенів НВ; ab_i – i-е антитіло, що обробляється;

$\text{Presentation}(AG, ab_i)$ – ІО представлення антигенів i-му антитілу;

$\text{Cloning}(ab_i, CL)$ – ІО клонування i-го антитіла;

$\text{Mutation}(CL)$ – ІО мутації клонів *CL*;

$\text{Presentation}(AG, CL)$ – ІО представлення антигенів клонам;

$\text{Editing}(CL, ab_i)$ – ІО редагування клонів и антитіл;

$\langle\ \rangle^1$ – позначення циклу, що виконується один раз для всіх антитіл.

При популяційній обробці антитіл на етапі ІМН дії ІО піддається відразу вся популяція антитіл. Після представлення антигенів антитілам відбувається клонування всієї популяції антитіл, а не кожного антитіла окремо. Після цього для всіх сформованих клонів виконується мутація. У процесі редагування популяції антитіл і множини клонів відбувається стиснення мережі шляхом вибору об’єктів з максимальними значеннями афінності до антигенів і специфічних клітин пам’яті. На рівні ІО процес навчання може бути описаний в такий спосіб:

$$\text{Iteration}\ (AB, AG) = \left\langle \begin{array}{l} \text{Presentation}\left(ag_i, AB\right) \rightarrow \\ \rightarrow \text{Selection}\left(AB, AB'\right) \rightarrow \\ \rightarrow \text{Cloning}\left(AB', CL\right) \rightarrow \\ \rightarrow \text{Mutation}\left(CL\right) \rightarrow \\ \rightarrow \text{Presentation}\left(ag_i, CL\right) \rightarrow \\ \rightarrow \text{Editing}\left(CL, AB'\right) \end{array} \right\rangle_{,}^{1} \quad (3)$$

де ag_i – обраний i-й антиген із популяції AG;

$\text{Presentation}\left(ag_i, AB\right)$ – ІО представлення i-го антигена антитілам;

$\text{Selection}\left(AB, AB'\right)$ – ІО первинного відбору антитіл для клонування;

$\text{Cloning}\left(AB', CL\right)$ – ІО клонування виділених антитіл AB';

$\text{Mutation}\left(CL\right)$ – ІО мутації клонів CL;

$\text{Presentation}\left(ag_i, CL\right)$ – ІО представлення i-го антигена клонам;

$\text{Editing}\left(CL, AB'\right)$ – ІО редагування клонів и множини антитіл AB'.

Слід зазначити, що використання послідовної обробки антитіл на етапі ІМН більш доцільно, оскільки при цьому спостерігається зниження кількості надлишкових обчислень, що виникають на етапі редагування популяції або супресії мережі антитіл, що характерно для популяційної обробки антитіл.

2. Класифікація даних на основі гібридного імунного методу kNN

Метод KNN є одним з найпростіших і найбільш поширених методів класифікації. Тому для підвищення точності класифікації в [5] був запропонований модифікований алгоритм kNN з імунним навчанням на основі алгоритму CLONALG. Оскільки модель штучної імунної мережі надає більше можливостей для організації взаємодії між імунними об'єктами, принцип роботи гібридного алгоритму, наведеного в [5], був використаний для алгоритму aiNETm. Таким чином, отриманий імунний гібрид aiNETmkn є модифікацією алгоритму aiNETm з імунним навчанням на основі моделі штучної імунної мережі. Роботу гібридного алгоритму aiNETmkn можна умовно розділити на наступні основні етапи:

1) визначення стимулюючих антитіл, які формують множину «найближчих сусідів» і їх областей стимуляції;

2) визначення цільових антигенів для стимулюючих антитіл;

3) класифікація стимулюючих антитіл в процесі ІМН;

4) класифікація антитіл з областей стимуляції;

5) відновлення популяції вихідних об'єктів і їх угрупування за результатами класифікації останньої популяції антитіл.

На рівні імунних операторів (ІО) запропонований гібридний алгоритм aiNETmkn представляється в такий спосіб:

$$\text{aiNETmkn}\,(AB, AG, T, k) = \left[\begin{array}{l} \text{NatCalculation}(AG, NAT) \rightarrow \\ \rightarrow \text{Presentation}(AB, AG) \rightarrow \\ \rightarrow \text{Selection}(AB, AG^{T}) \rightarrow \\ \rightarrow \text{Selection}(AB, AB^{S}, AB^{C}, AB') \rightarrow \\ \rightarrow \text{SpacesSelection}(AB', AB^{S}, k) \end{array}\right]^{PRP} \rightarrow$$

$$\rightarrow, \left[\begin{array}{l} {\scriptstyle AB^{S}} \left\langle \begin{array}{l} \text{Cloning}(ab_i^s, CL_i, k) \rightarrow \\ \rightarrow \text{Mutation}(CL_i) \rightarrow \\ \rightarrow \text{Presentation}(CL_i, AG_i^T) \rightarrow \\ \rightarrow \text{Suppression}(ab_i^s, CL_i) \end{array} \right\rangle \\ \rightarrow \text{ClassDetection}(AB^{S}, AG^{T}) \end{array}\right]^{LRN} \rightarrow \qquad (4)$$

$$\rightarrow \left[\begin{array}{l} \text{kNNclassDetection}(AB', AB^{S}, k) \rightarrow \\ \rightarrow \text{Reconstruction}(AB, AB^{C}) \end{array}\right]^{CLS},$$

де $\text{NatCalculation}(AB, NAT)$ – ІО визначення значень порогових афінностей для популяції об'єктів, що групуються;

$\text{Presentation}(AG, ab_i)$ – ІО представления антигенів i-му антитілу;

$\text{Selection}\,(AB, AG^{T})$ – ІО відбору цільових антигенів AG_i^T для кожного антитіла, що класифікується;

$\text{Selection}\,(AB, AB^{S}, AB^{C}, .AB')$ – ІО визначення класифікованих антитіл AB^{C}, стимулюючих антитіл AB^{S}, та антитіл, які будуть класифіковані після проведення імунного навчання AB';

$\text{SpacesSelection}(AB', AB^{S}, k)$ – ІО визначення k найближчих стимулюючих антитіл для кожного антитіла з множини AB';

$\text{Suppression}\,(ab_i^s, CL_i)$ – ІО супресії клонів и стимулюючих антитіл;

$\text{ClassDetection}(AB^{S}, AG^{T})$ – ІО завершення класифікації стимулюючих антитіл;

$\mathrm{kNNclassDetection}(AB', AB^S, k)$ – ІО класифікації антитіл із областей стимуляції на основі методу найближчих сусідів;

$\mathrm{Reconstruction}(AB, AB^C)$ – ІО відновлення і класифікації вихідних об’єктів за результатами класифікації популяції антитіл.

Таким чином, на першому етапі роботи алгоритму aiNETmkn завдяки значенням порогових афінностей антигенів відбувається визначення стимулюючих антитіл, які є найближчими сусідами для об’єктів, що характеризуються низькою афінністю до вихідних класів. Слід зазначити, що більшість стимулюючих антитіл класифікується за допомогою ІМН, однак, для деяких антитіл передбачається можливість класифікації без ІМН в разі, якщо вони характеризуються максимальними значеннями афінності до навчальних антигенів. Відповідно до цього в процесі ІМН беруть участь тільки стимулюючі антитіла, тобто антитіла, що входять до сфери стимуляції, не піддаються дії операторів клонування, мутації і редагування популяції імунних об’єктів.

Класифікація антитіл AB', що входять в області стимуляції, відбувається після завершення ІМН і класифікації всіх стимулюючих антитіл при використанні методу найближчих сусідів. Відповідно до цього, для кожного антитіла, що знаходиться в області стимуляції на початковому етапі роботи алгоритму aiNETmkn визначається k найближчих стимулюючих антитіл. Після завершення ІМН антитіла множина AB' класифікується на підставі значень початкових афінностей зі своїми найближчими стимулюючими антитілами. При цьому значення афінності використовуються в якості вагових коефіцієнтів. Таким чином, рішення про належність антитіла одному з вихідних класів визначається шляхом визначення максимальної суми афінностей належних йому стимулюючих антитіл, що входять в множину найближчих сусідів.

Для оцінки роботи запропонованого алгоритму aiNETmkn використовувалися набори даних, представлені в табл. 1. Кожен набір характеризується кількістю об’єктів і розмірністю матриць ознак. При цьому, якщо розмірність матриці ознак характеризується значенням 3 × 3, то кожен об’єкт, що класифікується, і об’єкт НВ описується дев’ятьма ознаками. При визначенні ефективності запропонованого алгоритму aiNETmkn було проведено порівняння результатів його роботи з результатами роботи алгоритмів kNN, CLONALGm і aiNETm, отриманими в [4, 5].

Результати порівняння перерахованих алгоритмів наведені в табл. 2, в

якій використовуються наступні умовні позначення: «T» характеризує швидкодію алгоритму класифікації (Time – в % до найповільнішого алгоритму); «A» використовується для відображення точності угруповання (Accuracy – в % до вихідних даних, що класифікуються); «S» характеризує стійкість алгоритму (Stability – в % до вихідних даних).

Таблиця 1

Характеристики наборів даних

Набори	Дані, що класифікуються	Навчальна вибірка	Кількість класів	Матриця ознак
Набір 1	500	100	3	3 × 3
Набір 2	500	100	3	5 × 5
Набір 3	1000	200	5	5 × 5
Набір 4	2000	400	5	5 × 5
Набір 5	5000	400	5	5 × 5
Набір 6	5000	400	5	10 × 10
Набір 7	10 000	500	10	5 × 5
Набір 8	15 000	500	10	5 ×5
Набір 9	20 000	500	10	5 ×5
Набір 10	20 000	500	10	10 × 10

Таблица 2

Результати класифікації даних на основі гібридного імунного методу kNN

Методи		Набори даних, що групуються									
		Н 1	Н 2	Н 3	Н 4	Н 5	Н 6	Н 7	Н 8	Н 9	Н 10
kNN (k = 5)	**T**	11.7	12.1	12.5	12.6	12.5	14.2	14.8	14.7	14.7	16.0
	A	96.2	96.3	93.8	91.6	90.5	90.5	87.2	86.1	84.3	84.4
	S	99.8	99.7	98.6	98.7	97.5	97.4	95.3	95.4	95.4	95.3
CLONALGm	**T**	39.5	39.6	39.8	39.9	39.8	41.3	41.5	41.4	41.5	43.4
	A	98.2	98.3	98.2	98.2	98.1	98.1	97.9	97.8	97.8	97.7
	S	99.8	99.7	99.2	99.2	99.1	99.1	98.8	98.8	98.7	98.7
aiNETm	**T**	38.9	38.9	39.0	39.0	39.1	39.4	39.4	39.4	39.5	39.6
	A	98.7	98.7	98.6	98.5	98.5	98.4	98.3	98.3	98.2	98.0
	S	99.8	99.8	99.7	99.7	99.6	99.6	99.5	99.5	99.4	99.4
aiNETmkn (k = 5)	**T**	37.9	37.9	38.1	38.1	38.1	38.2	38.2	38.3	38.3	38.4
	A	98.7	98.7	98.6	98.5	98.5	98.4	98.3	98.3	98.2	98.1
	S	99.8	99.8	99.7	99.7	99.6	99.6	99.5	99.5	99.4	99.4

З результатів експериментів випливає, що алгоритм aiNETmkn перевершує алгоритм aiNETm тільки за швидкодією, при цьому, не поступаючись йому за точністю і стійкістю класифікації. Слід зазначити, що при порівнянні даного алгоритму з алгоритмом kNN, алгоритм aiNETmkn перевершує kNN по точності і стійкості класифікації, поступаючись йому за швидкодією на 20-25%. Таким чином, використання гібридного алгоритму aiNETmkn для вирішення задачі класифікації з контрольованим навчанням, переважніше, ніж використання інших імунних алгоритмів, таких як CLONALGm, aiNETm або RLAISm.

3. Кластеризація даних на основі гібридного імунного методу k-means

Алгоритм k-means є одним із найпростіших і найбільш розповсюджених методів кластеризації. Даний алгоритм часто використовують як основу для формування гібридних методів інтелектуальної обробки інформації. Одним з найефективніших методів кластеризації даних є гібридний метод опорних векторів FCM [1], який функціонує на основі алгоритму k-means, і використовує принципи теорії нечітких множин при виділенні кластерів. Таким чином, для підвищення точності кластеризації в [5] було запропоновано гібридний алгоритм k-means, який функціонує на основі імунної моделі клонального відбору і використовує ІМН для виділення кластерів. У зв'язку з тим, що модель штучної імунної мережі надає більше можливостей для організації навчання, принцип роботи гібридного алгоритму, наведеного в [5], було використано для алгоритму aiNETm.

Отриманий імунний гібрид aiNETmkm є модифікацією алгоритму k-means, що використовує етап ІМН алгоритму aiNETm для підвищення швидкості формування кластерів. Робота гібридного алгоритму aiNETmkm починається з виділення стимулюючих антитіл як центрів кластерів, що формуються, і визначення областей стимуляції при використанні значення граничної афінності NAT. Кластеризація за допомогою даного алгоритму не має на увазі використання популяції антигенів, тому що модель штучно імунної мережі дозволяє використовувати взаємодію між антитілами в процесі ІМН. Таким чином, при використанні стимулюючих антитіл формуються пересічні області стимуляції, що містять об’єкти, які групуються.

В процесі ІМН антитіла, що знаходяться в областях стимуляції,

взаємодіють зі стимулюючими об'єктами як з антигенами. Стимулюючі антитіла не піддаються клонуванню і мутації, і змінюються лише після перевизначення центроїда для кожного кластера, що виділяється. За рахунок використання пересічних областей стимуляції в процесі супресії мережі клонованим антитілам і їх клонам представляється декілька стимулюючих антитіл-центроїдів. Навчання моделі aiNETmkm завершується при досягненні стану специфічності між усіма антитілами, які перебувають в областях стимуляції і виділеними центрами кластерів, або при досягненні максимальної кількості популяцій імунних об'єктів, які формуються в процесі ІМН.

Відповідно до цього метод aiNETmcm на рівні ІО представляється в такий спосіб:

$$\text{aiNETmkm}(AB, T, k) = \left[\begin{array}{l} \text{NatCalculation}(AB, NAT) \rightarrow \\ \rightarrow \text{Selection}(AB, k, AB^S, AB') \rightarrow \\ \rightarrow \text{SpacesSelection}(AB', AB^S) \end{array}\right]^{PRP} \rightarrow$$

$$\rightarrow, \left[\begin{array}{l} {\scriptstyle AB'} \left\langle\begin{array}{l} \text{Cloning}(ab_i^{'}, CL_i, k) \rightarrow \\ \rightarrow \text{Mutation}(CL_i) \rightarrow \\ \rightarrow \text{Presentation}(CL_i, AB_i^S) \rightarrow \\ \rightarrow \text{Suppression}(ab_i^{'}, CL_i) \end{array}\right\rangle \\ \rightarrow \text{ClassDetection}(AB^{'}, AB^S) \end{array}\right]^{T} \rightarrow \qquad (5)$$

$$\rightarrow \text{Reconstruction}(AB, AB^C),$$

де $\text{NatCalculation}(AB, NAT)$ – ІО визначення значень порогових афінностей NAT для популяції об'єктів, що групуються;

$\text{Selection}(AB, k, AB^S, AB')$ – ІО визначення k стимулюючих антитіл, які є початковими центрами кластерів;

$\text{SpacesSelection}(AB', AB^S)$ – ІО визначення антитіл в областях стимуляції;

$\text{Cloning}(ab_i^{'}, CL_i, k)$ – ІО клонування антитіл в областях стимуляції;

$\text{Presentation}(CL_i, AB_i^S)$ – ІО представлення клонів стимулюючим

антитілам;

$\mathrm{Suppression}(ab_i', CL_i)$ – ІО супресії антитіл и клонів;

$\mathrm{ClassDetection}(AB', AB^S)$ – ІО уточнення центроїдів AB^S в областях стимуляції.

Для оцінки роботи алгоритму aiNETmkm використовувалися набори даних, представлені в табл. 1, перетворені для проведення кластеризації. При визначенні ефективності алгоритму aiNETmkm було проведено порівняння результатів його роботи з результатами роботи алгоритмів k-means, CLONALGmc і aiNETmc, отриманими в [4, 5]. Результати порівняння перерахованих алгоритмів наведені в табл. 3.

Таблиця 3

Результати кластеризації даних на основі гібридного імунного методу k-means

Методы		Группируемые наборы данных									
		Н 1	Н 2	Н 3	Н 4	Н 5	Н 6	Н 7	Н 8	Н 9	Н 10
k-means	**T**	24.2	24.1	24.2	27.7	27.8	27.8	30.2	30.4	30.5	30.5
	A	99.8	99.7	99.7	99.6	99.5	99.4	99.5	99.4	99.4	99.3
	S	99.9	99.8	99.8	99.7	99.7	99.7	99.7	99.7	99.7	99.8
CLONALGmc	**T**	25.5	25.5	25.7	25.8	25.9	26.2	26.8	26.9	27.0	27.1
	A	94.8	94.8	94.6	94.7	94.6	94.1	94.0	94.0	94.1	94.0
	S	98.1	98.0	97.9	97.9	97.8	97.6	97.5	97.5	97.5	97.4
aiNETmc	**T**	25.3	25.3	25.5	25.7	25.7	26.0	26.5	26.5	26.9	26.9
	A	94.8	94.8	94.7	94.7	94.7	94.2	94.0	94.0	94.1	94.0
	S	98.0	98.0	98.0	97.7	97.9	97.7	97.6	97.6	97.6	97.5
aiNETmkm	**T**	20.0	20.0	20.1	22.4	22.4	22.7	22.9	23.0	23.2	23.2
	A	94.8	94.8	94.7	94.7	94.7	94.2	94.0	94.0	94.1	94.0
	S	98.0	98.0	98.0	97.7	97.9	97.7	97.6	97.6	97.6	97.5

Результати кластеризації свідчать, що модифікований алгоритм aiNETmkm зіставний з алгоритмом aiNETmc і перевершує його за швидкодією, збігаючись при цьому з ним по іншим характеристикам. При цьому метод aiNETmkm перевершує алгоритм k-means за швидкодією, але поступається йому в точності і стійкості кластеризації. Крім того, при використанні методу aiNETmkm відсутня необхідність у використанні апріорної кількості кластерів, що формуються, так як їх кількість

визначається імунними методами.

Таким чином, запропонований в результаті модифікації метод кластеризації aiNETmkm перевершує за швидкодією інші імунні методи, які використані в дослідженнях.

4. Автоматична класифікація даних на основі гібридної моделі імунної мережі і нечіткого походу

Використання принципів нечіткої логіки при вирішенні задач класифікації і кластеризації даних є одним з найбільш поширених методів підвищення точності класифікації. Це обумовлюється підвищенням кількості інформації про належність об'єктів до класів і кластерів, отриманої в результаті даного підходу. Використання нечіткого підходу в організації ІМН призводить до підвищення швидкості роботи або точності результату. В [5] запропоновано алгоритм автоматичної класифікації на основі моделі клонального відбору та нечіткого підходу.

У запропонованому методі використовувалися принципи конкурентно-цільового відбору і послідовної обробки антитіл в процесі ІМН. При цьому класифікація і виділення кластерів в [5] відбувається при використанні нечіткої гаусової функції належності. Для підвищення швидкості і точності результатів класифікації замість моделі клонального відбору може використовуватися модифікований метод aiNETma, що функціонує на основі моделі штучної імунної мережі aiNET. Отриманий гібридний метод aiNETmaf реалізує модель автоматичної класифікації і використовує гаусові функції при визначенні належності об’єктів, що класифікуються, вихідним класам і кластерам, які формуються.

Для класифікації антитіл в [4] була запропонована модифікація гаусової функції належності на основі порогових афінностей популяції антигенів і афінностей між класифікованим об’єктом і цільовими антигенами вихідних класів, а також центрами кластерів. Для підвищення точності класифікації пропонується зміна принципу формування аргументів гаусової функції належності, запропонованої в [4]. Гаусова функція належності має наступний вигляд:

$$\mu(x) = e^{\frac{-(x-b)^2}{2c^2}}, \qquad (6)$$

де x – деякий вхідний аргумент, щодо якого відбувається визначення належності множині;

b – максимальне значення у діапазоні зміни аргументу x;

c – коефіцієнт концентрації функції.

У методі aiNETmaf пропонується модифікована функція належності (6), в якій визначення коефіцієнта концентрації відбувається на основі порогових афінностей популяції антигенів і об'єктів, що належать класу, по відношенню до якого визначається належність. Функція належності, яка використовується в aiNETmaf, представляється в такий спосіб:

$$\mu_{ij} = e^{\frac{-(aff_{ij}-1)^2}{2(2-NAT-NAT_j)^2}}, \quad (7)$$

де μ_{ij} – належність i-го об'єкта j-му класу або кластеру;

aff_{ij} – афінність між i-м класифікованим антитілом і цільовим об'єктом j-го класу або кластера;

NAT – середня афінність між антигенами НВ;

NAT_j – середня афінність між об'єктами j-го класу або кластера.

В якості коефіцієнта концентрації функції використовується сума зворотних значень середніх афінностей між навчальними антигенами $(1-NAT)$ і об'єктами, що належать досліджуваному класу або кластеру $(1-NAT_j)$. Для підвищення швидкості роботи в aiNETmaf використовується стан умовної належності класу, яка визначається вхідним параметром B, змінним в діапазоні дійсних значень [0.51; 1.00]. Класифікація об'єкта відбувається в наступному випадку:

$$\forall ab_i \in C_j : \mu_{ij} \geq B. \quad (8)$$

У разі виконання цієї умови антитіло визначає приналежність до j-у класу і не піддається дії ІО в процесі ІМН. Після завершення етапу ІМН для всіх антитіл, що класифікуються, при використанні функції (7) відбувається визначення належності до кожного вихідного класу і сформованому кластеру для отримання більш детальної інформації про розбиття множини об'єктів.

Оскільки гібридний метод aiNETmaf є модифікацією мережевого методу автоматичної класифікації aiNETma, підготовчий етап і етап ІМН у даних методів ідентичні, за винятком способу угруповання об'єктів, тому що в методі aiNETma висновок про належність антитіла до класу робиться на основі значення афінності між ними, а в aiNETmaf для цього

використовується функція належності (7). Тому на рівні ІО метод aiNETmaf ідентичний методу aiNETma, за винятком додавання параметра порогу належності B, який використовується при класифікації, замість коефіцієнта зростання кластерів λ.

Для оцінки роботи методу aiNETmaf використовувалися набори даних, представлені в табл. 1. При визначенні ефективності методу aiNETmaf було проведено порівняння результатів роботи даного методу з результатами роботи методів aiNETma і CLONALGma, отриманими раніше. Результати порівняння перерахованих алгоритмів наведені в табл. 4.

Таблиця 4

Результати класифікації наборів даних на основі моделі імунної мережі і нечіткого походу

Методи		Набори даних									
		Н 1	Н 2	Н 3	Н 4	Н 5	Н 6	Н 7	Н 8	Н 9	Н 10
CLONALGma	**T**	100	100	100	100	100	100	100	100	100	100
	A	94.8	94.8	94.6	94.7	94.6	94.1	94.0	94.0	94.1	94.0
	S	98.1	98.0	97.9	97.9	97.8	97.6	97.5	97.5	97.5	97.4
aiNETma	**T**	99.6	99.7	99.6	99.6	99.7	99.6	99.5	99.6	99.7	99.6
	A	96.7	96.7	96.7	96.6	96.5	96.5	96.3	96.3	96.3	96.3
	S	99.9	99.9	99.9	99.8	99.8	99.8	99.8	99.8	99.8	99.8
aiNETmaf	**T**	99.7	99.7	99.7	99.7	99.8	99.7	99.6	99.8	99.8	99.8
	A	96.9	96.8	96.9	96.8	96.7	96.7	96.5	96.5	96.5	96.5
	S	99.9	99.9	99.9	99.8	99.8	99.8	99.8	99.8	99.8	99.8

З результатів класифікації випливає, що проведення модифікації методу aiNETma на основі методів нечіткої логіки призводить до незначного підвищення точності класифікації на 0.1-0.2% при незначному зниженні швидкодії.

Висновки

Підвищення вимог до моделей класифікації, що володіють здатністю адаптуватися до умов, що змінюються, і поєднувати особливості різних способів навчання, вимагає розробки моделей, що дозволяють змінювати угруповання вихідних даних. Використання ШІС для вирішення задачі класифікації дозволяє проводити угруповання даних різними способами, а також змінювати як структуру мережі імунних об’єктів, так і методи вирішення поставленої задачі. Рішення задачі класифікації на основі

імунного підходу зводиться або до вирішення задачі поділу даних між відомими групами, або формування нових груп з вихідної множини даних. При цьому процес класифікації умовно розбивається на три основних етапи: підготовчий етап PRP, етап імунного навчання LRN та етап уточнення меж вихідних класів CLS.

Гібридизація імунних методів і моделей інтелектуальної обробки інформації дозволяє підвищити їх ефективність, що виражається в підвищенні швидкодії, або точності їх роботи. Досліджено можливості розробки гібридних методів класифікації, що функціонують на основі різних імунних моделей та інших підходів. З цією метою в роботі запропоновані гібридні моделі класифікації даних на основі моделі штучної імунної мережі, що дозволяє змінювати як спосіб угруповання вихідних даних, так і структуру мережі імунних об'єктів. З використанням моделі штучної імунної мережі запропоновано гібридну модель класифікації даних на основі методу найближчих сусідів kNN, гібридну модель кластеризації даних на основі методу k-means, а також гібридну модель автоматичної класифікації на основі нечіткого походу.

Проведено порівняльні експериментальні дослідження запропонованих гібридних моделей класифікації даних з існуючими, які показали високу ефективність використання для цих цілей моделі штучної імунної мережі.

Література

1. Duda R.O. Pattern classification / R.O. Duda, P.E. Hart, D.G. Stork. – Wiley & Sons, 2010. – 738 p.

2. Ершов К.С. Анализ и классификация алгоритмов кластеризации / К.С. Ершов, Т.Н. Романова // Новые информационные технологии в информационных системах, 2016. – № 1. – С.274-279.

3. Dasgupta D. Immunological computation, Theory and applications / D. Dasgupta, L.F. Nino – Taylor & Francis Group, 2009. – 278 p.

4. Кораблев Н.М. Классификация данных с использованием иммунной модели клонального отбора / Н.М. Кораблев, А.А. Фомичев // Інформаційні системи та технології:: монографія / за заг. ред. В.С. Пономаренка. – Х.: ФОП Бровін О.В., 2019. – С. 185-199.

5. Кораблев Н.М. Классификация данных с использоваанием модели искусственной иммунной сети / Н.М. Кораблев, А.А. Фомичев // Інформаційні технології: сучасний стан та перспективи: монографія. За заг. ред. В.С. Пономаренка. – Х.: ТОВ «ДІСА ПЛЮС», 2018. – С. 86-101.

ГЛАВА 5

ОЦІНКА ЕФЕКТИВНОСТІ УПРАВЛІННЯ РОЗПОДІЛЕНИМИ КОМП'ЮТЕРНИМИ СИСТЕМАМИ В УМОВАХ НЕВИЗНАЧЕНОСТІ

Вступ і постановка задачі. Дедалі більшого поширення набувають нові технології, пов'язані з обробкою і передачею інформації по каналах зв'язку не тільки комп'ютерні дані, але голосову і відеоінформацію. Інтеграція різних видів інформаційного обслуговування в рамках однієї мережі є закономірним наслідком розвитку цифрових технологій. Розподілений характер великої мережі зі складною структурою унеможливлює підтримання її роботи на належному рівні без системи управління, яка також є складною системою.
В даний час система управління працює в автоматизованому режимі, виконуючи найпростіші дії з управління мережею автоматично, а складні рішення надаючи приймати людині на основі підготовленої системою інформації [1]. Система управління мережею повинна забезпечити, з одного боку, підтримку в робочому стані як мережу в цілому, так і окремих її складових, для того, щоб вона могла виконувати свої функції, а з іншого боку – розподіл і доставку інформаційних повідомлень за адресами з дотриманням різних вимог користувачів [2].

Досвід розробки та експлуатації складних систем, успіх їх оптимізації залежить не тільки від адекватності моделі процесу її функціонування і досконалості використовуваного математичного апарату для отримання точних і достовірних результатів оцінки характеристики системи, а й від обраного критерію ефективності системи.

На основі стандартної моделі взаємодії відкритих систем виділимо групи характеристик, які визначають ефективність КС. До таких характеристик відносяться [1]:

- тимчасові (час доставки, час обробки повідомлень);
- наведені цифри щодо (пропускна здатність, завантаженість елементів);
- характеристики зв'язності (тимчасова, просторова, протокольна);
- характеристика цілісності (стійкість, достовірність, точність передачі);
- характеристика безпеки (обсяг, тривалість, спосіб зберігання даних);
- характеристика надійності і живучості;

- характеристика безпеки зв'язку.

В даний час в Україні немає суворої концепції зі створення системи мережевого управління. Тому всі питання, пов'язані з розробкою такої системи, є надзвичайно актуальними. Питанням управління телекомунікаційними мережами присвячено достатню кількість робіт [3, 4].

В [2] висвітлюються загальні питання управління мережами зв'язку. Наводиться базова інформація по структурі і особливостям протоколів CMIP і SNMP, проведено огляд платформ і продуктів мережевого управління деяких іноземних компаній.

В [1,3] даються основи побудови систем управління телекомунікаційними мережами. Висвітлюються такі питання динамічного управління, як маршрутизація, принципи управління трафіком. Наведено багаторівнева архітектура мережевого управління.

Актуальними є питання, пов'язані з оцінкою ефективності мереж. Як правило, ефективність мережі обміну оцінюється за допомогою окремих приватних показників якості, таких як ймовірність помилки, час доставки, без урахування цінності переданої по мережі інформації і зв'язності елементів мережі. В [2] ставиться завдання узагальненого підходу до оптимального проектування систем управління мережею. Пропонується введення єдиного векторного показника ефективності. При цьому не вказується його формульне вираз. Не враховується можлива невизначеність стану елементів мережі.

У роботі [3] розглядається задача синтезу оптимальної системи управління мережею за допомогою мінімаксного методу, який компромісно оптимізує тільки обмеження на вхідні дані і спектр наявних умов.

В [2] аналізується можливість створення автоматичної системи управління мережею. Однак як приймати рішення ця система управління не розглядає.

Актуальними є питання, пов'язані з оцінкою ефективності розподілених інформаційних систем. Як правило, ефективність таких систем оцінюється за допомогою окремих приватних показників якості, таких як ймовірність помилки, час доставки пакетів повідомлень, без урахування цінності переданої по мережі. В [2] ставиться завдання узагальненого підходу до оптимального створення розподілених інформаційних систем. При цьому, пропонується введення єдиного векторного показника ефективності, але не наводиться його формульне вираз. Не враховується можлива невизначеність стану елементів інформаційної системи.

Швидке розширення областей застосування інформаційних систем призводить до постійного зростання вимог до їх ефективності. Фундаментальними причинами неможливості повного задоволення цих вимог є недосконаліть і обмеженість реалізованих концепцій і принципів побудови систем цього типу, які мають ряд недоліків і обмежують можливості подальшого підвищення ефективності систем управління. Це призводить до протиріччя з одного боку жорстким вимогам до оперативності управління інформаційними ресурсами і потоками даних, точності рішення задач розподілу ресурсів, з іншого боку нездатністю розподіленої інформаційної системи виконувати завдання управління і розподілу в масштабі реального часу, що викликана ускладненням логічної і фізичної структури систем.

В області теоретичних досліджень є значна кількість робіт, присвячених питанням підвищення ефективності комп'ютерних мереж. Частина з них орієнтована на дослідження інформаційних мереж, розробку моделей управління ними, методів їх аналізу і реалізованих в них протоколів [4]. Питанням розподілу мережевих ресурсів присвячена робота [5]. У наявних роботах в області динамічного управління в інформаційних мережах вирішуються завдання управління потоками мовних повідомлень [2], розроблені методи підвищення продуктивності мереж з мультимедійними послугами, методи підвищення ефективності протоколів доступу обчислювальних мереж [5], управління пропускною спроможністю мереж зв'язку, динамічного управління потоками інформації, а також оптимізації та динамічного регулювання навантаження інформаційних мереж [2]. Кожна з цих робіт присвячена вирішенню приватної конкретного завдання, але не спрямована на розробку комплексного підходу, що забезпечує необхідну ефективність функціонування всієї системи динамічного управління.

При оцінці ефективності роботи мережі в процесі управління необхідно враховувати якість обслуговування і пріоритетність передачі інформації. В [1,2] описані деякі питання реалізації заданого якості обслуговування в мережах, однак в них не враховується той факт, що інформація має свою цінність, причому зі збільшенням часу затримки в передачі або обслуговуванні цінність інформації зменшується, тобто відбувається її старіння.

Розглянемо методику оцінки ефективності системи управління комп'ютерної мережі з точки зору її продуктивності.

Ефективністю називається властивість процесу функціонування системи, що характеризує його пристосованість до вирішення поставлених перед системою завдань. Показником ефективності є її кількісна міра. Такий показник повинен характеризувати ступінь досягнення мети функціонування системи.

При виборі показника ефективності необхідно керуватися наступними вимогами:

- можливістю вимірювання параметрів, що входять в показник;
- системністю оцінки ефективності;
- можливістю обліку всіх існуючих параметрів;
- однозначністю оцінки;
- урахуванням особливостей вирішуваних завдань.

Крім перерахованих міркувань одним з основних вимог, яким керуються при виборі показників ефективності, є ступінь їх узгодженості з цілями функціонування системи і завданнями досліджень.

Основна частина

Основне призначення комп'ютерних систем складається в забезпеченні потреб користувачів у послугах з обробки і доставки в потрібний час необхідної інформації з необхідною якістю. Тому ефективність управління процесом обробки і доставки інформації пропонується оцінювати за результатами приросту ефективності комп'ютерної мережі при наявності управління $E_{ксу}$ щодо її ефективності без управління ($E_{кс}$)

$$E_{упр} = E_{ксу} / E_{кс} . \qquad (1)$$

Ефективність використання комп'ютерної системи оцінюється її продуктивністю, яка представляє собою відносну швидкість передачі і обробки інформації і дорівнює відношенню реальної швидкості $\lambda_{вик}$ до максимально можливої $\lambda_{мах}$:

$$E_{кс} = \frac{\sum_{i=1}^{N} \lambda_{викі}}{\sum_{i=1}^{N} \lambda_{махі}}. \qquad (2)$$

Інтенсивність передачі і-того потоку визначається за формулою:

$$E_{кс} = \frac{1}{T_{срі}} \Delta W_i (1 - P_{втрі}), \qquad (3)$$

де $T_{срі}$ – середній час доставки і-того повідомлення; ΔW_i – важливість

цього повідомлення при інформаційному обміні; $P_{втрі}$ – ймовірність втрати повідомлення при передачі і обробці.

Максимальна інтенсивність передачі інформаційної частини повідомлення дорівнює

$$\lambda_{махі} = \frac{1}{T_{пі}},$$

де $T_{пі}$ – тривалість і-того повідомлення без урахування впливу різної надмірності на час передачі.

Така максимальна інтенсивність передачі може бути забезпечена і при впливі системи управління.

Максимальна цінність повідомлення $\Delta W_{maxi} = 1$, отже

$$E_{кс} = \sum_{i=1}^{N} \frac{T_{пі}}{T_{срі}} \cdot \Delta W_{i} \cdot (1 - P_{втрі}). \qquad (4)$$

Слід зазначити, що цінність інформації може бути врахована тільки при наявності управління.

Ефективність комп'ютерної мережі при наявності управління здійснюється за формулою:

$$E_{кс} = \sum_{i=1}^{N} \frac{T_{пі}}{T_{усрі}} \cdot \Delta W_{уі} \cdot (1 - P_{увтрі}). \qquad (5)$$

В результаті ефективність управління визначається

$$E_{кс} = \sum_{i=1}^{N} \frac{T_{пі}}{T_{усрі}} \cdot \frac{\Delta W_{уі}}{\Delta W_{i}} \cdot \frac{(1 - P_{увтрі})}{(1 - P_{втрі})}. \qquad (6)$$

За умови забезпечення якості обслуговування, $P_{оп} \le P_{опдоп}$, , $T_{д} \le T_{доп}$.

Вираз (6) необхідно використовувати для вибору засобів і способів досягнення мети управління, що забезпечують максимум величини $E_{ксу}$.

Для його максимізації необхідно в процесі управління прагнути до мінімізації $T_{усрі}$ та наближення цього значення до $T_{пі}$, збільшення $\Delta W_{уі}$ і зменшення $P_{увтрі}$.

Середній час доставки і-того повідомлення включає в себе час затримки на центрі комутації $T_{зі}$ і час передачі по каналу зв'язку $T_{пері}$:

$$T_{срі} = T_{зі} + T_{пері}.$$

Обидві ці складові залежать від правильного розподілу повідомлень за напрямками передачі адресату, якщо таких напрямків може бути кілька. Крім того, на значення $T_{зі}$ і $T_{пері}$ впливають методи управління при доступі в мережу, при передачі по мережі, а також ширина вікна, тривалість обраного

тайм-ауту і маршруту. Тому $T_{усрі} \le T_{срі}$. Правильність розподілу повідомлень і обраний маршрут впливають на ймовірність втрати повідомлення. На цю характеристику впливають також методика розподілу ємності буферних пристроїв на вузлах комутації. В результаті має бути виконано нерівність $P_{увтрі} < P_{втрі}$.

Розробимо методику оцінки важливості потоків інформації при наявності і відсутності управління.

Припустимо, що на центр комутації надходить потік інформації з інтенсивністю

$$\lambda_{вхі} = \sum_{j=1}^{N} \lambda_{вхj},$$

де $\lambda_{вхj}$ – інтенсивність потоку від j-го джерела;

N – кількість джерел інформації.

Позначимо вагу і-го потоку даних джерела ΔW_i. Ця вага є відносної частиною потоку до сумарного потоку в мережі

$$\Delta W_i = f\left(\frac{\lambda_{вхі}}{\lambda_{вх}}\right) = f\left(\lambda_{вхі} \Big/ \sum_{i=1}^{N} \lambda_{вхі}\right), \qquad (7)$$

при $\lambda_{вхі} \le \lambda_{вхімах}$, де $\lambda_{вхі}$ – інтенсивність джерела і-го потоку, $\lambda_{вхімах}$ – максимально допустимий потік джерела і-го потоку. При $\lambda_{вхі} > \lambda_{вхімах}$ використовується механізм обмеження вхідного навантаження. Ліквідуються пакети, які не можуть бути обслужені з необхідною якістю.

Цей вираз чисельно визначає важливість (вагу) і-го потоку. При вирішенні конкретних завдань кожен потік має певну корисністю (цінністю). Тому при управлінні чисельне значення ΔW_i має враховувати не тільки $\lambda_{вхі}$, а й корисність (цінність) цього потоку, яку будемо враховувати коефіцієнтом K_i. З фізичної точки зору цей коефіцієнт повинен змінюватися в межах 0 – ∞. Якщо без і-го потоку рішення поставленого завдання неможливо, то $0 < K_i < \infty$. Якщо ж передається і-й потік при вирішенні поставленого завдання не потрібен, то $K_i = 0$. У ситуації, коли і-й потік не має визначального значення для вирішення поставленого завдання, його цінність знаходиться в межах $0 < K_i < \infty$.

З урахуванням вищесказаного отримаємо

$$\Delta W_i = f\left(\frac{\lambda_{вхі}}{\lambda_{вх}}, K_i\right).$$

При визначенні функції ΔW_i потрібно виходити з таких вимог:

значення ΔW_i має змінюватися в межах 0 ... 1; його величина повинна зростати зі збільшенням відносини $\lambda_{вхі}/\lambda_{вх}$; значення ΔW_i має дорівнювати 0 при K_i =0 і так само при K_i =1;

величина ΔW_i повинна дорівнювати 1 при $K_i = \infty$.

Зазначеним вимогам задовольняє функція

$$\Delta W_i = \frac{\lambda_{вхі}^{1/K_i}}{\lambda_{вх}}. \qquad (8)$$

Величини K_i в інтервалі від 0 до 1 визначаються помилками, допущеними при управлінні.

Визначимо методику оцінки корисності (цінності) інформації. Нехай по каналу зв'язку передається повідомлення A_j. Система управління не завжди діє відповідно до переданим повідомленням A_j,. Не завжди при передачі повідомлення A_j на приймальній стороні виконується дія a_j. В результаті управління здійснюється з помилками.

Середня корисність від передачі повідомлення визначається відповідно до виразу:

$$K_{ср} = \sum_{i=0}^{n} P_i(A_i) \cdot K_i(a_i / A_i).$$

Цінність даних повинна визначатися її збільшенням за рахунок прийому інформації. Тоді приріст корисності інформації може визначатися за формулою:

$$Ц = K_{ср} - K_{ср0} = \sum_{i=0}^{n} P_i(A_i) \cdot K_i(a_i / A_i) - \sum_{i=0}^{n} P_i(A_i) \cdot K_{i0}(a_i / A_i),$$

де $K_{ср0}$ – являє собою середню корисність при відсутності інформації. Наведені формули аналогічні виразами, які використовуються в теорії статистичних рішень, де кожній парі повідомлень і рішень ставлять у відповідність деяку втрату корисності.

Як відомо, взаємна інформація між і та j системами визначається формулою:

$$I(x_i, y_i) = \log \frac{P(x_i / y_i)}{P(x_i)}.$$

Імовірність $P(x_i / y_i)$ дорівнює $P(x_i / y_i) = n_{резі} / n$, де $n_{резі}$ - число можливих сприятливих результатів після отримання інформації (повідомлення y_j). Імовірність $P(x_i)$ дорівнює $P(x_i) = n_{рез} / n$, де $n_{рез}$ – апріорне

число можливих сприятливих результатів до отримання інформації. В результаті отримаємо рівняння:

$$I(x_i, y_i) = \log \frac{n_{резі}}{n_{рез}}. \qquad (9)$$

У цьому вираз $n_{резі}$ характеризує один сприятливий результат з $n_{рез}$ можливих. Це випадкова сприятлива подія. При управлінні таких подій може бути кілька. В цьому випадку $n_{рез\ i}$ є випадковою величиною, яка характеризує ймовірність успішного результату q_{1i} при наявності інформації і q_{0i} - при її відсутності.

В результаті відповідно до (9) цінність отриманої інформації оцінюється за формулою:

$$K_i = \log \frac{P_{1i}}{P_{0i}}. \qquad (10)$$

тобто визначається через відношення ймовірностей досягнення мети при наявності і відсутності інформації. Це визначення називається «приростом інформації». В результаті отримаємо:

$$\Delta W_i = \left(\frac{\lambda_{вхі}}{\lambda_{вх}} \right)^{1 / \log \frac{p_{1i}}{p_{0i}}}. \qquad (11)$$

Чисельні значення P_{0i} і P_{1i}, а також $\lambda_{вхі}/\lambda_{вх}$ повинні визначатися при доступі в мережу. При відсутності інформації про чисельні величини P_{0i} і P_{1i} можливе визначення їх відносини, яке може бути зазначено в заголовку пакета у вигляді коефіцієнта, що характеризує пріоритетність цього повідомлення. Цей коефіцієнт може бути заданий в залежності від адреси відправника і одержувача при інших рівних умовах.

Імовірність правильного рішення P_{1i} залежить від кількості отриманої інформації про стан об'єкта управління. При відсутності інформації ця ймовірність зменшується до величини P_{0i}. Таке зниження в залежності від отриманої інформації повинно з'являтися по якомусь закону. Цей закон повинен бути таким, щоб при наявності малої кількості інформації чутливість показника була вище, ніж при наявності великого її кількості. Зазначеним умовам задовольняє вираз виду:

$$P_{1i}(t) = 1 - (1 - P_{0i}) \cdot e^{\frac{I_{0i}(t)}{I_{max} - I_{0i}(t)}}, \qquad (12)$$

де $I_{0i}(t)$ - кількість отриманої інформації; I_{max} - максимальна кількість інформації.

Залежність $P_{1i}(t)$ визначається розв'язаним завданням і задається законом зміни функції $I_{0i}(t)$. Крім того, частина інформації може бути втрачена. Тоді кількість доставленої інформації можна представити в наступному вигляді:

$$I_{0i}(t) = I_{від}(t) \cdot (1 - P_{втр}), \quad (13)$$

де $I_{від}$ (t)- кількість відправленої інформації ($0 \le I_{0i} \le$ де $I_{від}$ (t) $\le I_{max}$),

$P_{втр}$ - імовірність втрати інформації.

Кількість доставленої інформації визначається кількістю ресурсів КС, виділених для вирішення завдань управління.

За час доставки інформація старіє. В даному випадку під старінням інформації розуміється зменшення з часом її цінності, тобто зменшення ймовірності $P_{1i}(t)$.

Визначимо можливості обліку старіння інформації. Нехай X(t) випадковий процес, що описує функціонування керованого об'єкта, вимірюється в момент часу t_0. Момент вимірювання від моменту використання відокремлює затримка z, пов'язана з доставкою та обробкою цього значення. За цей час вимірюваний процес прийме значення x(t+z), відмінне від значення $x(t_0)$ і невідоме спостерігачеві. Наскільки придатним для системи прийняття рішення є отримане значення $x(t_0)$ залежить, як від властивостей вимірюваного процесу, так і від апріорних знань і здібностей спостерігача обчислювати майбутні значення процесу X(t).

Якщо, наприклад, процес є детермінованим, а закон його зміни відомий спостерігачеві, тоді спостерігач може обчислити значення процесу в момент часу з високою точністю. Інформація про детерминированій процес для спостерігача, який здійснює ефективну екстраполяцію, не застаріла.

Якщо ж вимірюваний процес випадковий, безпомилкова екстраполяція його ставати в принципі неможливою. Процес екстраполяції пов'язаний з помилкою: передбачене значення $x_e(t_0+z)$ складається з точного значення $x(t_0+z)$ і помилки: $e(t_0+z)$:

$$x_e(t_0+z) = x(t_0+z)+ e(t_0+z).$$

Величина помилки зростає з часом. Відновлений результат вимірювання буде все більше відрізнятися від істинного значення $x(t_0+z)$, достовірність результату буде зменшуватися. Для оцінки якості прогнозування найбільш часто використовується показник середнього квадрата помилки

$$e^2(t_0+z) = M\{ [x(t_0+z) - x_e(t_0+z)]^2\}. \quad (14)$$

Інтуїтивно зрозуміло, що близькі за часом значення можна передбачити з більшою точністю, ніж віддалені, і що через це зі збільшенням часу прогнозу середній квадрат помилки зростатиме.

На сьогоднішній день існують різні алгоритми екстраполяції випадкових процесів серед них: прогноз за останнім значенням, по математичному очікуванню, статистичний прогноз, фільтр Калмана-Бьюси і ін. Процеси, які відбуваються в КС, є випадковими. При прогнозуванні цих процесів різні алгоритми володіють різною точністю. У загальному випадку, квадрат помилки залежить від кореляційної функції R(z*)* випадкового процесу X(t). У свою чергу кореляційна функція випадкового процесу є функцією інтервалу часу z. При малих значеннях z, коли R(z)≈1, передбачені значення мало відрізняються від реальних значень випадкового процесу:

$$x_e(t_0+z) \approx x(t_0+z).$$

З збільшенням відстані часу кореляційна функція убуває, передбачені значення все менше відрізняються від математичного очікування процесу, тобто прогноз здійснюється по математичному очікуванню. Фактично кореляційна функція приходить до нуля в кінцевий час – протягом тимчасового інтервалу τ_0.

Помилку можна зменшити, застосовуючи більш ефективний і більш складний алгоритм прогнозування. Межею зменшення помилки буде застосування оптимального прогнозу з повною інформацією про передісторію процесу і імовірнісних властивості вимірюваного процесу. Таким чином, ймовірність старіння інформації, а отже, і ймовірність $P_{1i}(t)$ є функцією часу, і її вигляд залежить від характеристик керованого процесу, від властивостей алгоритму прогнозування і від закону зміни функції $I_{дост}(t)$.

Можна вказати і деякі загальні вимоги, якими повинна задовольняти функція I_{0i}, а саме:

1) $\Delta T_i = I_{max}$ при $0 < t < t_k$, де t_k - допустимий час, протягом якого цінність інформації не зменшується. Так, наприклад, при передачі радіолокаційної інформації t_k визначається періодом огляду радіолокаційної станції;

2) $I_{0i}(t) = 0$ при $t > \Delta T_i$, де ΔT_i - допустимий інтервал, по закінченню якого інформація, що міститься в повідомленні, вважається повністю застарілою;

3) при $t_k < t < \Delta T_i$ цінність інформації ΔT_i повинна зменшуватися відповідно до якоїсь функцією від $I_{0i}(t)$ до 0. З деяким наближенням для

багатьох інформаційних задач ця функція може бути лінійною. Зазначеним вимогам задовольняє наступна функція:

$$I_{дост}(t) = \begin{cases} I_{max} & при \quad 0 \le t \le t_{k_i} \\ I_{max} \cdot (1 - \dfrac{t - t_{ki}}{\Delta T_i - t_{ki}}) & при \quad t_{k_i} \le t \le \Delta T_i \\ 0 & при \quad t > \Delta T_i \end{cases}. \qquad (15)$$

В даному вираженні значення t визначається часом доставки повідомлень в мережах ($T_{дост}$).

При використанні цього виразу досить задати лише один показник t_k і співвідношення t_k / ΔT. При t_k / ΔT = 1 цінність інформації в момент часу t_k стрибкоподібно змінює своє значення з 1 на 0. Якщо $0 < t_k/\Delta T < 1$, тоді цінність інформації з моменту часу монотонно (лінійно) убуває від 1 до 0 до моменту часу ΔT_i.

При неповноті отриманої інформації може виявитися, що завдання управління по отриманому повідомленню не може бути вирішена. В цьому випадку приймається $I_{дост}(t) = 0$ і $P_{дост}(t) = P_0(t)$.

Поняття старіння інформації нерозривно пов'язане з поняттям її цінності. Тому старіння інформації призводить до зміни від часу функції (15) і, отже, ймовірність своєчасного вирішення поставленого завдання. Ця залежність визначається вмістом конкретного завдання. Однак можна вказати і деякі загальні вимоги, яким повинна задовольняти функція, аналогічні вимогам до при обліку старіння інформації.

1) $P_{1i}(t) = P_{1i}$ при $0 < t < t_k$, де t_k - допустимий час, протягом якого цінність переданої інформації не зменшується;

2) $P_{1i}(t) = P_{0i}$ при $t > \Delta T_k$ де ΔT_k - допустимий інтервал, протягом якого інформація, що міститься в повідомленні не вважається повністю застарілою;

3) при $t_k < t < \Delta T_k$ ймовірність $P_{1i}(t)$ повинна зменшуватися за деякою функції від P_{1i} до P_{0i}. З деяким наближенням для багатьох інформаційних процесів ця функція може бути лінійною.

Зазначеним вимогам відповідає функція виду:

$$P_{1i}(t) = \begin{cases} P_{1i} & при \quad 0 \le t \le t_k; \\ (P_{1i} - P_{0i})\left(1 - \dfrac{t - t_{ki}}{\Delta T_{ki} - t_{ki}}\right) + P_{0i} & при \quad t_k < t \le \Delta T_k; \\ P_{0i} & при \quad t > \Delta T_k. \end{cases} \qquad (16)$$

Введений показник ефективності передбачає виконання обмежень $T_д \le T_{д_{доп}}$ и $P_{пом} \le P_{пом_{доп}}$. Якщо в окремих напрямках зазначені обмеження не

виконуються, то для цих напрямків ймовірність втрати пакета в вираженні (5) приймається рівною 1. Розглянемо спосіб оцінки ефективності КС в умовах невизначеності.

В умовах невизначеності не завжди можна задати значення параметрів, що входять в наведені вище вирази, і тим більше врахувати обмеження на якість інформації, що передається. Крім цього, за умов невизначеності, тобто при наявності неповної або нечіткої інформації, зазначені раніше обмеження можуть бути задані з деяким ступенем невпевненості або функціями належності. Наприклад, обмеження будуть мати вигляд $P_{пом} \le P_{пом_{доп}}$ з ймовірністю $P_{пом_{нч}}$, що знаходиться в інтервалі від $P_{пом_{нч1}}$ до $P_{пом_{нч2}}$, або з функцією приналежності $\mu_{нч}(P_{пом})$. Аналогічно можна записати обмеження для часу доставки з наперед визначеною імовірністю.

Якщо не передбачити можливість обліку таких ситуацій при розрахунку ефективності управління мережею, то для таких напрямків завдання може взагалі не вирішуватися або буде розраховуватися по (15) при $P_{втр1}=1$. В результаті оцінка ефективності управління мережею буде проведена невірно. Це призведе до тимчасового і матеріального збитків. Для усунення цього недоліку зазначені вище обмеження можна врахувати введенням коефіцієнта, який будемо називати коефіцієнтом довіри до інформації, що передається по і-му напрямку.

Нехай значення параметра описується чітким числом x = x0, тоді він знаходиться в одному зі своїх станів, які характеризуються функціями належності $\mu_i(x = x0)$. Якщо ж значення параметра представлено нечітким числом або інтервалом, то його характеристикою може бути як ступінь приналежності того чи іншого стану, так і ймовірність прийняття рішення про приналежність до того чи іншого стану. Перше поняття з теорії нечітких множин, друге – теорії ймовірностей. Обидва поняття характеризують рівень нашого незнання про справжній стан параметра.

Оскільки прийняття рішення про те, що параметр відноситься до якого-небудь стану – це подія, що має імовірнісну природу, то ймовірність приналежності до того чи іншого стану може служити додатковою оцінкою нечіткого параметра.

Визначити ймовірності приналежності параметра нечітким підмножини можна на основі аналізу чітких (або нечітких) інтервалів шляхом розбиття вихідних даних на α-рівні. При цьому кожне нечітке число, відповідне певному значенню лінгвістичної змінної, а також нечітке число, що

характеризує можливі значення параметра представляються сукупністю α-рівнів. На кожному α-рівні можна виконати порівняння інтервалів, оцінивши ймовірність їх належність чи неналежність один одному.

Оцінку поточного стану показника можна характеризувати належністю відповідній нечіткій множині. Згідно [3], аналізуючи досліджуваний показник, можна зробити висновок про ймовірність його приналежності якої-небудь нечіткій множині.

Відповідно до [1], чіткі ймовірності приналежності параметра x = xi нечітким множинам M1 і M2 визначаються за формулами:

$$P_{M1}(x = x_i) = \mu_1(x = x_i)p(x = x_i);$$
$$P_{M2}(x = x_i) = \mu_2(x = x_i)p(x = x_i). \qquad (17)$$

У формулі (17) використовуються такі символи:

РМ1, Рм2 – чіткі ймовірності приналежності параметра x = xi нечітким множинам M1 і M2;

p(x = xi) – ймовірність виникнення події x = xi;

$\mu_i(x)$ (x) (i = 1, 2) - функція приналежності показника відповідній нечіткій множині.

Якщо p(x = xi) = 1, то ймовірність приналежності показника заданого стану буде чисельно дорівнює значенню відповідної функції приналежності.

Якщо параметр (змінна) X визначений на інтервалі [x0, xк] і кожне з можливих чітких його значень може належати двом станам M1 і M2 з функціями належності $\mu_1(x)$ і $\mu_2(x)$ відповідно, то ймовірності приналежності таких станів визначаються за формулами [3]:

$$P_{M1}(x) = \frac{1}{x_k - x_0}\int_{x_0}^{x_k} \mu_1(x)\mu(x)dx;$$
$$P_{M2}(x) = \frac{1}{x_k - x_0}\int_{x_0}^{x_k} \mu_2(x)\mu(x)dx, \qquad (17)$$

де x_0 – початкове значення параметру на інтервалі $[x_0, x_к]$;

$x_к$ – кінцеве значення параметра на інтервалі $[x_0, x_к]$;

μ(x) – функція приналежності показника поточним можливих значеннях.

Визначимо функції приналежності $\mu_i(t)$ (i = 1, 2) параметра старіння інформації відповідній нечіткій множині.

Сформулюємо загальні вимоги, якими повинна задовольняти функції $\mu_i(t)$ (i=1;2):

1) $\mu_1(t) = 1$, $\mu_2(t) = 0$ при $0 < t < t_k$;

2) $\mu_1(t) = 0$, $\mu_2() = 1$ при $t > \Delta T_i$;

3) при $t_k < t < \Delta T_i$ функція $\mu_1(t)$ повинна зменшуватися відповідно від 1 до 0, а функція $\mu_2(t)$ – збільшуватись від 0 до 1. З деяким наближенням ці функції можна вважати лінійними.

Зазначеним вимогам задовольняють функції (рис.1):

$$\mu_1(t) = \begin{cases} 1 & \text{при} \quad 0 \le t \le t_{k_i} \\ \dfrac{\Delta T_i}{\Delta T_i - t_{ki}} \cdot (1 - \dfrac{t}{\Delta T_i}) & \text{при} \quad t_{k_i} \le t \le \Delta T_i \\ 0 & \text{при} \quad t > \Delta T_i \end{cases};$$

$$\mu_2(t) = \begin{cases} 1 & \text{при} \quad 0 \le t \le t_{k_i} \\ \dfrac{\Delta T_i}{\Delta T_i - t_{ki}} \cdot (\dfrac{t}{\Delta T_i} - 1) & \text{при} \quad t_{k_i} \le t \le \Delta T_i \\ 0 & \text{при} \quad t > \Delta T_i \end{cases}.$$

Вибір функції належності можливим поточним значенням середнього часу доставки повідомлення $\mu(t)$ визначається в результаті моделювання процесу обміну даними або статистичного аналізу експериментальних досліджень. На практиці часто застосовують такі функції належності: трикутні, трапецієподібні, нелінійні (функція Гауса, сигмоїдальна функція, сплайн). На рис.1 наведена трапецієподібна функція $\mu(t)$.

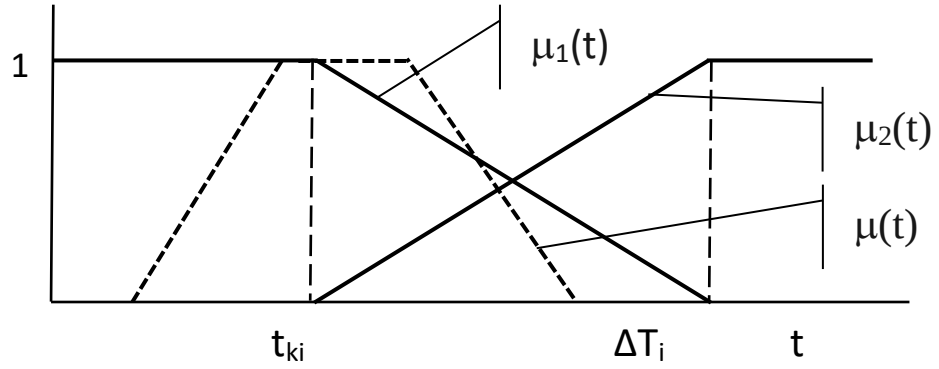


Рис.1. Функції належності середнього часу доставки повідомлення

Відповідно до (10) цінність отриманої інформації з урахуванням її старіння протягом часу t оцінюється за формулою:

$$K_1(t) = P_{M1}(t) \cdot \log \frac{P_{1i}}{P_{0i}}.$$

Ступінь втраченою цінності при отриманні інформації через час t визначається відповідно виразу:

$$K_2(t) = P_{M2}(t) \cdot \log \frac{P_{1i}}{P_{0i}}.$$

Тоді, вага і-го повідомлення даних джерела з урахуванням його старіння в процесі передачі даних визначається за формулою:

$$\Delta W_i = \left(\frac{\lambda_{вхі}}{\lambda_{вх}} \right)^{1 / P_{M1} \log \frac{p_{1i}}{p_{0i}}}.$$

Таким чином, пропонований показник ефективності системи управління дозволяє врахувати вплив інтенсивності і важливості інформаційного потоку, ймовірності втрати повідомлень як при відсутності, так і при наявності обліку старіння інформації.

Висновки

Управління ресурсами комп'ютерної мережі може здійснюватися вибором стратегії і тактики розподілу ресурсів (централізована, ієрархічна, децентралізована) методу інформаційного забезпечення, методу управління канальними, буферними, інформаційними і тимчасовими ресурсами. Приймаючи рішення по управлінню мережою необхідно вміти оцінювати наслідки цього управління.

З метою виявлення шляхів підвищення ефективності систем мережевого управління пропонується узагальнена математична модель технології централізованого способу управління інформаційними процесами в адаптивних комп'ютерних мережах. Технологія централізованого способу управління передбачає такі етапи, як збір інформації про стан мережі і окремих її елементах, рішення задачі розподілу завдань між вузлами комутації, доведення до всіх керованих об'єктів результатів цього рішення, виконання поставлених завдань центрами комутації, контроль виконання рішення і при необхідності коригування процесу управління. Модель дозволяє визначити середнє значення, дисперсію часу і ймовірність помилки управління в залежності від якості виконання завдання на кожному етапі технологічного процесу, як при наявності повної, так і неповної і застарілої інформації. Розроблений показник ефективності системи управління розподіленою інформаційною системою. Показано, що ефективність управління процесами обробки і доставки інформації необхідно оцінювати за

результатами приросту відносної швидкості передачі і обробки даних при наявності управління відносно її ефективності без управління за умовами забезпечення потрібної якості обслуговування. Розроблена методика оцінки важливості потоків інформації, яка враховує не тільки вагу цього потоку, але і зміну його цінності в процесі старіння з часом передачі даних для розв'язання завдань управління. Це дає можливість розробляти вимоги до характеристик комп'ютерих мереж.

Література

1. Лосев Ю. И. Автоматизация в сетях с коммутацией пакетов / Ю. И. Лосев, Ф. К. Яковец . – Київ. : «Техніка», 1994. – 212 с..
2. Методы и модели обмена информацией в распределенных адаптивных вычислительных сетях с временной параметризацией параллельных процессов: монография / Ю.И. Лосев, С.И. Шматков, К.М. Руккас.– Харьков: ХНУ им. В.Н.Каразина, 2011. – 204с.
3. Інформаційні технології: сучасний стан та перспективи: монографія / за заг. ред.. В.С. Пономаренка.– Харків: ТОВ «ДІСА ПЛЮС», 2018. – С. 102–118.
4. Закиров З.З. Методика определения эквивалентной вероятности ошибки, среднего значения и дисперсии времени при передечи кодовой комбинации в системах с обратной связью / З.З. Закиров // Системи обробки інформації: збірник наукових праць. – 2010. – Вип.1 (82). – С.34-36.
5. Minukhin S. V. Analysis of ways for exchanging data in networks with package commutation / S. V. Minukhin, D. E Sitnikov, M. U. Losev, // Radio Electronics Computer Science Control. – 2018. – №4. – С.196-204. https://doi.org/10.15588/1607-3274-2018-4-19.

ГЛАВА 6
МЕТОДИКА ПРОСУВАННЯ ІНТЕРНЕТ-МАГАЗИНУ В СОЦІАЛЬНИХ МЕРЕЖАХ

Вступ і постановка задачі

На сьогодні в умовах інформаційного суспільства важливого значення набувають соціальні мережі, використання яких створює широкі можливості для різних напрямків людської діяльності. Зокрема, значні переваги соціальні мереж надають для маркетингової та рекламної діяльності компаній, які здійснюють свою діяльність в мережі Інтернет. В контексті ефективного маркетингу інтернет-компаній центральне місце займає просування інтернет-магазину в соціальних мережах, а створення відповідної методики надає можливість формування науково обґрунтованого контент-плану та контент-стратегії організації.

Незважаючи на малий термін розвитку інтернет-реклами в Україні, вітчизняні рекламодавці вже набули достатнього досвіду та і реклама стала значущою реальністю, для вивчення якої необхідні конкретні наукові підходи.

У дослідженнях [1-4] пропонуються різноманітні види технологій створення інтернет-магазину та методичні рекомендації стосовно підвищення ефективності діяльності компаній в мережі Інтернет. Проте на сьогодні в спеціалізованій літературі відстутня комплексна науково обгрунтована методика підвищення ефективності інтернет-реклами.

Метою даного розділу є розробка методики просування інтернет-магазину в соціальних мережах.

Основна частина

Під рекламною кампанією слід розуміти стратегію проведення ретельно спланованих рекламних заходів і організації подій для просування компанії або продукту на ринок.

Основна мета інтернет-реклами – донесення якісних рекламних повідомлень про продукт до кінцевого споживача за допомогою різних видів реклами і відповідних рекламних носіїв [1]. Важливу роль відіграють зміст і форма подачі рекламних повідомлень, дизайн реклами, засоби поширення реклами, час виходу повідомлень, кількість публікацій і т. ін.

Методика підвищення ефективності інтернет-реклами має включати такі основні етапи:

1) аналіз ринка та визначення мети рекламної кампанії;
2) визначення цільової аудиторії;
3) виготовлення рекламних матеріалів;
4) визначення набору засобів інтернет-реклами;
5) вибір рекламних майданчиків і носіїв;
6) створення рекламного оголошення та проведення рекламної кампанії;
7) оцінка ефективності рекламної кампанії.

В ході експерименту було створено рекламну кампанію для інтернет магазину «Футбоголік».

«Футбоголік» – це інтернет-магазин молодіжного одягу з принтами, які здатні здивувати оточуючих своєю оригінальністю та індивідуальністю

Для визначення виду інтернет-реклами, який був використаний для рекламної кампанії інтернет-магазину скористуємось методом аналізу ієрархій.

Проаналізувавши всі види інтернет-реклами , із них виділимо три варіанти, які можуть вирішити поставлені задачі: контекстна, банерна, таргетингова реклама в соціальних мережах.

Визначимо найбільш важливі критерії для інтернет-реклами:

1) охоплення цільової аудиторії;
2) вартість рекламної кампанії;
3) емоційний вплив реклами;
4) складність реалізації.

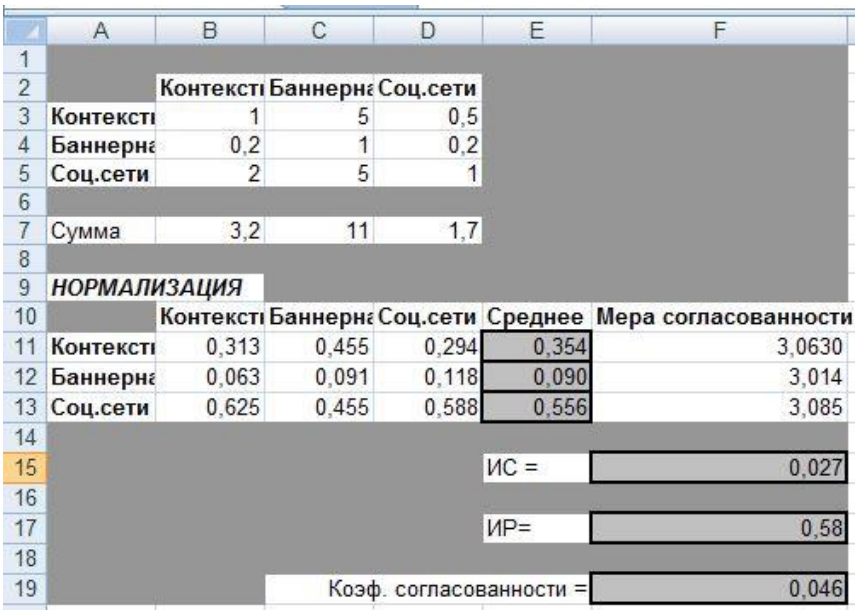
Перший крок процедури MAI полягає в попарному порівнянні видів інтернет-реклами за кожним критерієм. Для цього використовуємо стандартну шкалу порівняння, наведену в табл. 1.

Таблиця 1

Шкала порівняння

Рейтинг	**Опис**
1	Однакова перевага
3	Помірна перевага
5	Явна перевага
7	Очевидна перевага
9	Абсолютна перевага

Починаємо з першого критерію (охоплення цільової аудиторії) і вносимо в таблицю Excel дані, показані на рис. 1.



	A	B	C	D	E	F
1						
2		Контексті	Баннерна	Соц.сети		
3	Контексті	1	5	0,5		
4	Баннерна	0,2	1	0,2		
5	Соц.сети	2	5	1		
6						
7	Сумма	3,2	11	1,7		
8						
9	*НОРМАЛИЗАЦИЯ*					
10		Контексті	Баннерна	Соц.сети	Среднее	Мера согласованности
11	Контексті	0,313	0,455	0,294	0,354	3,0630
12	Баннерна	0,063	0,091	0,118	0,090	3,014
13	Соц.сети	0,625	0,455	0,588	0,556	3,085
14						
15					ИС =	0,027
16						
17					ИР=	0,58
18						
19				Коэф. согласованности =		0,046

Рис. 1. Попарне порівняння і нормалізація за показником охоплення цільової аудиторії

Після виконання всіх попарних порівнянь нормалізуємо матрицю шляхом підсумовування чисел в кожному стовпці і подальшого поділу кожного елемента стовпця на отриману для даного стовпця суму. Наступний крок полягає в обчисленні балів для кожного виду реклами за критерієм охоплення цільової аудиторії. Ці значення показані на рис. 1 в стовпці Е. Видно, що найвищий серед-ній бал за даним критерієм має інтернет-реклама в соціальних мережах (рис. 3.1). Завершивши нормалізацію матриці, вирахували коефіцієнт узгодженості і пере-вірили його значення. Для обчислення індексу узгодженості визначили середню міру узгодженості всіх трьох видів реклами, з неї відняли кількість можливих варіантів вирішення n і результат поділили на n-1. Індекс узгодженості ІС показаний на рис. 1 в осередку F15, його значення дорівнює 0,027.

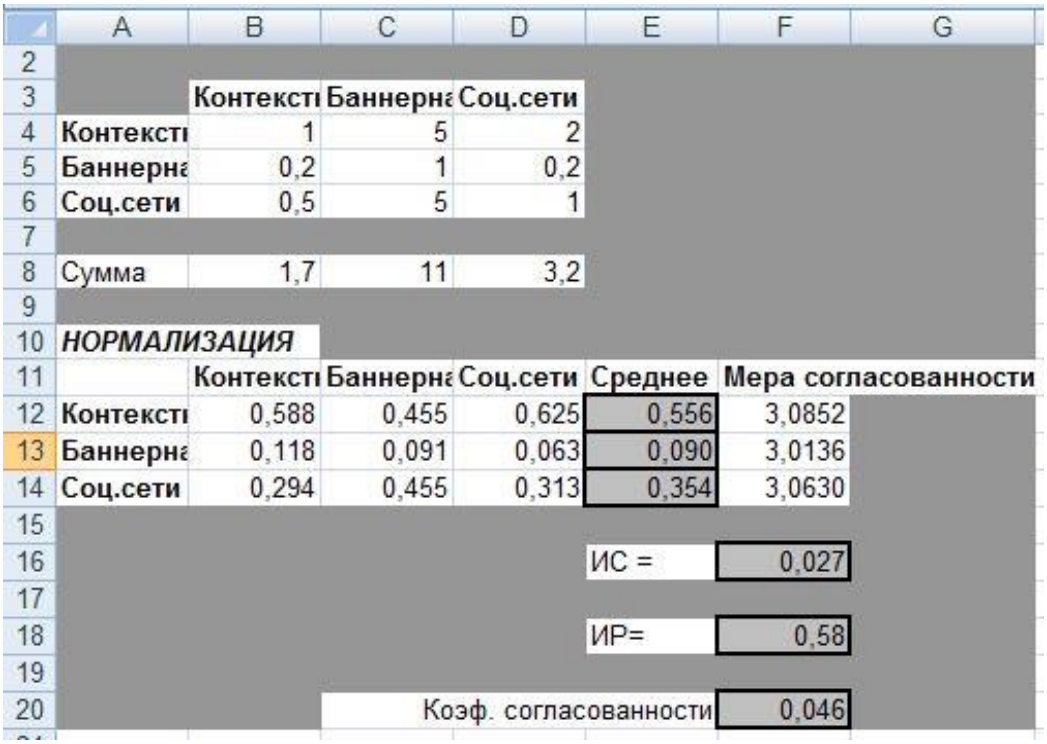
Останній етап визначення коефіцієнта узгодженості полягає в розподілі ІС на індекс рандомізації ІР, значення якого для різних значень n обчислюються в методі МАІ спеціальним чином і наведені в табл. 2. Коефіцієнт узгодженості дорівнює 0,046.

Таблиця 2

Значення індексу рандомізації

n	Індекс рандомізації
2	0,00
3	0,58
4	0,90
5	1,12
6	1,24
7	1,32
8	1,41
9	1,45
10	1,51

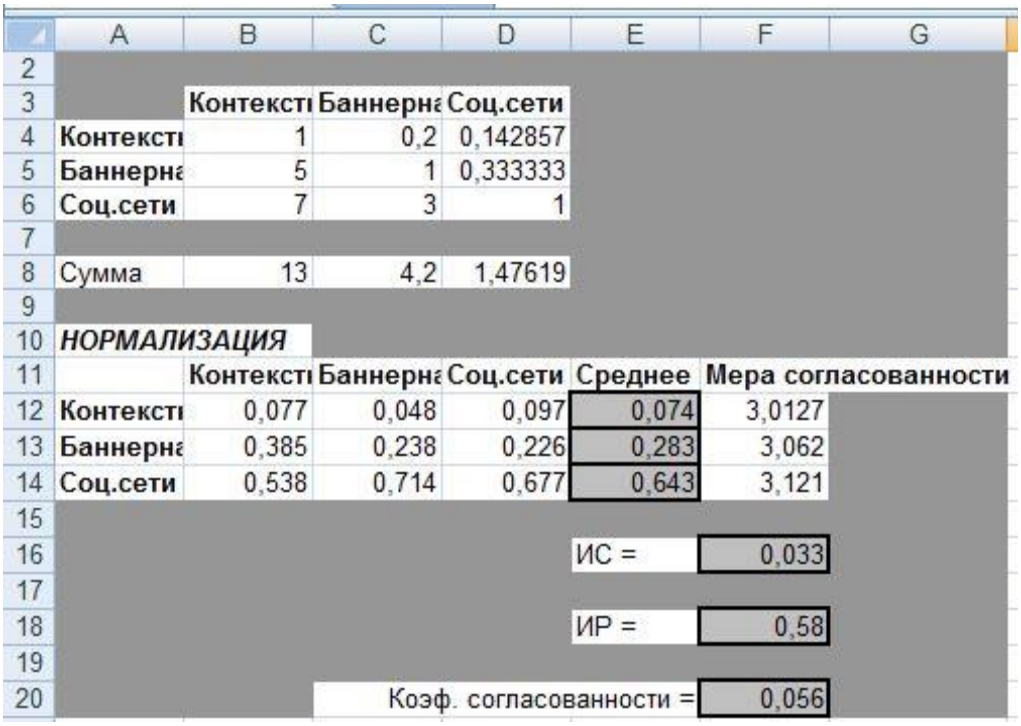
Далі за аналогією розраховуємо інші три критерії (рис. 2, рис. 3, рис. 4).



	A	B	C	D	E	F	G
2							
3		Контексті	Баннерна	Соц.сети			
4	Контексті	1	5	2			
5	Баннерна	0,2	1	0,2			
6	Соц.сети	0,5	5	1			
7							
8	Сумма	1,7	11	3,2			
9							
10	*НОРМАЛИЗАЦИЯ*						
11		Контексті	Баннерна	Соц.сети	Среднее	Мера согласованности	
12	Контексті	0,588	0,455	0,625	0,556	3,0852	
13	Баннерна	0,118	0,091	0,063	0,090	3,0136	
14	Соц.сети	0,294	0,455	0,313	0,354	3,0630	
15							
16					ИС =	0,027	
17							
18					ИР=	0,58	
19							
20				Коэф. согласованности		0,046	

Рис. 2. Попарне порівняння і нормалізація за показником вартість

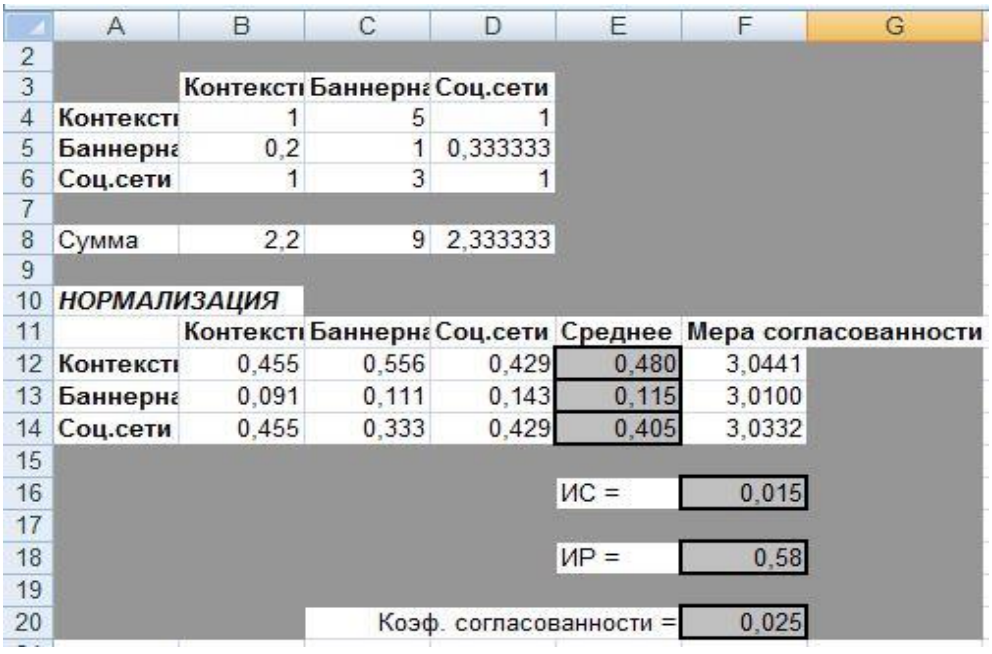
Попарне порівняння і нормалізацію критеріїв за показником «емоційний вплив» зображено на рис. 3.



	A	B	C	D	E	F	G
2							
3		Контексті	Баннерна	Соц.сети			
4	Контексті	1	0,2	0,142857			
5	Баннерна	5	1	0,333333			
6	Соц.сети	7	3	1			
7							
8	Сумма	13	4,2	1,47619			
9							
10	*НОРМАЛИЗАЦИЯ*						
11		Контексті	Баннерна	Соц.сети	Среднее	Мера согласованности	
12	Контексті	0,077	0,048	0,097	0,074	3,0127	
13	Баннерна	0,385	0,238	0,226	0,283	3,062	
14	Соц.сети	0,538	0,714	0,677	0,643	3,121	
15							
16					ИС =	0,033	
17							
18					ИР =	0,58	
19							
20				Коэф. согласованности =		0,056	

Рис. 3. Попарне порівняння і нормалізація за показником «емоційний вплив»

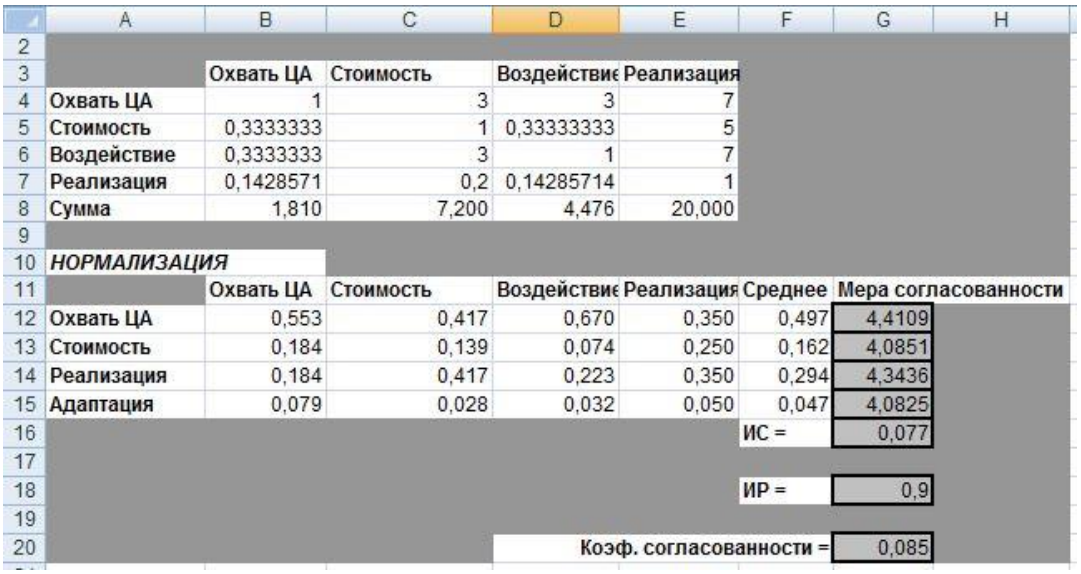
На рис.4 зображено попарне порівняння критеріїв і нормалізація за показником «складність реалізації».



	A	B	C	D	E	F	G
2							
3		Контексті	Баннерна	Соц.сети			
4	Контексті	1	5	1			
5	Баннерна	0,2	1	0,333333			
6	Соц.сети	1	3	1			
7							
8	Сумма	2,2	9	2,333333			
9							
10	*НОРМАЛИЗАЦИЯ*						
11		Контексті	Баннерна	Соц.сети	Среднее	Мера согласованности	
12	Контексті	0,455	0,556	0,429	0,480	3,0441	
13	Баннерна	0,091	0,111	0,143	0,115	3,0100	
14	Соц.сети	0,455	0,333	0,429	0,405	3,0332	
15							
16					ИС =	0,015	
17							
18					ИР =	0,58	
19							
20				Коэф. согласованности =		0,025	

Рис. 4. Попарне порівняння і нормалізація за показником «складність реалізації»

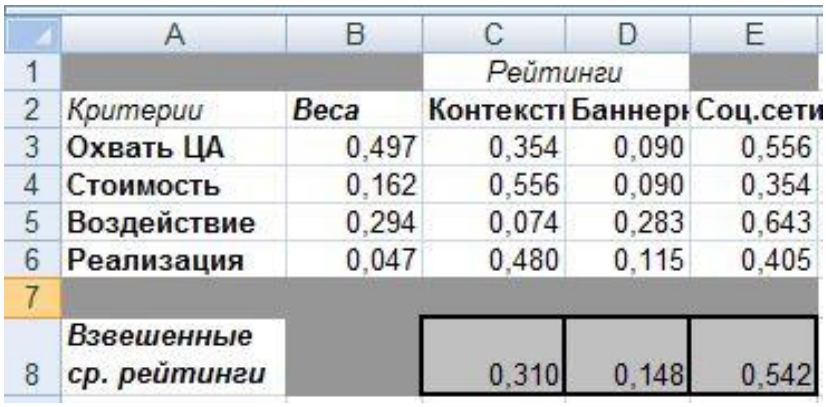
Далі здійснили аналогічні попарні порівняння критеріїв для визначення ваг критеріїв (рис. 5).



	A	B	C	D	E	F	G	H
2								
3		Охвать ЦА	Стоимость	Воздействие	Реализация			
4	Охвать ЦА	1	3	3	7			
5	Стоимость	0,3333333	1	0,33333333	5			
6	Воздействие	0,3333333	3	1	7			
7	Реализация	0,1428571	0,2	0,14285714	1			
8	Сумма	1,810	7,200	4,476	20,000			
9								
10	*НОРМАЛИЗАЦИЯ*							
11		Охвать ЦА	Стоимость	Воздействие	Реализация	Среднее	Мера согласованности	
12	Охвать ЦА	0,553	0,417	0,670	0,350	0,497	4,4109	
13	Стоимость	0,184	0,139	0,074	0,250	0,162	4,0851	
14	Реализация	0,184	0,417	0,223	0,350	0,294	4,3436	
15	Адаптация	0,079	0,028	0,032	0,050	0,047	4,0825	
16						ИС =	0,077	
17								
18						ИР =	0,9	
19								
20				Коэф. согласованности =			0,085	

Рис. 5. Коефіцієнт узгодженості для ваг критеріїв

Останній крок полягав в обчисленні зважених середніх оцінок для кожного варіанта рішення і застосуванні отриманих результатів для прийняття рішення про те, який вид інтернет-реклами буде обраний для задоволення поставленої мети для інтернет-магазину одягу «Футболік». Кінцеві обчислення представлені на рис. 6. На підставі отриманих результатів був зроблений висновок, що тар-гетингова реклама в соціальних мережах (показник 0,542 в комірці E8) перевершує контексну (0,310 в комірці C8), а банерна реклама має показник значно нижче (0,148 в комірці D 8).



	A	B	C	D	E
1			*Рейтинги*		
2	*Критерии*	*Веса*	Контекст	Баннер	Соц.сети
3	Охвать ЦА	0,497	0,354	0,090	0,556
4	Стоимость	0,162	0,556	0,090	0,354
5	Воздействие	0,294	0,074	0,283	0,643
6	Реализация	0,047	0,480	0,115	0,405
7					
8	*Взвешенные ср. рейтинги*		0,310	0,148	0,542

Рис. 6. Зважене середнє рейтингів з використанням ваг

На основі отриманого аналізу було прийнято рішення проводити рекламну кампанію в соціальних мережах.

На підставі проаналізованої інформації та виготовлених матеріалів переходимо до етапу налаштування та запуску рекламної кампанії. Реклама для Facebook та Instagram налаштовується за допомогою Ads Manager у Facebook.

Ads Manager – це інструмент Facebook, який дає можливість створювати рекламу і керувати нею. Ads Manager надає такі можливості:

1) створювати і показувати рекламу;
2) націлити реклами на потрібних людей;
3) налаштувати бюджет;
4) відслідковувати результативність реклами.

Наступним кроком Ads Manager пропонує провести A/B-тестування оформлення, плейсментів, аудиторії та стратегії оптимізації (рис. 7).

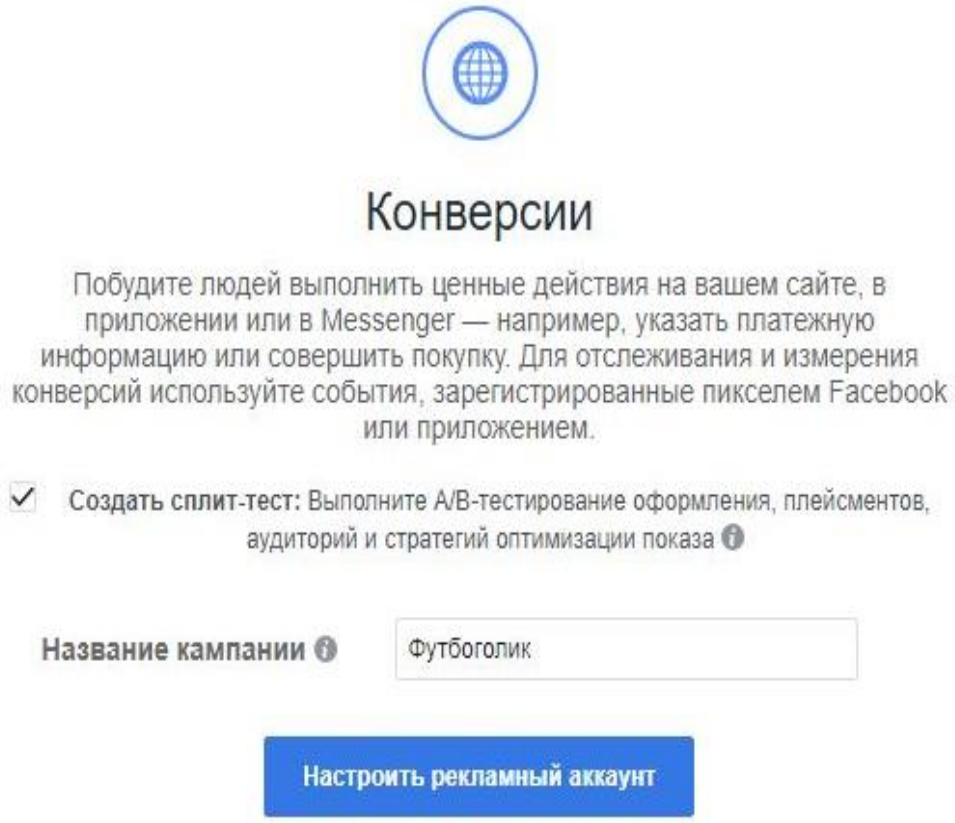


Рис. 7. Узгодження на створення спліт-тесту

Ґрунтуючись на обраній меті рекламної кампанії, для подальшого просування був обраний сайт та налаштован інструмент аналітики «Піксель» для відстеження цільової дії – покупки товару.

На сьогодні більшість оголошень в соціальних мережах також переглядаються на мобільних пристроях. В зв'язку з цими даними було прийнято рішення провести спліт-тест для плейсменту на мобільних пристроях (рис. 8).

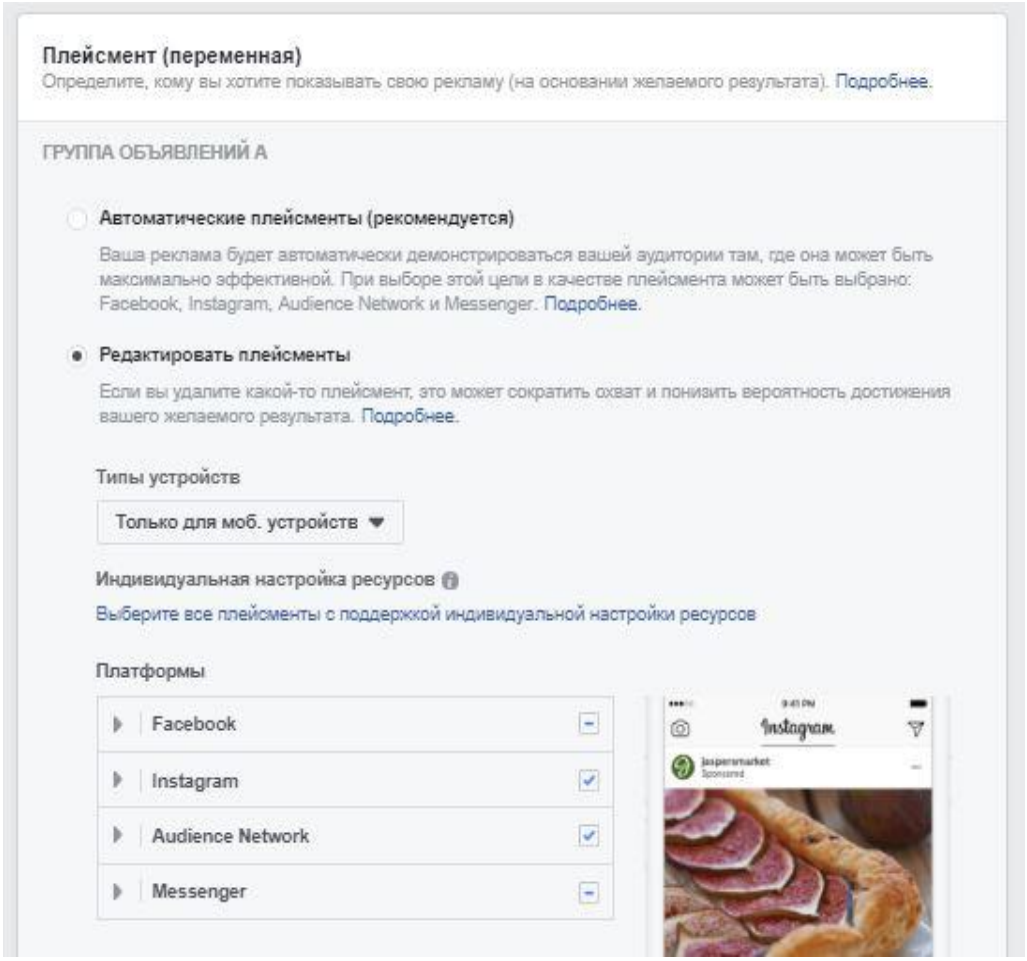


Рис. 8. Налаштування спліт-тесту для плейсменту на мобільних пристроях

Відповідно до проаналізованої та сформованої цільової аудиторії, налаштували таргетинг (рис. 9).

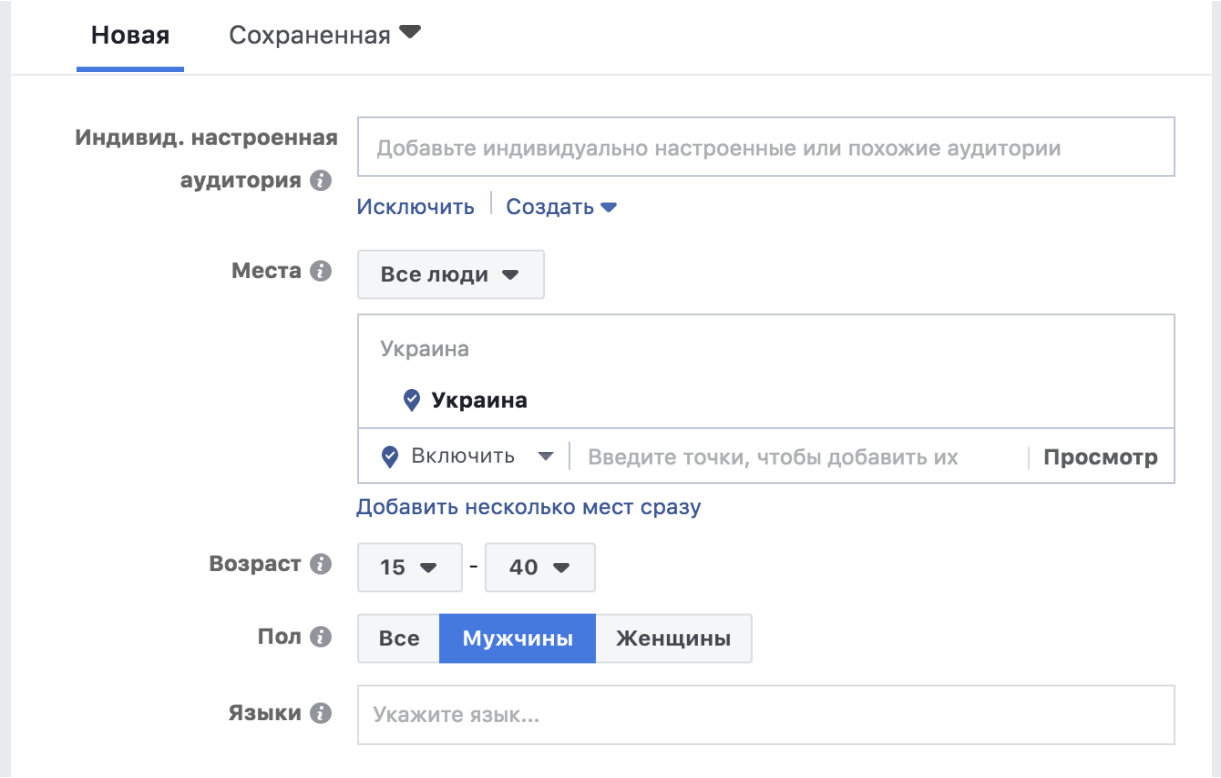


Рис. 9. Налаштування таргетингу

До детального таргетингу увійшли користувачі, які виразили інтерес або їм сподобалась сторінка, пов'язана з футбольним клубом «Динамо» (рис. 10).

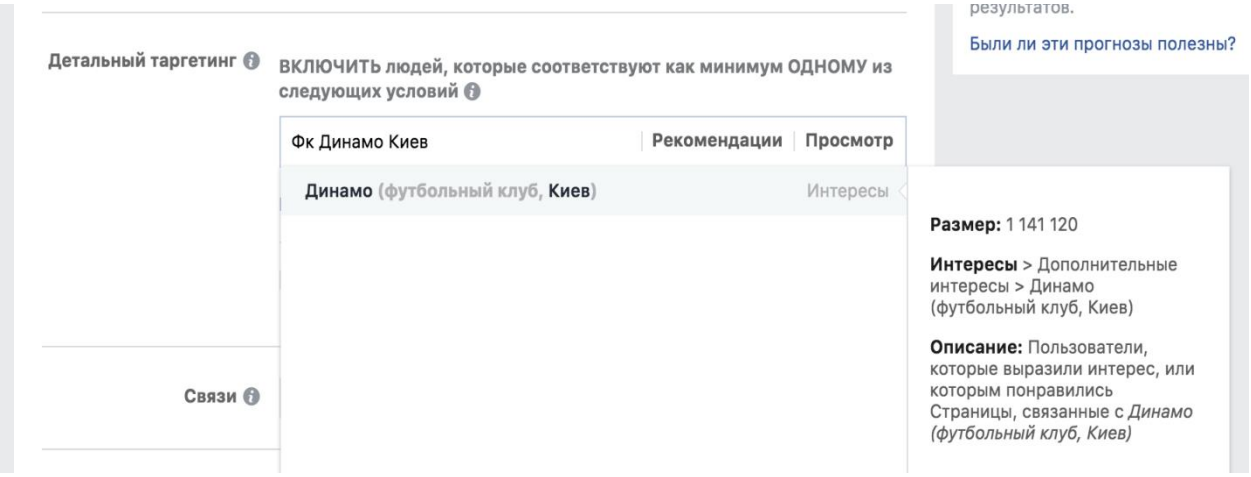


Рис. 10. Налаштування детального таргетингу

В оптимізації доставок було виявлено вікно конверсії, тобто, кількість часу, який звичайно проходить від моменту перегляду реклами або кліку до виконання цінного дії (конверсії). В спліт-тесті показники

підлаштовуються автоматично: стратегія ставок – найнижча ціна, оплата за показ, показ реклами постійно (рис. 11).

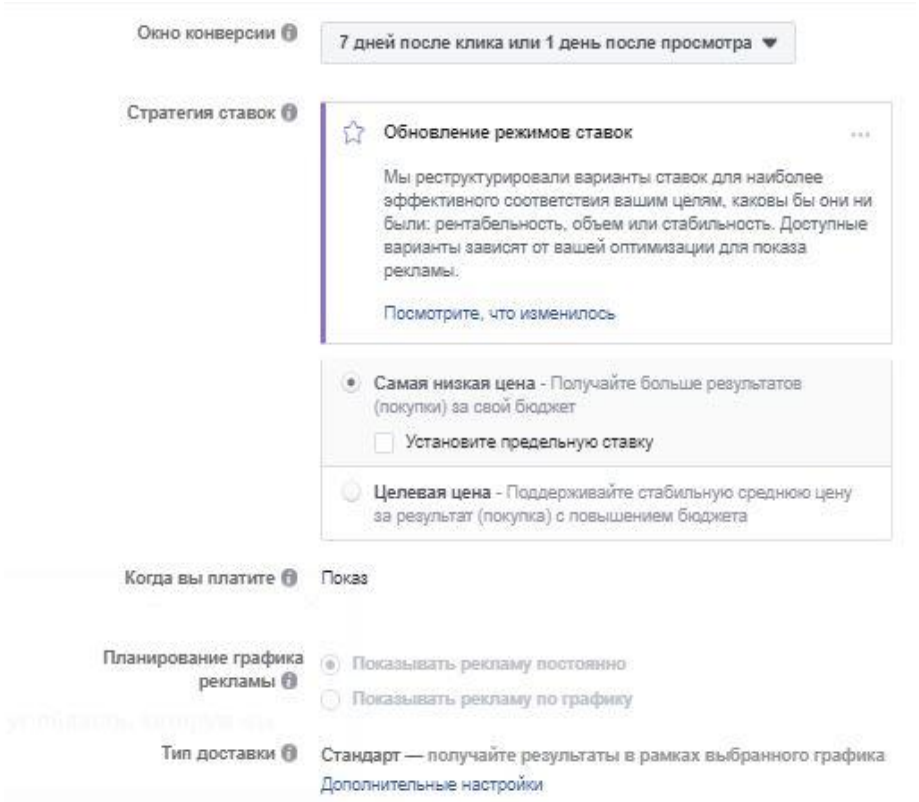


Рис. 11 Оптимізація доставки

На спліт-тест було виділено $50. Цей бюджет було розділено 50 % на 50 % між двома групами оголошень. Тривалість тесту 4 дні (рис. 12).

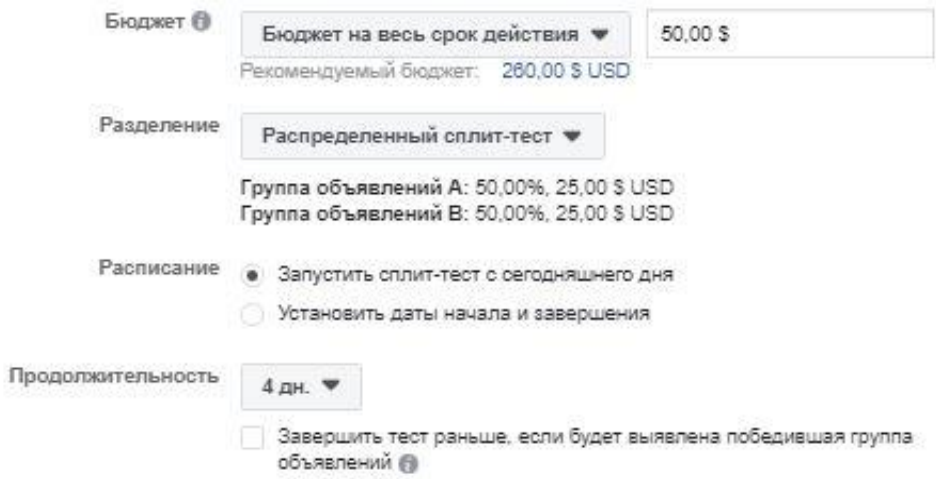


Рис. 12. Бюджет на спліт-тест рекламної кампанії

Наступний крок оформлення реклами, формат та вигляд реклами (заголовок, текст реклами, зображення). Оформлюємо вигляд реклами для Facebook та Instagram для групи А та В оголошень.

Після виконання всіх налаштувань був запущений спліт-тест для двох груп оголошень, де порівнювались показ реклами стрічка Instagram, стрічка Facebook, Instagram Stories на мобільному пристрої проти стрічка Facebook на ПК.

Група оголошень, яка перемогла, визначається шляхом порівняння ціни за результат для кожної групи оголошень. Беручи до уваги дані тестування, Facebook моделює використання кожної змінної десятків тисяч разів, щоб визначити, як часто результат який переміг може виграти. Результат був оцінений по отриманим конверсіям та їх вартості. В даному випадку ми отримали результат плейсменту на мобільних простоях 19 конверсій по $1,31, а плейсмент на ПК – 7 конверсій по $3,57. Хоча клік у другому випадку в 2 рази дешевше (рис. 13).

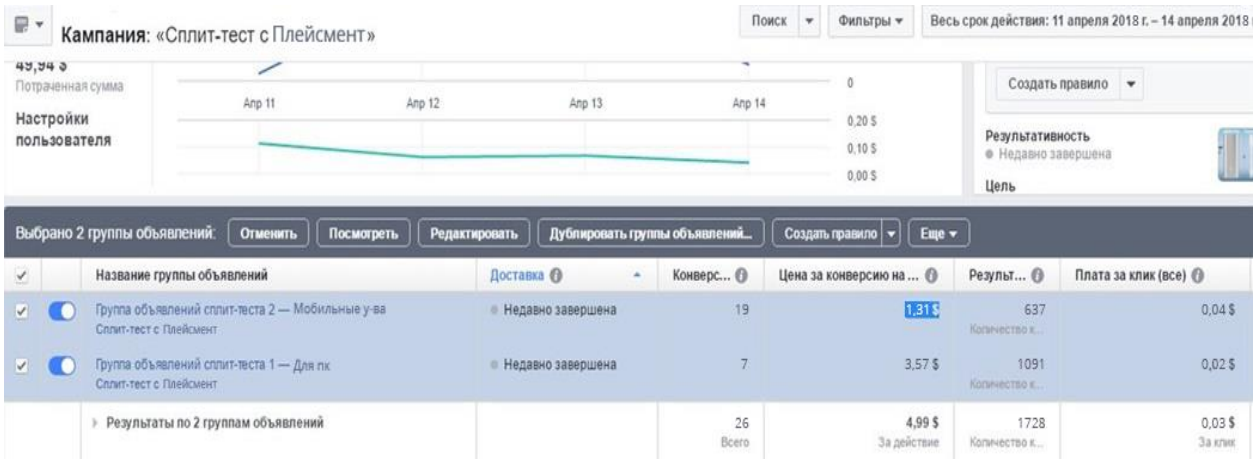


Рис. 13. Результат спліт-тесту

З проведеного спліт-тесту був зроблений висновок, що для даної цільової аудиторії ціна за конверсію, а саме за оформлення замовлення, нижча у плейсмента з мобільного пристрою, тому надалі рекламна кампанія буде проводитись лише для соціальних мереж на мобільних пристроях (стрічка Facebook, стрічка Instagram, Instagram Stories).

Facebook вивів приблизну аудиторію, яка відповідає заданим параметрам. Рекламна кампанія охоплює близько 75 000 людей, охоплення на день – 3 600-1 000 людей (рис.14).

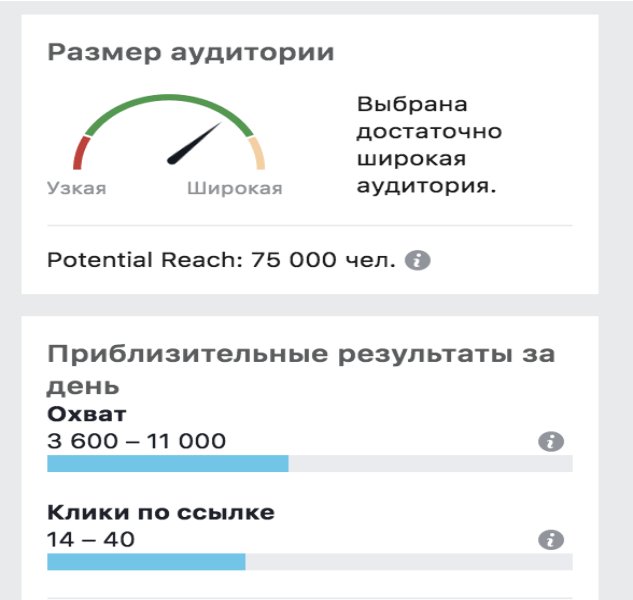


Рис. 14. Розмір аудиторії

Далі налаштували види плейсменту, згідно з проведеним спліт-тестом. Встановили бюджет рекламної кампанії (рис. 15).

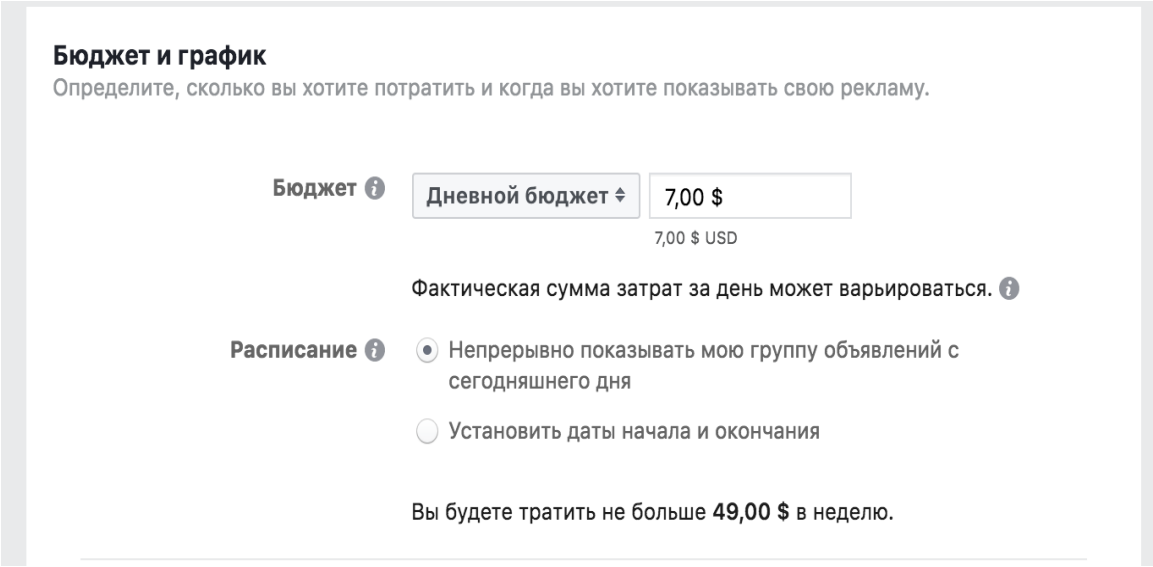


Рис. 15. Бюджет рекламної кампанії

Виконавши всі налаштування, була запущена рекламна кампанія інтернет-магазину «Футбоголік». Рекламна кампанія проходила один місяць (31 день). Виділений бюджет $220. Під час проходження рекламна кампанія редагувалась – раз в тиждень змінювалося головне рекламне зображення. Завдяки інструменту «Піксель» в особистому кабінеті бачимо кількість

конверсій на сайті. Статистичні дані після проведення рекламної кампанії представлені на рис. 16.

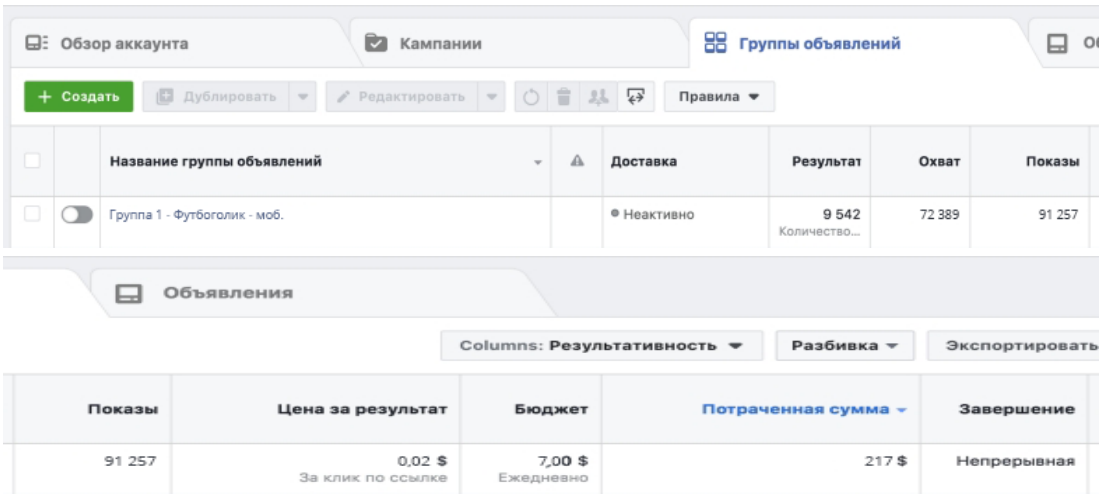


Рис. 16. Результативність групи оголошень

На рис. 17 відображено результати реклами окремих оголошень в особостому кабінеті.

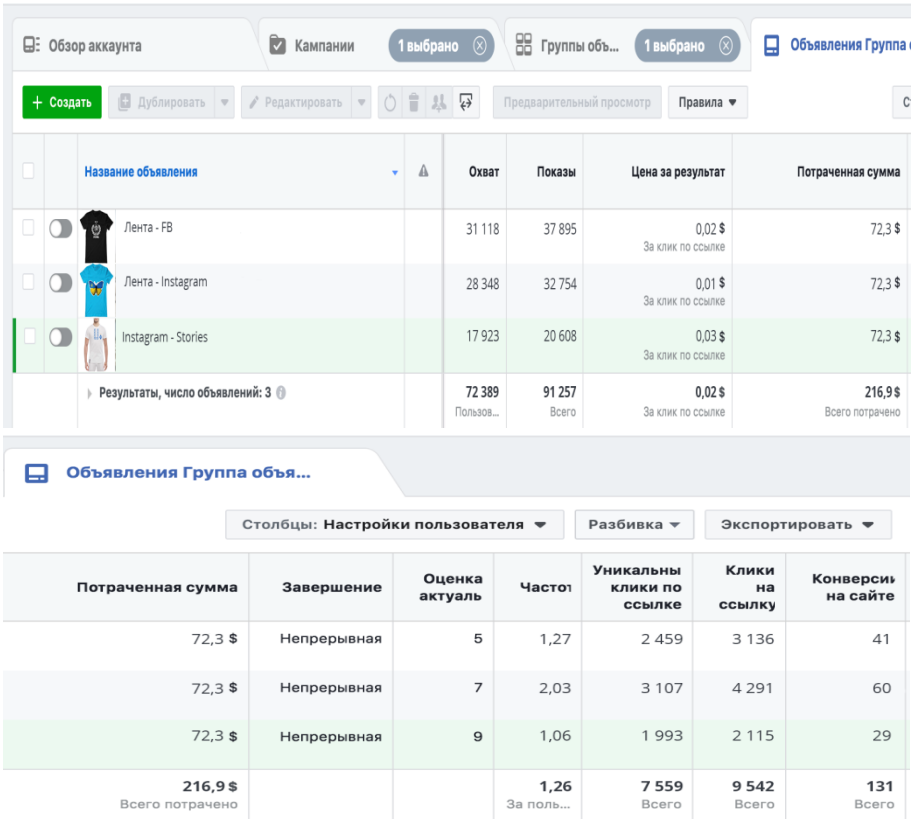


Рис. 17. Результативність окремих оголошень

Після проведення рекламної кампанії можемо зробити висновок, що найефективнішим виявився плейсмент стрічка Instadram. Дане рекламне оголошення зробило більше всього конверсій на сайті інтернет-магазину «Футбоголік» – 60 шт. Рекламна кампанія охопила майже всю орієнтовану аудиторію, а саме 72 382 людини, оголошення показані 91 257 разів. Загальна кількість людей, які перейшли за посиланням – 9 542 людини, середня вартість кліку за посиланням – 0,02 грн. За період рекламної кампанії було зроблено 131 замовлення товару.

Для того, щоб оцінити показники успіху і досягнення мети роботи, слід скористатися системою ключових показників ефективності інтернет-реклами.

Мета трафіку, що надходить на ресурс, зводиться до монетизації. Щоб довести якість трафіку, який надходить на ресурс, був разрахован показник конверсії сайту (1). Він являє собою відношення кількості замовлень до кількості відвідувачів, помножене на 100 %:

$$\text{Конверсія сайту} = \frac{131}{9542} \times 100\ \% = 1{,}4\ \%. \qquad (1)$$

Показник конверсії на сайті допоможе встановити, наскільки ефективно ресурс або канал реклами конвертує відвідувачів в клієнтів.

Розрахуємо витрати на залучення одного відвідувача. Для цього знайдемо співвідношення вартості витрат на рекламну кампанію на кількість відвідувачів сайту.

Показник вартості залучення одного покупця становить (2):

$$\text{CPO} = \frac{23090}{131} = 176\ \text{грн}. \qquad (2)$$

Дані показники повинні прагнути вниз.

За період запущеної рекламної кампанії було зроблено 131 замовлення загальний дохід якої склав 56 985 грн.

Важливим критерієм є повернення інвестицій, вкладених в рекламний канал (ROI). Розрахуємо його як різницю прибутку рекламної кампанії та витрат на рекламну кампанію по відношенню витрат на рекламну кампанію (3):

$$ROI = \frac{56985-23090}{23090} \times 100\ \% = 146\ \%. \qquad (3)$$

Аналізуючи ROI рекламної кампанії, можна зробити підсумковий висновок, що інвестиції в інтернет-рекламу окупаються. Продовження впровадження рекламної кампанії на підставі сформованої методики збільшить коефіцієнт окупаємості інвестицій. Створена рекламна компанія є ефективна, показники оцінки ефективності доводять, що впроваджена методика підвищення ефективності рекламної кампанії є результативною.

На рис. 18 зображена динаміка кількості замовлень у період проведення нової рекламної кампанії в порівняні до рекламної кампанії, яка була до впро-вадження методики.

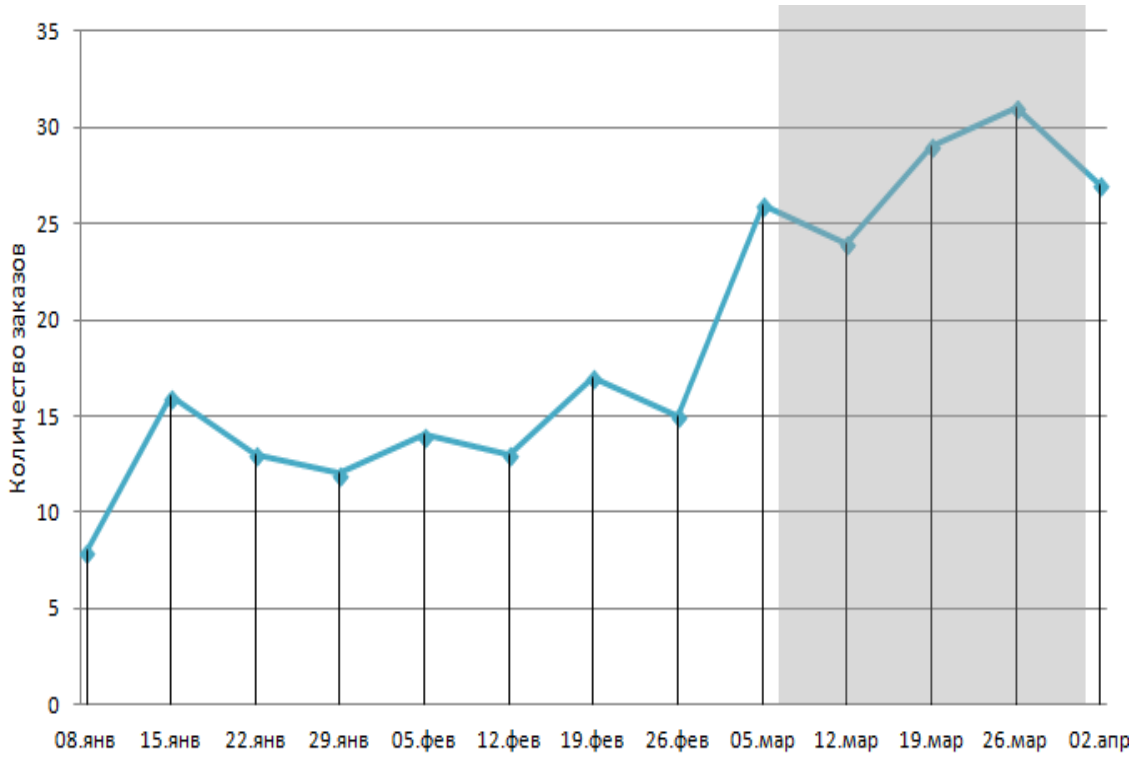


Рис. 18. Динаміка збільшення конверсії інтернет-магазину після застосування рекламної кампанії

На графіку показано, що за березень кількість конверсій збільшилось майже на 50 % проти січня і лютого.

Отже, інтернет-магазин одягу «Футболік» після застосування рекламної кампанії збільшив чисельність відвідувачів свого сайту, які виконали на ньому якісь цільові дії.

Висновки

Таким чином, вибір типу інтернет-реклами здійснюється з урахуванням специфіки підприємства (компанії, фірми), його цілей і завдань, а також переваг і недоліків кожного окремого засобу.

Існує кілька основних видів реклами в інтернеті, які відрізняються між собою схемою взаємодії сторін, складністю налаштування і ведення кампанії, плейсментом та іншими особливостями.

Розуміння їхньої суті, переваг і недоліків дозволить уникнути ризику. Розглянемо основні види реклами в інтернеті та їхні особливості.

Практично всі соціальні мережі надають достатній функціонал для інтернет-магазинів: конверсійні кнопки, картки товарів в фотоальбомах, створення вікі-розмітки, відеоогляди, призначеного для користувача контенту у вигляді відгуків про товари, роботи кур'єрів і т.ін. Отже, соціальні мережі є найбільш ефективним на сьогодні засобом рекламної компанії інтернет-магазинів.

Література

1. Aralova N. “The method of technology evaluation based on improved cost approach”, *Science and Innovation*, № 13(3), p. 65–76, 2017.

2. Martins P., Zacarias M. “A Web-based Tool for Business Process Improvement”, *International Journal of Web Portals*, № 9(1), p. 68–84, 2017.

3. Simola J., Kuisma J. “Perception of visual advertising in different media: from attention to distraction, persuasion, preference and memory” Lausanne : Frontiers Media SA, 2015. — 124 p.

4. Urbas R., Stankovič U. “Color differences and perceptive properties of prints made with microcapsules”, *Journal of graphic engineering and design*. — 2015. — № 1. — P. 15–21.

ГЛАВА 7
СУЧАСНІ ТЕХНОЛОГІЇ ЗБЕРІГАННЯ, РОЗПОВСЮДЖЕННЯ ТА ОПРАЦЮВАННЯ МУЛЬТИМЕДІЙНОГО КОНТЕНТУ СИСТЕМ ЕЛЕКТРОННОГО НАВЧАННЯ

Вступ і постановка задачі

Використання мультимедійних (тобто що містять інтегровану динамічну і статичну, графічну, текстову і звукову інформацію), компонент дидактичних комплексів (LMS Learning Management Systems) стають все більш і більш затребуваною формою представлення інформації освітнього характеру, поступово витіснивши традиційні текстові з незначною кількістю статичних ілюстрацій представлення. Стрімкий розвиток і масштабність застосування у сфері освітніх послуг мультимедійних компонент пояснюється, в першу чергу, наступними численними перевагами використання мультимедіа як інструмент.

Емоційна привабливість. Мультимедійні презентації дають можливість представити інформацію не лише в зручній для сприйняття послідовності, але і ефектно поєднувати звукові і візуальні образи, підбирати домінуючі кольори і колірні поєднання, які створять у глядачів позитивне відношення до інформації, що представляється.

Наочність. Наочність – це ключовий аргумент використання мультимедійних компонент. Відмітні особливості, специфічні властивості і переваги освітнього контенту можна гранично повно представити саме за допомогою сучасних мультимедійних технологій.

Інтерактивність. Можливість безпосередньо впливати на хід роботи з освітнім контентом - це одна з найважливіших переваг мультимедіа. Описати нелінійний порядок взаємодії з частинами контенту, що можливо містить численні повернення до блоків інформації (індивідуалізованої траєкторії формування компетентності) - незаперечне достоїнство мультимедійного представлення, яке дає можливість фокусувати увагу учня на необхідних ключових або не в повній мірі освоєних моментах.

Особливого значення мультимедійні компоненти освітньої інформації набувають зараз, у зв'язку з вибуховим зростанням продажів мобільних пристроїв, планшетів і смартфонів.

Метою даного розділу є аналіз сучасних технологій зберігання, розповсюдження та опрацювання мультимедійного контенту систем електронного навчання.

Основна частина

Є три вимоги до Web-scale додатків.

Багато даних: найбільші з Web-серверних додатків обробляють об'єми даних на порядки більше тих, що передбачалися для управління базами даних (Facebook – 50 терабайт для пошуку по вхідних повідомленнях; eBay – 2 петабайта).

Величезна кількість користувачів: обчислюються мільйонами, доступ до систем одночасно і постійно.

Складні дані: як правило, це не просто додатки обробки табличних даних, які можна знайти в багато комерційних і бізнес-додатках.

Технології реляційних баз даних, які домінували в IT-індустрії з 1980 року, почали показувати свої слабкі риси при переході до Web-серверних масштабів саме в цих трьох аспектах, тому все більше число людей почали шукати альтернативу. Так стали з'являтися NOSQL бази даних. Хоча різноманітність моделей зростає, усі вони мають загальні риси:

обробляють величезні об'єми даних, розділяючи їх між серверами;

підтримують величезну кількість користувачів;

використовують спрощену, гнучкішу, не обмежену схемою структуру бази даних.

Як одне з методологічних обґрунтувань підходу NoSQL використовується евристичний принцип, відомий як теорема CAP, що стверджує, що в розподіленій системі неможливо одночасно забезпечити узгодженість даних, доступність (англ. availability, в сенсі наявності відгуку по будь-якому запиту) і стійкість до розщеплення розподіленої системи на ізольовані частини. Таким чином, при необхідності досягнення високої доступності і стійкості до розподілу передбачається не фокусуватися на засобах забезпечення узгодженості даних, забезпечуваних традиційними SQL -орієнтованими СУБД з транзакційними механізмами на принципах ACID.

У базах NoSQL реалізується відмінна від традиційної (один сервер баз даних для декількох додатків) парадигма, при якій одному застосуванню або модулю застосування пропонується окреме рішення по роботі з даними. За своєю суттю архітектура рішень NoSQL орієнтована на боротьбу або з великим об'ємом даних, або з їх підвищеною складністю.

Відмінні риси NoSQL - нереляційні моделі даних, прості API або протоколи доступу, здатність до горизонтального масштабування на вимогу для деякого набору операцій на багатьох серверах, розподілене зберігання

даних, ефективне використання розподілених індексів і пам'яті для запитів, досить вільне поводження з такими непорушними для традиційних СУБД речами, як транзакційна цілісність. Загальним для NoSQL - проектів являються компроміси по відношенню до взаємно суперечливих вимог - наприклад, відхід від підтримки стандартних правил для забезпечення транзакційної цілісності ACID (atomicity, consistency, isolation, durability, атомарність, узгодженість, ізольованість, довговічність) на користь горизонтальної масштабованості. Відмова від подібних вимог була неможлива і є догмою для традиційних СУБД. Дійсно, не можна бути одночасно надійним, швидким, розподіленим і цілісним, проте у ряді конкретних випадків можливі варіанти.

Інше джерело компромісів - характер даних вирішуваної задачі. Саме тут важлива вимога до технології, яка повинна максимально використовувати особливості предметної області. Наприклад, якщо можна розпаралелювати обробку даних і використовувати принцип shared nothing ("без розподілу ресурсів"), то треба це ефективно застосовувати як для зберігання, так і для виконання запитів. В цьому випадку потрібно будувати модель зберігання і розподілу даних, яка спирається на цю можливість, проте в реляційних базах свободи вибору майже немає і доводиться використовувати те, що є, - скажімо, не можна зберігати дані по колонках на різних серверах

Усі найбільш успішні і відомі NOSQL бази даних були розроблені з нуля за останні декілька років. Але є вже існуючі, успішно реалізовані технології баз даних, які могли б забезпечити надійну основу для відповідності вимогам Web-scale. Це такі продукти – GT.M і Inter-systems Cache, чиї системи зберігання засновані на глобалах.

Глобал – це абстракція B-tree структури, яка зазвичай використовується в мові MUMPS для зберігання великих об'ємів даних.

Глобали спочатку були розроблені в 1966 році для підтримки управління великими об'ємами складних і слабоструктурованих медичних даних, і були спеціально спроектовані для ефективної підтримки великої кількості конкурентних інтерактивних користувачів на серйозно обмежених ресурсах PDP міні-комп'ютерів. Це не досягає Web-серверного масштабу, проте, глобали стали досягненням, якого інші технології баз даних у той час досягти не могли.

Основні характеристики глобалів наступні:

вони не обмежені схемою;

ієрархічно структуровані;

розріджені;

динамічні.

Однією з сучасних реалізацій цієї архітектури є так звані колонкові СКБД.

Основна ідея колонкових СКБД — це зберігання даних не по рядках, як це роблять традиційні СКБД, а по колонках. Це означає, що з точки зору SQL-клієнт дані представлені як завжди у вигляді таблиць, але фізично ці таблиці є сукупністю колонок, кожна з яких по суті є таблицею з одного поля. При цьому фізично на диску значення одного поля зберігаються послідовно один за одним.

Така організація даних наводить до того, що при виконанні select в якому фігурують лише 3 поля з 50 полів таблиці, з диска фізично будуть прочитані лише 3 колонки. Це означає що навантаження на канал введення-виводу буде приблизне в 50 / 3 = 17 разів менше ніж при використанні традиційного підходу.

Найбільш відомою з колонкових СКБД це BigTable — високопродуктивна пропрієтарная база даних, що підтримує стискування, побудована на основі Google File System (GFS), Chubby Lock Service і деяких інших продуктах Google. Зараз BigTable використовується в різного роду додатках Google, таких як MapReduce, яке часто використовується для створення і модифікації даних тих, що зберігаються в BigTable, Google Reader, Google Maps, Google Book Search, Search_History, Google Earth, Blogger.com, Google Code hosting, Orkut і YouTube.

Сховище Windows Azure storage забезпечує розробникам можливість зберігання даних у хмарі. Додаток може виконувати доступ до своїх даних у будь-який момент часу з будь-якої точки планети, зберігати будь-який об'єм даних і як завгодно довго. При цьому дані гарантовано не будуть пошкоджені і загублені. WindowsAzureStorage пропонує багатий набір абстракцій даних:

WindowsAzureBlob – забезпечує сховище великих елементів даних.

WindowsAzureTable – забезпечує структуроване сховище станів сервісу.

WindowsAzureQueue – забезпечує диспетчеризацію асинхронних завдань для реалізації обміну даними між сервісами.

Найбільш часто використовувані СУБД мультимедійного контенту використовують так звану key - value архітектуру. Це максимально спрощена база даних. Усі дані зведені до: ключ або індекс, і самі дані, які, зазвичай представлені у вигляді простого рядка, в деяких випадках можуть бути ще

числа. Тобто, усі дані це просто список ключів і зіставлених з ними рядків даних. Дані яких типів існують в базі відомо тільки застосуванню. Інтерфейс доступу до такої бази є простими командами get (отримати дані по ключу), set (записати дані з ключем), delete (вилучає ключ і його дані), update (оновлює вже існуючі дані).

Найголовнішою перевагою є те, що складність таких операцій (тобто, час обчислення результату) буде заздалегідь відомою і не залежить від об'єму даних або кількості серверів. Також усі операції зазвичай атомарні (у звичайних SQL базах даних аналог транзакції) - задаючи команду можна бути упевненим, що вона або успішно відпрацює або відразу поверне помилку, а інші дані залишаться незайманими, при цьому інші користувачі не перешкодять, навіть якщо намагатимуться зробити те ж саме.

У відкритої системи управління базами даних MySQL 5.6 реалізований механізм прискореного доступу до даних за принципом технології NoSQL програмний інтерфейс API, який дозволить стороннім застосуванням безпосередньо звертатися до даних з ядра обробки даних InnoDB, а не через SQL-інтерфейс. Новий API повторює функції відкритої технології кешування Memcache, яку великі сайти ніби Facebook використовують для прискорення доступу до великих об'ємів даних. Цей новий вбудований програмний інтерфейс ставить MySQL в один ряд з базами даних NoSQL, які стають все популярніше в web-додатках.

Ще одна нова функція, яка робить MySQL привабливішою платформою для NoSQL-систем, реалізована в новому наборі операцій ADD, які дозволяють застосуванням записувати дані в БД без блокування інших операцій з доступом до індексу БД. Ці функції будуть особливо корисні для швидкорослих web - сервісів.

Частиною стандартів розвинутих під година реалізації HTML 5.0 є IndexedDB - стандарт зберігання великих об'ємів структурованих даних на клієнтові. IndexedDB служить для зберігання великих об'ємів структурованих даних, з можливістю індексації. Потреба в такому інструментарії назріла давно, що і привело до його появи в специфікаціях HTML5.

Об'єкт JavaScript аналогічний запису в реляційній базі даних, то властивості цього об'єкту можна вважати стовпцями (чи полями) в таблиці. В результаті інтерфейс IndexedDB дозволяє визначити індекси, що ідентифікують ті властивості об'єктів, які можна використовувати для пошуку записів в сховищі об'єктів. Таким чином, індекси дозволяють перебирати і шукати дані IndexedDB за допомогою значень атрибутів на

об'єкті JavaScript. Дані знаходяться в сховищах об'єктів, що є колекціями об'єктів JavaScript, атрибути яких містять індивідуальні значення.

API IndexedDB складається з декількох об'єктів, кожен з яких призначений для певних завдань.

Усі об'єкти JavaScript в сховищі, мають загальний атрибут - key path. Значення цього атрибуту називається значенням ключа (чи просто ключем). Значення ключів служать єдиними ідентифікаторами записів в сховищі об'єктів.

У покажчику об'єкти впорядковані за значенням загального атрибуту. Покажчики повертають набір значень ключів, які можна використовувати для отримання окремих записів з початкового сховища об'єктів.

Курсор представляє набір значень. Курсор, визначений покажчиком, представляє набір повернених цим покажчиком значень ключів. Курсор, визначений сховищем об'єкту, представляє набір збережених в нім записів.

Атрибут keyRange визначає діапазон значень покажчика або набору записів в сховищі об'єктів. Діапазони ключів дозволяють фільтрувати результати курсору.

Бази даних містять сховища об'єктів і покажчики, а також управляють транзакціями.

Запит представляє окремі дії з об'єктами в базі даних. Відкриття бази даних веде до об'єкту запиту. Щоб відреагувати на результати запиту, треба визначити обробники подій об'єкту цього запиту.

У стандарті і більшості реалізацій реляційних баз даних існує спеціальній тип BLOB (англ. Binary Large OBject - двійковий великий об'єкт) - масив двійкових даних. У СУБД BLOB - спеціальний тип даних, призначений, в першу чергу, для зберігання зображень, аудіо і відео, а також програмного коду, що компілює.

На клієнтському рівні найчастіше використається як сервер SQLite.

SQLite - легковага вбудовувана реляційна база даних. Початковий код бібліотеки переданий в громадське надбання.

Слово "вбудовуваний" означає, що SQLite не використовує парадигму клієнт-сервер, тобто движок SQLite не є окремо працюючим процесом, з яким взаємодіє програма, а надає бібліотеку, з якою програма компонується і движок стає складовою частиною програми. Таким чином, як протокол обміну використовуються виклики функцій (API) бібліотеки SQLite. Такий підхід зменшує накладні витрати, час відгуку і спрощує програму. SQLite зберігає усю базу даних (включаючи визначення, таблиці, індекси і дані) в

єдиному стандартному файлі на тому комп'ютері, на якому виконується програма. Простота реалізації досягається за рахунок того, що перед початком виконання транзакції запису увесь файл, що зберігає базу даних, блокується; ACID – функції досягаються у тому числі за рахунок створення файлу журналу.

Декілька процесів або потоків можуть одночасно без яких-небудь проблем читати дані з однієї бази. Запис в базу можна здійснити тільки у тому випадку, якщо ніяких інших запитів в даний момент не обслуговуються; інакше спроба запису закінчується невдачею, і в програму повертається код помилки. Іншим варіантом розвитку подій є автоматичне повторення спроб запису протягом заданого інтервалу часу.

Web SQL Database - це нова технологія, що є частиною HTML5. Суть полягає в тому, що браузер, підтримувальний HTML5 повинен містити локальну базу даних, робота з якою відбувається використовуючи javascript. Маніпуляції з даними відбуваються використовуючи SQL синтаксис.

Web sql database найчастіш використається для двох типів завдань : створення офлайн застосування і прискорення роботи з великими об'ємами інформації при повільному і нестабільному з'єднанні. Web Sql Database працює асинхронно. Тобто виконати ланцюжок запитів непросто.

На серверному рівні веб додатку найчастіше використається СКБД MySQL - система управління базами даних (СУБД) з відкритим кодом. MySQL є власністю компанії Oracle Corporation, що отримала її разом з поглиненою Sun Microsystems, що здійснює розробку і підтримку застосування. Поширюється під GNU General Public License або під власною комерційною ліцензією. Окрім цього розробники створюють функціональність за замовленням ліцензійних користувачів, саме завдяки такому замовленню майже в самих ранніх версіях з'явився механізм реплікації.

MySQL є рішенням для малих і середніх додатків. Входить до складу серверів WAMP, AppServ, LAMP і в портативні складки серверів Денвер, XAMPP. Зазвичай MySQL використовується як сервер, до якого звертаються локальні або видалені клієнти, проте в дистрибутив входить бібліотека внутрішнього сервера, що дозволяє включати MySQL в автономні програми.

Гнучкість СУБД MySQL забезпечується підтримкою великої кількості типів таблиць : користувачі можуть вибрати як таблиці типу MyISAM, що підтримують повнотекстовий пошук, так і таблиці InnoDB, підтримувальні транзакції на рівні окремих записів. Більше того, СУБД MySQL

поставляється із спеціальним типом таблиць EXAMPLE, що демонструє принципи створення нових типів таблиць. Завдяки відкритій архітектурі і GPL - ліцензуванню, СУБД MySQL постійно удосконалюється і обростає новим функціоналом.

Незалежні типи таблиць (MyISAM для швидкого читання, InnoDB для транзакцій і посилальної цілісності);

Maria (починаючи з версії 5.2.x - Aria) - новий тип таблиць MySQL для зберігання даних. Maria є розширеною версією сховища MyISAM, з додаванням засобів збереження цілісності даних після краху.

У разі краху проводиться відкат результатів виконання поточної операції або повернення в стан до команди LOCK TABLES. Можливість відновлення стану з будь-якої точки, включаючи підтримку CREATE/DROP/RENAME/ TRUNCATE. Може бути використано для створення інкрементальних бекапів, через періодичне копіювання.

Підтримка усіх форматів стовпців MyISAM, розширена новим форматом "rows - in - block", що використовує сторінковий спосіб зберігання даних, при якому дані в стовпцях можуть кешуватися.

В майбутньому буде реалізовано два режими: транзакційний і без віддзеркалення транзакцій, для некритичних даних.

Розмір сторінки даних рівний 8Кб (у MyISAM 1Кб), що дозволяє досягти вищої продуктивності для індексів по полях фіксованого розміру, але повільніше у разі індексування ключів змінної довжини.

Засоби обробки мультимедійного контенту

Архітектура систем обробки мультимедійної інформації в значной мірі визначається вибором мови обробки і бібліотек класів - фреймворків. Коротко охарактеризуємо найбільш популярні складові.

Згідно рейтингу TIOBE Software BV - голландської компанії, яка відома своїми щомісячними оцінками популярності мов програмування, мови програмування у 2019 році розташовані таким чином.

1. Java. Java залишається домінуючою платформою для розробки додатків. Кількість робочих місць для Java - програмістів за рік збільшилося приблизно на 50%. Після переходу під керівництво компанії Oracle, мова продовжує розвиватися. Були представлені дві нові специфікації, які будуть реалізовані в найближчі декілька років. Використання було значно підштовхнуто популярністю планшетів та смартфонів на платформі Андроид.

2. C - одна з найбільш популярних мов за багато десятиріч, яка використовується для системного програмування, а також для розробки

додатків для систем, що вбудовано. Незважаючи на свій поважний вік, вона як і раніше затребувана, хоча кількість вакансій за рік знизилася на 11.

3. C++ - це розширена версія C, що надає програмістові доступ до класів. Ця мова швидко стала одним з найпопулярніших і залишається таким до цього дня. C++ використовується для розробки системного ПО, додатків, драйверів, програм для що вбудовано, високопродуктивних серверних і клієнтських додатків, відеоігор і багато чого іншого. Втім, кількість вакансій для програмістів зменшилася на 13%.

4. C#. C# був розроблений компанією Microsoft як альтернатива Java, і включає виразні засоби Java, C, C++ і Delphi. Кількість вільних місць для програмістів виросла за півтори роки приблизно на 50%.

5. JavaScript. Мова широко використовується в побудові сайтів для виконання скриптів на стороні клієнта в браузері. Інтернет стає усе більш мультимедійним, що сприяє зростанню популярності цієї мови. Втім, він використовується і за межами веба - в PDF - документах, віджетах і навіть для розробки розширень для великих додатків (наприклад, в Adobe Illustrator). Кількість вакансій виросла приблизно на 75%.

6. Perl. Perl - мова висого рівня загального призначення, що інтерпретується, розроблена в 1987 році для Unix -систем.

7. PHP. PHP використовується при розробці сайтів на серверному боці, для чого її і було створено. Нових програмістів вимагається майже на 60% більше.

8. Visual Basic. VB третього покоління заснований на подіях і вбудований в середу розробки Microsoft Visual Studio. Він замислювався як відносно простий в навчанні і використанні мова, що і привело його до популярності серед програмістів. Досить багато "серйозних людей" плюються, коли чують його назву, але працедавці думають інакше - зростання вакансій склало більше 110%!

9. Python. Динамічна мова, використовувана в застосуваннях різного профілю. Він дозволяє швидко писати код, а програмісти, як відомо, люди ледачі. Число вакансій виросло майже на 70%.

10. Ruby. Ruby фокусується на спрощенні розробки і підвищенні продуктивності. Він має елегантний синтаксис, близький до природної мови. CMS Ruby on Rails набирає популярність серед веб - розробників. За рік кількість місць для програмістів збільшилася майже на 80%.

11. Objective C, в основному використовуваному для створення додатків для систем від Apple. Популярність зросла на 70 % за рахунок успіху пристроїв iPhone і iPad.

Скриптові мови швидко стають мовами загальної реалізації для багатьох областей, саметам, де час розробника важливіший, ніж час виконання (і навіть там, де важливий час виконання; наприклад, завдяки вбудованим операціям високого рівня швидкодія програм, написаних на Python, таке ж, або навіть швидше, ніж програм, написаних на Java).

Скриптові мови дозволяють розробникам зчіплювати разом різні пакети програм, а також погоджувати отримані в результаті системи.

Все частіше скриптові мови самі по собі використовуються як повноцінні базові інструментальні платформи. Наприклад, багато великих комерційних Інтернет-додатків зараз програмуються переважно на мовах Perl, Python або PHP.

Природно, скриптові мови використовуються для автоматизації завдань системного адміністрування.

Мови, що інтерпретуються або оперативно компільовані, все більше і більше заступатимуть на зміну заздалегідь компільованим мовам. Компіляцію з часом розглядатимуть просто як інструмент оптимізації (чим вона власне і являється), використання якого в усіх випадках навряд чи розумно. Межа між компіляцією і інтерпретацією, яка завжди була трохи довільна, буде розмита ще більше. У мови Perl вже є фаза оперативної компіляції перед інтерпретацією. Майбутнє буде за сумісністю платформ, так що компілятори все частіше будуть націлені на абстрактні "віртуальні машини" (як JVM у Sun або CLR у Microsoft), які нашаровуються на апаратні засоби.

Скриптові мови мають складніший інструментарій і підтримують прогресивнішу техніку програмування. Наприклад, можливості сортування даних в Perl вбудовані прямо в мову. Те, що в мову вбудовані усі основні інструменти програмування, позбавляє від необхідності створювати їх самостійно і означає, що для вирішення конкретної проблеми треба писати менше коду, що збільшує продуктивність розробника.

Кількість людей, що не мають підготовки, яку мають традиційні комп'ютерні фахівці, але що можуть зайнятися написанням скриптів, стала на порядок більше. Інакше кажучи, програмуванню на скриптових мовах простіше навчитися. Щоб стати середнім програмістом на C++, потрібний

більший досвід роботи, чим для того, щоб стати середнім програмістом на PHP.

Розглянемо недоліки скриптових мов. Є області, де швидкість занадто важлива, щоб можна було програмувати безпосередньо на скриптовій мові. Ця проблема зазвичай вирішується тим, що код ретельно вибраної частини застосування (скажімо, 10-30%) пишеться мовою низького рівня (такому, як C або C++); наприклад, в Python є розвинені механізми для того, щоб вставити такий код (як і в більшості інших динамічних мов).

Загальною проблемою усіх скриптових мов є відсутність хорошого інтегрованого середовища розробки (IDE). Звичайно, якісь інтегровані середовища розробки існують, проте в них бракує потужності, як у Visual Studio.

Ключовим нетехнічним, проте важливим недоліком є відсутність маркетингового бюджету. Багато динамічних мов ідеально підходять для багатьох проектів, проте їм важко конкурувати з такими локомотивами маркетингу, як Sun (Java) і Microsoft (C#), які продовжують просувати свої технології як єдино можливі. У історії є приклади того, як технічна перевага пригнічується чудовим маркетингом.

В більшості випадків при згадці JavaScript мається на увазі так званий клієнтський JavaScript, інтерпретатор якого вбудований в Web -браузери. Проте JavaScript спочатку був розроблений як універсальна мова програмування для вбудовування в будь-яке застосування і забезпечення можливості написання в нім сценаріїв. Наприклад, ActionScript, мова сценаріїв, доступна в Macromedia Flash 5 і MX, також змодельована відповідно до стандарту ECMAScript. На серверному боці найчастіше використовується скриптова мова програмування PHP.

Tcl (Tool Command Language) - скриптова мова програмування, достатньо високого рівня.. Tcl орієнтований переважно на автоматизацію рутинних процесів ОС і великих програмних систем і складається з потужних команд, орієнтованих на роботу з абстрактними об'єктами, що не типізуються. Принципова відмінність Tcl від командних мов ОС полягає в незалежності від типу системи (коли не потрібно утрудняти себе вивченням нової командної мови) і, найголовніше, він дозволяє створювати мобільні програми з графічним інтерфейсом (GUI).

Tcl дуже часто застосовується спільно з бібліотекою Tk (Tool Kit).. Tcl/Tk поширюється в початкових текстах безкоштовно. Tcl/Tk розроблявся одночасно як мова і бібліотека. Tk - це популярний графічний інструментарій, що дозволяє

дуже швидко створювати графічні програми. Варіанти Tcl/Tk доступні для безлічі платформ (Windows, Macintosh, практично усе UNIX - платформи, включаючи Linux).. Бібліотека Tk містить стандартизований набір команд підтримки GUI. Елементи, що керують, зберігаються в Tk, називаються виджетами (widgets). Велику кількість нетипових віджетів можна знайти в мережі.

Tcl - розширювана мова. Можна самостійно визначати нові команди мови. На Tcl написана оболонка Visual Tcl, яка дозволяє розробляти кросплатформене ПО для UNIX, Windows і Macintosh. Фірмою Sun розроблена версія Tcl, написана на Java, - Jacl (JAva Command Language).

PHP. PHP (пи-эйч-пи) - скриптова мова програмування, що інтерпретується, створена для генерації HTML - сторінок на веб - сервері і роботи з базами даних. У області веб - програмування PHP являється на сьогодні однією з найпоширеніших технологій (разом з Perl, ASP/.NET і Python) завдяки простоті, швидкості виконання і багатої функціональності. PHP поширюється вільно. Синтаксис мови схожий на синтаксис C++. PHP підтримується переважною більшістю провайдерів мережевого хостінгу.

За своє життя PHP значно змінювався. Однією з найсильніших сторін PHP є можливість розширення ядра. Інтерфейс написання розширень притягнув до PHP безліч сторонніх розробників, що працюють над своїми модулями, що дало PHP можливість працювати з величезною кількістю баз даних, протоколів, підтримувати велике число API. PHP підтримує прототипне програмування (деструкції, відкриті, закриті і захищені члени і методи, final - члени і методи, інтерфейси і клонування об'єктів) та засоби розбору XML.

Значна частина функціональності сучасних інтерактивні сайти забезпечується бібліотеками - розширення скриптових мов (фреймворків), розглянемо основні фреймворки, на базі яких будується динаміка сучасних сайтів.

Коли JavaScript тільки з'явився, він використовувався переважно для створення досить простих ефектів, таких як прокручування тексту в статусбаре браузеру або створення широко відомого ефекту rollover для посилань. Останніми роками ситуація навкруги JavaScript істотно змінилася: з приходом ери Web 2.0 і технології Ajax ця скриптовый мова стала тим технічним базисом, на якому будується динамічна і інтерактивна частина більшості світових сайтів. Не дивлячись на те, що Javascript достатньо потужна мова, створення на нім складних сучасних додатків - непросте

завдання для більшості програмістів, наприклад, із-за вимоги обов'язкової сумісності з усією безліччю основних браузерів, що вимагає дуже скрупульозних знань особливостей кожного окремого браузеру.

JavaScript framework - це бібліотека класів або набір готових утиліт і функцій, які реалізують основну функціональність подібних інтерактивних сайтів, при цьому забезпечують крос - браузерную сумісність цих рішень на усіх рівнях роботи JavaScript. Дані фреймворки пишуться співтовариством розробників і поширюються, як правило, безкоштовно, тому будь-який бажаючий веб - розробник може підключити усі ці можливості до свого сайту вже в готовому виді, замість того щоб створювати і тестувати їх самостійно.

Існує величезна кількість різних платформ дистанційного навчання, як з відкритим кодом, так і пропрієтарних. Забезпечити передачу інформації між різними платформами можна лише при використанні стандартних форматів обміну. Найпоширенішим з них є SCORM.

Sharable Content Object Reference Model (SCORM) - збірка специфікацій і стандартів, розроблена для систем дистанційного навчання. Містить вимоги до організації учбового матеріалу і усієї системи дистанційного навчання. SCORM дозволяє забезпечити сумісність компонентів і можливість їх багатократного використання : учбовий матеріал представлений окремими невеликими блоками, які можуть включатися в різні учбові курси і використовуватися системою дистанційного навчання незалежно від того, ким, де і за допомогою яких засобів вони були створені. SCORM заснований на стандарті XML.

Найпопулярнішою LMS з відкритим кодом є Moodle. Moodle - система управління курсами (система управління вмістом) також відома як система управління навчанням або віртуальне повчальне середовище. Це відкритий (поширюється за ліцензією GNU GPL) веб - додаток, що надає можливість створювати сайти для онлайн- навчання, орієнтована передусім на організацію взаємодії між викладачем і учнями, хоча підходить і для організації традиційних дистанційних курсів, а також підтримки очного навчання. Moodle написана на PHP з використанням SQL - бази даних (MySQL, PostgreSQL, Microsoft SQL Server та ін. БД - використовується ADOdb XML). Moodle може працювати з об'єктами SCO і відповідає стандарту SCORM. Moodle зберігає мультимедійну інформацію у вигляді файлів, поширення і переглядання которіх виробляється зовнішніми по відношенню до Moodle засобами.

З систем пропрієтарних найбільш розвинені засоби розробки мультимедійної інформації має Adobe eLearning Suite 2. Для поширення електронного повчального контенту служить Adobe Connect.

Adobe eLearning Suite – це ккомплексний набір інструментів для професійної розробки та розповсюдження електронного повчального контенту - учбових і повчальних програм, моделювання, а також редагування медіаконтенту. Adobe eLearning Suite підтримує стандартні формати - SWF, HTML, PDF, AVI, і SCORM - що дозволяє легко поширювати контент в мережі Інтернет, на настільних комп'ютерах, мобільних пристроях і в системах управління процесом навчання (Learning Management System).

Adobe eLearning Suite включає наступні продукти.

Adobe Captivate.

Adobe Flash® Professional.

Adobe Deamweaver® .

Adobe Photoshop® .

Adobe Acrobat® Pro.

Adobe Presenter .

Adobe Soundbooth® .

Більшість з перерахованих додатків широко відома. Опишемо функції компонент, специфічних для завдань дистанційної освіти.

Adobe Presenter є доповненням для застосування Microsoft PowerPoint і дозволяє швидко розробляти вміст для електронного навчання. Після установки Adobe Presenter на стрічці Microsoft PowerPoint з'явиться додаткова закладка:

Використовуючи інструменти на закладці Adobe Presenter можна додати виразності звичайним слайдам PowerPoint, доповнивши їх Flash - контентом, аудіо, відео, запитальниками і тому подібне. Таким чином, Adobe Presenter можна використовувати для створення інтерактивних матеріалів для електронного навчання, а також для ефективніших презентацій різного роду. Наприклад, використовуючи кнопку Record (запис), можна швидко записати звук високої якості і потім синхронізувати цю аудіо доріжку і анімації в PowerPoint.

Використовуючи кнопку Insert SWF, можна доповнити презентації Presenter демонстрацією програмних продуктів або інтерактивними сімуляціями, створеними за допомогою Adobe Captivate.

Коли проект, створений за допомогою Adobe Presenter, закінчено - його можна опублікувати в будь-яку систему електронного навчання (Learning

Management System або LMS), адже Adobe Presenter підтримує формати AICC і SCORM. Також Adobe Presenter може публікувати готовий вміст безпосередньо в застосування Adobe Connect 8 (для онлайнового навчання на платформі Adobe), а також у файл PDF - при цьому зберігаються усі анімації.

Adobe Captivate - програма електронного навчання яка може бути використана для демонстрації програмного забезпечення, запису відео уроків, створення симуляції програми, створення учбових презентацій і різних тестів в .swf форматі. Можливо конвертувати .swf в .avi що згенеровано Adobe Captivate, для завантаження на сайти видео- хостінги. Для створення симуляцій програм, Captivate може використати праву і ліву кнопку миші і натиснення клавіш.

Adobe Captivate також можна використовувати для створення скрінкастов, подкастов, і конвертації презентацій Microsoft PowerPoint у формат Adobe Flash.

За допомогою Captivate можна створювати і редагувати інтерактивні демонстрації програм, симуляції, подкасти, скринкасты, ігри і уроки. Для демонстрацій програм, можливий запис в реальному часі. Створені за допомогою Captivate скрінкасты займають набагато менше місця, чим повноцінні записи з екрану.

Користувачі можуть редагувати Captivate презентації для додавання ефектів, активних точок, текстові області, відео і так далі. Автори можуть редагувати вміст і змінювати час появи того або іншого елементу. Натиснення на активні точки може переводити як на інші слайди, так і на зовнішні посилання.

Captivate підтримує імпорт зображень, презентацій PowerPoint, відео, .flv і аудіо в будь-який слайд проекту.

Adobe Connect. Програмне забезпечення Adobe Connect є гнучкою системою веб- комунікацій з високою мірою безпеки, яка пропонує рішення для проведення тренінгів, дистанційного навчання і веб - конференцій, дозволяє проводити віртуальні конференції з голосовою і відеозв'язком, а також можливістю показувати презентації і інші мультимедійні матеріали, проводити онлайн - зустрічі - заміна відряджень, при якій скорочуються як витрати, тимчасові і матеріальні, так і риски, пов'язані, наприклад, із затримкою або відміною рейсу.

Може використовуватися як для зустрічей у вузькому крузі, з невеликою кількістю учасників, так і для масштабних конференцій (до 2500 учасників). Учасники діляться на три категорії: ініціатор, що презентує і

слухачі. Ініціатор - спочатку запрошує користувачів і може передавати право виступу різним слухачам. Що презентує - може зробити видимим для слухачів увесь свій робочий екран або окреме застосування, а також транслювати голос. Слухачі - можуть ставити питання усім іншим або особисто, а також сигналізувати про стандартні прохання - поставити питання, говорити поголосніше і так далі.

Зображення і звук в синхронному виді передаються як струменеві відеофайли, які формуються на сервері Adobe Connect Server.

Висновки

Таким чином, розглянуті основні засоби керування мультимедійною інформацією забезпечують ефективне зберігання, розповсюдження та опрацювання мультимедійного контенту систем електронного навчання.

Відзначається, що не існує універсального вибору сукупності мультимедійних технологій кожного з етапів обробки контенту, що призводить до необхідності проводити вибір технологій для кожної системи e - learning , залежно від сукупності її характеристик, таких як масштабу і об'єму інтерактивного контенту.

Технологія передачі струменевого відео у веб - браузер користувача може бути розширена і використана для широкого круга завдань, пов'язаного з дистанційним навчанням (e - learning) і підготовкою навчальнвих матеріалів.

Література

1. H. Allende, “Artificial neural networks in time series forecasting: a comparative analysis”, *Kybernetik*a, № 6, p.685–707, 2002.

2. S. Robertson, “Placing teachers in global governance agendas”, *Comparative Education Review*, № 56, p. 584–607, 2012.

3. Unlock the future of smart eLearning design [Electronic resource]. – Access mode: https://www.adobe.com/products/captivate.html.

4. В. Лапінський, "Електронні засоби навчального призначення – світовий досвід й українська освіта", *Педагогіка вищої школи: методологія, теорія, технології,* № 3, с.487-495, 2013

ГЛАВА 8
СИСТЕМА ВІДДАЛЕНОГО МОНІТОРИНГУ ТА ПРОГНОЗУВАННЯ СТАНУ ЗДОРОВ'Я ПРАЦІВНИКА

Вступ та постановка задачі

Відвідування лікаря – основна складова забезпечення здоров'я людини, не тільки коли вона хворіє, занедужала, але і для регулярних медичних (диспансерних або профілактичних) оглядів. Однак, саме відвідування лікаря може бути тривалим, дорогим, а іноді і неприємним. Багато захворювань вимагають постійних відвідувань лікаря для контролю та спостереження показників стану зоров'я, наприклад, артеріального тиску, частоти серцебиття, та інш. На сьогодні, для звернення до певного лікаря людині, можливо, доведеться проїхати велику відстань, очікувати в черзі, і т.ін., в кінцевому підсумку це може зайняти більшу частину дня, а то й весь день.

При відвідуванні лікаря люди часто скаржаться на минулі стани або епізоди. Проте, лікар може тільки перевірити поточний стан пацієнта та поставити йому запитання, щоб згадати, що пацієнт відчував у минулому. Облікові записи пацієнтів можуть бути неінформативними та ненадійними. Пацієнти можуть не пам'ятати такі речі, як миттєвий пульс, кров'яний тиск, температура, тощо. Проблеми зі здоров'ям можуть виникати у пацієнтів, як внаслідок способу життя, так і внаслідок професійних або парапрофесійних (виробничо-обумовлених) захворювань. Гіпертонія, ішемічна хвороба серця, хвороби опорно-рухового апарата саме ці захворювання є прикладами виробничо-обумовлених захворювань. Факторами ризику їх розвитку є шкідливі професійні чинники, фізичні чи нервово-психічні перевантаження.

Розвиток спеціальних технологій виявлення та відстеження стану людини дозволяє реалізувати віддалений моніторинг стану працівників підприємств задля прийняття рішення для своєчасного реагування при виникненні погіршення стану працівника або інциденту чи екстреної ситуації. Основна увага приділяється розробці і вдосконаленню моделей і методів кількісного оцінювання та аналізу стану конкретного співробітника з точки зору безпеки його праці.

Однак, в даний час практично невирішеними залишаються питання, що пов'язані з прогнозом і попередженням виникнення нещасних випадків під час виконання виробничих процесів на підприємстві. Дана ситуація виникла через те, що більшість існуючих моделей і методів можна вважати елементами апостеріорного аналізу безпеки праці. Апостеріорний аналіз

виконується вже після того, як відбулися події, викликані впливом шкідливих факторів виробничого середовища і призвели до травми або погіршення здоров'я. У зв'язку з цим актуальним є розробка нових і вдосконалення вже існуючих моделей і методів, що дозволяють вирішувати завдання попередження нещасних випадків або усунення зниження продуктивності процесів пвдприємтсва внаслідок погіршення стану співробітників.

Для вирішення цієї проблеми пропонується система віддаленого моніторингу та прогнозування стану здоров'я працівника в процесі виробничої діяльності (СВМП), яка безперервно стежить за станом здоров'я людини в її повсякденному робочому середовищі.

Узагальнена модель системи віддаленого моніторингу та прогнозування стану здоров'я працівника в процесі виробничої діяльності

Дійовими особами пропонованої СВМП є працівник, для якого вирішується завдання кількісного контролю і аналізу стану як показника професійної безпеки праці і професійного здоров'я. Працівник повинен бути зареєстрований в комп'ютеризованій медичній установі для включення в систему, мати біодатчики, вимірювальні прилади та відеокамеру, які зчитують найбільш важливі показники здоров'я і з'єднуються з мобільним пристроєм, який доставляє отримані дані за допомогою бездротового зв'язку на основний сервер. На основі цих показників аналізуються:

– шкіра;
– дихальна система;
– серцево-судинна система;
– система органів травлення;
– сечовидільна система;
– опорно-рухова система;
– ендокринна система;
– нервова система і органи чуття.

Після реєстрації на Web-ресурсі підприємства працівник отримує унікальний ідентифікатор і на його мобільний пристрій встановлюється спеціальний додаток.

Трирівнева архітектура системи віддаленого моніторингу та прогнозування стану здоров'я працівника [1] наведена на рисунку 1.

На першому (вимірювальному) рівні системи розміщується апаратна частина (вимірювальні прилади, камери відеоспостереження, фотокамери та

т. п.). Тут показання приладів переводяться в форму, придатну для подальшої обробки і відправляються через Bluetooth до мобільного пристрою, яке пересилає отримані дані на Slave-сервер другого рівня, що підключений до мережі зв'язку та налаштований на те, щоб:

– приймати від пристрою мобільного зв'язку дані про стан здоров'я;

– зареєструвати медико-діагностичну інформацію;

–передати отриману інформацію на хмарний Master-сервер третього рівня.

На Master-сервері вирішуються наступні завдання:

– на основі отриманих даних визначається профіль здоров'я працівника, що включає одну або кілька прогнозованих проблем зі здоров'ям і ризики для здоров'я, в яких кожна або декілька прогнозованих проблем визначається як потенціальна проблема зі здоров'ям, з якої співробітник зіткнеться в майбутньому, і в якій прогнозована проблема зі здоров'ям включає прогнозоване фізичне пошкодження;

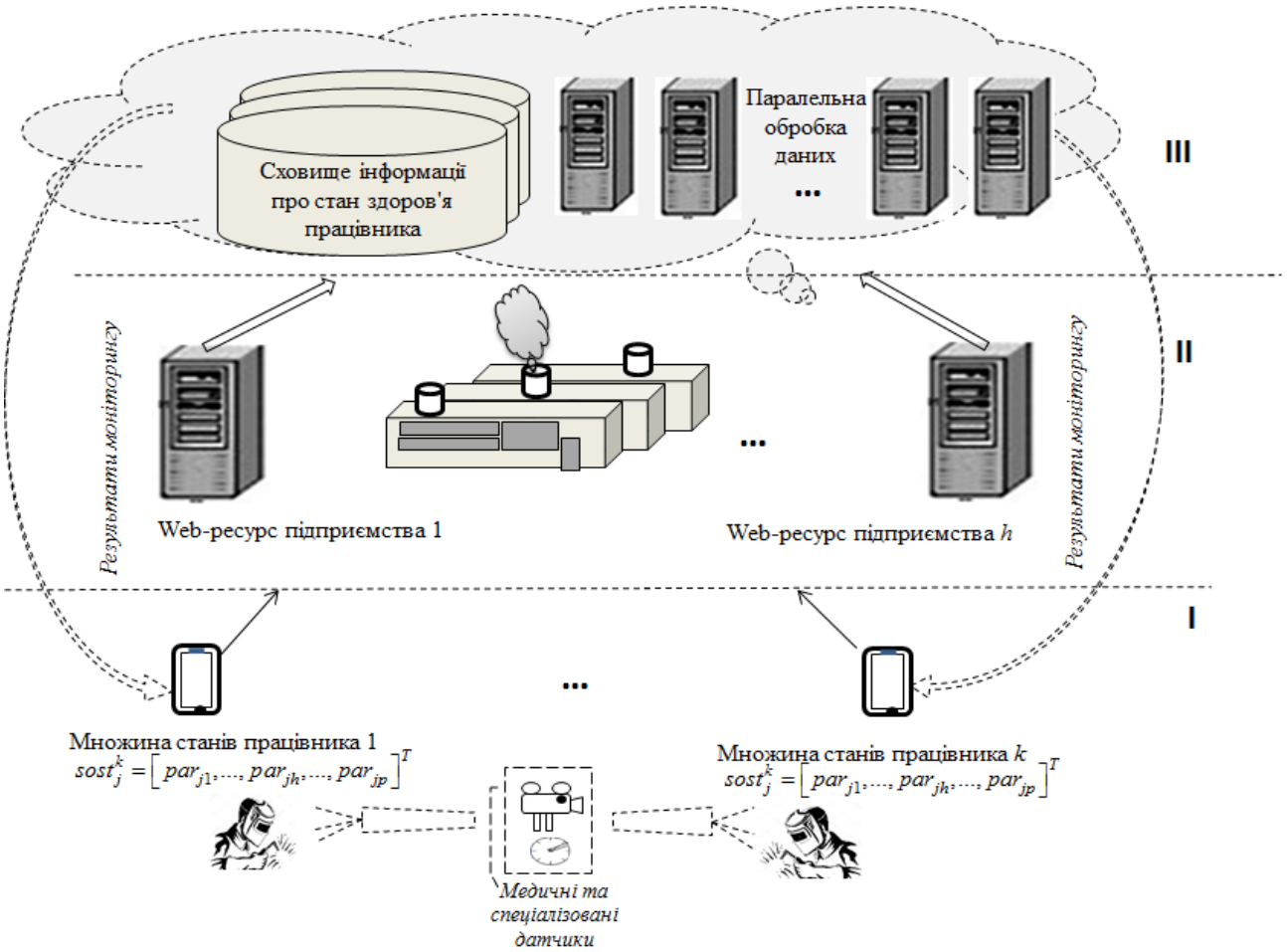


Рис. 1. Архітектура системи віддаленого моніторингу та прогнозування стану здоров'я працівника

– оновлюється інформація про стан здоров'я, що зберігається в базі даних для відображення профілю здоров'я працівника;

– на підставі відеозйомки порівнюється положення тіла (окремих частин тіла) співробітника з попередньо визначеною позицією для визначення шкідливого відхилення від попередньо визначеної позиції тіла;

– подавати в пристрій мобільного зв'язку для відображення співробітнику вміст звіту про стан його здоров'я, що містить, щонайменше, одну з характеристик, ризикі для здоров'я і профіль стану здоров'я, що вказує на одну або кілька передбачених проблем.

Модель СВМП Ξ виражається як перетворення вхідних значень H у вихідні величини Y [2]:

$$\Xi \subset H \times Y, \tag{1}$$

де $H(i) = \{H_D(i)\}$, $(i = \overline{1, M})$ – матричне подання вхідних даних:

– медико-діагностичні дані вимірювальних приладів,

– зображення, у якого значення яскравості елементів позначаються h_{kj} – $(h_{kj} = \overline{0, 255}, \quad k = \overline{1, m}, \quad j = \overline{1, n}, \ )$;

$Y = \{Y_D\}$, $(Y \subset \Upsilon)$ – множина формалізованих властивостей.

Таким чином, простір (універсум) $\Sigma = X \times \Upsilon$ включає $\Xi \subset (H \times Y)$, це означає, що існує така підмножина H, $(H \subset X)$ і відносин між ними, на яких будується модель Ξ $(\Xi \subset \Sigma)$.

Для вихідних величин Y_D побудовано множину завдань, вирішення яких належить множині $D_\Xi = \{ D_{Hr}, D_{Hst}, D_{Hdn}, D_{Hem} \}$,

де $D_{Hr} = \{Par_1, Par_2, \ldots, Par_n\}$ – задача обробки медико-діагностичних даних вимірювальних приладів (Par_1 – рівень систолічного артеріального тиску, Par_2 – рівень діастолічного артеріального тиску, Par_3 – частота серцевих скорочень, Par_4 – час реакції на світловий подразник, ...);

$D_{Hst} = \{M_1, M_2, \ldots, M_{st}\}$ – задача обробки зображень об'єктів в стані спокою (M_1 – виділення області інтересу, M_2 – бінаризація, M_3 – скелетонізація, …);

$D_{Hdn} = \{D_1, D_2, \ldots, D_{dn}\}$ – задача обробки зображень об'єктів в стані руху (прямолінійного, обертального, поступального, равноускоренного та інших видів);

$D_{Hem} = \{F_1, F_2, \ldots, F_{em}\}$ – задача розпізнавання емоцій (F_1 – здивування, F_2 – страх, F_3 – щастя, …).

Відображення $T: H_D \to Y_D$ дозволяє для кожного $H_D(i)$ знайти таке $Y_j \in Y_D$ ($j = \overline{1,Q}$, Q – кількість класів), яке є рішенням задачі D_H.

Значення $Y_j \in Y_D$ використовуються для формування рішення про стан здоров'я людини за допомогою нейромережевого класифікатора і вироблення подальшої тактики поведінки.

Стан k-го працівника визначається як $sost^k{}_j \in SOST$, $j = \overline{1, n}$.

Будь-який стан з набору SOST визначається набором параметрів функціонування організму k-го працівника

$$sost^k{}_j = \begin{bmatrix} par_{j1} \\ ... \\ par_{jh} \\ ... \\ par_{jp} \end{bmatrix} \qquad (2)$$

де par_j – значення *j* - го параметра організму *k*-го працівника, j= 1,…h,…,p.

Тоді

$$SOST = \left[\begin{bmatrix} par_{11} \\ ... \\ par_{1p} \end{bmatrix} , ..., \begin{bmatrix} par_{n1} \\ ... \\ par_{np} \end{bmatrix} \right]. \qquad (3)$$

Для того, щоб описати зміну стану працівника під впливом чинників зовнішнього середовища необхідно враховувати початковий стан працівника та комплекс тих чинників, що безпосередньо діють на працівника $ф^1$, ..., $ф^m$, i = $\overline{1,m}$, $ф^i$ –i -й чинник, що діє на організм.

$$sost^k{}_j \in SOST = \vec{w}(t_0) + \Delta\vec{w}(t), \qquad (4)$$

де $\vec{w}(t_0)$- стан організму працівника у початковий момент часу; $\Delta\vec{w}(t)$ - зміна стану організму працівника під впливом комплексу чинників за час t, $t \in \lfloor 0,..., t_j,..., T \rfloor$.

Для визначення зміни стану організму працівника під впливом комплексу чинників скористаємося моделлю Гаммерштейна [4] для системи (організму) з n внутрішніми параметрами – вектор стану системи та $\vec{ф}(t)$ – m зовнішніми впливами, що залежать від часу:

$$\Delta\vec{w}(t) = \Gamma\ (\vec{\phi}(t), t_1, t_2) = \int_0^{t_2 - t_1} \vec{w}(\tau) \cdot \vec{f}(\vec{\phi}(t_2 - \tau))\, d\tau \qquad (5)$$

де $\vec{f}(\vec{\phi}(t_2 - \tau))$ - векторна функція перетворення вхідного впливу чинників на організм людини в опис реакції певного організму.

Даний вираз можна розглядати як модель зміну стану працівника при сумісному впливу комплексу виробничих чинників. Виходячи з припущення, що у початковій момент процесу роботи внутрішній стан працівника залишається незмінним, модель (5) прийме вигляд [5]:

$$\Delta\vec{w}(t) = \Gamma\ (\vec{\phi}(t), 0, T) = \vec{w}(\tau_0) \cdot \int_0^T \sum_{i=1}^{m} \left[\int_0^{\infty} \omega(\tau) \phi^{ik}(t - \tau) d\tau \right], \qquad (6)$$

де $\vec{w}(\tau_0)$ – векторна функція, що визначає внутрішній стан організму людини в початковий момент часу τ_0, причому будь який стан визначається набором параметрів з виразу (2); $\omega(\tau)$ – імпульсна перехідна матриця-функція розміром m×n, яка відображає конкретний та незмінний взаємозв'язок між чинниками та набором параметрів, що характеризують стан людини в момент часу τ; ϕ^{ik} - і-ий виробничий чинник, що діє на k-го співробітника підприємства.

Основним методом вирішення задачі знаходження перехідної функції $\omega(\tau)$, яка встановлює конкретний вид залежності між результатами вимірювання впливу виробничих чинників і реакцією організму спостережуваного співробітника на цей вплив, є складання рівняння Вінера-Хопфа. Існує ряд методів рішення рівняння Вінера-Хопфа, які ґрунтуються на подальшій параметризації завдання шляхом розкладання $\omega(\tau)$ по заданій системі функцій, або переходу до дискретного часу. Ця функція дозволяє встановити залежність реакції будь-якого організму на спільне вплив виробничих факторів.

Визначаючи стан співробітника по набору з параметрів у відповідності з виразом (2) і з огляду на всі запропоновані вище удосконалення, модель (6) можна представити таким чином:

$$\Delta w(\tau) = \Gamma_2(\mu_{\phi^k(t)}, t_1, t_2) = \int_0^{t_2-t_1} \begin{bmatrix} par_1(0) \\ \dots \\ par_h(0) \\ \dots \\ par_n(0) \end{bmatrix} \cdot \begin{bmatrix} \sum_{i=1}^{m} \int_0^{\infty} \omega_1(\tau)\, \mu_{\phi^{i\,k}(t)}(t-\tau)d\tau \\ \dots \\ \sum_{i=1}^{m} \int_0^{\infty} \omega_h(\tau)\, \mu_{\phi^{i\,k}(t)}(t-\tau)d\tau \\ \dots \\ \sum_{i=1}^{m} \int_0^{\infty} \omega_n(\tau)\, \mu_{\phi^{i\,k}(t)}(t-\tau)d\tau \end{bmatrix} dt, \qquad (7)$$

де $par_1(0),...,par_h(0),...,par_n(0)$ - параметри організму працівника для визначення стану в початковий момент часу; $\omega_1(\tau),...,\omega_h(\tau),...,\omega_n(\tau)$ - імпульсна перехідна матриця-функція опису реакції організму на вплив діючих виробничих чинників; $\mu_{\phi^{ik}(t)}$ - показник шкідливості процесу на працівника для m значень величини і-го чинника, що діє на k-го працівника в момент часу.

Таким чином, стає можливим визначити за допомогою виразів (2-4) стан співробітника по вимірюється параметрами організму співробітника безпосередньо перед початком його трудової діяльності. Модель (7) дозволяє визначити зміну стану організму співробітника під впливом виробничих факторів за час виконання цим працівником виробничих завдань.

Для оцінки якості процесу функціонування пропонованої системи використовується сукупність критеріїв оцінки ефективності $K = \{k_1, k_2\}$. Найбільш важливим критерієм є час відгуку системи при вирішенні загального завдання D_Ξ

$$k_1 : \tau = \min(\tau^r, \tau^{st}, \tau^{dn}, \tau^{em}),$$

де τ^r – час обробки показників стану організму людини, τ^{st} – час обробки зображень в стані спокою, τ^{dn} – час обробки зображень в стані руху, τ^{em} – час розпізнавання емоцій.

Іншим критерієм є точність вирішення загальної задачі

$$k_2 : \varsigma = \min(\varsigma^r, \varsigma^{st}, \varsigma^{dn}, \varsigma^{em}),$$

де ς^r – помилка рішення задачі обробки показників стану організму людини, ς^{st} – помилка рішення задачі обробки зображень в стані спокою, ς^{dn} – помилка рішення задачі обробки зображень в стані руху, ς^{em} – помилка рішення задачі розпізнавання емоцій.

Критерієм ефективності розробки є задоволення таким вимогам

$$K : \forall (D_i \in D_\Xi)[(\tau < \tau^{max}) \& (\varsigma < \varsigma^{max})] \Rightarrow \Xi, \quad (8)$$

де D_i – підзадача спільного завдання D_Ξ $(i = \{H_r, H_{st}, H_{dn}, H_{em}\})$, τ^{max} – максимально допустимий час вирішення завдання ς^{max} – максимально допустима помилка.

Імітаційне моделювання системи віддаленого моніторингу та прогнозування стану здоров'я працівника

Для оцінки ефективності алгоритмів створено експериментальний кластер Hadoop, що складається з 9 комп'ютерів (Intel Core i3-7100, 3,9 ГГц, ОС Microsoft Windows 12) з фізичною топологією «зірка», яка складається з 8 вузлів Datanodes та одного Namenode. Завдання керувалися менеджером YARN.

Реалізована приватна постановка завдання: визначення зміни стану здоров'я бригади зварників, що виконували електрогазове зварювання особливо складних і відповідальних конструкцій і трубопроводів з високовуглецевої сталі, призначених для роботи під динамічними і вібраційними навантаженнями і високим тиском. Будь-який зварювальний процес завжди супроводжується цілою низкою чинників, які становлять небезпеку для здоров'я як зварника, так і людей, що знаходяться поблизу під час зварювання. Найбільш небезпечним по впливу на людину елементом даного процесу слід вважати електричну дугу, так як інтенсивність її випромінювання дуже висока. Крім того, при будь-якому вигляді зварювання в тій чи іншій мірі присутні такі шкідливі фактори:

- ультрафіолетове випромінювання;
- сліпуча яскравість видимого світла;
- інфрачервоне випромінювання:
- іскри і бризки розплавленого металу;
- шкідливі речовини, що виділяються в процесі зварювання в вигляді аерозолів і газів (залежать від виду зварювання, виду електрода, виду виконуваних робіт і матеріалів, що зварюються).

Вплив комплексу вказаних шкідливих факторів на людину може виявитися критичною для серцево-судинної і нервової систем його організму.

Результати замірів параметрів стану організму одного зі співробітників бригади зварників наведені в табл. 1.

Таблиця 1

Результати вимірів параметрів стану організму співробітника бригади зварників

№ п/п	Час початку зміни	День тижня	Систолічний артеріальний тиск	Діастолічний артеріальний тиск
1	8.00	понеділок	130	85
2	8.00	вівторок	135	90
3	8.00	середа	135	90
4	8.00	четвер	125	84
5	8.00	п’ятниця	140	90

В ході виконання зазначених робіт вимірювання діючих виробничих факторів проводились за допомогою наступних вимірювальних приладів. Для вимірювання енергетичних характеристик був використаний вимірювач теплової опромінення РАТ-2П, який призначений для забезпечення вимірювань інтегральних характеристик опромінення у всьому спектральному діапазоні. Даний вимірювач застосовується для проведення санітарно-гігієнічних досліджень при атестації робочих місць (рекомендований Міністерством охорони здоров'я України). Температура виробничого середовища вимірювалася за допомогою універсального вимірювача параметрів мікроклімату «Метеоскоп-М», призначеного для проведення моніторингу середовища в житлових і виробничих приміщеннях, на відкритих територіях. «Метеоскоп-М» використовується для контролю параметрів мікроклімату, атестації робочих місць на підприємствах, в офісах і громадських установах. Результати вимірювань за один робочий день (четвер) наведені в табл. 2.

Дані про зовнішні чинники були інтерпольовано між змінами значенням рівня ЕМВ = 0 і значенням температури повітря = 20° С. Дані про вимірювані параметри стану співробітника інтерпольованого за методом Ейткена-Лагранжа. Були отримані часові ряди для досліджуваних параметрів, які були відцентровані. Рішення рівняння Вінера-Хопфа проводилося шляхом його дискретизації і відомості до чотирьох системам лінійних рівнянь. Нумерація факторів зовнішнього середовища: 1 - рівень ЕМВ; 2 - температура повітря. Нумерація: 1 - рівень систолічного артеріального тиску; 2 - рівень діастолічного артеріального тиску.

Таблиця 2

Результати вимірювань

День тижня, в який проводилися вимірювання	Час проведення виміру	Рівень ЕМВ, Вт/м^2	Температура, °С
Четвер	8.00	0	26
Четвер	10.00	83	29
Четвер	12.00	86	31
Четвер	14.00	95	32
Четвер	16.00	131	32

З метою регуляризації рішення було застосовано згладжування отриманих результатів. За допомогою графічного аналізу результатів для функцій $\omega_{11}(\tau)$, $\omega_{21}(\tau)$ була обрана формула

$$\omega(\tau) = Ce^{-a\tau}. \quad (9)$$

Для функції $\omega_{22}(\tau)$, була обрана формула

$$\omega(\tau) = C\tau^b e^{-a\tau}. \quad (10)$$

Для функції $\omega_{12}(\tau)$ була обрана формула $\omega(\tau) = C = Const$.

Співвідношення (9), (10) були лінеаризовані шляхом логарифмування, коефіцієнти визначалися методом найменших квадратів, в результаті чого були отримані наступні співвідношення:

$\omega_{11}(\tau) = 0{,}083e^{-0{,}1613\tau}$;

$\omega_{21}(\tau) = 5{,}6 \cdot 10^{-3} e^{-0{,}3402\tau}$;

$\omega_{12}(\tau) = -0{,}00324$;

$\omega_{22}(\tau) = 0{,}277 \cdot \tau^{1{,}409} e^{-0{,}5498\tau}$;

Розглянемо детальніше застосування моделі (7) на прикладі розрахунку, виробленого на основі даних, виміряних в четвер, в момент початку зміни (див. табл.1 і 2). Спочатку розглянемо приклад розрахунку оцінки зміни стану співробітника в процесі професійної діяльності під впливом виробничих чинників через 2 години після початку зміни. Реакція організму людини на вплив виробничих факторів розраховується за допомогою:

$$f\left(\mu_{\phi^{ik}(t)}\right) = \sum_{i=1}^{N} \omega(\tau_i)\mu_{\phi^{ik}(t)}(t-\tau_i)\Delta\tau_i = \sum_{i=1}^{N} \begin{bmatrix} (R_{\mu\mu})_{11} & (R_{\mu\mu})_{12} \\ (R_{\mu\mu})_{21} & (R_{\mu\mu})_{22} \end{bmatrix} \cdot \begin{bmatrix} \omega_{i1} \\ \omega_{i2} \end{bmatrix}, \quad (11)$$

де $\omega(\tau)$ - імпульсна перехідна матриця-функція опису реакції організму на вплив факторів розміром m×n; $\mu_{\phi^{ik}(t)}$ - показник шкідливості процесу на співробітника для m значень величини i-го фактора, що діє на k-го співробітника в момент часу; $R_{\mu\mu}$ - елемент матриці для розрахунку сумарного показника шкідливості; N - кількість замірів факторів.

Тоді

$$\Delta\vec{w}(\tau)=\begin{bmatrix}par_1(0)\\ par_2(0)\end{bmatrix}\int_0^T\left(\int_0^\tau\begin{bmatrix}(R_{\mu\mu})_{11} & (R_{\mu\mu})_{12}\\ (R_{\mu\mu})_{21} & (R_{\mu\mu})_{22}\end{bmatrix}\times\begin{bmatrix}\sum_{i=1}^{N}\omega_{11}(\tau)+\omega_{12}(\tau)\\ \sum_{i=1}^{N}\omega_{21}(\tau)+\omega_{22}(\tau)\end{bmatrix}d\tau\right)dt=$$

$$=\begin{bmatrix}125\\ 84\end{bmatrix}\cdot\begin{bmatrix}0 & 13\\ 13 & 26\end{bmatrix}\cdot\begin{bmatrix}\sum_{i=1}^{2}0{,}083e^{-0{,}1613\tau}-0{,}00324\\ \sum_{i=1}^{2}5{,}6\cdot10^{-3}e^{-0{,}3402\tau}+0{,}277\cdot\tau^{1{,}409}e^{-0{,}5498\tau}\end{bmatrix}+$$

$$+\begin{bmatrix}83 & 112\\ 112 & 29\end{bmatrix}\cdot\begin{bmatrix}\sum_{i=2}^{2}0{,}083e^{-0{,}1613\tau}-0{,}00324\\ \sum_{i=2}^{2}5{,}6\cdot10^{-3}e^{-0{,}3402\tau}+0{,}277\cdot\tau^{1{,}409}e^{-0{,}5498\tau}\end{bmatrix}=\begin{bmatrix}3\\ 0{,}3\end{bmatrix}.$$

Результати обчислень зміни значень параметрів стану співробітника при впливі на нього виробничих чинників через 2 години після початку зміни наведені в табл. 3.

Таблиця 3

Результат виконання обчислень для прогнозу зміни параметрів стану співробітника через 2 години після початку зміни

Використовуваний параметр	Значення параметра на момент початку зміни	Розрахована величина зміни параметра	Прогнозоване значення параметра
$par_1(t)$	125	+3	128
$par_2(t)$	84	+0,3	84,3

За результатами, наведеними в табл. 3 можна зробити наступний висновок. В результаті професійної діяльності протягом 2 годин після початку зміни параметри стану організму співробітника змінилися. Збільшення параметрів стану організму свідчить і про збільшення величини $sost^k_j \in SOST$ з виразу (2).

Виходячи з припущення, що зростання значення цієї величини відповідає погіршення стану, ми приходимо до висновку про деяке погіршення стану співробітника після 2 години після початку зміни.

Таким же чином був проведений розрахунок оцінки зміни стану співробітника в процесі професійної діяльності за допомогою математичної моделі оцінки зміни стану співробітника під впливом виробничих факторів через 4 і 6 годин після початку зміни. Результати даних розрахунків наведено в табл. 4.

Таблиця 4

Результат виконання обчислень

Параметр	Значення параметра на момент початку зміни	Розрахована величина зміни параметра через 2 години	Розрахована величина зміни параметра через 4 години	Розрахована величина зміни параметра через 6 годин	Прогнозоване значення параметра на момент закінчення зміни
$par_1(t)$	125	+3	+1,02	+1,78	+130,8
$par_2(t)$	84	+0,3	+1,9	+2	+88,2

Таким чином, ґрунтуючись на припущенні про неможливість визначення зміни стан організму співробітника за результатами прямих вимірювань була вирішена задача оцінки зміни стану організму (артеріального тиску) за результатами спостереження впливу факторів виробничого середовища за допомогою вдосконаленої математичної моделі зміни стану співробітника підприємства (7).

Виконано моделювання, що засноване на спостереженнях за працівником за показниками датчиків. Опис динамічних характеристик виконане за допомогою мови темпорального трасування (Temporal Trace Language, TTL). Для вираження динаміки в TTL важливими поняттями є стани, моменти часу й трасування. Модельовані процеси були переведені в здійсненну підмножину TTL, що називається leadsto. Спрощений формат leadsto дозволяє моделювати прямі тимчасові залежності між двома характеристиками стану в такий спосіб. Нехай α і β характеристики стану кон'юнкції літералів ‘ (де літерал – це атом або узгодження атома), і e, f, g, h – ненегативні дійсні числа. У мові *leads to* вираз

$\alpha \rightarrow\rightarrow_{e, f, g, h} \beta$ означає:

якщо характеристика стану α знаходиться на

певному інтервалі часу із тривалістю g

тоді після деякої затримки (між e і f)

характеристика стану β перебуватиме на певному

інтервалі часу довжиною h

Реалізована приватна постановка завдання – нагляд за рівнем артеріального тиску працівника та міри для збереження здоров'я.

Для моделювання та реалізації використана мова LEADSTO, Leads To Editor, а також TTL Editor для перевірки правильності моделювання та побудови правильних слідів.

У завданні, що пов'язано з артеріальним тиском, змінні пов'язані саме з тиском: high_stress_lvl, normal_stress_lvl, low_stress_lvl. Також, змінні для позначення того, що саме потрібно робити з показниками тиску: home_care, call_amb, do_not_call_amb. Задано час для кінцевого моделювання та додано інтервали для кожної зі змінних артеріального тиску (наприклад, 10 одиниць), а також початок сліду та кінець (рис. 2). Кінець сліду означає початок наслідку для даної змінної. Змінна high_stress_lvl сигналізує, що потрібно викликати швидку та надавати негайну допомогу.

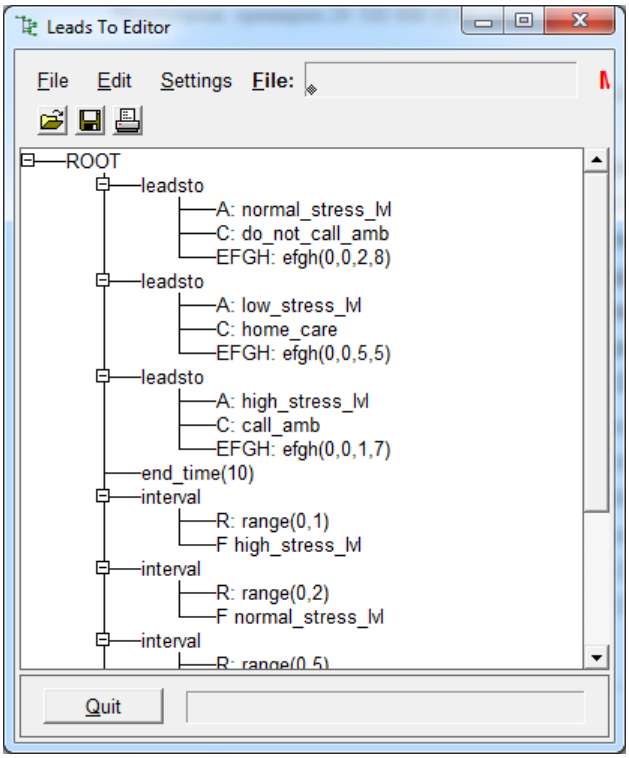


Рис.2. Часові інтервали для усіх Leadsto-умов

Усіма часовими відмітками пояснене, що якщо вираз носитиме правдивий характер з тривалістю g, то весь вираз матиме значення тривалістю h із затримкою 0.0. Для перевірки коректної роботи специфікації запускається LEADSTO Simulation Tool і, якщо під час введення специфікації не виникало

ніяких помилок, то побачимо такий результат (рис.3,4). Розроблена експериментальна система добре узгоджується із запропонованою архітектурою системи віддаленого моніторингу та прогнозування стану здоров'я працівника [3].

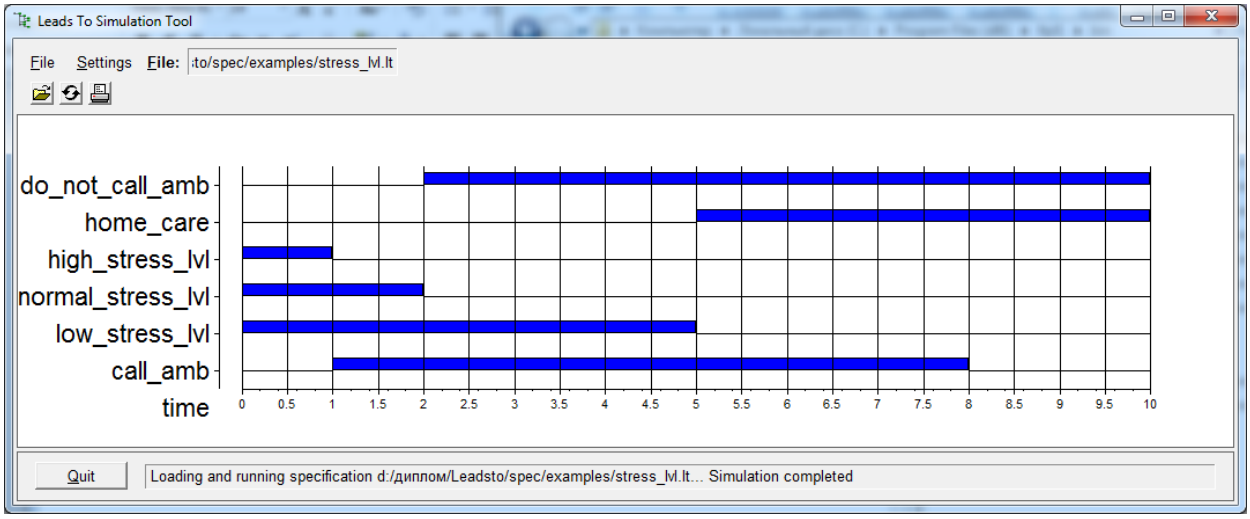


Рис.3 Успішно виконана специфікація (1)

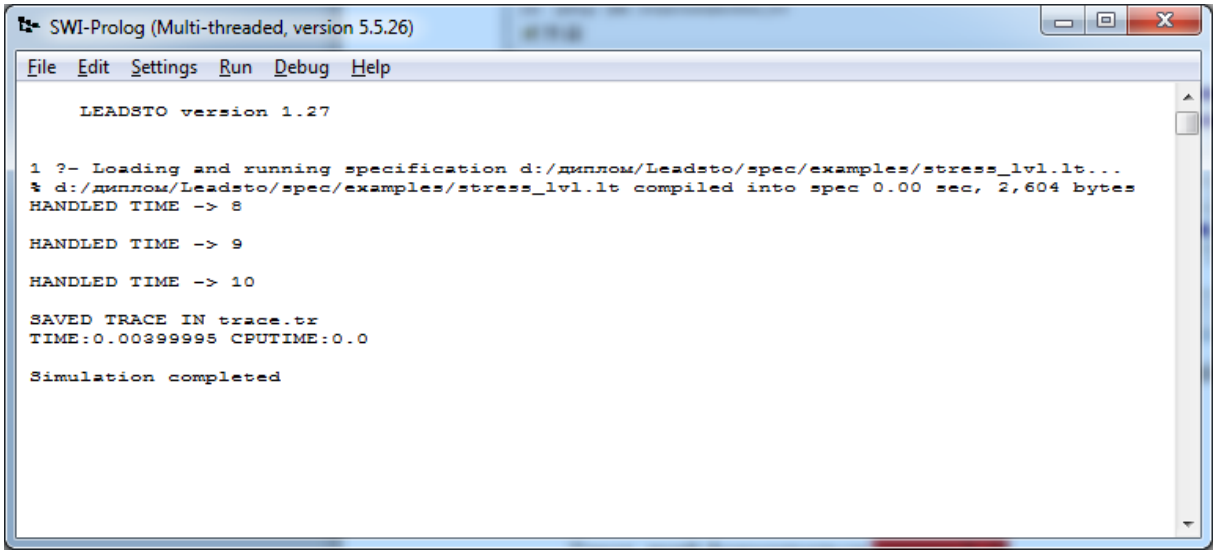


Рис.4. Успішно виконана специфікація (2)

Ця система включає чотири типи програмного забезпечення.

1. Персональне програмне забезпечення встановлюється на особистому цифровому пристрої пацієнта, та взаємодіє з Hadware (датчики, вимірювальні пристрої тощо), які збирають інформацію про співробітника та цифровану інформацію відправляє Master серверу.

2. Програмне забезпечення обслуговуючого персоналу з’єднується з Master сервером, і отримує дані відповідно до вимог користувача. Це програмне забезпечення дозволяє взаємодіяти з даними про стан співробітника не залежно від місця розташування спостерігача.

3. Серверне програмне забезпечення встановлено на Master сервері на

платформі .NET Framework із застосуванням об’єктно-орієнтованої мови програмування C #. Воно обробляє і зберігає дані про співробітників. Для зберігання інформації про користувачів обрана база даних MS SQL Server. Підключення до бази даних здійснюється за допомогою ORM (Object-Relational Mapping).

4. Резервне копіювання даних і їх відтворення це механізм, що забезпечує безпеку даних при збої в роботі серверів.

Висновок

Пропонується комплекс взаємопов’язаних завдань, вирішення яких полягає у розробці моделей та методів організації ефективної взаємодії людини та сервіс-орієнтованої системи на різних архітектурних рівнях, як в горизонтальній і вертикальній площинах.

Запропонована архітектура системи віддаленого моніторингу та прогнозування стану здоров'я співробітника дозволяє взаємодіяти з великою кількістю різнорідних джерел, включаючи цифрові фотографії та медико-діагностичну інформацію, і своєчасно реагувати на зміну стану здоров'я.

По результатам реалізації було виявлено, що запропонована модель СВМП:

– використовує та своєчасно управляє знаннями про стан здоров’я працівника;

– забезпечує взаємодію між діючими сторонами;

– необхідну інформацію відправляє користувачам у потрібний час;

– забезпечує найбільш раціональні рішення відповідно до належних запитів.

Розроблено програмні модулі на основі запропонованої моделі СВМП, які інтегровані в єдиний комплекс сервіс-орієнтованої комп’ютерної системи, що дозволило підвищити її продуктивність за якістю рішень і термінів отримання результатів.

Література

1. Axak N. Cloud-fog-dew Architecture for Personalized Service-oriented Systems / N. Axak, D. Rosinskiy, O. Barkovska, I. Novoseltsev // The 9th IEEE International Conference on Dependable Systems, Services and Technologies, DESSERT’2018, Kyiv, Ukraine – 2018. – P. 80–84.

2. Axak N. G. Development of multi-agent system of neural network diagnostics and remote monitoring of patient. /N. G. Axak //Eastern-European

Journal of Enterprise Technologies. – 2016. - 4/9 (82) – P. 4-11.

3. Axak N., Korablyov M., Rosinskiy D. MapReduce Hadoop Models for Distributed Neural Network Processing of Big Data Using Cloud Services //International Conference on Computer Science and Information Technology. – Springer, Cham, 2019. – C. 387-400.//doi.org/ 10.1007/978-3-030-33695-0

4. Сердюк Н.Н. Модели типа Гаммерштейна для описания нелинейного воздействия группы факторов на организм человека // Радиоэлектроника и информатика. − 2006. − № 1. − С.111–113.

5. Сердюк Н.Н. Функциональная задача оценки влияния вредных производственных факторов на человека // Восточно-Европейский журнал передовых технологий. – 2013. − № 4/4 (64). − С. 22-25.

ГЛАВА 9
АВТОМАТИЗОВАНЕ АНОТУВАННЯ ЕЛЕКТРОННИХ ТЕКСТІВ З ВИКОРИСТАННЯМ МЕТОДУ ЗВ'ЯЗНИХ СЛОВОФОРМ

Вступ та постановка задачі

Формальний опис основного змісту електронних текстових документів в зручному для програмної обробки форматі є одним з найбільш важливих напрямків використання активно розроблюваних в останнє десятиліття семантичних технологій [1]. Документи, що мають при цьому бути аналізовані і описані, належать зазвичай до веб-ресурсів або до ресурсів електронних бібліотек. Під семантичними електронними бібліотеками (СЕБ) будемо розуміти електронні бібліотеки, які використовують семантичні технології для організації всіх процесів своєї роботи, таких як опис ресурсів, ведення каталогів, опис профілів користувачів, пошук і рекомендація ресурсів користувачам тощо. Однією з важливих функцій семантичних електронних бібліотек є надання можливості анотування ресурсів, що публікуються. Формування анотацій ресурсів СЕБ та веб-ресурсів може виконуватися із застосуванням різних семантичних засобів, зокрема шляхом виділення наборів ключових слів та аналізу зв'язків між словоформами анотованого тексту. Особливе значення має застосування анотування веб-ресурсів під час роботи інформаційно-пошукових систем (ІПС). Принцип роботи сучасних ІПС заснований на концепції інвертованого індексу, який ставить у відповідність термінам ті частини документа, в яких вони зустрічаються. Процес створення інвертованого індексу включає ряд етапів, найважливішими з яких є: створення списку нормалізованих лексем для кожного документа; сортування списку, в результаті якого терміни розташовуються в алфавітному порядку; угруповання багаторазово повторюваних термінів. В результаті індекс фактично є словником, що зберігає терміни, розташовані в алфавітному порядку; частоту появлення деякого терміну в документах, а також номери документів, в яких зустрічається цей термін. Вважається, що така структура інвертованого індексу є досить ефективною для пошуку текстової інформації за довільним запитом. Однак організація інвертованого індексу з анотацій документів дозволить істотно скоротити обсяг оброблюваної інформації і підвищити швидкість роботи ІПС. Як відомо, основними компонентами ІПС є модуль індексування, база даних та пошуковий сервер, з'єднані за кільцевою схемою (рис. 1).

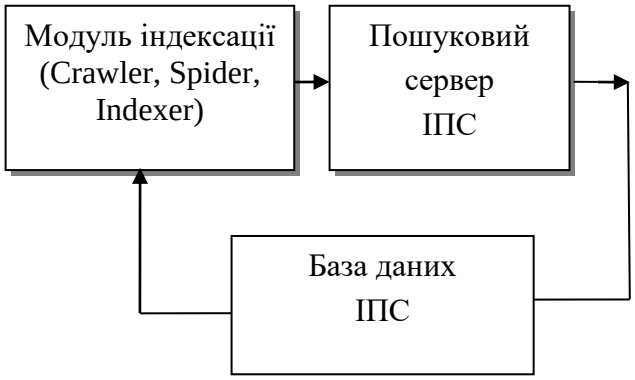


Рис. 1. Основні компоненти ІПС

Модуль індексування, в свою чергу, складається з трьох допоміжних програм: Spider (програма, яка здійснює скачування HTML-кодів Web-сторінок); Crawler (програма, що виділяє посилання, присутні на сторінці, і визначає напрямок подальшого пошуку, тобто здійснює пошук нових документів, яких ще немає в системі); Indexer (програма, що очищає текст індексованої сторінки від всіх нетекстових елементів, зокрема графічних фрагментів та тегів розмітки HTML).

Потім здійснюється перегляд сторінок з метою вибірки слів і видалення інформації, яка словами не вважається (прогалини, знаки пунктуації). Відповідно до пропонованого далі підходу, доцільно до функцій Indexer'a, таких як: аналіз тексту, заголовка та спеціальних HTML-тегів, додати функцію анотування документа. В цьому разі алгоритм роботи пошукового сервера зазнає незначних змін, а саме: користувальницький запит буде підвергнуто морфологічному аналізу з використанням модуля ранжирування. Далі має бути згенеровано розширений сніппет, в якому до звичайного заголовку додаватимуться два-три речення, які містять ключові слова і посилання на сам документ, що дає користувачеві можливість при наведенні курсору на посилання, крім перегляду збереженої в кеші копії документа або схожих документів, ознайомитися з анотацією. Таким чином, користувач отримує можливість уже на початковому етапі пошуку мати уявлення про документи, що містять потрібну інформацію, і в разі необхідності переглянути тексти документів. Така технічна реалізація дозволить суттєво скоротити час, що витрачається системою на пошук потрібної інформації.

На даний момент одним з найбільш перспективних підходів до складання анотацій до електронних тестових документів є семантичне стиснення аналізованого тексту.

Метою статті є розробка та програмна реалізація метода формування контекстних множин ключових слів та зв'язних словоформ, призначеного для подальшого автоматизованого анотування електронних текстових документів шляхом їх семантичного стиснення. Відповідно до цієї мети далі вирішуються наступні завдання: аналіз моделі семантичного стиснення текстових документів, які анотуються; модифікація алгоритму Гінзбурга, що дозволяє формувати матрицю зв'язків між словоформами анотації; тестування методу формування контекстних множин ключових слів та зв'язних словоформ.

Математична модель семантичного стиснення текстової інформації

Семантична складова процедури стиснення тексту вимагає фундаментальних змін в методології дослідження процесу стиснення тексту. У зв'язку з цим виникає необхідність застосування моделі, що дозволяє, з одного боку, коректно здійснювати стиснення тексту, а з іншого – гарантувати збереження смислової складової вихідного тексту.

Процес стиснення тексту включає ряд етапів. Оригінальний текст піддається ряду перетворень на шляху його стиснення. Весь процес поетапного перетворення вихідного тексту в анотацію може бути представлений у вигляді моделі. Модель семантичного стиснення тексту повинна включати ряд компонентів а саме: вхідні і вихідні дані, а також характеристики функціональних перетворень вхідних даних в анотацію. У моделі, що розробляється, пропонується ввести новий компонент, що характеризує рівень стиснення тексту. Цей компонент дасть можливість користувачеві самостійно визначати розмір анотації, в той час як модель повинна гарантувати коректне формування самої анотації [2].

Таким чином, модель семантичного стиснення (semantic compression) тексту може бути представлена у вигляді кортежу:

$$M^{SC} = \langle X, Y, F, CR \rangle, \quad (1)$$

де X – множина вхідних даних; Y – множина вихідних даних; F – множина функцій перетворення вхідних даних у вихідні; CR – множина значень рівня стиснення (compression rate).

Модель семантичного стиснення тексту (1) дає можливість, по-перше, поетапно представити процес перетворення вхідних даних в анотацію, а по-

друге – дозволяє користувачеві задавати рівень стиснення і при необхідності змінювати його значення [3]. Слід зазначити, що користувач може впливати тільки на розмір анотації, а сама модель гарантує включення в анотацію речень, які найбільш повно відображають зміст вхідного тексту, і не залежить від рівня стиснення. Аналіз моделі грунтується на дослідженні її компонентів. У моделі (1) множина вхідних даних X представлена у вигляді такого кортежу:

$$X = \langle ST_x, W_x \rangle, \tag{2}$$

де ST_x – упорядкований набір речень (sentences); W_x – упорядкований набір слів вхідного тексту (words).

Такий опис вхідних даних дозволяє зберегти порядок проходження слів у кожному реченні, порядок проходження пропозицій в тексті, а, отже, зберегти семантичну складову моделі (1). Кортеж речень в початковому тексті ST_x є упорядкованим набором елементів, кожному з яких присвоєно номер

$$ST_x = \langle st_{x1}, st_{x2}, ..., st_{xi}, ..., st_{xN} \rangle, \overline{i = 1, N}, \tag{3}$$

де st_{xi} – і-е речення у вхідному тексті; N – кількість речень вхідного тексту.

У свою чергу, набір слів в кожному і-му реченні вхідного тексту також може бути представлений у вигляді кортежу

$$W_{xi} = \langle w_{x1}, w_{x2}, ..., w_{xij}, ..., w_{xiM} \rangle, \overline{j = 1, M}, \tag{4}$$

де w_{xij} – j-е слово в і-му реченні вхідного тексту; M – кількість слів в і-му реченні.

Весь набір слів вхідного тексту з урахуванням місця розташування кожного слова в реченні і речення в тексті може бути представлений у вигляді

$$W_x = \sum_{i=1}^{N} \sum_{j=1}^{M} w_{xij}. \tag{5}$$

У моделі (1) множина вихідних даних Y представлена як упорядкований набір речень ST_y та кортеж слів W_y, що ввійшли до анотації.

$$Y = \langle ST_y, W_y \rangle. \tag{6}$$

Кортеж речень ST_y, які увійшли в анотацію, як і в випадку з реченнями вхідного тексту (3), представлений в моделі (1) у вигляді впорядкованого набору, однак кількість речень анотації може бути рівною або менше кількості речень в початковому тексті. Аналогічно (4) може бути описаний кортеж слів анотації W_y з тією різницею, що прив'язка здійснюється згідно з порядком розташування слів і номерами речень в анотації.

У моделі (1) вся множина слів вхідного тексту W_x включає множину ключових слів KW_x (keywords), множину стоп-слів SW_x (stop-words) і множину звичайних слів OW_x, які не увійшли до жодної з перших двох множин:

$$W_x = \{KW_x, SW_x, OW_x\}. \tag{7}$$

Ключовим словом будемо називати слово, яке входить в анотацію, і має високе значення рангу (ранг слова дорівнює кількості повторень слова в початковому тексті). Стоп-словом в інформаційному пошуку вважається слово, що не несе самостійного змістового навантаження (прийменники, сполучники, займенники). Вони можуть бути виключені з вихідного тексту без шкоди для його семантичної складової. Модель (1) передбачає виключення стоп-слів тільки на етапі обробки вхідного тексту. Включені в анотацію речення можуть містити слова з повного набору слів, в тому числі і стоп-слова.

Множина слів вхідного тексту W_x, з урахуванням (7), може бути представлена як об'єднання множин

$$W_x = KW_x \cup SW_x \cup OW_x. \tag{8}$$

По аналогії з (7) та (8) може бути представлена множина слів анотації W_y, що об'єднує множину ключових слов KW_y, множину стоп-слов SW_y та множину звичайних слів OW_y, що ввійшли в анотацію. Модель (6) передбачає, що всі елементи множини вихідних даних (6) входять в множину елементів вхідних даних (2). Іншими словами, множина речень анотації ST_y є частиною множини речень вхідного тексту ST_x:

$$ST_y \subset ST_X, \tag{9}$$

а множина слів анотації входить до множини слів вхідного тексту:

$$W_y \subset W_X, \tag{10}$$

У моделі (1) передбачено поетапне перетворення вхідних даних в анотацію. Внутрішнє представлення тексту в моделі кардинально відрізняється від початкового. Такий підхід дозволяє здійснити більш точну і швидку обробку даних. Він дає можливість видалити з тексту «шумову» складову у вигляді стоп-слів і здійснити угруповання слів по їх основам.

Процес перетворення вхідних даних представлено в моделі (1) множиною функцій перетворення F, що може бути надана в загальному вигляді як

$$F = \langle F_1, F_2, ..., F_i, ..., F_T \rangle, \overline{i = 1, T}, \tag{11}$$

де T – загальна кількість функціональних перетворень, передбачених моделлю (1).

Слід зауважити, що загальна їх кількість T буде змінюватися динамічно, оскільки користувач має можливість не тільки задавати рівень стиснення, але і змінювати це значення. Множину значень рівня стиснення CR визначимо як

$$CR = \langle cr_1, cr_2, ..., cr_i, ..., cr_B \rangle, \overline{i = 1, B}, \tag{12}$$

де B – кількість значень коефіцієнта стиснення.

Рівень стиснення CR є базовим поняттям моделі (1) і кількісно

визначає, наскільки має бути стиснутий вхідний текст.

У моделі передбачено дискретне завдання користувачем коефіцієнта стиснення в процентному відношенні з точністю до десятої частки відсотка, що дозволяє обробляти тексти великих і надвеликих обсягів. Для детального визначення набору функцій (11) необхідно поетапно розглянути всі функціональні перетворення тексту в анотацію, передбачені моделлю (1).

Так, на першому етапі модель передбачає розбиття вихідного тексту на речення і присвоювання кожному реченню номера ID_S (sentence identifier). Результатом роботи етапу є упорядкований набір речень ST_x (3).

Другий етап передбачає подання кожного і-го речення впорядкованим набором слів W_{xi} (4), а всього вхідного тексту – загальним набором слів W_x (10). Таким чином, модель (1) враховує місце розташування кожного слова в реченні і, відповідно, розташування кожного речення в початковому тексті.

Третій етап передбачає видалення стоп-слів з початкового тексту. У моделі (1) така процедура дозволяє спростити і прискорити подальший семантичний аналіз вихідного тексту.

Функціональне перетворення F_1 з множини перетворень (11) здійснює видалення стоп-слів і може бути представлено у вигляді:

$$F_1 : W_x \to W', \tag{13}$$

де W' – набір слів вхідного тексту, з якого видалені стоп-слова.

На четвертому етапі текст, отриманий в результаті перетворень перших трьох етапів, піддається процедурі стемінгу (приведення слова до його основи). В результаті набір слів W', отриманий після третього етапу, буде перетворений і представлений у вигляді:

$$F_2 : W' \to W'', \tag{14}$$

де W'' – набір слів, отриманий в результаті приведення слів набору W' до їх основ.

На п'ятому етапі отриманий набір основ слів W'' буде перетворений до виду модифікованого інвертованого індексу E:

$$E = \langle e_1, e_2, ..., e_i, ..., e_{LI} \rangle, \overline{i = 1, LI}, \tag{15}$$

де LI – кількість елементів в модифікованому інвертованому індексі.

Визначення 3. Модифікованим інвертованим індексом назвемо інвертований індекс, побудований в порядку зменшення рангів слів, що містить посилання на номер речення в документі.

Функціональне перетворення основ слів в елементи модифікованого інвертованого індексу може бути представлено у вигляді відображення:

$$F_3 : W'' \to E. \tag{16}$$

На шостому етапі здійснюється формування анотації. Цей процес проводиться поетапно і складається з відбору ключових слів і подальшого відбору речень, що містять найбільшу кількість таких слів. Такі речення і будуть представляти кінцеву анотацію. Функціональне перетворення елементів модифікованого інвертованого індексу в анотацію може бути представлено відображенням:

$$F_4 : E \to ST_y. \tag{17}$$

Слід зауважити, що функціональне перетворення (17) безпосередньо залежить від заданого користувачем значення рівня стиснення тексту CR:

$$F_4 : F(CR). \tag{18}$$

У моделі (1) передбачена можливість завдання значення рівня стиснення тексту CR відповідно до (12), отже, розмір анотації буде залежати від заданого значення.

Розглянуту модель можно розширити з застосуванням для анотування так званих зв'язних словоформ (в поєднанні з набором ключових слів).

Під зв'язними словоформами будемо розуміти слова (словосполучення) аналізованого текста, що можуть бути поєднані явним (зв'язним словом) або неявним семантичним зв'язком.

В даній роботі для виділення ключових слів і формування контекстних множин будемо використовувати модифікований алгоритм Гінзбурга та алгоритм контекстно-залежного реферування.

Формування ключових слів та зв'язних словоформ з використанням модифікованого алгоритму Гінзбурга

Перед тим, як перейти до опису змістовної структури тексту, необхідно визначити, яким чином ключові слова пов'язані між собою. З цього випливає, що для виявлення співвідношень між ключовими словами і визначення їх семантичної близькості необхідно мати інструмент для їх аналізу.

Алгоритми аналізу ключових слів дозволяють кількісно оцінювати зв'язок між словами і словоформами в потрібному нам тексті. Зокрема алгоритм Гінзбурга служить для пошуку контексту певного слова в досліджуваному тексті. Основна процедура алгоритму полягає в наступному [4]:

–в запропонованому тексті T визначаються відносні частоти RFT (x) для кожної словоформи;

–в тексті T знаходяться частини даного тексту (T') з однаковим розміром, які містять слово W. Для всіх цих слів будується частотний словник Voc (T'), який так само містить в собі відносну і абсолютну частоти. Відносну частоту словоформи x в Voc (T') позначимо як RFT' (x);

– порівнюються відносні частоти T і T' і вводиться індекс значущості словоформи x в контексті слова W, який позначимо як SI (x), що обчислюється таким чином:

$$SI(x) = \frac{RFT'(x)}{RFT(x)}. \qquad (19)$$

– словоформу відносимо до контексту слова W, якщо індекс значимості перевищує одиницю.

Після визначення контекстних множин обчислюється сила залежності між двома словоформами. Щоб обчислити силу зв'язку (BF) між словоформами x_1 та x_2 потрібно сформувати контекстні множини S_1 та S_2, які мають позитивний індекс значимості. Число словоформ в S_1 та S_2 позначимо відповідно як N_1 та N_2.

Згідно з алгоритмом обробки контекстних множин здійснюються наступні кроки:

– по модулю підсумовуються специфічні частини SS;

– виділяється різниця для кожної частини в загальній частині:

$$SI(S_1) - SI(S_2). \quad (20)$$

– по модулю підсумовуються отримані різниці SG;

– обчислюється загальна сума ST:

$$ST = SS + SG. \quad (21)$$

– розраховується сума всіх індексів значущості $SumS_1$ всіх словоформ, які мають приналежність до множини S_1;

– розраховується сума всіх індексів значущості $SumS_2$ всіх словоформ, які мають приналежність до множини S_2;

– визначається сила зв'язку між словоформами x_1 та x_2:

$$BF = 1 - \frac{ST}{SumS_1 + SumS_2}. \quad (22)$$

В результаті отримуємо значення в діапазоні від 0 до 1. Якщо сила зв'язку дорівнює 0, це вказує на відсутність загальних слів в контексті, а якщо 1 – на повний збіг. Величину сили зв'язку між словоформами також можна інтерпретувати і як величину, зворотну відстані між словоформами в семантичному просторі досліджуваного тексту. Таким чином, використання алгоритму Гінзбурга дозволяє виділити ключові слова і контекстні множини словоформ, які будуть використані для подальшої більш точної побудови анотації тексту. Також алгоритм Гінзбурга дає можливість визначити силу зв'язку між словоформами та їх частоту присутності в тексті. Саме ці параметри необхідні для аналізу входження ключових слів і слів з контекстної множини в сформовану анотацію. Задавання обмежень значень цих характеристик (мінімальне значення) дозволяє отримати анотацію тексту, яка буде включати в себе словоформи з найбільшою частотою входження і силою зв'язку, вказуючи тим самим на суттєве смислове навантаження отриманого тексту. Модифікація базового алгоритму Гінзбурга полягає в його доповненні процедурою формування зв'язних словоформ. Алгоритм встановлення зв'язків між зв'язними словоформами, що базується на використанні по методу головного концепту, наведеного в [4]. Головною процедурою якого є вибір зв'язної словоформи і її нормалізація її до одного слова, для якого слід сформувати трійку «слово-зв'язок-слово» за допомогою

настроюваних коефіцієнтів (зменшувати в разі знаходження великої кількості непотрібної інформації і збільшувати в разі недостатньої кількості зв'язків в тексті). До елементів такої трійки віднесемо: слово (словосполучення), що означає зв'язок між двома словоформами (L_1) і власне дві словоформии (W_1 та W_2), кожна з яких може бути представлена одним словом або словосполученням. Таким чином, трійку можна представити у вигляді: «слово№1, зв'язок, слово№2»: $W_1 \leftrightarrow L_1 \leftrightarrow W_2$.

Розглянутий метод формування контекстних множин ключових слів та зв'язних словоформ, призначений для подальшого анотування електронних текстових документів, назвемо методом зв'язних словоформ.

Програмна реалізація алгоритму анотування електронних текстових документів

Запропонований алгоритм анотування електронних текстів з використанням методу зв'язних словоформ включено до складу комплексного програмного модуля Booker VI, що дозволяє здійснювати класифікацію текстових документів по категоріям з можливістю подальшого формування їх анотацій [5]. Реалізовано програму побудови анотації тексту заданої довжини на основі модифікованого методу Гінзбурга для пошуку контекстної множини слів в рамках розглянутого тексту, що включає в себе:

– визначення контекстних множин ключових слів та зв'язних словоформ;

– реалізацію методу (обробка тексту і виділення анотації);

– запис обробленої інформації в БД;

– можливість перегляду повного тексту;

– відображення результатів роботи програми (анотації в зручному вигляді з можливістю редагування та збереження в форматі .txt).

На основі розробленого набору інструментальних і алгоритмічних засобів, що дозволяють аналізувати текст з метою подальшого його анотування, було розроблено додаток, що підтримує такі функції:

–побудова контекстної множини за алгоритмом Гінзбурга для словоформи, що аналізується;

– обчислення сили зв'язку між двома словоформами;

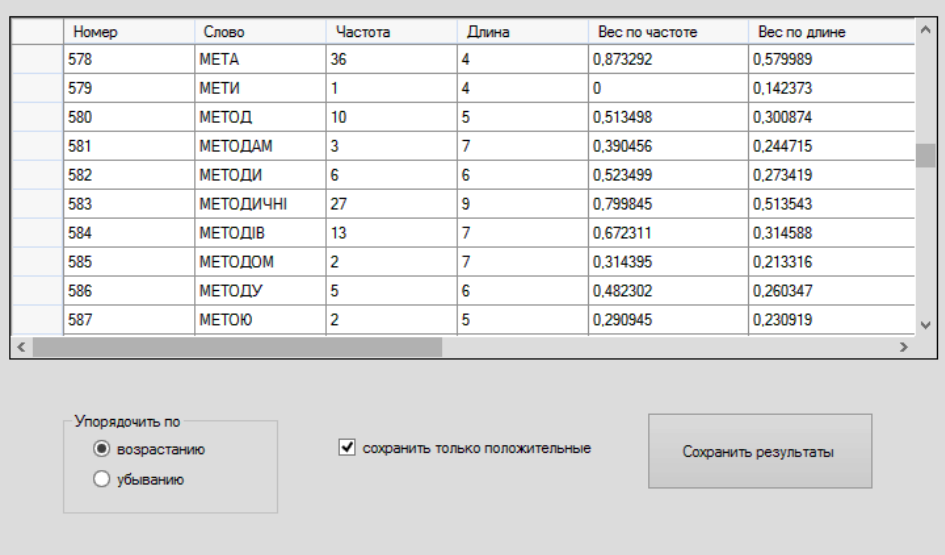
– побудова для заданого числа ключових слів з максимальними індексами значущості матриці зв'язків між словоформами;

– побудова анотації тексту, яка включає в себе речення, що містять словоформи в заданому діапазоні індексів значущості;

-– побудова анотації заданого розміру;

–отриманого результату і збереження його у форматі .txt.

Перша частина розробленої програми дозволяє відображати обраний документ для аналізу і анотування в окремому вікні для зручного перегляду або читання. На рис. 2 наведено фрагмент вікна програми для виділення тематично маркованої лексики. Слова розміщуються в алфавітному порядку з відображенням індексу тематичного маркування та інших критеріїв, таких як частота, вага по частоті, вага по довжині тощо.



Номер	Слово	Частота	Длина	Вес по частоте	Вес по длине
578	МЕТА	36	4	0,873292	0,579989
579	МЕТИ	1	4	0	0,142373
580	МЕТОД	10	5	0,513498	0,300874
581	МЕТОДАМ	3	7	0,390456	0,244715
582	МЕТОДИ	6	6	0,523499	0,273419
583	МЕТОДИЧНІ	27	9	0,799845	0,513543
584	МЕТОДІВ	13	7	0,672311	0,314588
585	МЕТОДОМ	2	7	0,314395	0,213316
586	МЕТОДУ	5	6	0,482302	0,260347
587	МЕТОЮ	2	5	0,290945	0,230919

Рис. 2. Фрагмент вікна перегляду результатів з відображенням індексів тематичного маркування

На рис. 3, рис. 4 наведено фрагменти прикладу вікон для побудови для заданого числа ключових слів з максимальними індексами значущості матриці зв'язків між словоформами.

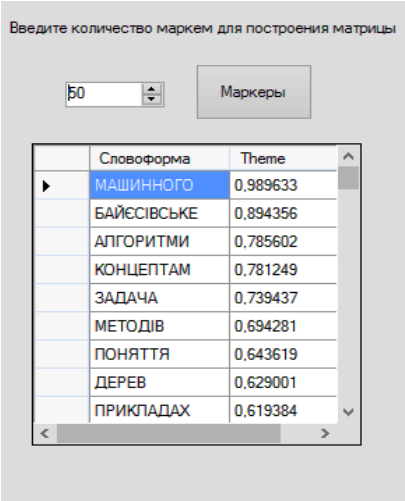


Рис. 3. Фрагмент прикладу вікна формування переліку ключових слів (словоформ)

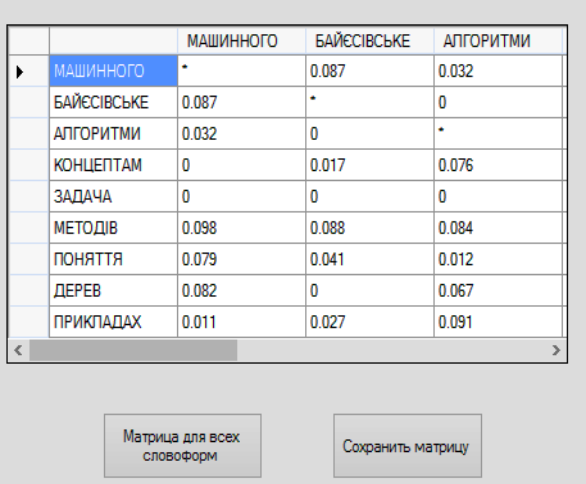


Рис. 4. Фрагмент прикладу вікна формування матриці зв'язків між словоформами

Під час тестування за допомогою програми були оброблені тексти методичних матеріалів і побудовані матриці зв'язків словоформ (до 100 ключових слів з максимальним індексом тематичного маркування).

У таблиці 1 наведено приклад фрагменту матриці ключових слів (контекстної множини словоформ), що мають величину сили зв'язку більше 0,1 для слова «НАВЧАННЯ».

Таблиця1

Матриця словоформ, пов’язаних зі словом «Навчання»

Словоформа	Сила зв’язку
машинного	0,989633
байєсівське	0,894356
алгоритми	0,785602
концептам	0,781249
задача	0,739437
методів	0,694281
поняття	0,643619
дерев	0,629001
прикладах	0,619384

Можна бачити, що отримані результати (табл. 1) дійсно відображають особливості сполучуваності слів і взаємозв'язок понять в аналізованих методичних матеріалах. Таким чином, розроблена програма дозволяє виявляти контекстні множини (слова, близькі за контекстом) для конкретного слова в аналізованих текстах. З використанням цих множин можна отримати інформацію про взаємозв'язок між значущими словоформами тексту.

Друга частина реалізованої програми дозволяє побудувати анотацію. Так як для складання анотації використовуються результати аналізу ключових слів і виділених контекстних множин, то користувачеві пропонується вибрати мінімальну частоту зустрічі слова в тексті, мінімальну силу зв'язку між словоформами, а також задати довжину анотації виходячи з кількості речень, які вона може містити. Можливий вибір трьох значень довжини анотації: коротка (2-3 речення); середня (4-6 речень); довга (7-9 речень). Анотація буде побудована відповідно вибраній довжині. Кількість речень для будь-якого з розмірів може коливатися. Це залежить від насиченості речень словоформами з контекстної множини. Якщо максимальна кількість речень для обраного розміру анотації не може бути виділена, довжина анотації буде зменшена до отримання набору речень із заданою частотою зустрічі словоформ з контекстної множини (але не менше за мінімум для обраного розміру анотації).

Приклад отриманого результату роботи програми наведено на рис. 5. Анотація представлена в окремому вікні з можливістю редагування. Це дозволяє виправити смислові та семантичні помилки в тексті, доповнити або скорегувати отриманий результат. Результат роботи програми зберігається в текстовому документі у форматі .txt.

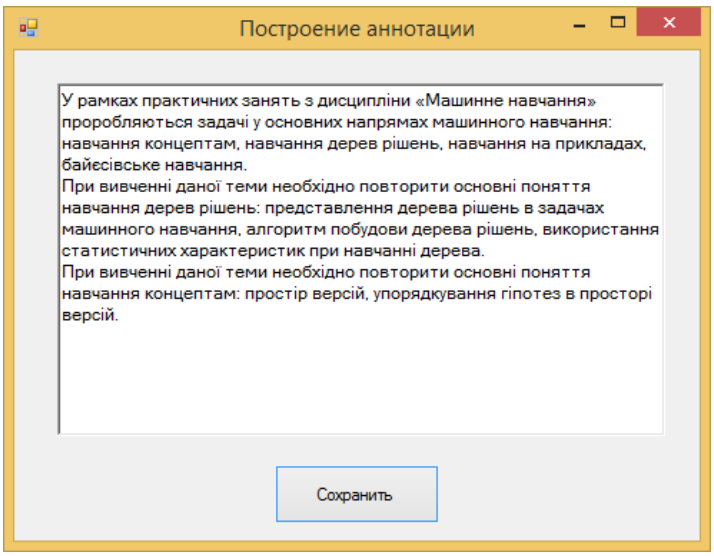


Рис. 5. Вікно формування та редагування результату

Отриманий результат обробки текстової інформації включає в себе словоформи, які мають силу зв'язку вище 0,6 і зустрічаються в тексті найчастіше: «машинне навчання», «байєсівське навчання», «дерева рішень», «навчання дерев», «навчання концепту». В цілому анотація містить речення, які відображають ключові завдання документа та є коротким опис розглянутого тексту. Втім сформована анотація не завжди вдало відображає переходи між реченнями. Це передбачає доцільність незначного редагування користувачем отриманого тексту з виправленням семантичних і лексичних помилок.

Таким чином, за допомогою розробленої системи можна отримати результат обробки тексту в вигляді анотації. Якість цього результату частково залежить від попередньої обробки тексту: виділення ключових слів, контекстних множин, визначення сили зв'язку і частоти зустрічі словоформ. Метод зменшує можливість виникнення похибок в визначенні ключових слів і дозволяє видалити шумові словоформи, які можуть вплинути на вибір речень для анотації. При задовільних умовах роботи алгоритму користувач може отримати анотацію, яка несе в собі коротку інформацію про аналізований текст. Анотація може бути відкоригована користувачем для виправлення семантичних і лексичних помилок.

Даний підхід є простим в реалізації, але досить точним, так як можливість вибору мінімальних значень частоти словоформ, сили зв'язку між ними та розміру анотації дозволяє користувачеві регулювати не тільки розмір отриманої анотації, але й розширювати або звужувати коло понять, які

потрібно включати в анотацію тексту.

Висновки та перспективи подальших досліджень

В роботі запропоновано підхід до автоматичного анотування текстів з використанням методу зв'язних словоформ. Основою цього методу є модифікований алгоритм Гінзбурга, що дозволяє виділяти ключові слова тексту, а також зв'язані словоформи, які мають схожий зміст і загальну тематику. За допомогою запропонованого методу підвищується якість виділення речень, які увійдуть в анотацію і будуть містити в собі найбільшу кількість ключових слів зі встановленою силою зв'язку між словоформами і частотою їх зустрічі в тексті. Програмно реалізований метод зв'язних словоформ включено до складу комплексного програмного модуля Booker VI, що дозволяє здійснювати класифікацію текстових документів по категоріям з можливістю подальшого формування їх анотацій (з функціями редагування та збереження в форматі .txt). Роботу розробленого методу було протестовано на прикладі аналізу електронних навчальних документів (методичних вказівок). Отримана анотація може потребувати часткової корекції в разі наявності синтаксичних або семантичних помилок, але є досить точним описом основного сенсу аналізованого тексту. Перспективним розвитком розглянутого підходу є його поліпшення і модернізація за допомогою комбінування ключових параметрів визначення ключових слів і контекстних множин, а також використання інших алгоритмів для підвищення якості формування анотацій.

Література

1. Kumar B. Automatic Keyword Extraction for Text Summarizationin Multi-document Articles / B. Kumar, B. Sathya, P. Anima // European Journal of Advances in Engineering and Technology. – 2017. – Vol. 4 (6). – P. 410 – 427.

2. Автоматическая обработка текстов на естественном языке и компьютерная лингвистика / Большакова Е.И., Клышинский Э.С., Ландэ Д.В., Носков А.А., Пескова О.В., Ягунова Е.В./ – М.: МИЭМ, 2011. – 272 с.

3. Domingue J. Handbook of Semantic web Technologies. / D. Fensel, J.A. Hendler. – Heidelberg; Dordrecht; London; N.Y.: Springer, 2011. – 1077p.

4. Чалая Л.Э. Метод поиска релевантных связей между концептами проектируемых онтологий / Л.Э. Чалая, А.В. Чижевский, Е.Б. Волощук// АСУ и приборы автоматики. – 2015. – № 172. – С. 48–54.

5. Удовенко С Г. Метод аналізу звязків між концептами предметних онтологій / С. Г. Удовенко, Л. Е. Чала, О.Є. Гриньова // Матеріали XVII Міжнародної науково-практичної конференції «Математичне та програмне забезпечення інтелектуальних систем». – 2019.– С. 264-265.

ГЛАВА 10
ПРОБЛЕМИ СИМПЛІФІКАЦІЇ МЕДИЧНИХ ТЕКСТІВ УКРАЇНСЬКОЮ МОВОЮ

Вступ

Удосконалення систем документообігу (в тому числі електронного) в різних організаціях має на меті, з одного боку, оперативне проходження службових документів по інстанціях, а з інший – уніфікацію документів для підвищення якості управлінських рішень. Це, в свою чергу, передбачає ефективний менеджмент по синхронізації документарних потоків серед виконавців і керівників, а значить – автоматизоване прогнозування трудовитрат на підготовку і опрацювання документів. Особливо це стосується звітних та інформаційно аналітичних документів, обсяги яких у великих організаціях можуть бути значними, внаслідок чого якість їх підготовки істотно впливає на час ознайомлення з ними і прийняття рішень керівниками.

Відомо, що невиправдано довгі речення або складні лексичні конструкції ускладнюють сприйняття тексту документа, тому вимоги ділового та наукового стилю містять положення про стислість і лаконічність тексту. Однак для об’єктивної кількісної оцінки складності тексту документа, що дозволяє прогнозувати ефект від підготовлюваної в організації документації, необхідний обґрунтований вибір і програмна реалізація відповідного методу, адекватного щодо мови і стилістики документообігу.

Медицина, біологія, фармакологія та інші суміжні області наповнені складними термінами та поняттями. Медичні тексти містять багато запозичених слів, наприклад, з латині. Більш того, терміни можуть мати декілька синонімів, що робить тексти ще складнішими. Читачі, які не є експертами, можуть отримати додаткову користь від використання спрощених медичних текстів у різних ситуаціях. По-перше, коли людина не має медичної освіти, спрощений текст може стати корисним, щоб зрозуміти, що означають деякі медичні інструкції. Наприклад, коли в медичному приписі вказано зробити деякі аналізи, назва тесту може бути досить складною (як, наприклад, біохімічний аналіз крові), тоді як для пацієнта це буде означати лише аналіз крові, який вимагає певної підготовки до нього. По-друге, людина, можливо, захоче отримати всебічне пояснення свого діагнозу, записаного у медичну картку. Спрощений текст у цьому випадку може допомогти зрозуміти характер розладу. По-третє, після отримання

пацієнтом припису від лікаря, який містить багато складних медичних термінів, може виникнути багато питань. Тому спрощення тексту може допомогти людині виконувати вказівки лікаря та отримувати чіткі пояснення, що означають конкретні методи лікування.

Медична інформація доступна в Інтернеті в інформаційних системах медичних закладів, порталах охорони здоров'я, медичних бібліотеках і т. д. Все більше і більше читачів-неекспертів прагнуть використовувати цю інформацію. Цифровий характер та доступність цих даних разом із широким колом потенційних читачів спонукають до розробки інформаційної системи спрощення медичних текстів.

Проблема спрощення тексту виникає, коли початковий текст має бути змінений, щоб зробити його більш зрозумілим для аудиторії [1]. Можуть бути різні причини, чому початковий текст виглядає неприйнятним для використання [2]. Найбільш вірогідні випадки є результатом самої складності тексту або особливостей аудиторії, яка хоче використовувати текст. У будь-якому разі спрощення текстів вимагає сучасних методів обробки мови та аналізу даних [3, 4]. Складність синтаксичної та лексичної структури тексту може спричинити чимало незручностей для читачів. Наприклад, текст може бути надто складним для людей з певними вадами і порушеннями, які ускладнюють сприйняття інформації. Інші категорії потенційних читачів, які можуть зіткнутися з проблемами під час використання складних текстів, включають дітей, людей з низьким рівнем грамотності, іншомовних студентів тощо. Для всіх них спрощений текст буде хорошим рішенням для отримання ідеї та використання текстової інформації .

Іншою сферою, де проблема спрощення текстів є досить важливою, є машинна обробка текстових даних. Оригінальні тексти можуть бути досить складними для методів обробки природних мов (NLP). Тому, щоб вирішити деякі проблеми обробки мови, було б зручно застосувати алгоритми NLP до текстів, які вже були спрощені раніше. Такі проблеми включають пошук та аналіз інформації, узагальнення та анотацію інформації, машинний переклад тощо.

Постановка здачі дослідження

Основна ідея даного дослідження – спрощення медичного тексту, залежить від складності цього тексту та особи, яка вивчає цей текст. Так, для пацієнтів такі частини як паспорт протоколу, список довідок, можуть бути опущені. Для пацієнтів найбільший інтерес представляють ті частини

протоколу, які описують симптоми захворювання, епідеміологію, необхідні дії лікаря і, особливо, рекомендації. Слід також зазначити, що вся медична документація надається державною мовою. Як результат, медичний текст наповнений не лише спеціальними латинськими термінами, а й складними медичними словами української мовт. Аналіз українського тексту з точки зору лінгвістики є непростим завданням. У цьому випадку проблема ускладнюється величезною кількістю медичної термінології. При цьому текст містить також слова з предметної області, які не потребують спрощення. Запропонована реалізація процесу (рис. 1) є досить універсальною для таких завдань.

Припускається використання морфологічних шаблонів для ідентифікації складних медичних слів. Метою даного дослідження є реалізація підходу до складного процесу ідентифікації термінів для українських медичних текстів, що у подальшому можуть бути використані для симпліфікації медичних текстів.

Рис. 1. Процес симпліфікації медичних текстів

Підходи до оцінки складності тексту

Під складністю, в загальному випадку, розуміють складання об’єкта з декількох частин; різноманітність за складом вхідних елементів і зав’язків між ними. Відповідно текст, що складається з безлічі елементів, зв’язаних різного роду зв'язками, описується такою характеристикою, як складність. Складність довільного об'єкта може бути описана в рамках одного з п’яти підходів до визначення складності, рис. 2.

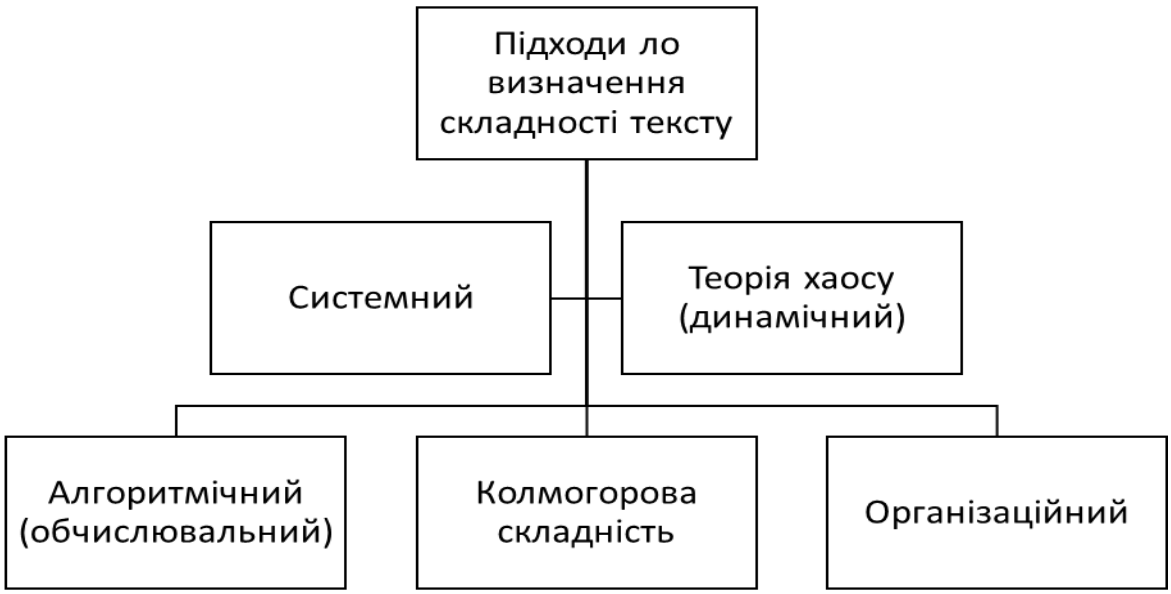


Рис. 2. Підходи до визначення складності тексту

В рамках кожного підходу розглядаються теоретичні аспекти складності і математичний апарат для отримання чисельних оцінок складності. У системному підході складність системи тим вище, чим більше число і різноманітність її елементів і зав’язків між ними, внаслідок чого складна система набуває безліч нових властивостей, непритаманних окремим підсистемам. Теорія хаосу оперує поняттям нелінійної динамічної системи, яка характеризується нестаціонарністю, тобто випадковими стрибкоподібними змінами характеристик в часі. З позицій теорії алгоритмів, обчислювальна складність є функція обсягу операцій, що виконуються деяким алгоритмом, від розміру вхідних даних. Ідея колмогорівської складності полягає в тому, що чим складніший об'єкт, тим більший його математичний (формальний) опис. Тут мірою складності є довжина мінімально можливого опису об'єкта. Організаційний (кібернетичний) підхід оперує поняттям керуючої підсистеми, здатністю системи до активної

адаптації. Чим важче ідентифікувати поведінку системи і процеси управління в ній, тим складніше система.

У XX столітті при розгляді поняття складності тексту виявляються такі синонімічні поняття, як легкість для читання, читабельність, труднощі тексту і милозвучність. Вони відображають, наскільки зручним для зорового або слухового сприйняття є текст, а факторами виступають розмір букв, колір шрифту і фону, наявність жаргонізмів і неологізмів і т.п. Однак такий підхід до складності тексту не дозволяє оцінити зміст тексту, його структуру і характеристики. Крім того, дослідження складності тексту багато в чому велися психологами, які враховували особистісні характеристики суб'єкта при оцінці природи тексту.

З початком XXI століття при описі складності тексту став переважати підхід, який оперує загальною ідеєю розуміння складності як кількості витрачених ресурсів для опису будь-якого об'єкта. Складності тексту відповідає абсолютна (об'єктивна) складність, яка залежить від обраного підходу до її оцінки. Втім, виділення суб'єктивних і об'єктивних факторів складності тексту досить умовне. Наприклад, інформативність і абстрактність тексту можуть бути оцінені як щодо, так і безвідносно до суб'єкту.

Процес спрощення тексту та збереження вихідної інформації та сенсу є основними проблемами спрощення тексту. Спрощення тексту може використовуватися для багатьох цілей: вивчення іншої мови, допомога людям з обмеженими можливостями, для автоматичного вилучення даних із рецептів лікаря, лабораторних результатів та інших медичних документів [5-7]. Для правильного представлення текстової інформації в обчислювальних формах для певного завдання, наприклад класифікації, кластеризації, аналізу емоцій, рекомендацій та пошуку інформації, використовуються різні способи спрощення тексту.

В даний час автоматичні системи лексичного спрощення або не мають достатнього рівня охоплення (контрольовані підходи), або виконують лише заміну одного-двох слів і, таким чином, не можуть спростити довші лексичні фрази, також, вони не виконують жодного виду переупорядкування слів [8]. У статті [9] було створено початковий набір даних для автоматизованого спрощення тексту з використанням вдосконаленого набору вказівок для ручного спрощення та розроблена методологія розширення набору даних. Семантична оцінка терміну є важливою для конкретних застосувань машинного навчання [4].

У роботі [11] досліджено використання спрощення тексту як етапу попередньої обробки для статистичного машинного перекладу. Досліди перекладу з англійської на сербської показують, що такий підхід може покращити грамотність перекладу та зменшити технічні зусилля після редагування (кількість операцій після редагування). Спрощення можна застосувати на лексичному, синтаксичному та дискурсному рівнях [12]. Деякі лексичні плагіни [13] дозволяють використовувати різні синоніми, щоб уникнути повторення, а у випадку синтаксичного спрощення користувач бачить усі сполучники в тексті та може відокремити складні речення кількома простим кліком.

На початковому етапі вивченням складності займалися вчені лінгвісти, реалізуючи свої методи для оцінки складності навчальної літератури шкільних навчальних посібників і підручників. З 20-х років XX століття робилися численні спроби чисельної оцінки складності тексту індексами удобочитабельності, спочатку для англійської мови, а потім для європейських мов. Перший індекс легкості читання був запропонований в 1923 році американські вчені-лінгвісти Б.Лайвлі і С. Пресси. Подальші дослідження в цьому напрямку були зроблені в 1947-1958 рр., 1967-1976 рр., А також з початку 2000-х років спостерігається сплеск інтересу до даного питання.

В основу більшості запропонованих формул лягла лінійна регресійна модель (ЛРМ), змінними якої виступають статистичні параметри тексту. До переваг даної моделі відносять простоту, гнучкість, а також однаковість аналізу і проектування, що проводиться за допомогою даних моделей. При використанні ЛРМ результат прогнозування може бути отриманий швидше, ніж при використанні інших моделей. Крім того, перевагою є прозорість моделювання, тобто доступність для аналізу всіх проміжних обчислень. Основним недоліком ЛРМ є складність визначення виду функціональної залежності, а також трудомісткість визначення параметрів моделі. До недоліків також відносять низьку адаптивність і відсутність здатності моделювання нелінійних процесів, а також високу кореляцію результатів для різних методів оцінки.

На початку 70-х років свої методи оцінки складності текстів стали пропонувати психологи і соціологи, які розглядали складність з точки зору віку людини і її психологічних особливостей. Пропонувалися альтернативні варіанти формул розрахунку складності, а також графи і таблиці. Так, в 1969 році Маклогін запропонував індекс «БМОС» читабельності тексту, яким

розраховував вік читача прозового твору через квадратний корінь від частки складних слів в тексті. Також в 70-х року були запропоновані індекси читабельності для російськомовних текстів і обґрунтовано методику їх побудови.

Існують дослідження, в яких висвітлено роль спрощення тексту. Показано, що розуміння прочитаного покращується, коли тексти вручну змінюються, щоб зробити мову більш доступною або зробити зміст більш прозорим завдяки явним дискурсним відносинам [14, 15]. Також було встановлено, що зміна тексту надала змогу читачам з низьким рівнем знань предметної області розуміти текст на рівні читача що має певний рівень спеціалізованих знань [16], а також, що переформулювання причинно-наслідкових зав’язків для порівняно складних текстів має суттєвий ефект для читачів що погано розуміють текст [17]. Подібні результати були отримані для читачів із низьким рівнем досвіду в спеціалізованій області [18]. Крім того, різні способи упорядкування слів можуть покращити сприйняття тексту.

З початку 2000-х років відзначається повернення інтересу до оцінки читабельності текстів, перш за все, в Україні. Крім адаптації відомого індексу Флеша до української мови зроблені перспективні спроби запропонувати індекси для текстів «дорослої аудиторії». Так, представляють інтерес роботи М.М. Невдаха І Ю.Ф. Шпаковського. На сучасному етапі вивчення складності тексту робляться спроби автоматизації процесу її розрахунку, дослідження проводяться на аудиторії студентів вузів. У глобальній мережі Інтернет з'явилися онлайн сервіси, що дозволяють оцінити складність тексту, проте в них реалізовані лише окремі формули, що часто не відповідають рівню аналізованого тексту.

На сучасному етапі вивчення складності тексту робляться спроби автоматизації процесу її розрахунку, дослідження проводяться на аудиторії студентів вузів. У глобальній мережі Інтернет з'явилися онлайн сервіси, що дозволяють оцінити складність вводиться користувачем тексту, проте в них реалізовані лише окремі формули, часто не що відповідають рівню аналізованого тексту.

Потрібно автоматизувати розрахунок вищевказаних методів і перевірити придатність їх використання до текстів типових інформаційно-аналітичних документів українською мовою.

Тому актуальною є задача автоматизації розрахунку вищевказаних методів і перевірка їхньої придатності до використання щодо текстів типових інформаційно-аналітичних документів українською мовою.

Отримані результати

Медичні тексти включають рецепти до ліків, медичні записи, аркуші справ, медичні довідники та навчальні матеріали, сертифікати тощо. Щоб вирішити проблему спрощення медичного тексту, спочатку необхідно виділити особливості таких текстів. У цьому дослідженні ми спираємось на тексти медичних клінічних протоколів. З метою прискорення розробки та впровадження державних стандартів у галузі охорони здоров'я Міністерство охорони здоров'я України затверджує медико-технологічні документи на основі доказової медицини. Такі документи включають уніфікований клінічний протокол медичної допомоги, а також клінічне випробування, засноване на випробуваннях. Залежно від захворювання, план лікування та профілактичні заходи можуть відрізнятися, що також передбачено законодавством у місцевих протоколах профілактики та лікування.

В Україні, яка активно інтегрується в європейську та світову спільноту, були прийняті реформи охорони здоров'я. Ці реформи забезпечують правову основу для доказової медицини. Універсальний клінічний протокол забезпечує основу для функціонування медичних установ та приватних лікарів. Тому в цій роботі саме тексти медичних протоколів використовуються для побудови корпусу медичних текстів для аналізу мовної складності.

На першому етапі розглядається типовий медичний протокол. Розглянемо як приклад протокол профілактики серцево-судинних захворювань [14]. Відповідно до наказу Міністерства охорони здоров'я України цей протокол визначає ознаки та критерії діагностики захворювання; умови, в яких повинна надаватися медична допомога; діагностична програма, що складається з обов'язкових та додаткових досліджень; медична програма; рекомендації щодо подальшої профілактики медичної допомоги.

Клінічні протоколи, а також інші документи публічно доступні в Інтернеті [14]. Вони є посібником для лікарів-практиків, адміністрації закладів охорони здоров'я, а також є джерелом інформації для пацієнтів. Усі протоколи до затвердження проходять багаторазову колективну експертизу. Дані, що вводяться в протокол, відповідають медичним стандартам національного та міжнародного рівнів.

Аналіз показує, що всі протоколи мають спільну структуру: вступ, скорочення, паспортна частина, загальна частина, основна частина, опис етапів медичної допомоги, ресурсна підтримка впровадження протоколу, показники якості медичної допомоги , перелік посилань та додатків.

Існує двадцять клінічних протоколів [14] для експериментів, які містять понад 2500 тисяч слів. Протоколи взяті з сайту "Реєстр медико-технологічних документів", і вони мають декілька напрямків, такі як дерматовенерологія, генетика, гастроентерологія, дерматологія, алергологія та гематологія. Тексти протоколів мають кілька обов'язкових частин, проте всередині абзаців вони слабо структуровані. Приклад оригінального тексту представлений на рис. 3, де виділено такі частини документа: жовтий колір – це абревіатури (АГ); синій колір - це складні медичні слова (кортикостероїди, гепатопротектори), а рожевий колір – це спеціальні медичні терміни (цироз печінки, портальна гіпертензія).

В рамках даного дослідження було розроблено програмне рішення, що надає можливість розрізнити складний для сприйняття текст медичної тематики від простого тексту. Це дозволить в подальшому розробити алгоритм для спрощення складного тексту. Розроблений програмний компонент базується на сторонній бібліотеці засобів Pymorphy2 (морфологічного аналізатору, https://github.com/kmike/pymorphy2) при створенні звітів та обробці текстових даних.

Даний програмний продукт слугуватиме як частина комплексу програмних продуктів що визначають складність тексту та зменшують її, роблячи текст більш доступним для читача. В свою чергу це допоможе покращити розуміння складних медичних інструкцій та рекомендацій для медичного персоналу що збільшить вірогідність правильного виконання цих інструкцій.

3.2. ДЛЯ ЗАКЛАДІВ ОХОРОНИ ЗДОРОВ'Я, ЩО НАДАЮТЬ ВТОРИННУ (СПЕЦІАЛІЗОВАНУ) МЕДИЧНУ ДОПОМОГУ

Положення протоколу Обгрунтування Необхідні дії лікаря

1. Профілактика

Обов'язковою умовою попередження рецидиву або переходу АГ у хронічну форму є повна відмова від вживання алкоголю пацієнтів, у яких був хоча б один епізод АГ.

Здоровий спосіб життя та повноцінне харчування є важливими доповненнями до абстиненції. Існують беззаперечні докази стосовно прогресування хвороби при подальшому вживанні алкоголю навіть у незначних кількостях.

Крім того Білково-калорійна незбалансованість раціону є фактором несприятливого прогнозу при АГ. Обов'язкові:

1.1. Призначити корекцію факторів ризику розвитку АГ, що пов'язані зі способом життя та зловживанням алкоголем.

1.2. Надавати рекомендації щодо здорового способу життя.

1.3. Рекомендувати пацієнтам і діагностованим АГ повну відмову від вживання алкоголю.

1.4. Надавати рекомендації щодо харчування пацієнтів із АГ (воно має бути повноцінним, підвищеної калорійності та збалансованим стосовно білків, вітамінів і мікроелементів).

2. Госпіталізація

Госпіталізація здійснюється при:

2.1. алкогольному гепатиті помірної або високої активності;

2.2. неефективності амбулаторного лікування;

2.3. ускладненнях АГ (цироз печінки, портальна гіпертензія, кровотеча з варикозно розширених вен стравоходу). Госпіталізація пацієнтів з АГ у стаціонар здійснюється при важкому протіканні АГ, що потребує призначення кортикостероїдів, гепатопротекторів та проведення адекватної дезінтоксикаційної терапії та парентерального харчування в умовах стаціонару. Необхідні дії лікаря Обов'язкові:

При плановій госпіталізації заповнюється відповідна медична документація:

- направлення на госпіталізацію.

Рис. 3. Приклад уніфікованого клінічного протоколу первинної, вторинної (спеціалізованої) медичної допомоги при алкогольному гепатиті.

Передбачається що користувачі цього програмного продукту мають навички роботи з терміналом операційної системи, що робить його доступним лише людям з певним рівнем комп'ютерної грамотності. В першу чергу користувачі цієї програми – це розробники програмного забезпечення, які на основі даного компоненту будуть створювати систему програмних продуктів.

Даний програмний продукт можна використовувати як окрему бібліотеку, для інтеграції з іншими програмами.

Також одна з властивостей – це можливість аналізувати тексти незалежно від їх ієрархії розташування, якщо тексти знаходяться нижче програми або на одному рівні з нею в файловій системі.

Для експерименту список стоп-слів був створений з різних ресурсів і містить такі частини мови, як займенники, прикметники, прийменники, вигуки, суфікси, комбінацію літер тощо. На першому кроці було використано просту частоту для ідентифікації терміну. Приклади найбільш і найменш вживаних слів наведені відповідно у таблицях 1 та 2. Ці результати були незадовільними, потрібні слова були у різних частинах списку термінів.

У таблиці 2 містяться не тільки спеціальні медичні терміни, такі як сероконверсія, гломерулонефрит, тощо, а також слова загальної медичної лексики – вагітність, спектроскопія. Таким чином, ми не можемо використовувати просту частоту для знаходження спеціальних медичних термінів.

Таблиця 1

Фрагментний список найбільш часто вживаних слів

Слово	Кількість
Лікування	1427
Пацієнт	1084
МОЗ	726
Україна	717
Здоров’я	476
Індикатор	473
Протокол	440
Лікар	421
Дитина	379
Розвиток	349
Проведення	334
Обстеження	329

Таблиця 2

Фрагментний список найменш часто вживаних слів

Слово	Кількість
Серконверсія	4
Гломерулонефрит	4
Опоїд	4
Моксифлоксацин	4
Сертралін	4
Остеопенія	4
Вагітність	4
Спектроскопія	4
Пентоксифілін	4
Гептагідрат	4

Наступним кроком були обрані іменники із загальних лексиконів частоти. Це дозволило знайти в тексті терміни, які мають таку характерну структуру: іменник + іменник або прикметник + іменник (прикметник + прикметник + іменник). Лексикони складених членів конструкції іменника + іменника та іменника + прикметника представлені в таблицях 3 та 4.

Таблиця 3

Список складених термінів (іменник + іменник)

Складений термін	Кількість
Охорона здоров'я	875
Кваліфікація хвороб	782
Дитина інвалід	218
Інфаркт міокарда	203
Забезпечення якості	194
Синдром Дауна	167
Стан здоров'я	154
Хвороба Гоше	48
Емфізема легень	29
Цироз печінки	14

Таблиця 4

Список складених термінів (іменник + прикметник)

Складений термін	Кількість
Практика сімейних	43
Контрольний рівень	42
Хірургічне втручання	35
Підвищений ризик	35
Шлунково-кишковий тракт	33
Хронічний кашель	27
Протозойна інфекція	26
Бронхіальна астма	18
Вірусний гепатит	18
Церебральний параліч	12

Відповідно було отримано статистику щодо лексичних шаблонів. Ця інформація представлена в таблиці 5. Загальна кількість іменників становить 5973.

Таблиця 5

Список лексичних конструкцій

Лексична конструкція	Кількість
Іменник	1322
Іменник+іменник	127
Прикметник+іменник	283
Прикметник+прикметник+іменник	46

Проблема полягає в якості та кількості даних у шаблонах. Набір даних було використано для оцінки отриманих результатів. Точність, повнота та F-міра представлені в таблиці 6.

Таблиця 6

Оцінка якості

Лексична конструкція	Точність	Повнота	F-міра
Іменник	0,53	0,48	0,5
Іменник+іменник	0,5	0,43	0,46
Прикметник+іменник	0,6	0,67	0,63

Аналіз якості експерименту показав незадовільні результати. Основною причиною цього є те, що було проаналізовано набір даних українською мовою. Морфологічний аналізатор працює неналежним чином на українських текстах. Список стоп-слів було змінено, але це не призвело до кращих результатів у пошуку спеціальних слів. Результати показують, що шаблони трапляються часто в текстах, але для їх автоматичного вилучення необхідний морфологічний аналізатор. Крім того, доцільно досліджувати структуру у медичних тестах. Вищезгадані проблеми – це теми для майбутніх досліджень.

У даній роботі аналіз частот показав, що конкретні та складні терміни використовуються рідше, що дозволило відкинути слова, які зустрічаються найчастіше. Таким чином результати експериментів показують, що можна виділити дві основні групи, які потребують спрощення:

1) конкретні слова (наприклад, глюкокортикостероїд), які можна замінити більш простими «розмовними» синонімами;

2) складні терміни, які можна спростити, уточнивши їх значення. Це вплине на синтаксичну структуру речення.

Для першої групи доцільно використовувати словник синонімів. Для роботи з другою групою необхідно створити спеціальну термінологічну онтологію чи інший лексичний ресурс.

Експеримент показує, що спеціальні медичні тексти, такі як протоколи, написані відповідно до певного шаблону, в результаті чого слова та фрази, такі як "лікування", "пацієнт" або "Україна", зустрічаються в сотні разів частіше, ніж інші іменники. Серед слів із середньою частотою (близько 100 в нашому експерименті) зустрічаються як прості слова (професор – 91, документ – 91), так і складні (клінічний огляд – 91, синдром – 178), а також діагнози (гепатит – 123, туберкульоз – 141). Слід зазначити, що серед слів із низькою частотою (менше 60) частка спеціальних медичних термінів становить 0,87. Це підтверджує гіпотезу про те, що частота слова в корпусі медичних текстів є ознакою його складності. У той же час слова, знайдені таким чином, потребують додаткового дослідження іншими методами.

Висновки

Виходячи з отриманих результатів експерименту, видно, що більшість методів при оцінці україномовних інформаційно-аналітичних документів дають оцінки, що виходять як за інтерпретований діапазон значень, так і за референсні значення. Можна також відзначити, що тексти маркетингових документів в середньому мають найбільшу складність, а організаційні - найменшу.

На підставі проведеного аналізу надається можливим зробити наступні висновки. Отримані результати характеризуються високим ступенем кореляції, в силу використання розробниками однієї математичної моделі (ЛРМ), а також одноманітності застосовуваних в них параметрів тексту (середня довжина слова, середня довжина пропозиції).

Реалізація автоматичної оцінки складності тексту є досить складним і мовозалежним завданням на етапі підрахунку статистики по тексту (наприклад, обчислення довжин фрагментів, виражених в слотах). Жоден з розглянутих методів не спрямований на оцінку сприйняття тексту дорослою людиною, що працює зі службовими документами. У професіонала не повинно виникати труднощів з розумінням складних слів. В кінцевому підсумку фактором складності виступає семантика тексту і абстрактність його викладу.

Перевірені індикатори читабельності тексту демонструють низьку интерпретованість, оскільки не можуть безпосередньо бути використані для

прогнозування часу обробки тексту тією чи іншою людиною. Тому доцільно продовжити дослідження в рамках сформульованих завдань, в тому числі реалізувати розрахунок і перевірку показників складності текстів на макрорівні представлення текстів.

Література

1. Siddharthan A. A survey of research on text simplification / A. Siddhartan // International Journal of Applied Linguistics. – Peeters Publishers. – 2014. – Vol. 165. – P. 259-298.

2. Shardlow M. A Survey of Automated Text Simplification / M. Shardlow // International Journal of Advanced Computer Science and Applications. – 2014. – P.58-70.

3. Services for Text Simplification and Analysis / J. Falkenjack et al. // Proceedings of the 21st Nordic Conference of Computational Linguistics. – Gothenburg, Sweden. – 2017. - P.309-313.

4. Matsuo R., Tu Bao Ho. Semantic Term Weighting for Clinical Texts / R. Matsua, Tu Bao Ho // Expert Systems With Applications. – 2018. – Vol. 114. – P. 543-551.

5. Popolov D., Barr J. R. Units of meaning in medical documents / D. Popolov, J.R. Barr // IEEEInternational Conference on Semantic Computing. – 2014. - P. 320-323.

6. NegAIT: A new parser for medical text simplification using morphological, sentential and double negation / P. Mukherjee et al. // Journal of Biomedical Informatics. – 2017. – Vol. 69. - P.55-62.

7. Sridevi M., Arunkumar B. R. Information Extraction from Clinical Text using NLP and Machine Learning: Issues and Opportunities / M. Sridevi, B.R. Arunkumar // International Journal of Computer Applications. – 2016. – P.11-16.

8. Stajner S., Saggion H., Ponzetto S.P. Improving lexical coverage of text simplification systems for Spanish / S. Stajner, H. Saggion, S.P. Ponzetto // Expert Systems with Applications. – 2018. – Vol. 118. – P.80-91.

9. Djamasbi S. Improving Manual and Automated Text Simplification / S. Djamasbi. – Doctoral dissertation. – Umass Medical School. – 2017.

10. Optimizing Statistical Machine Translation for Text Simplification / W. Xu et al. // Transactions of the Association for Computational Linguistics. – 2016. – Vol. 4. – P. 401-415.

11. Baltic J. Can Text Simplification Help Machine Translation / J. Baltic // Modern Computing. – 2016. – Vol. 4(2). – P. 230-242.

12. Stajner S., Glavas G. Leveraging event-based semantics for automated text simplification / S. Stajner, G. Glavas // Expert Systems With Applications. – 2017. – Vol. 82. – P. 383-395.

13. Integration of lexical and syntactic simplification capabilities in a text editor / R. Hervas et al. // Procedia Computer Science. – 2014. – Vol. 27. – P. 94-103.

14. Реєстр медико-технологічних документів // https://dec.gov.ua/mtd/home, 02.02.2020.

15. Medical text simplification using synonym replacement: Adapting assessment of word difficulty to acompounding language / E. Abrahamsson et al. // Proceedings of the 3rd Workshop on Predicting and Improving Text Readability for Target Reader Populations (PITR). – 2014. - P. 57-65.

16. Natural language processing to extract symptoms of severe mental illness from clinical text: the Clinical Record Interactive Search Comprehensive Data Extraction (CRIS-CODE) / R.G. Jackson et al. // Project BMJOpen. – 2017.

17. Explanation of “taghtirolbol” in traditional medical texts Which one Dribbling or Pollakiurea / F. Nojavan et al. // Journal of Islamic and Iranian Traditional Medicine. – 2015. – Vol. 6(2).

18. A Natural Language Processing System That Links Medical Terms in Electronic Health Record Notes to Lay Definitions: System Development Using Physician Re- views / J. Chen et al. // JMedInternetRes. – 2018.

ДОДАТОК

ВІДОМОСТІ ПРО АВТОРІВ ТА АНОТАЦІЇ

ГЛАВА 1

ВИКОРИСТАННЯ ЧАТ-БОТА @ES_ECONOMY_KARKAS_BOT ДЛЯ ОНЛАЙН КОНСУЛЬТАЦІЇ В ЕКОНОМІКО-ФІНАНСОВІЙ СФЕРІ

В. П. Бурдаєв

Анотація. *В роботі представлені результати інтегрування чат-бота @es_economy_karkas_bot з прототипами експертних систем для організації консультування в режимі онлайн. Дано опис архітектури і реалізація інтеграції чат-бота месенджера телеграм і системи "КАРКАС". Розглянуто структуру взаємодії чат-бота і агентів експертної системи в онлайн режимі. Наведено приклад онлайн консультації експертної системи у фінансовій та економічній предметній області на прикладі вибору комерційного банку.*

Ключові слова: *чат-боти, агенти, повідомлення, база знань, експертна система.*

Abstract. *The paper presents the results of the integration of the @es_economy_karkas_bot chat bot with prototypes of expert systems for organizing online counseling. The architecture description and implementation of the integration of the chat bot of the TELEGRAM messenger and the KARKAS system are given. The structure of the interaction of the chat bot and agents of the expert system in the online mode is considered. An example of online consultation of an expert system in the financial and economic subject field is given on the example of choosing a commercial bank.*

Keywords: *chat bots, agents, messages, knowledge base, expert system.*

ГЛАВА 2

ПРО ОДИН АЛГОРИТМ НАВЧАННЯ АДАЛІНИ В ЗАДАЧІ ОЦІНЮВАННЯ НЕСТАЦІОНАРНИХ ПАРАМЕТРІВ

О. Г. Руденко, О. О. Безсонов

Анотація. *На основі використання методів теорії ідентифікації досліджено властивості модифікованого регуляризованого алгоритму Качмажа. Визначено умови збіжності регуляризованого алгоритму Качмажа при оцінюванні настаціонарних параметрів при наявності завад*

вимірів. Отримано неасимптотичні та асимптотичні оцінки, як є досить загальними і залежать як від ступеня нестаціонарності об'єкту, так і від статистичних характеристик завад. Якщо ж ці параметри не відомі, слід скористатися будь-якої рекурентною процедурою їх оцінювання і використовувати одержувані оцінки для уточнення параметрів, що входять в алгоритми.

Ключові слова: *Алгоритм навчання, оцінювання, збіжність, нестаціонарний, завада, критерій, характеристики, рекурентна процедура*

Abstract. *Using the methods of the theory of identification, the properties of the modified regularized Kaczmarz algorithm are investigated. The conditions of convergence of the regularized Kaczmarz algorithm for estimation of the accretionary parameters in the presence of measurement interferences are determined. Non-asymptotic and asymptotic estimates are obtained, as they are quite general and depend on both the degree of non-stationarity of the object and the statistical characteristics of the interference. If these parameters are not known, then any recurrent estimation procedure should be used and the estimates obtained should be used to clarify the parameters included in the algorithms.*

Keywords: *algorithm of training, evaluation, convergence, non-stationary, interference, criterion, characteristics, recurrent procedure*

ГЛАВА 3
ГРАДІЄНТНІ АЛГОРИТМИ НАВЧАННЯ ЗГОРТАЛЬНИХ НЕЙРОННИХ МЕРЕЖ

К. О. Олейник, О. С. Романюк

Анотація. *Проведено порівняльний аналіз градієнтних алгоритмів навчання згортальних нейронних мереж при вирішенні задачі візуального розпізнавання. Дослідження показують, що для цієї задачі досить ефективним є простий алгоритм градієнтного спуску. Використання імпульсів в розглянутих модифікаціях призвело до деякого поліпшення процесу розпізнавання, але також збільшило вартість обчислень. У розглянутих задачах найбільш ефективним алгоритмом є Adamax. У той же час, однак, проблема вибору оптимальних значень параметрів алгоритмів, що забезпечують максимальну швидкість навчання, залишається відкритою.*

Ключові слова: *Алгоритм градієнтного спуску, навчання, розпізнавання, збіжність, критерій, швидкість навчання, параметр алгоритму*

Abstract. *A comparative analysis of gradient learning algorithms of convolutional neural networks in solving visual recognition problem was held in this report. Studies show that for this task quite effective is a simple gradient descent algorithm. Pulse usage in the considered modifications led to some improvement in the recognition process, but it also increased the computation cost. In the considered problems the most effective algorithm is Adamax. At the same time, however, the problem of selecting the optimal values of the algorithms parameters that provide top speed of learning still remains open.*

Keywords: *Algorithm of gradual descent, learning, recognition, criterion, speed of learning, parameter of the algorithm*

ГЛАВА 4
КЛАСИФІКАЦІЯ ДАНИХ НА ОСНОВІ ГІБРИДНИХ МОДЕЛЕЙ ШТУЧНИХ ІМУННИХ СИСТЕМ

М. М. Корабльов, О. О. Фомічов

Анотація. *Розглянуто рішення задач класифікації і кластеризації даних на основі гібридних моделей штучних імунних систем, які дозволяють змінювати як спосіб угруповання вихідних даних, так і структуру мережі імунних об'єктів. Гібридизація імунних методів і моделей інтелектуальної обробки інформації дозволяє також підвищити їх ефективність, що виражається в підвищенні швидкодії і точності їх роботи. Запропоновано гібридні імунні моделі класифікації даних на основі методу k-найближчих сусідів, нечіткого підходу, а також кластеризації даних на основі методу k-середніх з використанням моделей штучних імунних систем. Використання адаптивної імунної моделі дозволяє групувати дані або на основі навчальної вибірки, або в процесі імунної навчання. Проведено експериментальні дослідження, що підтверджують ефективність запропонованих гібридних моделей класифікації даних.*

Ключові слова: *класифікація, кластеризація, гібридна модель, штучна імунна система, навчальна вибірка, імунне навчання, афінність, антитіло, антиген.*

Abstract. *The solution to the problems of data classification and clustering based on hybrid models of artificial immune systems, which allow you to change*

both the method of grouping the source data and the structure of the network of immune objects, is considered. Hybridization of immune methods and models of intelligent information processing can also increase their effectiveness, which is expressed in increasing the speed and accuracy of their work. Hybrid immune models for data classification based on the method of k-nearest neighbors, a fuzzy approach, as well as clustering of data based on the k-means method using models of artificial immune systems are proposed. Using an adaptive immune model allows you to group data either on the basis of a training sample, or in the process of immune training. Experimental studies have been carried out confirming the effectiveness of the proposed hybrid data classification models.

Keywords: *classification, clustering, hybrid model, artificial immune system, training set, immune training, affinity, antibody, antigen.*

ГЛАВА 5

ОЦІНКА ЕФЕКТИВНОСТІ УПРАВЛІННЯ РОЗПОДІЛЕНИМИ КОМП'ЮТЕРНИМИ СИСТЕМАМИ В УМОВАХ НЕВИЗНАЧЕНОСТІ

М. Ю. Лосєв

Анотація. *В роботі розглядаються питання оцінки ефективності управління розподіленими інформаційними системами. При цьому пропонується критерій, який дозволяє враховувати імовірнісно-часові характеристики, навантаження мережі, а також цінність інформації, що передається. Розглядається методика нечітко-множинного аналізу старіння інформації в процесі її передачі в розподілених інформаційних системах. Методика дозволяє визначати нечіткі імовірнісні характеристики старіння даних. Можливо використання методики для оцінювання ефективності управління в умовах невизначеності.*

Ключові слова: *розподілена інформаційна система, ефективність управління, цінність інформації, старіння інформації, архітектура мережі, модель управління мережею, ресурси мережі, розподілена стратегія.*

Abstract. *The paper deals with the issues of evaluation of the management of distributed information systems. It offers a criterion that allows to take into account the probabilistic and temporal characteristics, network load, value of the transmitted information. The technique of fuzzy-multiple analysis of information aging in the process of its transmission in distributed information systems is*

considered. The technique allows to determine fuzzy probabilistic characteristics of data aging. It is possible to use a methodology to evaluate the effectiveness of management under uncertainty.

***Keywords**: distributed information system, management efficiency, value of information, aging of information, network architecture, model of network management, network resources, distributed strategy.*

ГЛАВА 6
МЕТОДИКА ПРОСУВАННЯ ІНТЕРНЕТ-МАГАЗИНУ В СОЦІАЛЬНИХ МЕРЕЖАХ

О. І. Пушкар, Є. М. Грабовський

***Анотація.** В розділі запропоновано методику просування інтернет-магазину в соціальних мережах. Визначено мету рекламної кампанії для інтернет-магазину одягу «Футболік», проведено налаштування спліт-тесту рекламної кампанії, виконано налаштування та запуск рекламної кампанії для плейсменту (стрічка Facebook, стрічка Instagram, Instagram Stories).*

Здійснено оцінку ефективності створеної рекламної кампанії та прогнозування поведінки інтернет-магазину після застосування рекламної кампанії.

***Ключові слова:** інтернет-магазин, просування, рекламна кампанія, плейсмент, спліт-тест.*

***Abstract.** The section offers a technique for promoting the online store on social networks. The purpose of the advertising campaign for the online clothing store "Footballgolik" was determined, the split test of the advertising campaign was made, the advertising campaign for the placement (Facebook feed, Instagram feed, Instagram Stories) was set up and launched.*

The performance of the created advertising campaign and the prediction of online store behavior after the application of the advertising campaign were evaluated.

***Keywords:** online store, promotion, advertising campaign, placement, split test.*

ГЛАВА 7
СУЧАСНІ ТЕХНОЛОГІЇ ЗБЕРІГАННЯ, РОЗПОВСЮДЖЕННЯ ТА ОПРАЦЮВАННЯ МУЛЬТИМЕДІЙНОГО КОНТЕНТУ СИСТЕМ ЕЛЕКТРОННОГО НАВЧАННЯ

О. К. Пандорін

Анотація. *В розділі наведено аналіз сучасних технологій зберігання, розповсюдження та опрацювання мультимедійного контенту систем електронного навчання. Розглянуті особливості застосування key - value архітектури у технологіях розповсюдження мультимедійного контенту. Систематизовано переваги та недоліки скрипкових мов. Окрему увагу наділено технології передачі струменевого відео у веб - браузер користувача.*

Ключові слова: *Мультимедійний контент, електронне навчання, скриптові мови, веб-базована технологія.*

Abstract. *This section provides an analysis of current technologies for storing, distributing and processing multimedia content of e-learning systems. Features of application of key - value architecture in technologies of distribution of multimedia content are considered. The advantages and disadvantages of violin languages have been systematized. Particular attention is given to streaming video technologies to the user's web browser.*

Keywords: *Multimedia content, e-learning, scripting languages, web-based technology.*

ГЛАВА 8
СИСТЕМА ВІДДАЛЕНОГО МОНІТОРИНГУ ТА ПРОГНОЗУВАННЯ СТАНУ ЗДОРОВ'Я ПРАЦІВНИКА

Н. Г. Аксак, Н. М. Сердюк

Анотація. *Розроблено систему для надання дистанційних послуг на виробництві для контролю стану здоров'я співробітника. Система реалізована на основі запропонованої трирівневої архітектури для сервіс-орієнтованих систем, що характеризується поєднанням розподілених методів та засобів збору, зберігання, обробки різнорідних даних, що дозволяє використовувати результати віддаленого моніторингу при прийнятті рішень для своєчасного реагування при настанні екстреної ситуації. Одним з ключових компонентів системи є застосування моделі Гаммерштейна, що*

дозволило кількісно оцінити зміну стану здоров'я працівника при виконанні професійної діяльності підприємства.

***Ключові слова*:** *сервіс-орієнтована система, віддалений моніторинг, стан організму співробітника, модель Гаммерштейна.*

Abstract. *A system for the provision of remote production services to control the health of the employee has been developed. The system is implemented based on the proposed three-tier architecture for service-oriented systems, characterized by a combination of distributed methods and means of collecting, storing, processing heterogeneous data, which allows to use the results of remote monitoring in making decisions for timely response in case of emergency. One of the key components of the system is the application of the Hammerstein model, which allowed us to quantify the change in the health of the employee while performing the professional activity of the enterprise.*

Keywords: *service oriented system, remote monitoring, employee body condition, Hammerstein model.*

ГЛАВА 9
АВТОМАТИЗОВАНЕ АНОТУВАННЯ ЕЛЕКТРОННИХ ТЕКСТІВ З ВИКОРИСТАННЯМ МЕТОДУ ЗВ'ЯЗНИХ СЛОВОФОРМ

С. Г. Удовенко, Л. Е. Чала

Анотація. *Запропоновано та програмно реалізовано метод формування контекстних множин ключових слів та зв'язних словоформ, призначений для подальшого автоматизованого анотування електронних текстових документів. Метод базується на використанні моделі семантичного стиснення текстів та модифікованого алгоритму Гінзбурга, що дозволяє формувати матрицю зв'язків між словоформами анотації. Наведено результати тестування розробленого методу на прикладі анотування електронних методичних матеріалів. Отримана анотація може потребувати часткової корекції в разі наявності синтаксичних або семантичних помилок, але є досить точним описом основних положень сенсу аналізованого тексту.*

***Ключові слова*:** *анотування текстів, зв'язні словоформи, семантичне стиснення, модифікований метод Гінзбурга, програмний модуль*

Abstract. *A method for forming contextual sets of keywords and related word forms, intended for further automated annotation of electronic text*

documents, is proposed and implemented. The method is based on the use of the semantic text compression model and the modified Ginsburg algorithm, which allows to form a matrix of links between the word forms of the abstract. The results of testing of the developed method on the example of annotation of electronic methodological materials are given. The resulting abstract may need partial correction in the case of syntax or semantic errors, but it is a fairly accurate description of the main meaning of the text being analyzed.

Keywords: *text annotation, related word forms, semantic compression, modified Ginsburg method, program module*

ГЛАВА 10
ПРОБЛЕМИ СИМПЛІФІКАЦІЇ МЕДИЧНИХ ТЕКСТІВ УКРАЇНСЬКОЮ МОВОЮ

О. Ю. Чередніченко, О. В. Янголенко

Анотація. *Медичні тексти характеризуються високою складністю та вживанням великої кількості специфічних термінів. Читачами медичних текстів не завжди є лікарі та медичний персонал. Потреба прочитати та зрозуміти медичний текст також часто виникає у людей, що не мають необхідного рівня підготовки для адекватного сприйняття викладеного матеріалу. Інструкції лікарських засобів, дані медичної карти, приписи лікарів, документація медичних приладів – всі ці документи найчастіше містять слова та словосполучення, які не обов'язково є зрозумілими для широкого загалу. Тому задача симпліфікації медичних текстів є достатньо актуальною, враховуючи кількість потенційних зацікавлених осіб та відсутність інструментів, які б її вирішували щодо текстів українською мовою. Дана робота аналізує підходи до оцінки складності текстів. В якості прикладу розглянуто тексти медичних протоколів України, проаналізовано термінологію, яка в них вживається. Зроблено висновок, що складність тексту залежить від частоти вживаних у ньому термінів. Разом з тим, аналіз текстів українською мовою вимагає розробки нових програмних інструментів, що мають враховувати мовні особливості.*

Ключові слова: *медичний текст, медичний протокол, складність тексту, оцінка складності, симпліфікація тексту, частота вживання слів.*

Abstract. *Medical texts are characterized by high complexity and use of a large number of specific terms. The readers of medical texts are not always*

doctors and medical staff. The need to read and understand the medical text also often arises among people who do not have the necessary level of training to adequately perceive the material presented. Medication instructions, medical records, doctors' prescriptions, medical device documentation - all of these documents most often contain words and phrases that may not be understood by the general public. Therefore, the task of simplifying medical texts is quite relevant taking into account the number of potential stakeholders and the lack of tools for Ukrainian language. This paper analyzes approaches to assessing the complexity of texts. The texts of the medical protocols of Ukraine are considered as an example, the terminology used in them is analyzed. It is concluded that the complexity of the text depends on the frequency of terms used in it. However, the analysis of texts in Ukrainian requires the development of new software tools that must take into account the language features.

Keywords: *medical text, medical protocol, text complexity, estimate of text complexity, text simplification, terms usage frequency***.**

НАУКОВЕ ВИДАННЯ

ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ТА СИСТЕМИ

Монографія

під ред. д.е.н., проф. Пономаренка В. С.

Відповідальний за випуск **Бурдаєв В. П.**

Відповідальний редактор **Грабовський Є. М.**

Підп. до друку 2.05.2020 Формат 60 × 90 1/16. Папір. Друк офсетний.
Ум.-друк. арк. 13,25 Обл.-вид. арк. Тираж 150 екз. Зам. №
Свідоцтво про внесення до Державного реєстру суб'єктів видавничої справи
Видавець і виготівник – ПП «Стиль-іздат»