МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
«ЗАПОРІЗЬКА ПОЛІТЕХНІКА»

ТИЖДЕНЬ НАУКИ-2021.
Факультет комп'ютерних наук і технологій

Збірник тез доповідей щорічної
науково-практичної конференції викладачів, науковців,
молодих учених, аспірантів та студентів

19–23 квітня 2021 року

Електронне видання на DVD-ROM

м. Запоріжжя

УДК 004(06)
Т39

Рекомендовано до видання Вченою радою
Національного університету «Запорізька політехніка»
(Протокол №10/21 від 12.04.2021 р.)

Упорядник Голуб Т.В.

Тези доповідей друкуються методом прямого відтворення тексту, представленого авторами, які несуть відповідальність за його форму і зміст.

Т39 **Тиждень науки-2021. Факультет комп'ютерних наук і технологій.** Тези доповідей науково-практичної конференції викладачів, науковців, молодих учених, аспірантів та студентів, Запоріжжя, 19–23 квітня 2021 р. [Електронний ресурс] / Редкол. :В. В. Наумик (відпов. ред.) Електрон. дані. – Запоріжжя : НУ«Запорізька політехніка», 2021. – 1 електрон. опт. диск (DVD-ROM); 12 см. – Назва з тит. екрана.
ISBN 978-617-529-317-1.

Зібрані тези доповідей, заслуханих на щорічній науково-практичній конференції викладачів, науковців, молодих учених, аспірантів та студентів. Збірка відображає широкий спектр тематики наукових досліджень, які проводяться на факультеті комп'ютерних наук і технологій Національного університету «Запорізька політехніка». Збірка розрахована на широкий загал дослідників та науковців.

ISBN 978-617-529-317-1.

ЗМІСТ

УДК 004.42:535.361
Точилін С.Д.[1]
[1]канд. фіз.-мат. наук, доц. НУ «Запорізька політехніка»

КОМП'ЮТЕРНА ПРОГРАМА ДЛЯ АНАЛІЗУ РІЗНИЦЕВИХ СПЕКТРІВ ВТОРИННОГО ВИПРОМІНЮВАННЯ

Останнім часом при оптичних дослідженнях для кількісного порівняння об'єктів вивчення використовують різницеві спектри (РС) $I_X^E(\lambda)$ [1], які визначають за допомогою виразу:

$$I_X^E(\lambda) = 1 - \left| I_X(\lambda) - I_E(\lambda) \right|, \tag{1}$$

де λ - довжина хвилі, $I_E(\lambda)$ та $I_X(\lambda)$ - нормовані спектри вторинного випромінювання (СВВ) для еталонної речовини (E) і речовини, яку аналізують (X), відповідно.

Оцінка ідентичності або відмінності об'єктів дослідження здійснюється за допомогою коефіцієнта відповідності [1]:

$$K = \frac{1}{N}\sum_{i=1}^{N}\left(1 - \left| I_X(\lambda) - I_E(\lambda) \right|\right)_i = \frac{1}{N}\sum_{i=1}^{N} I_X^E(\lambda_i), \tag{2}$$

де N - число замкнутих інтервалів $\Delta\lambda$ у спектральному діапазоні довжин хвиль від λ_I до λ_{II} із шириною B, $B = \lambda_{II} - \lambda_I$, $N = B/\Delta\lambda$.

Згідно (2), при $\Delta\lambda \to 0$, коефіцієнт K має значення:

$$K = \frac{1}{B}\int_{\lambda_I}^{\lambda_{II}} I_X^E(\lambda)\,d\lambda. \tag{3}$$

У даній роботі, з допомогою мови програмування JavaTM, для аналізу РС була розроблена комп'ютерна програма ST_DS. Програма мала графічний інтерфейс користувача. За допомогою меню «File» СВВ еталонної та речовини, яку аналізують, завантажувалися з *.csv файлів і відображалися у вигляді графіків на панелі додатка. Для ініціалізації спектрального діапазону формування РС використовувалися поля вводу «Low value, nm» і «High value, nm». У той час при натисканні на кнопку «Calculate» програма ST_DS розраховувала різницевий спектр. Потім визначала коефіцієнт K за

допомогою виразу (3). Крім того, розраховувала мінімальне I_{min} та максимальне I_{max} значення $I_X^E(\lambda)$. При цьому РС відображався на панелі додатка в графічному вигляді, а його дані поміщалися в таблицю додатка. У той час значення K, I_{max} і I_{min} з'являлися в полях вводу «К», «I max» і «I min», відповідно.

На рис.1, як приклад, зображене вікно програми ST_DS з $I_X^E(\lambda)$, K, I_{max} і I_{min} при порівнянні спектрів фотолюмінесценції (ФЛ) таблеток медпрепаратів - дротаверина (еталон) та дипіридамола.

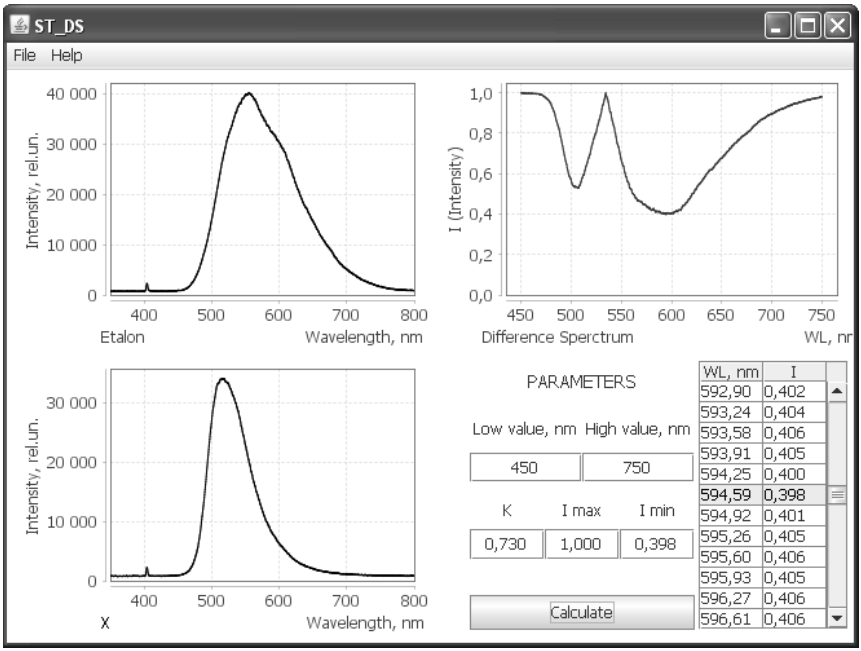


Рисунок 1 – Вікно програми ST_DS при розрахунку РС, K, I_{max} і I_{min}

ФЛ збуджувалася випромінюванням фіолетового лазера, яке попередньо проходило через інтерференційний світлофільтр (СФ), з максимумом пропущення на довжині хвилі 405 нм. Її сигнали, пройшовши СФ ЖЗС5, реєструвалися спектрометром ВТС-110S.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Умаров М.Ф. Оптическая спектроскопия биоактивных препаратов / М.Ф. Умаров, В.С. Горелик. - Вологда: ВоГУ, 2014. - 147 с.

УДК 004.04
Голуб Т.В.[1], Зеленьова І.Я.[2]
[1] старш. викл. НУ «Запорізька політехніка»
[2] доц. НУ «Запорізька політехніка»

СКОРОЧЕННЯ ЧАСОВИХ ВИТРАТ НА ОБРОБКУ ІНФОРМАЦІЇ

Із збільшенням кількості інформації, що потребує обробки та аналізу, підвищуються вимоги до розрахункових можливостей апаратних ресурсів, на які покладено вирішення цієї задачі. При цьому очевидно, що збільшення обсягу вхідної інформації при сталих розрахункових ресурсах супроводжується зростанням часових витрат на даний процес.

При наявності обмежень збільшення можливостей апаратних ресурсів логічно виникають умови для необхідності вдосконалення певних етапів обробки інформації.. При чому вдосконалення має проводитися з акцентом на скорочення часових витрат як безпосередньо, так і опосередковано шляхом зменшення кількості розрахунків. Одним із шляхів вдосконалення є також реалізація програмного та/або апаратного розпаралелювання даного процесу. Програмне розпаралелювання, реалізоване на базі розрахункової машини, вимагає перерозподілу з виділенням додаткових ресурсів на вирішення даної задачі. Апаратне розпаралелювання, при своєму незначному збільшенні початкових витрат на реалізацію цього рішення, дозволяє суттєво прискорити процес обробки вхідної інформації.

Прикладом обробки інформації з метою її подальшого аналізу є класифікація текстів, що представляє собою складний процес (рис.1). Одночасно з тим, що на кожному його етапі можна досягти певного скорочення часових витрат, перенесення частини розрахункового навантаження на апаратну платформу дозволяє досягти значних результатів. В якості апаратної платформи можна використати FPGA (Field-Programmable Gate Array) [1].

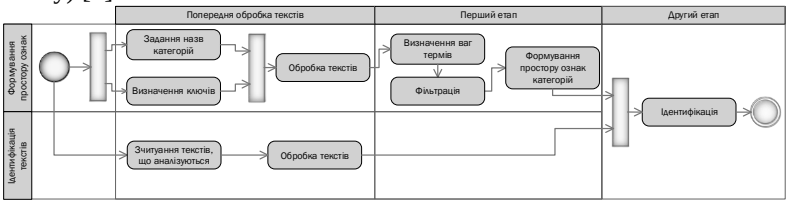


Рисунок 1 – UML-діаграма дій процесу класифікації текстів

Зокрема FPGA надає можливість паралельного виконання однотипних дій над різними вхідними даними. Апаратне розпаралелювання може бути використане як на етапі ідентифікації категорії, так і на кроках видалення

«стоп-слів» та стемінга слів. Це доцільно у випадку, коли зазначені етапи також реалізуються апаратно на тому ж кристалі FPGA.

UML-діаграма розгортання ресурсів класифікатору, реалізованого на зазначених принципах, наведена на рис. 2.

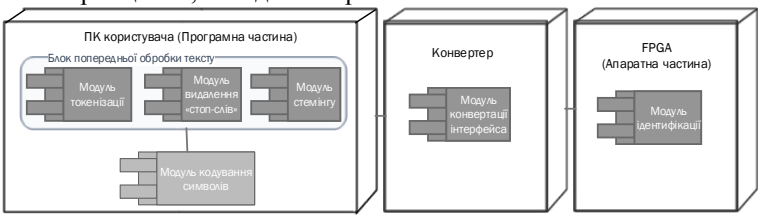


Рисунок 2 - UML-діаграма розгортання програмно-апаратного комплексу

Для даного класифікатору було розроблено різні засоби вдосконалення процесу на кожному із основних етапів. Використання окремо кожного із розроблених засобів на відповідному етапі класифікації текстів призводить прямо або опосередковано до скорочення часових витрат, а комплексне їх застосування дозволяє досягти ще більшого ефекту.

Результати застосування розроблених засобів наступні:

а) в рамках програмної частини:

1) в результаті модифікації алгоритму стемінга досягнуто зменшення часу на стемінг на 22%;

2) при фільтрації простору ознак категорій досягнуто зменшення зазначеного простору в середньому на 24%;

б) в рамках апаратної частини: паралельна організація схеми блоку ідентифікації текстів на основі апаратної реалізації забезпечила можливість прискорення процесу класифікації в *n* разів.

Комплексне використання засобів скорочення часових витрат на основі програмної та апаратної частин дозволяє досягти зменшення часу класифікації текстів на 32%.

Таким чином, на прикладі виконання класифікації текстів було показано, що використання комплексного підходу до обробки вхідної інформації із аналізом і вдосконаленням кожного з кроків, що потребують виконання, дозволяє досягти значного скорочення часових витрат певного складного процесу в цілому.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

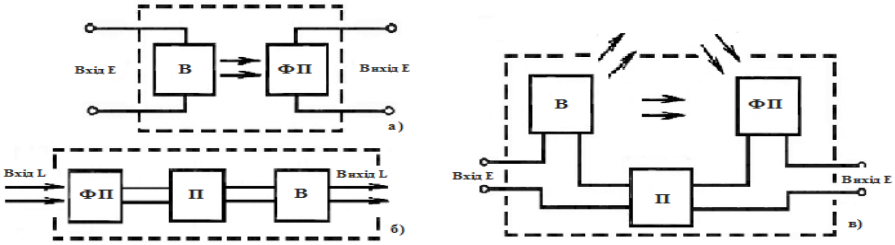
1. Програмно-апаратний спосіб прискорення процесу класифікації текстових документів / Т.В. Голуб, І.Я. Зеленьова, С.С. Грушко, М.А. Павлішин, А.О. Котенко // міжнар. наук.-практ. конф.: «Technical sciences: history, the present time, the future, EU experience», 27-28 september, 2019: тези доп., - Wlolawek, Republic of Poland, 2019. – P. 90-93.

УДК 671.327.1.
Проскурін М.П.
к.т.н., доц. каф. КСМ НУ «Запорізька політехніка»

ЕЛЕКТРОННІ І ОПТОЕЛЕКТРОННІ ЛОГІЧНІ СХЕМИ: ПОРІВНЯННЯ ЇХ СХЕМОТЕХНІЧНОЇ ПОБУДОВИ

Відомі переваги сучасних способів оптичного зв`язку (ОЗ) [1] над традиційними (за допомогою дротів) для передавання швидкісних цифрових потоків (ЦП) у вигляді використання: волоконно оптичних каналів (ВОК) і хвилеводів [2] на поверхні підкладок у складі фотонних інтегральних схем (ФІС) [3]; підвищення їх продуктивності за рахунок втілення WDM технологій [4]; оптичної взаємодії об`єктів у космічному середовищі (в разі відповідного розташування їх у полі локації) когерентними джерелами випромінювання [5].

Однак є ще один напрямок, який стосується деяких переваг при створенні і порівнянні схемотехніки логічних схем (ЛС) електронної і оптоелектронної логіки (типу: nАБО, nАБО-НІ; виключне nАБО, nАБО-НІ, ін.), які створюють двійкові базиси (відповідно: Пірса і Жегалкіна) і (разом із базисами nI-HI-nАБО Буля та nI-HI Шеффера) лежать в основі побудови комбінаційних схем (КС) - найбільшою складової частини обчислювальних цифрових пристроїв (ЦП). Щоб зрозуміти про які переваги йде мова, розглянемо типи спрощених моделей оптронів (рис.1 а, б, в), які широко використовуються і складають основу ОЗ.



а) для електричної (гальваничної) роз`язки; б) з оптичним входом і виходом; в) регенеративний. Символами E (Electrical) та L (Light) означено їх відповідні входи і виходи, а стрілками – напрям оптичних (L) зв`язків
Рисунок 1 – Типи оптронів.

На рис.1а-в зображено функціональні різновиди оптронів. У склад пристроїв з їх використанням можуть входити додатково електронні, оптичні, оптоелектронні, механічні елементи для виконання заданої функції. Оптичні,

електричні зв'язки мають в них довільний вид (прямі, зворотні, позитивні, негативні, зовнішні, внутрішні, їх комбінації). У оптронів по рис.1а - вхідні, вихідні зв'язку електричні, а між випромінювачем (В) і ФП - оптична; по рис. 1б оптичний зв'язок між В та ФП розірваний, але присутній електричний зв'язок. Він може бути використаний, як перетворювач сигналів - оптоелектронний підсилювач спектру (наприклад, некогерентного випромінювання в когерентне або навпаки, ін.). Пристрій по рис. 1в з електричними і оптичними зв'язками носить назву регенеративного оптрона. За рахунок зворотного зв'язку між В і ФП на його ВАХ є ділянки з від`ємним опором (з можливістю часткового або повного відновлення вхідного сигналу, ін.). Різновидом оптрона з малою протяжністю ВОК (0,001...10 мм) є оптопара, функціонально - це елемент гальваничної розв'язки (є оптичний, відсутній – електричний зв'язок між В і ФП, рис. 1.5а).

Розглянемо деякі електронні схеми (на прикладі діодно-резистивної логіки, ДРЛ, елемент nАБО, рис. 2а-в [6]) і оптоелектронну (елемент nАБО, рис. 3 [7]), які обидві доповнено інвертором (справа) і формально порівняємо їх схемотехнічні рішення. Додавання інвертора в означені схеми розширює їх до мінімального базису nАБО-НІ (базис Пірса).

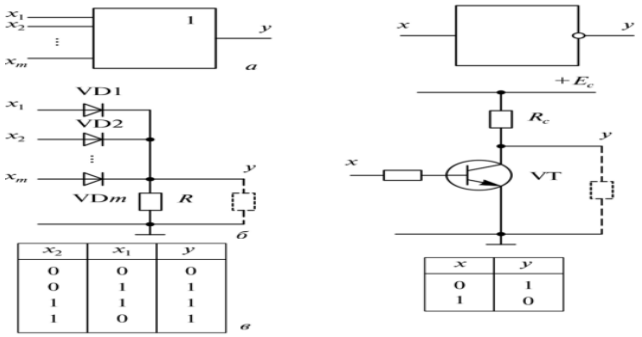


Рисунок 2 - Елемент nАБО-(НІ) на схемі ДРЛ з m вхідними діодами. Справа додано інвертор.

Як виходить з порівняння кількості елементів для електронної і оптоелектронної схем, остання має суттєві переваги, бо функція «монтажне nАБО» не потребує активних компонентів (дорівнює нулю) і формується виключно пасивним чином фактично шляхом механічного з`єднання Y-подібних розгалужувачів. Схема ДРЛ nАБО-(НІ) рис.2 наведена, як приклад і зараз не використовується в цифрових ІС (вхідні 1,2…m діоди замінено на багатоеміттерні транзистори, ін.), від чого вона стає ще складнішою.

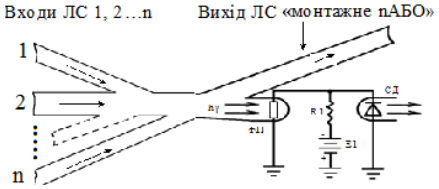


Рисунок 3 – Ескіз оптоелектронної схеми nАБО-(НІ) «монтажне nАБО» (зліва; виконано на Y-подібних розгалужувачах, які з`єднано з одним фотодіодом - ФД). Справа додано інвертор.

Відомий російський вчений, д.т.н., проф. Шустов М.А. (є автором багатьох робіт, печатних видань, підручників, ін. з цифрової схемотехники), останній час приділив увагу саме використанню оптичних зв`язків, оптронів при формуванні як елементарних логічних схем (ЛС), так і деяких типів комбінаційних схем (КС) [8-10]. Порівняльний аналіз їх схемотехнічних якостей з електронними аналогами (по типу наведеного на рис. 2, 3) також вказує на суттєві переваги при використанні оптоелектронних компонентів.

Висновок. Порівняння елементарних КС типу nАБО-(НІ) у вигляді електронного (рис. 2а-в) і оптоелектронного (рис. 3) варіантів вказує на те, що останні можуть бути виконані з меншою кількістю елементів і з`єднань, а частина функціональних електричних зв`язків може бути доповнена/змінена на оптичні. Вказані переваги може бути реалізована при створенні як елементарних ЛС, так і в побудові більш складних КС (шифратори/ дешифратори, мультиплексори/демультиплексори, компаратори, ін.). Це пояснюється тим, що крім звичайного «електронного» компонента перетворення/взаємодії сигналів з елементами КС на активних перетворювачах (діоди, транзистори, їх комбінації) і пасивних елементах (резистори, з`єднувальні дроти) до електронної дії додається оптична складова (рис. 1в) і/або їх комбінації (електронна і оптична), що забезпечує меншу кількість елементів у схемах і підвищує їх швидкодію.

СПИСОК ВИКОРИСТАННОЇ ЛІТЕРАТУРИ.

1. Функциональные элементы и устройства оптоэлектроники / В.П. Кожемяко, Л.И. Тимченко, Г.Л. Лысенко, Ю.Ф. Кутаев. – Киев: УМК ВО, 1990. – 251 с.

2. Интегральный оптический волновод на основе пористого кремния. / В.П.Бондаренко, В.С.Вариченко, А.М. Дорофеев и др. - Письма в ЖТФ, т.19, в.14, – 1993г., – С.73-76.

3. Технологія Holey Optochip фірми IBM. Режим доступу: https://researcher.watson.ibm.com/.../view_group.php?id=70..

4. Технологія Light Peak фірми Intel Режим доступу: https://en.wikipedia.org/wiki/Thunderbolt_(interface)

5. НАСА: тест космической лазерной связи прошел успішно. Режим доступу: https://www.osp.ru/news/articles/2014/24/13041668

6. Логические элементы и логические схемы на транзисторах Режим доступу: https://studref.com/532624/tehnika/logicheskie_elementy_logicheskie_shemy_tranzistorah

7. Проскурін М.П. Швидкодіючий універсальний оптоелектронний логічний елемент НАБО-НІ типу SUPROSTD / Проскурін М.П, Проскурін В.П., Дериведмідь В.М., декл. патент №32921А, Заявка №98073980; Заявл. 21.07.98; Опубл. 15.02.2001, Бюл.№1.- 4 с.

8. Шустов М.А. Оптоэлектронные элементы «Эквивалентность» и «Эквивалентность-НЕ»/ М.А.Шустов //Ж.Радиолоцман, № 07, 08-2020, – С.50-52.

9. Шустов М.А. Оптоэлектронные логические элементы 3XOR/3XNOR » / М.А.Шустов // Ж. Радиолоцман, № 09, 10-2020, – С.52-53.

10. Шустов М.А. Оптоэлектронный цифровой компаратор и аналитический цифровой компаратор-дешифратор / М.А.Шустов // Ж. Радиолоцман, №11, 12-2020, – С.25-27.

УДК 004.75
Дьячук Т.С.[1]
[1]асист. каф. НУ «Запорізька політехніка»

НЕВЗАЄМОЗАМІННІ ТОКЕНИ ЯК ЧАСТИНА ЦИФРОВОЇ ЕКОНОМІКИ

Невзаємозамінні токени (NFT, non-fungible token) [1] — це цифровий сертифікат, який представляє деякий унікальний об'єкт. NFT грає одну з ключових ролей у новій цифровій економіці, заснованої на блокчейні. NFT можуть являти собою власність на цифрові або фізичні активи, будучи різновидом криптографічних токенів. Головна особливість і відмінність в тому, що кожен екземпляр унікальний, має свою ідентифікацію: їх не можна обміняти або замістити іншим аналогічним токеном, хоча зазвичай за своєю природою токени взаємозамінні.

Права на оригінал належать тільки покупцеві NFT, при цьому інші копії все одно можуть вільно циркулювати в інтернеті. Звіти про покупку будь-якого NFT, крім того, що відображають права власності, відстежують його історію: кожен NFT має метадані, які засвідчують, коли він був зроблений, хто його створив та інші параметри. В даний час, коли людина набуває NFT,

актив є власністю, але не авторським правом. Теоретично правовласник може окремо передати авторські права, але за окремим контрактом, дотримуючись права інтелектуальної власності.

Торгівля NFT-токенами ведеться на спеціально створених цифрових маркетплейсах. Найпопулярніші з них OpenSea [1], Rarible [2], Zora, Mintbase. Є й нішеві майданчики, наприклад, для продажу контенту (Mirror, Audius, EulerBeats), предметів мистецтва (SuperRare, Async Art, Known Origin, Nifty Gateway та інші). Тут можна стати власником цифрового варіанту зображення, музичного треку, скульптури, доменного імені або ж персонажа улюбленої комп'ютерної гри.

Виділимо найбільш перспективні сфери застосування NFT:

1. Цифрове-мистецтво. Попит на цифрові картини зростає з кожним роком. Цінність твору визначається культурним контекстом навколо нього. При поширенні копій роботи, про неї буде дізнаватися все більше і більше людей, а культурна цінність роботи буде рости. Блокчейн фіксує цю цінність та її приналежність, так як дозволяє підтвердити право володіння і оригінальність будь-якої роботи. Покупець, таким чином підкреслює свій статус в цифровому світі, що є аналогією з членством в елітному клубі;

2. Комп'ютерні ігри. Перспективним напрямком у розвитку NFT-ігор представляється інтероперабельність, що дозволяє використовування предметів або персонажів в різних ігрових середовищах за межами тієї ігровий «пісочниці», де цей персонаж або предмет були створені;

3. Шоу-бізнес. Токен, як сертифікат, гарантує справжність і оригінальність альбому виконавця. Також токени можуть стати своєрідним цифровим автографом з обмеженою пропозицією;

4. Спорт. Токенізація спортивних досягнень спортсменів, картки спортсменів, автографи - все це буде цінністю для шанувальників;

5. Smart-токени (наприклад, доменні імена). NFT-токени вже почали входити в сферу нерухомості. Вже активно продаються преміум-домени на основі блокчейна;

6. Предмети колекціонування. Якщо людина хоче продати на аукціоні колекційний товар - з цим завданням чудово впораються NFT-токени: емітент може включити інформацію про майно, що продається в опис свого токена, доступне для перегляду в блокчейні;

7. Освіта. Проблема підробки дипломів, сертифікатів та інших документів легко вирішується за допомогою випуску NFT-токенів, які відображали б необхідну інформацію в блокчейні. ВНЗ можуть раз на рік (а будь-які інші державні установи - в міру необхідності) випускати NFT-токени, за якими будуть закріплені всі ідентифікаційні дані.

Для випуску невзаємозамінних токенів в контексті Ethereum, найбільшої NFT-екосистеми, потрібні:

- Ethereum-гаманець наприклад MetaMask, де є підтримка ERC-721 та інших стандартів невзаємозамінних токенів;
- деяка сума в ETH для оплати транзакційних комісій.

Існує безліч маркетплейсов, що дозволяють підключити гаманець і завантажити файл для його перетворення в NFT. Одні з найбільш відомих платформ: OpenSea [1], Rarible [2], Mintable. На OpenSea та багатьох інших маркетплейсах можна налаштовувати різні параметри. Наприклад, автори можуть включити опцію «контент, що розблоковується», подивитися який можуть лише покупці.

Таким чином, сучасні IT-фахівці для побудови успішної кар'єри повинні не тільки розбиратися в основах і тенденціях цифрової економіки таких як блокчейн, цифрові валюти, NFT та і використовувати ці знання у своїй роботі. Представляючи фізичні активи в цифровому світі, NFT токени можуть стати важливою частиною не тільки блокчейн-екосистеми, але і економіки в цілому.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. The Non-Fungible Token Bible: Everything you need to know about NFTs – [Електронний ресурс] Режим доступу: https://opensea.io/blog/guides/non-fungible-tokens/

2. rarible.com FAQ – [Електронний ресурс] Режим доступу: https://www.notion.so/rarible-com-FAQ-a47b276aa1994f7c8e3bc96d700717c5

УДК 681.5.01

Єфименко М.В.[1], Луценко Н.В.[2]
[1] гол. конструктор НВП "Хартрон-Юком", д.т.н., доцент
[2] старш. викл. НУ "Запорізька політехніка"

КВАТЕРНІОНІ МОДЕЛІ В ЗАДАЧАХ КЕРУВАННЯ ОРІЄНТАЦІЄЮ КОСМІЧНОГО АПАРАТА

Успіх у вирішенні завдань керування кутовим рухом КА багато в чому залежить від обраної моделі кутового руху КА. В даний час найпоширенішою є модель, в якій динаміка описується рівнянням Ейлера (1), а кінематика - кинематичним рівнянням в параметрах Родріга-Гамільтона (2):

$$J\dot{\omega}+\omega\times J\omega=M_u\,, \tag{1}$$

$$\dot{\Lambda}=\frac{1}{2}\begin{bmatrix}0 & -\omega^T\\ \omega & -\Phi(\omega)\end{bmatrix}\cdot\Lambda\,, \tag{2}$$

де ω – абсолютна кутова швидкість обертання КА; J – момент інерції КА; Mu – керуючий момент; Λ – вектор параметрів Родріга-Гамільтона (нормований кватерніон), що визначає кутове положення КА в просторі; $\Phi(\omega)$ – лінійний кососиметричний оператор векторного добутку, що визначається рівністю $\Phi(\omega)\square y = x \times y$.

Переваги моделі – відсутність обчислювальних особливостей і мінімальна надмірність вектора стану. Недоліки – нелінійність моделі, що істотно ускладнює синтез законів керування. Задачу синтезу можна істотно спростити, якщо при побудові керування в якості моделі кутового руху КА використовувати модель, побудовану на основі динамічних рівнянь обертального руху твердого тіла в параметрах Родріга-Гамільтона (3). У цьому випадку в якості компонентів вектора стану КА використовуються параметри Родріга-Гамільтона і їх похідні [1-4].

$$\ddot{\Lambda} = (I_4 - \Lambda\Lambda^T)U - \left\|\dot{\Lambda}\right\|^2 \Lambda . \qquad (3)$$

У зазначених роботах математична модель обертального руху КА є векторне диференціальне рівняння і описується із застосуванням апарату векторно-матричного обчислення. Параметрам Родріга-Гамільтона можна поставити у відповідність гіперкомплексне число – кватерніон зі скалярною частиною $\lambda 0$ і векторної частиною λ :

$$\Lambda = \lambda_0 + \lambda ; \qquad \lambda = (\lambda_1 \quad \lambda_2 \quad \lambda_3)^T .$$

У доповіді розглянуті питання опису обертального руху твердого тіла з застосуванням кватерніонів і отримані кватерніоні диференціальні рівняння для опису, яких використовується тільки кватерніоне обчислення. На прикладі рішення задачі стабілізації та задачі термінальної переорієнтації показано, що застосування розроблених моделей істотно спрощує задачу синтезу керування, яка в цьому випадку зводиться до синтезу керування для системи, що є сукупністю інтегруючих ланок другого порядку. У багатьох випадках для таких систем задачу синтезу має аналітичне рішення.

Отримані при цьому алгоритми керування реалізуються значно простіше, ніж алгоритми, отримані при використанні традиційної моделі

Запропонований підхід пройшов експериментальну перевірку в системі керування КА «Egyptsat-1» і показав свою високу ефективність (рис. 1,2).

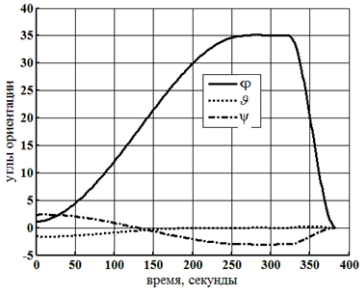


Рисунок 1 - КА«Egyptsat-1». Трасова зйомка. Кути орієнтації

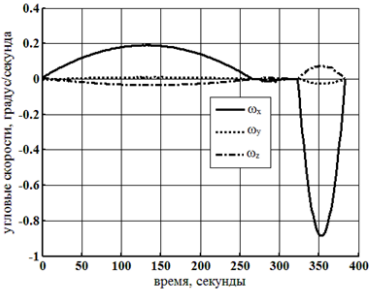


Рисунок 2 - КА«Egyptsat-1». Трасова зйомка. Кутові швидкості

Робота може бути корисною для фахівців, що займаються створенням систем керування КА.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Ефименко Н.В. Синтез алгоритмов управления пространственной пере-ориентацией космического аппарата с использованием динамических уравнений вращательного движения твердого тела в параметрах Родрига-Гамильтона / Н.В. Ефименко // Проблемы управления и информатика. 2015. № 3. С. 145–155.

2. Ефименко Н.В. Синтез оптимального по времени пространственного разворота космического аппарата с использованием динамического уравнения вращательного движения твердого тела в параметрах Родрига-Гамильтона / Н.В. Ефименко // Проблемы управления и информатика. 2017. № 3. С. 109–128.

3. Ефименко Н.В. Математическая модель углового движения космического аппарата в параметрах Родрига-Гамильтона и ее свойства / Н.В. Ефименко // Электронное моделирование. 2018. Т.4. № 6. С. 21–36.

4. Ефименко Н.В. Управление переориентацией космического аппарата посредством маховиков // Проблемы управления и информатики. 2008. №5. С. 121-128.

УДК 004.056.5
Левченко А.М. [1], Паромова Т.О. [2], Луценко Н.В.[2]
[1] студ. гр. КНТ-528 НУ «Запорізька політехніка»
[2] старш. викл. НУ «Запорізька політехніка»

АНАЛІЗ КРИПТОСТІЙКОСТІ АЛГОРИТМУ ШИФРУВАННЯ RSA

Сьогодні кожен з нас зустрічається з необхідністю захисту даних, бо зростає обсяг конфіденційної інформації в мережі. Шифрування забезпечує захист від випадкового або навмисного втручання, що може стати причиною втрати даних або їх несанкціонованої зміни. Тому забезпечення високої криптостійкості є першочерговим завданням при створенні надійної та безпечної системи.

В основі криптографічного алгоритму RSA (Rivest–Shamir–Adleman) лежить факторизація великих цілих чисел, що є однобічною функцією з потайним входом. Це така функція, що легко обчислюється в одному напрямку, але важко обчислюється у зворотному без спеціальної інформації (секрету), тому злом методом «грубої сили» потребує значні обчислювальні потужності. Криптосистема RSA стала першою системою, придатною і для шифрування, і для цифрового підпису.

На рис.1 зображено прогрес розв'язання задачі факторизації цілих чисел, що є ключем для злому алгоритму RSA, де вісь OY показує довжину ключа, а OX – це вісь часу. Пряма лінія – це функція, що демонструє приблизне зростання вимог до довжини ключа при збільшенні обчислювальних потужностей комп'ютерів з кожним роком. Кружечками позначені криптоаналізи науковців алгоритму RSA, а зірочками позначені відомі спроби вдалих атак [1].

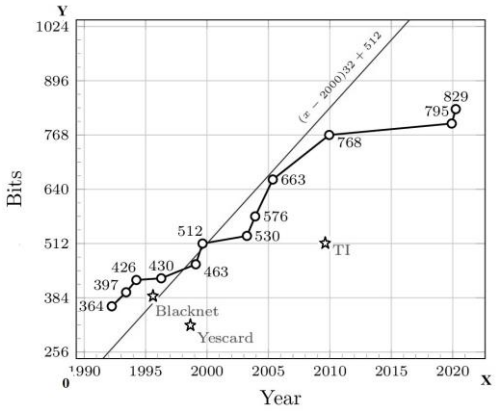


Рисунок 1 – Прогрес факторизації цілих чисел

Поточним рекордом факторизації цілих чисел є 829-бітний RSA-250, що був обрахований наприкінці лютого 2020 року. Загальний час обчислень становив приблизно 2700 базових років при використанні Intel Xeon CPUs Gold 6130 як еталон (2,1 ГГц) [2]. Тобто на даний час неможливо обчислити настільки великі обсяги даних без використання суперкомп'ютерів. Але навіть з огляду на цей факт час для злому залишається дуже великим, тому завжди є можливість збільшення криптостійкості алгоритму за допомогою збільшення довжини ключа.

Одним з напрямків для криптоаналізу алгоритму RSA є використання квантових комп'ютерів. У 1994 році Пітер Шор розробив алгоритм, що дозволяє, використовуючи квантовий комп'ютер, обчислити факторизацію цілого числа M за час $O(log^3 M)$, використовуючи $O(log\ M)$ логічних кубітів. У 2001 році його працездатність була продемонстрована групою фахівців IBM. Число 15 було розкладено на множники 3 і 5 за допомогою квантового комп'ютера з 7 кубітами [3].

У 2015 році дослідники зробили висновок, що для досить швидкого злому 2048-бітного RSA шифрування квантовому комп'ютеру знадобиться мільярд кубітів. У 2019 році Крейг Гідні та Мартін Екеро визначили, що комп'ютер з 20 млн кубітів зможе впоратися з цим завданням всього за 8 годин. За прогнозами вчених такий квантовий комп'ютер можливо розробити через 25 років, але беручи до уваги той факт, що ці прогнози не можна вважати повністю вірними, то можливо очікувати такі машини раніше зазначеного терміну [4].

На цей час алгоритм RSA є криптостійким та може постійно удосконалюватися за допомогою збільшення довжини ключа. Квантові комп'ютери можуть стати значною загрозою у майбутньому для алгоритму, тому необхідно заздалегідь вжити необхідні заходи для запобігання криптографічних атак.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. How big an RSA key is considered secure today? [Електронний ресурс]. – Режим доступу: https://crypto.stackexchange.com/questions/1978/how-big-an-rsa-key-is-considered-secure-today

2. Factorization of RSA-250 [Електронний ресурс]. – Режим доступу: https://lists.gforge.inria.fr/pipermail/cado-nfs-discuss/2020-February/001166.html

3. Алгоритм Шора [Електронний ресурс]. – Режим доступу: https://ru.wikipedia.org/wiki/ Алгоритм_Шора

4. How a quantum computer could break 2048-bit RSA encryption in 8 hours [Electronic resource]. – Access mode: https://www.technologyreview.com/ 2019/05/30/65724/how-a-quantum-computer-could-break-2048-bit-rsa-encryption-in-8-hours/

УДК 004.92
Білоусов В.В.[1], Киричек Г.Г.[2]
[1] студ. гр. КНТ-518сп, НУ «Запорізька політехніка»
[2] канд. техн. наук, доц., НУ «Запорізька політехніка»

КОМП'ЮТЕРНА ГРАФІКА ТА СУЧАСНИЙ СВІТ

Перша комп'ютерна техніка не мала спеціальних засобів для роботи з графікою, але вже використовувалась для отримання і обробки зображень. Програмуючи пам'ять перших електронних машин, створену на основі матриці ламп, програмісти вже могли отримувати візуальне представлення об'єктів. Комп'ютерна графіка має велике значення, оскільки надає сучасному суспільству методи і програмні засоби представлення та візуалізації даних в графічному вигляді за допомогою програмно-апаратних засобів. Вона, на даний час, є найбільш швидкозростаючим сегментом в області інформаційних технологій [1]. Комп'ютерна графіка реалізує широкий спектр можливостей, починаючи з систем відеоспостереження і закінчуючи, розрахованими на велику кількість користувачів: онлайн іграми; переглядом відеоматеріалів; використанням таблиць, схем, графіків, діаграм та інших зображень [1, 2].

Метою дослідження є проведення аналізу основних видів та методів, які застосовуються в комп'ютерній графіці та визначення їх особливостей. Об'єктом дослідження є процес визначення основних методів та прийомів створення графічної інформації. Предметом – методи та програмні засоби реалізації візуального представлення інформації.

В залежності від методів, програмних засобів і прийомів створення графічної інформації виділяють наступні види комп'ютерної графіки: растрову, векторну, фрактальну та тривимірну (3D) [1].

Растрові зображення, зазвичай, створюються при скануванні кольорових ілюстрацій та фотографій, коли використовуються цифрові фото- і відеокамери, з застосуванням растрового графічного редактора. Вони формуються з пікселів різного кольору у матричному вигляді та забезпечують високу точність передачі кольорів і відтінків. Недоліком растрових зображень є їх великий розмір та погіршення якості зображення при масштабуванні. Переваги: можливість створення графіки будь-якої складності; швидке оброблення зображень та те, що растровий формат є природним для більшості пристроїв: моніторів, принтерів і сканерів. Найбільш розповсюдженими програмами для роботи з растровою графікою є Paint та Adobe Photoshop [2].

Редактори для роботи з векторною графікою, використовуються для створення і редагування зображень з чіткими контурами. Вони дозволяють зображувати не тільки плоскі, а й об'ємні об'єкти: куля, циліндр, куб та ін.

Перевагами векторної графіки є можливість збільшення об’єктів без втрати якості та те, що векторні зображення займають невеликий обсяг пам'яті. Недоліками є неможливість отримати реальне зображення фотографічної якості. Найбільш розповсюджені векторні графічні редактори: Corel Draw, Adobe Illustrator та AutoCAD [2, 3].

Фрактальна графіка – вид комп'ютерної графіки, яка застосовує самоподібні структури (фрактали). Вона відносно молода в порівнянні з растровою і векторною графікою, але найбільш швидко розвивається і є перспективним видом комп'ютерної графіки. Математичною основою фрактальної графіки є фрактальна геометрія. Вона застосовується при зображенні дерев, сніжинок та інших об'єктів, окремі елементи яких успадковують якості батьківських структур. Основними перевагами фрактальної графіки є малий розмір та будь-яка складність зображення; масштабованість та деталізація; можливість об’єднання складних форм з відносною простотою та висока якість зображення. Недоліками є: велике навантаження на обчислювальні модулі (використання досить складних математичних понять і формул) та абстрактність зображень. До програмних засобів можна віднести продукти Golden SoftWare (Surfer, Grapher, Map Viewer) та пакети Iris Explorer і Earth Watch [4].

Тривимірна графіка пов`язана з зображенням об'єктів у тривимірному просторі. Для отримання об'ємного зображення, вона може використовувати полігональне моделювання, сплайнове або NURBS моделювання. У першому випадку поверхню об'єкта представляють у формі простих двовимірних геометричних фігур – полігонів (багатокутників). У другому – модель створюється за допомогою сплайнів (тривимірні криві). При цьому лінії є тривимірним набором контрольних точок в просторі, які визначають гладкість кривої. Моделювання за технологією Non-Uniform Rational B-Spline використовує неоднорідні раціональні B-сплайни та створює плавні форми і моделі, які не мають гострих країв, як у полігональних моделей. Саме через це NURBS використовують для створення органічних об'єктів і моделей (людей, тварин та рослин). Основними перевагами є реалістичність; можливість використання тривимірних об'єктів для створення додатків та свобода трансформацій об'єктів. Недоліки: значний обсяг файлів; програмна залежність та висока вартість 3d редакторів. Програми для роботи з тривимірною графікою: Autodesk: 3d max; Maya; Softimage; Mudbox та ZBrush, Lightwave 3D, Modo, Cinema 4d [4, 5].

На даний час складно визначити, який із методів та прийомів комп`ютерної графіки краще забезпечує ту або іншу функціональність в процесі її використання. Кожен з них доводить свою життєздатність і необхідність. Всі вони мають характерні особливості і можливості, які

роблять їх незамінними в роботі. Знання особливостей, тонкощів кожної з цих технології важливо для сучасного дизайнера.

У процесі дослідження проведено опитування серед 60 випадкових людей різного віку. Результати опитування наведені на рисунку 1.

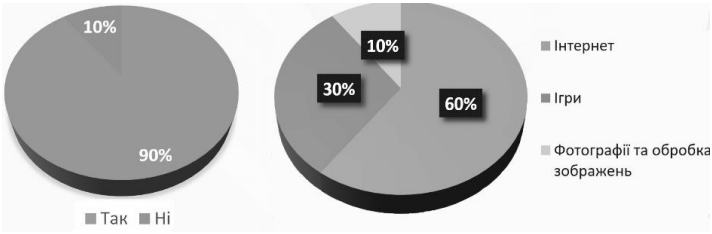


Рисунок 1 – Застосування комп'ютерної графіки

На питання «Чи використовуєте ви комп'ютерну графіку?» 54 опитуваних відповіли «Так», 6 - «Ні». Таким чином, робимо висновок, що кожна дев'ята людина постійно стикається з застосуванням різних методів та прийомів комп'ютерної графіки. При чому однозначним лідером у використанні комп'ютерної графіки є Internet, його використовує постійно 60 відсотків опитаних. Також, третина вибрала категорію комп'ютерних ігор, які найчастіше ними використовуються, а решта більш застосовує ці методи та прийоми у сфері фотографії.

Отже, усі області застосування, будь то інженерна чи наукова, бізнес або мистецтво є сферами використання комп'ютерної графіки. Тому вона є гігантською областю сучасних комп'ютерних знань, що означає збільшення необхідності широкого її використання та впровадження нових методів, прийомів та графічних програмних засобів різного призначення.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Quevedo, W. X. Virtual reality system for training in automotive mechanics /W. X. Quevedo, J. S. Sánchez, O.Arteaga, M. Álvarez, V. D. Zambrano, C. R. Sánchez, V. H. Andaluz //International Conference on Augmented Reality, Virtual Reality and Computer Graphics. – Springer, Cham, 2017. – P. 185-198.

2. Marschner, S. Fundamentals of computer graphics / S. Marschner, P. Shirley. – CRC Press, 2018. – 723 p.

3. Kirichek, G. Implementation quadtree method for comparison of images /G. Kirichek, V. Kurai //2018 14th International Conference on Advanced Trends in Radioelecrtronics, Telecommunications and Computer Engineering (TCSET). – IEEE, 2018. – P. 129-132.

4. Hildenbrand, D. Introduction to Geometric Algebra Computing: Computing with Circles and Lines /D. Hildenbrand. – CRC Press, 2020. – 193 p.

5. Aristidou, A. Inverse kinematics techniques in computer graphics: A survey /A. Aristidou, J. Lasenby, Y.Chrysanthou, A. Shamir //Computer Graphics Forum. – 2018. – vol. 37. – №. 6. – P. 35-58.

УДК 004.45

Касьян К.М.[1], Головіна К.П.[2]
[1] канд. техн. наук. НУ «Запорізька політехніка»
[2] студ. гр. КНТ-510м НУ «Запорізька політехніка»

ПОРІВНЯЛЬНИЙ АНАЛІЗ СУЧАСНИХ СИСТЕМ ВИЯВЛЕННЯ ВТОРГНЕНЬ

Метою є виявлення поширених методів виявлення мережевих атак і вибір найбільш ефективного на даний момент рішення. У роботі наводиться методика порівняння ПЗ систем виявлення вторгнень, обґрунтування вибору характеристик для порівняння і результати проведення порівняння на основі отриманої методики. В результаті порівняльного аналізу виявилося кілька кращих продуктів: ПЗ з відкритим вихідним кодом Snort і розробка Фонду Відкритої Інформаційної Безпеки Suricata

Мережеві атаки - неминуча загроза будь-якого пристрою, підключеного до мережі Інтернет. В результаті успішної реалізації такої атаки можливий несанкціонований доступ до комп'ютера, вихід з ладу сервера, витік даних. безумовно, це одна з головних проблем інформаційної безпеки.

Для запобігання подібних атак актуальним завданням стає виявлення спроби здійснення мережевої атаки. Клас програм, що вирішують цю задачу, отримав назву «система виявлення вторгнень (СВВ)».

Система виявлення вторгнень - програмне (апаратне, програмно-апаратний) засіб, що виявляє факт несанкціонованого, несанкціонованого доступу до пристрою. Стандартна архітектура СВВ включає наступні підсистеми:

— Сенсорна підсистема. Складається з «датчиків», які збирають інформацію про абітурієнта трафіку і стан об'єкта, що захищається.

— Підсистема аналізу. Отримує дані від сенсорної підсистеми і виносить вердикт про факт спроби мережевої атаки.

— Місце. Накопичує дані про події інформаційної безпеки, отриманих від підсистеми аналізу.

— Панель управління. Програмний модуль, дозволяє конфігурувати СВВ, спостерігати поточний стан системи, переглядати інциденти безпеки. [1]

Система виявлення вторгнень має кілька можливих виконань:

— Мережева СВВ. Відстежує атаки, аналізуючи мережевий трафік і спостерігаючи за декількома хостами. Отримує доступ до трафіку, будучи підключена до хабу або світчу. Одним з найяскравіших прикладів мережевої СВВ є open-source ПЗ Snort.

— СВВ, заснована на протоколі Protocolbased IDS. Аналізує комунікаційні протоколи зв'язку. У разі виконання на веб-сервері в основному аналізує HTTP і HTTPS з'єднання. При аналізі HTTPS з'єднання СВВ повинна мати доступ до трафіку до його шифрування і відправлення.

— СВВ, заснована на прикладних протоколах Application Protocol-based IDS. Аналізує специфічні для певного додатка протоколи. У разі захисту SQL бази даних аналізу піддаються всі SQL запити.

— Вузлова СВВ (Host-based IDS). Будучи розташованої на хості, відстежує вторгнення, аналізуючи системні журнали подій, логи додатків, модифікації файлів. Найбільш популярним програмним рішенням даного виду СВВ OSSEC.

— Гібридна СВВ. Поєднує два і більше підходів до визначення атак. Є найбільш кращим варіантом з огляду на створення найбільш повного уявлення про безпеку комп'ютерної мережі. Прикладом гібридного виконання СВВ є ПЗ Prelude. [2]

З класифікації можна зробити висновок: основним моментом порівняння ПЗ є вид виконання СВВ. Однак метод визначення атак в СВВ ключовим моментом в порівнянні.

Основні класи методів визначення мережевих атак:

— Кореляційні. Методи, засновані на машинному навчанні. Характеризуються здатністю виявляти не закладені в базу аномалії, але також наявністю високого рівня помилкових спрацьовувань.

— Сигнатурні методи. Засновані на порівнянні поточного стану мережі з шаблонами, наявними в базі сигнатур. Характеризуються відсутністю помилкових спрацьовувань, але також вкрай низькою ймовірністю визначення аномалії, що не міститься в базі.

СВВ також має два подання з режиму роботи: пасивний і активний. Визначення мережевої атаки - основне завдання пасивної СВВ. Активний режим роботи присутній у «системи запобігання вторгнень », яка веде захисні дії в разі виявлення мережевої атаки.

Наявність даного режиму в програмному продукті є безумовною перевагою. Крім специфічних для СВВ характеристик варто відзначити такий

важливий фактор для порівняння як кросплатформеність. [3] Таким чином, наведемо характеристики, по яким був здійснений порівняльний аналіз роботи СВВ:

— вид виконання СВВ;
— метод визначення атак;
— наявність активного режиму;
— кросплатформеність.

Для порівняння обрані 4 найпопулярніших програмних продуктів СВВ: Snort, OSSEC, Suricata, AIDE.

Підсумки порівняння програмних продуктів представлені в таблиці 1.

Таблиця 1 – Порівняльна таблиця програмних продуктів СВВ

СВВ	Вид виконання	Метод визначення атаки	Активний режим	Підтримувані ОС
Snort	Мережева	Сигнатурний зі зворотним зв'язком	+	GNU/Linux, Windows
OSSEC	Хостова	Сигнатурний метод, пошук аномалій	-	Кросплатформеність
Suricata	Мережева	Сигнатурний зі зворотним зв'язком	+	GNU/Linux, Windows
Bro (Zeek)	Гібридна	Граф переходів, відповідний атаці	-	Unix, GNU/Linux

Таким чином, найбільш популярним методом виявлення мережевих атак є клас сигнатурних методів зважаючи на низький рівень помилкових спрацьовувань. Всі розглянуті рішення мають підтримку під ОС Linux, також існують СВВ з підтримкою ОС Windows. Кращими показали себе продукти Snort і Suricata. Їх метод визначення атаки - сигнатурний зі зворотним зв'язком, що дозволяє поліпшувати результат визначення найбільш часто виникають атак.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Бірюков А.А. Інформаційна безпека. / А.А. Бірюков – М.: ДМК, 2017. – 434 с.

2. Система виявлення вторгнень. — Режим доступу: https://uk.wikipedia.org/wiki/Система_виявлення_вторгнень

3. Технології виявлення атак — Режим доступу: http://ypn.ru/448/intrusion-detection-technologies/

УДК 004.31

Діденко А.Є.[1], Зеленьова І. Я.[2]
[1]студ.гр. КНТ-527 НУ «Запорізька Політехніка»
[2] к.т.н., доц. каф. КСМ НУ «Запорізька Політехніка»

АНАЛІЗ МЕТОДІВ ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ МНОЖЕННЯ ДВІЙКОВИХ ЧИСЕЛ

Множення є однією з основних операцій, що виконується в арифметико-логічних пристроях комп'ютерів. Оскільки операція множення використовується в багатьох задачах, які реалізуються на ПЛІС, то від ефективності виконання алгоритму множення залежить і ефективність пристрою, що може бути виражена у швидкості роботи, апаратних витратах, вартості розробки, тощо.

Базовий алгоритм множення двох двійкових чисел x та y довжини n виконується за наступною формулою:

$$p = \sum_{i=0}^{n-1} x \cdot y_i 2^i \qquad (1)$$

З формули (1) видно, що для того, щоб перемножити два числа довжини n, необхідно спочатку вирахувати n значень xy_i2^i, а потім виконати $n-1$ операцій додавання часткових сум.

Суто архітектурним методом покращення алгоритму множення є зменшення кількості суматорів. Для покращення алгоритму, представленого у формулі (1), можна зменшити кількість суматорів до одного і використовувати його повторно упродовж всього процесу множення. Таким чином, алгоритм множення зміниться наступним чином [1]:

$$p[i+1] = 2^{-1}(p[i] + x \cdot y_i \cdot 2^n), \ i = \overline{0, n-1}, \qquad (2)$$

де $p[0]=0$, $p = p[n]$.

Кожен крок в формулі (2) складається з додавання до попередньої суми значення xy_i2^n, та зсуву отриманого результату на одну позицію вправо. Множник 2^n означає зсув вліво на n позицій для вирівнювання множника x із найбільш значущою половиною регістру добутку p (регістр добутку p містить $2n$ біт).

Можна виділити три методи покращення алгоритму множення [2], що представлений формулою 1, а саме:

— зменшення кількості часткових сум;

— прискорення обчислення часткових сум;

— прискорення додавання часткових сум.

Метод прискорення додавання часткових сум заснований на використанні так званого суматору із збереженням переносу(Carry-Save Adder, CSA), що дозволяє виконувати складання трьох та більше операндів, зменшуючи кількість міжразрядних переносів. При аналізі структурної схеми суматору зі збереженням переносу, що використовується в схемі помножувача [2], стає очевидно, що часткові суми, позначені літерами M, додаються одночасно на двох суматорах CSA. Далі виконується передача відповідних векторів сум Z та векторів переносів C на інші суматори CSA. При формуванні кінцевої суми P використовується один із можливих паралельних суматорів.

Для зменшення кількості часткових сум використовують метод множення, що називається алгоритмом Бута. Алгоритм Бута дозволяє зменшити кількість часткових сум при появі групи послідовних нулів чи одиниць в множнику. При появі послідовності нулів накопичена сума просто зсувається вправо на кількість нулів. При появі послідовності, наприклад, m одиниць, кількість часткових сум буде менше m[4]. Так, послідовність одиниць, яку можна записати як …0{11…11}0… можна представити наступним чином:

$$\dots 0\{11 \dots 11\}0 \dots = \cdots 1\{00 \dots 00\}0 \dots - \cdots 0\{00 \dots 01\}0 \dots \qquad (3)$$

Виходячи з формули 3, замість додавання m часткових сум виконується одне додавання, одне віднімання та зсув на m біт вправо.

Таким чином, застосовуючи методи покращення алгоритму множення двійкових чисел, можна підвищити ефективність роботи обчислювального пристрою. Це дає підставу для подальшого використання структурних властивостей сучасних ПЛІС при проектуванні та реалізації схеми множення цілих чисел.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Milos D. Ercegovac, Tomas Lang. Digital Arithmetic / Milos D. Ercegovac, Tomas Lang. - San Francisco: Morgan Kaufman Publisher, 2004. - 709 p.
2. Wallace C. S. A Suggestion for a Fast Multiplier / C.S. Wallace // IEEE Trans. Electron. Comput. – New York: IEEE, 1964. – Vol. EC-13. – P.14-17.
3. Vladutiu M. Computer Arithmetic Algorithms and Hardware Implementations. - Berlin: Springer-Verlag, 2012. - 267 p.
4. Israel K. Computer Arithmetic Algorithms. - Natick, Massachusetts: A K Peters, 2002. - 281 p.

УДК 004.05

Дмитренко К.О.[1], Киричек Г.Г.[2]
[1] студ. гр. КНТ-527, НУ «Запорізька політехніка»
[2] канд. техн. наук, доц., НУ «Запорізька політехніка»

JIRA ДЛЯ ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ РОЗРОБКИ

На даний час, велика кількість компаній-розробників програмного забезпечення, має потребу скоротити час на організацію праці IT-команд. Тому застосування спеціальних програмних засобів для реалізації бізнес процесів у IT-компаніях є актуальним та допомагає підвищити ефективність ведення бізнесу в IT-індустрії [1].

У сучасному світі існує безліч IT-компаній із сотнями та тисячами робочих місць управління в яких, ще зовсім недавно, зводилося до видачі завдань співробітникам на щоденних засіданнях. Однак, організація персоналу, отримання повної звітності та перевірка виконання поставлених задач займає багато часу, ефективність роботи не представлена наочно, і це значно ускладнює проведення аналізу продуктивності компанії в цілому.

Метою дослідження є визначення особливостей систем та методів, які застосовуються для управління проектами та підтримки командної роботи при розробці програмного забезпечення. Об’єктом дослідження є система автоматизації процесів розробки програмного забезпечення. Предметом – моделі, методи та програмні засоби управління проектами та забезпечення ефективності командної роботи при розробці програмного забезпечення.

Jira є, орієнтованим на розробників, інструментом управління проектами та підтримки командної роботи. В основу її роботи покладені методи Agile, Scrum та Kanban [2]. Для проходження усіх етапів розробки в Agile, використовується спеціальний інструмент, який дозволяє реєструвати помилки і організовувати спільну роботу команди (рис.1). Система дозволяє:

створювати задачі за проектом; видавати ці завдання членам команди та переглядати їх в процесі виконання. На даний час проектні дошки з офісу перемістилися у тікет-систему, де кожен може подивитись результати та оформити звітність по кожному спринту [3].

Різні статуси завдань, які постійно змінюються, допомагають структурувати роботу розробників, скоротити час з інформування о готовності конкретних задач та регулювати строк їх виконання. При цьому система надає можливість спілкуватися прямо у потрібному вікні [4].

Особливістю роботи з Jira при створенні програмного забезпечення є використання станів завдання, які вибираються в залежності від типу проекту та роботи його учасників. Їх найчастіше помічають різними кольорами та підлаштовують під кожний проект, що надає можливість контролювати стадії виконання потрібних задач [5].

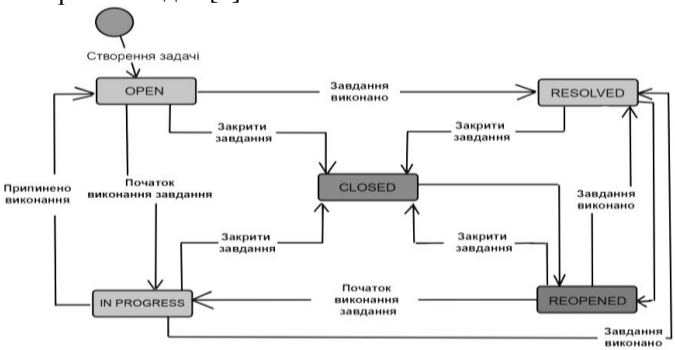


Рисунок 1 – Алгоритм роботи з задачами

Основними конкурентами Jira є системи Trello, Rally, Asana, тощо. Важливими відмінностями в роботі з Trello є: проблеми з формування звітності; робота при виконанні дуже великих проектів і відстежування часу. Asana краще підтримує спілкування в системі та зберігання файлів в хмарі. Rally більш схожий з Jira, але не є таким же універсальним інструментом. Jira все ж таки легше підлаштовується під потреби проекту.

При проведенні досліджень виконано порівняльний аналіз систем для автоматизації процесу розробки. Майже усі IT-компанії користуються засобами Jira, або інших, схожих платформ. Це означає, що даний спосіб ведення проектів та роботи у команді є дуже ефективним. У порівнянні з аналогами, система показує більшу ефективність в роботі з великими проектами, має ширший функціонал для формування звітності та контролю виконання завдань.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Kuruvilla, J. JIRA Development Cookbook /J. Kuruvilla. – Packt Publishing Ltd, 2016. – 551 p.

2. Calefato, F. A case study on tool support for collaboration in agile development /F. Calefato, A. Giove, F. Lanubile, M. Losavio //Proceedings of the 15th International Conference on Global Software Engineering. – 2020. – P. 11-21.

3. Doar, M. Practical JIRA Plugins: Using JIRA Effectively: Custom Development /M. Doar. – " O'Reilly Media, Inc.", 2011. – 113 p.

4. Maddox, S. Confluence, Tech Comm, Chocolate: A wiki as platform extraordinaire for technical communication /S.Maddox. – XML Press, 2012. – 488 p.

5. Humble, J. Continuous delivery: reliable software releases through build, test, and deployment automation /J.Humble, D.Farley. – Pearson Education, 2010. – 484 p.

УДК 004.4

Касьян К.М.[1], Жигулін Д.І.[2]

[1] канд. техн. наук. НУ «Запорізька політехніка»

[2] студ. гр. КНТ-510м НУ «Запорізька політехніка»

АНАЛІЗ ШВИДКОСТІ РОБОТИ УТИЛІТ АВТОМАТИЗОВАНОГО ТЕСТУВАННЯ

Метою роботи є аналіз сучасних утиліт автоматизованого тестування: Selenium WebDriver, Puppeteer і Playwright с використанням мови програмування JavaScript. Використання утиліт автоматизованого тестування допомагає значно прискорити час регресійного тестування. Методи дослідження засновані на розробці скриптів головного функціоналу і end-to-end скриптів демо-проекту. Так само проведено порівняльний аналіз швидкості роботи утиліт автоматизованого тестування.

Тест, що включає Puppeteer / Playwright і Selenium WebDriver, в значній мірі є некоректне порівняння: ці інструменти мають суттєво різні сфери застосування, і будь-хто, хто їх оцінює, повинен знати про їх відмінностях, перш ніж розглядати швидкість.

Selenium WebDriver - це веб-структура, яка дозволяє виконувати кросбраузерні тести. Цей інструмент використовується для автоматизації тестування веб-додатків для перевірки його належної роботи. Selenium WebDriver дозволяє вибрати мову програмування для створення тестових сценаріїв. Як вже обговорювалося раніше, подолання кількох обмежень - це прогрес у порівнянні з Selenium RC. Selenium WebDriver не здатний

обробляти компоненти вікна, але цей недолік можна подолати за допомогою таких інструментів, як Sikuli, Auto IT тощо. [1].

Puppeteer - це бібліотека Node, яка забезпечує API високого рівня для управління безголовим Chrome або Chromium за протоколом DevTools. Він також може бути налаштований на використання Chrome або Chromium [2].

Playwright - це бібліотека Node.js для автоматизації Chromium, Firefox та WebKit за допомогою єдиного API. Playwright створений для того, щоб забезпечити постійну зелену, здатну, надійну та швидку автоматизацію веб-браузерів. Оскільки Microsoft Edge побудований на веб-платформі з відкритим кодом Chromium, Playwright також може автоматизувати Microsoft Edge [3].

Для коректного збору даних буде виконано 1000 успішних послідовних виконань одного і того ж скрипта.

Всі тести проводилися на пристрої під управлінням Windows 8.1 з наступними характеристиками:

— Процесор: 4 ядерний процесор Intel Core i3-3217U;

— Швидкість: 1,8 ГГц;

— Кількість процесорів: 1;

— Загальна кількість ядер: 4;

— Пам'ять: 6 ГБ.

Виділимо основні метрики:

— Середній час виконання (Mean);

— Стандартне відхилення (SD): міра мінливості часу виконання;

— Коефіцієнт варіації (CV): безрозмірний коефіцієнт, що показує мінливість результатів по відношенню до середнього значення;

— P95 (вимір 95-го процентиля): найбільше значення, що залишилося, коли відкидаються верхні 5% чисельно відсортованого набору зібраних даних. Цікаво зрозуміти, як може виглядати неекстремальна, але все ж висока точка даних.

Перший тест порівнювався з демонстраційним сайтом Advantage Online Shopping [4]. Advantage Online Shopping (AOS) - це вигадана компанія, яка використовується Micro Focus як “тестовий додаток” під час демонстрації продуктів, спрямованих на клієнтів. У цьому першому сценарії, при виконанні процедур входу в систему, очікувалося, що час виконання буде складено всього кілька секунд, що відмінно підходить для виявлення потенційних відмінностей при швидкому запуску між реальними інструментами.

Результати перевірки часу на тест логіну в системі в таблиці 1.

Таблиця 1 – Співвідношення швидкості роботи утиліт автоматизованого тестування в коротких тестах

Метрики	Selenium WebDriver	Playwright	Puppeteer
Mean (s)	3.566	3.227	2.211
SD (s)	0.655	1.225	0.433
Cv (s)	0.15	0.334	0.18
P95 (s)	4.677	5.732	2.881

Перше, що відповідає за увагу - це велика розбіжність між середнім часом виконання для " Playwright " і " Puppeeteer", завдяки чому останній майже на 30% швидше і демонструє менші варіації в його продуктивності. Так тест показує, що Selenium WebDriver показує більш низьку швидкість, ніж інші утиліти. Можливо, менш індивідуально, що запуск Puppeteer без якої-небудь доданої високоуровневої інфраструктури допоможе нам значно зменшити час виконання цього дуже короткого сценарію.

Наступний сценарій це E2E-тест (наскрізний). Сценарій проходить через вхід в систему, вибирає товар, додає в кошик та оформляє замовлення. Результати перевірки часу на тест логіну в системі в таблиці 2.

Таблиця 2 – Співвідношення швидкості роботи утиліт автоматизованого тестування в end-to-end тестах

Метрики	Selenium WebDriver	Playwright	Puppeteer
Mean (s)	37.56	31.5	32.39
SD (s)	2.8	1.88	2.71
Cv (s)	0.159	0.055	0.071
P95 (s)	40.77	35.19	35.44

Ми бачимо, що розбіжність між " Playwright " та іншими інструментами тепер означає значно більше: середній час виконано більше 1000 ітерацій на Selenium WebDriver значно менше, ніж у інших інструментів. Виходичя з цього, використання " Playwright " у наскрізних тестах працює ефективніше ніж в коротких тестах.

Досить ефективно використовувати Puppeteer в коротких тестах, а Playwright довів свою ефективність в end-to-end тестах. Selenium WebDriver показав найгірші результати при порівнянні утиліт. Автоматизоване тестування покликане допомогти робити ті ж завдання регресійного тестування, швидше і оперативніше випускати нові релізи продукту і так само скорочувати витрати на тестування. Якщо зазвичай на тестування йде 3-

4 дні (залежить від специфіки проекту), з автоматизацією цей час скоротиться до декількох годин.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Selenium WebDriver documentation [Electronic resource]. – Access mode: https://www.browserstack.com/guide/selenium-webdriver-tutorial#:~:text=Selenium%20WebDriver%20is%20a%20web,language%20to%20create%20test%20scripts

2. Puppeteer documentation [Electronic resource]. – Access mode: https://developers.google.com/web/tools/puppeteer#:~:text=Puppeteer%20is%20a%20Node%20library,%2Dheadless)%20Chrome%20or%20Chromium

3. Playwright documentation [Electronic resource]. – Access mode:https://docs.microsoft.com/en-us/microsoft-edge/playwright/#:~:text=Playwright%20is%20a%20Node.,capable%2C%20reliable%2C%20and%20fast

4. Advantage Online Shopping [Electronic resource]. – Access mode:https://advantageonlineshopping.com/

УДК 004.3`2

Сгадов С. О. [1], Ковтун М.І. [2]
[1] ст. викладач НУ «Запорізька політехніка»
[2] студ. гр. КНТ-519сп НУ «Запорізька політехніка»

ОРГАНІЗАЦІЯ КАСТОМНОЇ РІДИННОЇ СИСТЕМИ ОХОЛОДЖЕННЯ ДЛЯ ГРАФІЧНОГО ПРОЦЕСОРА В СИСТЕМНОМУ КОРПУСІ

В сучасний час відеокарта вважається одним з найдорожчих і надважливих комп'ютерних комплектуючих в системному корпусі. Для неї в обов'язковому порядку необхідна добре продумана система охолодження для досягнення еталонних температур і прийнятному рівні шуму, стабільної роботи в робочих завданнях, 3D рендеринге і особливо у майнінгу під 98-100% навантаженням. Продумана система охолодження графічного процесора дозволяє не тільки збільшити термін служби всієї відеокарти в цілому, а також вирішити питання енергоспоживання і прийнятний рівень шуму при досягненні гарної співвідношенні ціна-якість. Тепер варто приділити увагу про вирішення проблеми роботи відеокарти у майнінгу і відповідно проблему терміну експлуатації графічного процесора в цілому. Майнінг в наш час стає все більш актуальним і в той же самий час одним з найнебезпечніших видів діяльності для відеокарти. Майнінг небезпечний для графічних процесорів тим, що часто відеокарти у майнінгу працюють при

максимальному навантаженні практично цілодобово, і традиційне активне охолодження не завжди справляється з такими навантаженнями протягом великого відрізку часу. Для того, щоб відеокарта працювала при максимальному навантаженні довгий час, їй необхідно досягти еталонних температур, для цього потрібно розібратися в тому, що кожен графічний чіп індивідуальний за своїми технічними характеристиками, для кожного графічного чіпа відеокарти існують різні споживання Ват енергії і виділення Ват тепла, а також різні робочі температури. Але в загальному їх усіх об'єднує одна схожа характеристика - стандартний набір температур в простої (в мінімальному робочому навантаженні), і в режимі максимальної продуктивності.

Еталонні температури для кожного компонента графічного процесора:

- Графічний процесор (в простої 30 ± 3 С ° і максимальним навантаженням 60 ± 3 С °)
- Відеоконтролер (в простої 30±3 С° і максимальним навантаженням 60±3 С°);
- Відеопам'ять та Відео-BIOS (в простої 30±3 С° і максимальним навантаженням 55±60 С°);
- Цифро-аналоговий перетворювач (в простої 30±3 С° і максимальним навантаженням 50±55 С°);

Ці температури впливають на стабільність роботи всього графічного процесора як єдиної робочої станції. Під цим терміном мається на увазі робота відеокарти під максимальним навантаженням тривалий час з еталонними для неї температурами. Стабільності роботи можна домогтися за допомоги продуманої системи відводу тепла. Бажано щоб графічний процесор мав гарне співвідношення ціна-якість і був ідеальним рішенням для будь-якого користувача, потрібно створити відеокарту з хорошою системою охолодження, щоб не було надлишку кулерів, трубок, виробництво було масове і легко даний графічний процесор міг обслуговуватися. За даними критеріями вийшла еталонна відеокарта з такими головними особливостями: ціна на рівні будь-якого конкурента, це можна досягти шляхом масовості даного продукту і відповідно, зниження кінцевої ціни на виробництво, легкість в технічному обслуговуванні та правильно підібрані розміри кулерів, трубок, їх форма і вихідна потужність для досягнення співвідношення продуктивність-шум. Загальний термін служби відеокарти залежить від правильної системи охолодження, а продумана система є головною складовою в цьому процесі. Відповідно, чим краще система охолодження графічного процесора, тим більше часу він зможе пропрацювати.Останній критерій полягає в тому, щоб вирішити проблему з шумом відеокарти в цілому, для цього потрібно знайти баланс між продуктивністю кулерів,

рівнем шуму і грамотно налаштованої рідинної системи охолодження. Таким чином, можна досягти тихої роботи відеокарти (30 дб) під максимальним навантаженням. Характеристики еталонної кастомної відеокарти: варіант установки - горизонтальний; довжина відеокарти: 330 мм; висота відеокарти: 135 мм; кількість вентиляторів: 3;матеріал: сталь; роз'єми: 3 x Display Port 1.4;1 x HDMI 2.1; вага-800 гр.

Отже, пропоную вирішити проблему з традиційним активним охолодженням графічного процесора в наступний спосіб. За основу кастомної відеокарти будуть взяті 3 кулери, розташовуватися вони будуть на передній стороні графічного процесора, а саме модель від фірми NZXT Kraken Z63 280mm. Ці кулера ідеально справляються з відведенням тепла за рахунок підшипника з гвинтовим різьбленням. На задній стороні відеокарти буде стояти помпа, куди буде надходити рідина для додаткового охолодження графічного процесора, також в самій відеокарті будуть розташовуватися так звані "доріжки" куди рідина буде рівномірно розподілятися по всьому графічному процесору, в результаті чого температура буде знижена на 30-40%. Основний резервуар, де буде зберігатися рідина, буде розташований на передній стороні всередині системного блоку. В результаті відеокарта при максимальному навантаженні матиме наступні показники: шум відеокарти становитиме - 30 дб, а температура – 60 С°.

УДК 004.3
Сгадов С. О. [1], Ренгевич Д.С. [2]
[1] ст. викладач НУ «Запорізька політехніка»
[2] студ. гр. КНТ-519сп НУ «Запорізька політехніка»

ОРГАНІЗАЦІЇ ПОВІТРЯНИХ ПОТОКІВ У ЗАКРИТОМУ КОРПУСІ СИСТЕМНОГО БЛОКУ ЗІ СТАНДАРТНИМ НАБОРОМ КОМП'ЮТЕРНИХ КОМПЛЕКТУЮЧИХ

Ціль роботи є створення оптимального (еталонного) комп'ютерного корпуса системного блоку з організацій повітряних потоків для забезпечення стабільності роботи стандартного набору комп'ютерних комплектуючих, зниження ціни для кінцевого споживача, розв'язок проблеми «деградації кремнію» у наслідки нагрівання, а також зниження шуму вхідного й вихідного потоку повітряних потоків .

У кожному системному блоці закритого типу в сучасний час обов'язково необхідна добре продумана система відводу тепла (циркуляція повітря) для досягнення еталонних температур кожного комп'ютерного

компонента, також для досягнення стабільної роботи всіх електронних компонентів у робочих завданнях . Розв'язати проблеми з побільшати термін служби всієї системи в цілому, також розв'язати питання всіх споживачів у такому ключі «скільки потрібно комп'ютерних кулеров у їхню систему?» і як досягтися співвідношення ціни-якості . Окремо варто приділити увага питанню про розв'язку проблеми «деградація кремнію» і проблему «шуму комп'ютерної системи».

Для того щоб досягтися еталонних температур потрібно розібратися в тому що кожні комп'ютерні комплектуючі індивідуальні по своїх технічних характеристиках для них існують різне споживання кількості енергії й виділення тепла, а також різні робочі температури. Але в цілому їхніх усіх поєднує одне –стандартний набір температур що в простої (у мінімальному робочому навантаженні) і в режимі максимальної продуктивності (у піковому навантаженні). За основи були взято 3 кулера для вдува холодного корпуса з передньої сторони корпуса, а саме модель від фірми be quiet! Pure Wings 2 PWM 120mm. Ці кулера ідеально справляються з відводом тепла за рахунок підшипник із гвинтовим різьбленням, який у свою чергу має безліч режимів роботи для тихого й продуктивного складання залежно від режиму роботи користувача. Дев'ять оптимізованих для тихої роботи лопат вентилятора забезпечують великий повітряний потік і підвищений тиск зі зниженої турбулентності Також були взято два кулера цієї ж фірми на видувши гарячого потоку повітря (один із задньої сторони системного блоку й один зверху після проходження потоку гарячого повітря через процесорний кулер. Також ці кулера в пікою потужності мають до 1500 про./мін. Потік повітря: 51.4 CFM, 87 м³/рік що забезпечує співвідношення продуктивність-шум. Максимальний рівень шуму перебуває в межах 10-20 Дб.

Також за основи були взяті магнітні пилові фільтри, який будуть установлені з передньої сторони вдува холодного повітря й з верхньої сторони выдува гарячого потоку повітря. Також у системному блоці використовуються додатковий пиловий фільтр із нижньої сторони системного блоку.

Принцип роботи всієї системи продуву комп'ютерних комплектуючих :

1 Етап. 3 передніх корпусних кулера будуть вдмухувати потік холодного повітря з навколишнього середовища через пиловий фільтр. При цьому нічого не буде заважати забору повітря й проходженню його через пиловий фільтр.

2 Етап. Самий нижній корпусний кулер вдуву холодного повітря буде наганяти повітря в нижнею частина системного блоку, потік повітря буде проходити через Жд-Диск і Ssd-Накопичувач для ефективного обдува виділюваного тепла пристроями.

3 Етап. Центральний кулер передньої панель корпуса буде вдмухувати потік холодного повітря й потім повітря буде проходити через чипсет системної плати й буде наповнювати свіжим повітрям кулера графічного процесора й виходити через задню частину корпуса поблизу графічного процесора.

4 Етап. Самий верхній корпусний кулер вдува холодного повітря буде вдмухувати повітряний потік у верх материнської плати, який буде проходити через оперативну пам'ять, установлень M2 через PCI Express 3.0 x4, а також буде підхоплюватися процесорним кулером через який буде вже проходити холодний потік повітря, яке у свою чергу буде проходити через гарячий радіатор процесора й видуватися через верхнею й задню частину корпусного кулера працюючого на видувши.

5 Етап. Два корпусні кулера працюючих на видмувши гарячого повітря будуть видувати всі потоки гарячого повітря через верхній пыливой фільтр.

УДК 004

Сологубов І.Я. [1], Іващенко В.Р. [1], Паромова Т.О. [2], Луценко Н.В. [2]
[1]студ. гр. КНТ-528 НУ "Запорізька політехніка"
[2]старш. викл. НУ "Запорізька політехніка"

ПОБУДОВА ДОДАТКІВ ЗА АРХІТЕКТУРОЮ SINGLE ACTIVITY

Сьогодні Android додатки можна побудувати за двома видами архітектури: Multiple Activity Architecture та Single Activity Architecture. Основним елементом додатку є Activity - ~~не~~ окремий екран в Android. Це вікно додатку для робочого столу, або фрейм в програмі на Java. Activity дозволяє розмістити всі компоненти інтерфейсу користувача.

У світі довгий час при розробці Android додатків використовується підхід Multiple Activity. Архітектура Multiple Activity додатка показана на рис.1. При побудові додатку за цією архітектурою додаток створює декілька окремих екранів (Activity), на яких можуть розміщуватись всі компоненти інтерфейсу користувача [1, 2]. Кожен екран (Activity) має свій життєвий цикл, він може знаходиться в різних станах в залежності від того, що відбувається з додатком.

На екрані користувача може відображатися тільки один Activity. Коли користувач переходить на інший екран, то Activity, що відображається в цей момент, зупиняється і стає неактивним, а його життєвий цикл передається у керування іншому Activity.

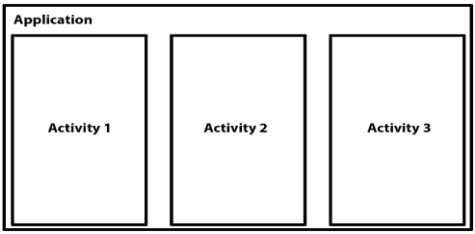


Рисунок 1 – Архітектура Multiple Activity додатка

У 2018 році компанія Google запропонувала інший підхід для побудови мобільних додатків - Single Activity Architecture. Архітектура Single Activity додатка показана на рис.2. Особливість такого підходу полягає в створенні одного Activity, що буде головним модулем для керування життєвим циклом додатка, а всі екрани інтерфейсу користувача створюються на фрагментах - частинах Activity та мають свій життєвий цикл. У фрагменті знаходяться елементи інтерфейсу і весь функціонал для взаємодії з користувачем [3].

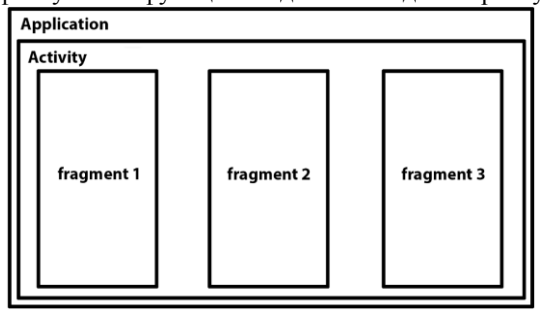


Рисунок 2 – Архітектура Single Activity додатка

В роботі було розроблено два мобільних додатка, які мають однаковий функціонал, але побудовані за різною архітектурою. Під час роботи вивчені особливості розробки додатку за архітектурою Single Activity, а також досліджені переваги такого підходу у порівнянні з архітектурою Multiple Activity. Наприклад, саме використання фрагментів дозволило реалізувати навігації компонентів для переходу по екранам додатку. При цьому Activity стає рамкою зі своїми компонентами, усередині якої розміщуються фрагменти. Вкладеність фрагментів обмежується фізичними можливостями пристрою. Вкладений фрагмент сам може мати навігацію і свої вкладені фрагменти [4].

На рис.3 продемонстровано перехід користувача між екранами за методом Single Activity, з використанням навігаційної панелі, яка прив’язана

до Activity і завжди знаходиться на екрані. Activity, після отримання команди користувача, змінює активний фрагмент на вибраний.

Рисунок 3 – Перехід між екранами за допомогою панелі навігації

На рис.4 продемонстровано перехід користувача при відмові від навігаційної панелі, коли при переході між вікнами необхідно повертатися до головного екрану додатку.

Рисунок 4 – Перехід між екранами без панелі навігації

У деяких випадках відмова від фрагментів може значно погіршити навігаційні можливості додатку. Наприклад, фрагменти представляють собою по суті компоненти екрану, в той час як Activity цілий екран. Ту ж навігаційну панель довелося б малювати на всіх екранах, куди переходить користувач. У випадку з фрагментами, можна змінювати тільки частину екрану (фрагмент), при цьому збільшується швидкість реакції екрану на маніпуляції користувача.

До переваг архітектури Single Activity можна віднести:

— просте керування життєвим циклом додатку;

— легший спосіб передачі даних між фрагментами;

— більш гнучкий динамічний інтерфейс.

Останнього часу біли виправлені помилки та незручності в проектуванні за архітектурою Single Activity, тому кожен додаток можна

удосконалити шляхом переробки його у відповідності до цієї архітектури. Це зробить його швидшим, гнучким і легшим у його майбутній підтримці та масштабуванні.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Dawn Griffiths, David Griffiths. Head First Android Development: A Brain-Friendly Guide. – 2nd Edition. – O'Reilly Media, 2017. – 704 p.

2. Клифтон, Ян. Проектирование пользовательского интерфейса в Android: реализация принципов материального дизайна для разработчиков / Пер. с англ. – М.: ДМК Пресс, 2017. – 451 с.

3. Лицензия на вождение болида, или почему приложения должны быть Single-Activity [Електронний ресурс] – Режим доступу: https://habr.com/ru/company/ redmadrobot/blog/426617.

4. Одна активность и все остальные фрагменты Oh! Android [Електронний ресурс] – Режим доступу: http://www.ohandroid.com/x444-2.ht

5. Филлипс Б., Стюарт К., Марсикано К. Android. Программирование для профессионалов. – 3-е изд. – СПб: Питер, 2019. – 688с.

УДК 004

Фінько В.С.[1], Скрупський С.Ю.[2]
[1] студ. гр. КНТ-527 НУ «Запорізька політехніка»
[2] доц. НУ «Запорізька політехніка»

АЛГОРИТМ БЕЗПЕРЕРВНОЇ ІНТЕГРАЦІЇ ТА БЕЗПЕРЕРВНОЇ ДОСТАВКИ

Традиційно випуск нових версій ПЗ був пов'язаний з чималими труднощами. Проблеми починалися на ранніх етапах розробки. Окремі групи працювали над своїми модулями незалежно один від одного і перед випуском нових версій повинні були спочатку інтегрувати їх разом. У процесі інтеграції, коли модулі вперше об'єднувалися один з одним, система часто навіть не компілювалася. Витрачалося багато часу для того, щоб інтегрувати всі зміни. Також процес перенесення був дуже трудомістким. Вигоди від впровадження якої-небудь однієї функції не виправдовували зусиль, що витрачалися на інтегрування і розгортання.

За допомогою технологій безперервної інтеграції і безперервного розгортання можна запобігти виникненню подібних проблем. Знижується трудомісткість інтеграції, система стає більш передбачуваною, несправності виявляються і виправляються швидко, розгортання здійснюється частіше. Це

скорочує час введення в дію нових можливостей, що дає суттєву перевагу для бізнесу: він отримує можливість швидше реагувати на зміни ринку.

CI / CD (Continuous Integration, Continuous Delivery - безперервна інтеграція і доставка) - це технологія автоматизації тестування та доставки нових модулів, розроблюваного проекту зацікавленим сторонам (розробники, аналітики, інженери якості, кінцеві користувачі і ін.) [1].

Процеси розгортання мають високий рівень автоматизації, тому результати легко відтворити. Тобто після розгортання і перевірки системи в тестовому оточенні, той же результат буде отримано в робочому оточенні, тому що самі оточення повністю ідентичні. Це дозволяє практично повністю усувати будь-які помилки і, відповідно, зменшувати ризики.

Крім того, сам процес тестування програмного забезпечення стає простішим завдяки автоматизації його виконання. Це ще більше підвищує якість, тому що тестування може виконуватися набагато частіше.

Чим частіше проводиться розгортання нових версій, тим нижче ризики, тому що з кожним разом розгортається менша кількість змін і нововведень. Чим менше змін потрапляє в робоче оточення, тим нижче ризик появи помилок [2].

Для організації процесу CI / CD на виділеному сервері запускається служба, в завдання якої входять: отримання вихідного коду зі сховища, визначення гілки проекту, збірка проекту, виконання тестів, відправка образів у віддалений репозиторій, видалення використаних образів, розгортання готового проекту.

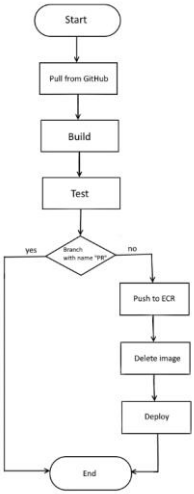


Рисунок 1 – Блок схема запропонованого алгоритму

В роботі пропонується наступний алгоритм. На першому етапі витягується Dockerfile з репозиторію на GiHub. Команда docker build будує образ, звертаючись до файлу Dockerfile, який знаходиться в поточній директорії, на що вказує точка в кінці.

На наступному етапі тестуємо образ, використовуючи команду docker run, яка запускає контейнер на основі побудованого образу.

Надалі визначаємо належність до певної гілки. Після цього відправляємо образ до віддаленого репозиторію.

Наступна дія – це видалення образу. Останній етап – це розгортання образу у сервісі ECS. Створюємо набір змін для оновлення стеку CloudFormation за допомогою заданого шаблону.

Якщо дотримуватися такого алгоритму, можна забезпечити побудову стійкої та налагодженої роботи будь-якої системи, що збільшує прибуток компанії і зменшує час простою обладнання та робочої сили.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Поняття CI/CD [Електроний ресурс]. Режим доступу: https://selectel.ru/blog/what-is-ci-cd

2. Еберхард В. Continuous delivery. Практика безперервних апдейтів. – СПб.:Пітер, 2018. – 320с.

УДК 004.5

Іващенко Н.А.[1], Федорова В.Ю.[1], Луценко Н. В.[2], Паромова Т. О.[2]
[1] студ. гр. КНТ-518 НУ «Запорізька політехніка»
[2] старш. викл. НУ «Запорізька політехніка»

ТЕХНОЛОГІЇ, ЩО ВИКОРИСТОВУЮТЬСЯ В РОЗУМНИХ БУДИНКАХ

З розвитком сучасних технологій з’явилася ідея їх використання в побуті. Концепція розумного будинку з кожним роком все більш привертає увагу як звичайних користувачів, так і людей, які працюють в сфері технологій. Використання систем управління побутовими приладами може дати до 40% економії комунальних витрат на будинок, квартиру. Технологія розумного будинку – це збірний термін для інформаційних і комунікаційних технологій, що використовуються в будинках, де різні компоненти зв'язуються через локальну мережу. Технологія може бути використана для контролю, попередження та виконання функції відповідно до обраних критеріїв, надає можливість автоматичного зв’язку з навколишнім середовищем через Інтернет або мобільні телефони. Система розумного

будинку використовує передові комп'ютерні технології (мережеві та комунікаційні) та технології автоматичного управління, яка об'єднує підсистеми управління освітленням, контролю температури, управління пожежною і охоронною сигналізацією, системи комутації головного будинку [1].

Але при всій актуальності цієї теми готової системи розумного будинку ще не існує, але вже є багато прототипів. На це є ряд причин. По-перше, велика кількість різновидів пристроїв, які використовується у побуті, та багато виробників, що використовують різні технології для реалізації керування пристроями. Це підвищує витрати на розробку системи та ускладнює управління пристроями. По-друге, окремі технології складно об'єднувати в одну систему розумного будинку, вони дуже швидко розвиваються незалежно одна від одної. Це означає, що і вся система повинна швидко адаптуватися до змін кожного пристрою.

Кожна система розумного будинку включає в себе три основних складові, а саме: контролер (або хаб), датчики та актуатори. Хаб – це керуючий пристрій системи, він об’єднує всі елементи системи та дозволяє керувати їх роботою. Датчики або сенсори використовуються для отримання даних приборів, які потрібні системі для аналізу подальшої роботи. Актуатори – це виконавчі пристрої, що безпосередньо виконують команди. Це найчисленніша група приладів, в яку входять автоматичні вимикачі, розетки, клапани для труб, сирени, клімат-контролери і так далі [2].

Один з варіантів побудови схеми розумного будинку показано на рис.1.

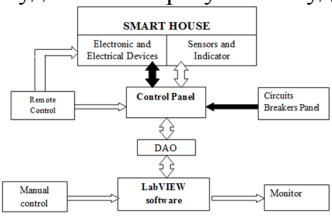


Рисунок 1 – Блок-схема розумного будинку

У розумному будинку два інтерфейси: комп'ютерний і пульта дистанційного керування [3]. З точки зору розумного будинку систему керування можна представити як сукупність електронних, електричних пристроїв (Electronic та electrical devices), їх індикаторів та сенсорів (Sensor and indicator), панелі вимикачів (Circuit breakers panel), які підключені до панелі керування (Control panel). Основна мета збору даних - виміряти електричні параметри (напруга, струм, потужність), а також температуру, звук, тощо. Всі зібрані параметри пристроїв подаються на панель керування. Устаткування DAO діє як інтерфейс між мікроконтролером і підсистемами.

Воно, в першу чергу, функціонує як пристрій, який вхідні аналогові сигнали перетворює в цифрові для подальшого аналізу мікроконтролером.

Для аналізу вхідних сигналів та отримання вихідних сигналів керування для всієї системи будинку використовується програмне забезпечення LabVIEW, яке призначено для збору та обробки даних в системах, а також для управління технічними об'єктами і технологічними процесами [3].

Кожен компонент виконує одну функцію і не синхронізується з іншими компонентами. Пульт Remote control використовується для виконання деяких операцій в системі за допомогою бездротової технології з програмним забезпеченням LabVIEW. Він призначений для управління і перемикання приладів в кожній кімнаті будинку за допомогою приймача кімнатного блоку.

Існує чи мало технологій, які використовують для побудови розумного будинку. Але єдиної системи розумного будинку досі не існує, хоча сучасні технології це позволяють. Компанії, що встановлюють такі системи, рекомендують це робити на стадії початку ремонту або взагалі початкової стадії побудови будинку.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Чарли Уинг. Как работает ваш дом / Пер. с англ. – М.: ДМК Пресс, 2020. – 206с.

2. Xu, Jingzhi. "Smart Home". Business Aspects of the Internet of Things, 13, 2009.

3. Guneet Kour, Jaswanti, "Lab VIEW Based Alarm Systems In Home" / ISSN 2250-0987 Jaswanti et al, UNIASCIT, 2012. – Vol 2, Issue-3. – P.305-307.

УДК 004.384

Костецький Д.В.[1], Луценко Н.В.[2], Паромова Т.О.[2]
[1] студ. гр. КНТ-518 НУ "Запорізька політехніка"
[2] старш. викл. НУ "Запорізька політехніка"

СОРТУВАННЯ ФАЙЛІВ З ОДНОЧАСНИМ СТВОРЕННЯМ ПАПОК

У зв'язку із зростанням обсягів інформації, що зберігається на комп'ютерах користувачів, питання про підвищення ефективності при пошуку і завантаженні потрібної інформації стає дуже актуальним.

Кожному напевно доводилося опинитись у ситуації, коли витрачається багато часу, щоб знайти потрібну інформацію. При цьому на комп'ютері є папки з величезною купою файлів, що розкидані хаотично, і які займають багато вільного місця. Якщо з'явилися проблеми з пошуком необхідних файлів, рекомендується навести порядок на своєму комп'ютері.

Вже давно було помічено, що правильна організація роботи дозволяє значно заощадити час, який доводиться витрачати на пошук файлів. До того, час від часу на комп'ютері з'являються дублікати файлів, тому ж треба позбавлятися від даних, що вже не потрібні. Для того, щоб правильно організувати зберігання файлів, доведеться пройти кілька етапів. Однак, як показує практика, всі ці витрати часу виправдані. Якщо правильно організувати зберігання файлів, то можна більш ефективно використовувати ємність жорсткого диску, якій використовується для збереження файлів, що призведе до підвищення ефективності роботи комп'ютеру у цілому.

Найпростішим способом рішення цієї проблеми може бути використання різних видів сортування файлів. У операційній системі Windows передбачено сортування файлів у межах однієї папки. Тому для скорочення часу пошуку можна додати сортування файлів різного типу з одночасним розподіленням їх по окремим папкам.

Для розширення можливостей сортування було розроблено додаток, що сортує файли в певній директорії. Для розробки такої програми була обрана мова Python, яка може використовуватись з різним апаратним забезпеченням та у середовищі різних операційних систем [1] . Python - інтерпретована мова, до запуску це звичайний текстовий файл. Мова має відносно простий синтаксис, добре спроектована та логічна, тобто програму не потрібно компілювати та один і той же код можна запускати на різних платформах. Програмування на мові Python значно простіше, ніж на багатьох інших мовах, тому потребує меншого часу, і відмінно підходить для новачків [2, 3].

Розроблений додаток забезпечує сортування, як вже існуючих в директорії файлів, так і файлів, що додаються після запуску програми. Відсортовані файли розміщуються у відповідних папках, які створюються під час роботи програми. Цей додаток було зібрано в exe-файл, що сортує файли у поточній директорії.

При створенні программи використовувались бібліотеки watchdog і os. За допомогою бібліотеки watchdog проглядаються зміни в певній директорії, при появі нових файлів викликається повідомлення-івент. Потім за допомогою бібліотеки os відбувається перегляд файлів в поточній директорії. Далі в циклі з імені файлу відділяється розширення і передається у відповідну функцію, що обирає потрібну папку для перенесення файлів. [4, 5]. Результати виконання програми надані на рис. 1, на якому видно, що після сортування за допомогою розробленої програми файли відповідного типу на диску упорядковані у окремі папки. Це значно спрощує пошук необхідної інформації.

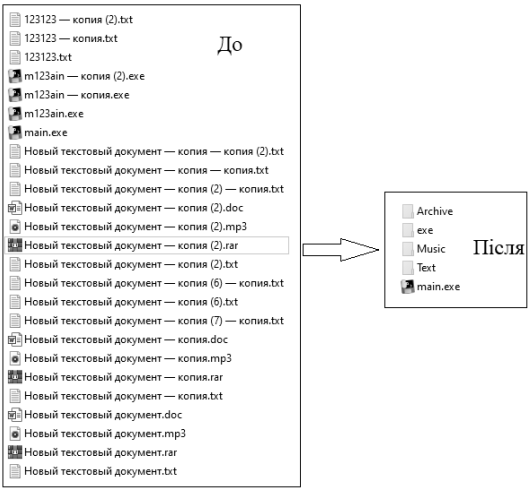


Рисунок 1 - Результати виконання програми

Таким чином, розроблена програма забезпечує можливість використання ще одного варіанту сортування у середовищі операційної системи Windows.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Лутц М. Изучаем Python / Пер. с англ. – Киев.: ООО "Диалектика", 2020. –870 с.

2. Марк Саммерфилд. Python на практике / Пер. с англ. – М.: ДМК Пресс, 2014. –338 с.

3. Васильев А.Н. Python на примерах. – СПб: Наука и Техника, 2017. – 432 с.

4. Бизли, Дэвид М. Язык программирования Python. Справочник. – Киев: ДиаСофт, 2000. – 336 с.

5. Сузи Р. А. Python. Наиболее полное руководство (+CD). – СПб.: БХВ-СПб, 2002. – 768 с.

УДК 004.056.5
Мороз А.Р. [1], Паромова Т.О. [2]
[1]студ. гр. КНТ-528 НУ "Запорізька політехніка"
[2]старш. викл. НУ "Запорізька політехніка"

ПОРІВНЯННЯ МЕТОДІВ ШИФРУВАННЯ

Криптографія є одним з важливих напрямків розвитку IT-технологій, якій націлені на підвищення та удосконалення захисту інформації. Вона розвивається в декількох напрямках, таких як забезпечення конфіденційності, забезпечення цілісності даних, автентифікації і шифрування.

Одним з методів захисту інформації є шифрування, яке застосовується для зберігання важливої інформації в ненадійних джерелах і передачі її по незахищеним каналам зв'язку. Удосконалення методів шифрування дозволить підвищити як безпеку, так і швидкість передачі даних.

Існує два види шифрування: симетричне і асиметричне [1].

У шифруванні симетричним методом використовується один і той же ключ як для шифрування повідомлення, так і для його дешифрування. Існує декілька методів симетричного шифрування, наприклад, DES, 3DES, Skipjack. Кращим симетричним алгоритмом шифрування на даний момент визнається AES, який і буде використаний для подальшого аналізу. Основним недоліком симетричного методу є те, що один і той же ключ використовується як для шифрування, так і для дешифрування. А так як ключ передається по відкритому каналу, то ймовірність його перехоплення дуже висока [2, 3].

При асиметричному методі шифрування використовується публічний ключ для шифрування і закритий для дешифрування. Так як ключі пов'язані в пару, навіть втрата публічного ключа ніяк не загрожує безпеці системи. Асиметричний метод шифрування реалізується стандартним алгоритмом RSA і новим методом шифрування, заснованим на епілептичних кривих ECC [2, 4].

Кожен алгоритм шифрування має свої переваги та недоліки і свою галузь застосування. Для використання потрібного алгоритму шифрування необхідно знати надійність його ключів і часові витрати. Дані порівняння було отримано аналітичним методом (таблиця 1) та на практиці (таблиця 2)

Аналіз даних таблиці 1 показує, що час злому шифру навіть другого рівня безпеки є занадто великим (понад 10^5 років) для успішного використання стандартних дешифраторів у вигляді стаціонарних комп'ютерів. Однак існує можливість використання суперкомп'ютерів, які забезпечують продуктивність на 5-10 і більше порядків. Для подальшого підвищення швидкості обчислень існує можливість об'єднати суперкомп'ютери в кластери, підсумовуючи їх обчислювальні потужності.

Таким чином, не дивлячись на збільшення часу шифрування і дешифрування, ефективним є використання шифрів підвищеної складності (4-5 і вище) рівня безпеки. Для порівняння часових витрат було використано файл розміром 500 Мб.

Таблиця 1 – Порівняння надійності ключів

Симетричний алгоритм	Біт безпеки	RSA публічний (біт безпеки)	ECC публічний (біт безпеки)	Час на зламування (при $1 * 10^{21}$ комбінацій/с)	Рівень безпеки
Skipjack	80	1024	160	20 хвилин	1
3DES	112	2048	224	164 464 років	2
AES-128	128	3072	256	$10.8*10^{9}$ років	3
AES-192	192	7680	384	$199 * 10^{27}$ років	4
AES-256	256	15360	512	$3 * 10^{48}$ років	5

Таблиця 2- Порівняння часових витрат

Рівень безпеки	Алгоритм	Час генерації ключа (мс)	Час шифрування (мс)	Час дешифрування (мс)
1	Skipjack	33	34	10
	RSA	36	30	27
	ECC	20	10	0
2	3DES	35	33	17
	RSA	40	35	30
	ECC	22	22	15
3	AES	39	32	25
	RSA	42	39	39
	ECC	28	32	35
4	AES	41	32	30
	RSA	46	42	41
	ECC	30	37	39
5	AES	44	33	33
	RSA	50	47	50
	ECC	33	40	43

Аналіз даних таблиці 2 показує, що час, необхідний для створення ключа, шифрування і дешифрування даних розміром 500 мб вкрай мало, однак при використанні в промислових і корпоративних масштабах, обсяг шифрованих даних можуть становити більш терабайта, що робить цей процес

неприпустимо тривалим. Тому необхідно вибирати між більшою швидкістю або безпекою.

У порівнянні з іншими алгоритмами асиметричного шифрування найбільш швидкодіючим ECC, проте на відміну від інших алгоритмів алгоритми випадкових чисел для генерації еліптичних кривих викликають сумнів, так як метод їх генерації залишається невідомий [5].

Існує ще один метод шифрування - гібридний метод. Суть даного методу полягає у тому, що дані шифруються симетричним методом, наприклад, за алгоритмом AES, а ключ для симетричного методу ще раз шифрується асиметричним методом (алгоритм RSA або ECC), і по незахищеному каналу передається вже зашифрований відкритий ключ.

Аналіз таблиці 1 та таблиці 2 показує, що метод гібридного шифрування об'єднує у собі простоту реалізації та швидкість шифрування основного обсягу даних із допомогою симетричного методу шифрування та криптографічну стійкість асиметричного методу шифрування для шифрування симетричного ключа.

Поєднання цих факторів дозволяє підвищити швидкість виконання криптографічних операцій та захищеного обміну даними, що робить метод симетричного шифрування вкрай перспективним для вивчення і подальшого розвитку.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Баричев С.Г. и др. «Основы современной криптографии». - М.: «Горячая линия – Телеком», 2001 - 120 с.

2. Криптография простым языком: разбираем симметричное и асимметричное шифрование на примере сюжета Звездных войн (Updated) [Електронний ресурс] - Режим доступу: https://habr.com/ru/post/452042/

3. Саломаа А. «Криптография с открытым ключом Пер. с англ. - М.: Мир, 1995. - 318 с.

4. Асимметричное шифрование на практике [Електронний ресурс] - Режим доступу: https://habr.com/ru/post/449552/

5. Daniel J. Bernstein, Tung Chou, Chitchanok Chuengsatiansup and others «How to manipulate curve standards: a white paper for the black hat».- Ch.: University of Illinois at Chicago, 2015 – 44 с.

6. Баричев. С.Г. «Криптография без секретов» - М.: Горячая линия - Телеком, 2004. - 43 с.

УДК 004
Демченко А.С.[1], Скрупський С.Ю.[2]
[1] студ. гр. КНТ-517 НУ «Запорізька політехніка»
[2] к.т.н., доц. НУ "Запорізька політехніка"

АЛГОРИТМ РОЗРАХУНКУ ВАРТОСТІ РОБОЧОГО ЧАСУ

Кожна людина планує свій день, зокрема, роботу, відпочинок та інші буденні справи. Даний алгоритм призвана оптимізувати всі або частину складових вашого життя, ким би ви не працювали та чим би не займались.

Як саме, розглянемо по пунктах.

Основна ціль алгоритму розрахувати вартість однієї години користувача, що б на основі цих даних надати йому статистичну інформацію що-до продуктивності робочого дня та допомогти в раціональному плануванні витрати власного часу.

З точки зору роботи найбільшу користь алгоритм несе для «фрілансерів»[1]. Люди, які працюють таким чином не раз стикались з проблемою оптимальної оцінки своєї праці. Так, спираючись на час, який виконавець витратив на виконання поставленої задачі, розрахує суму до оплати для замовниках [2, 3]. Алгоритм розрахує вартість години праці виконавця спираючись на надану стартову інформацію користувачем алгоритму. Після чого з часом використання алгоритм відстежує та корегує суму вартості однієї години праці спираючись на доходи користувача. А замовник матиме можливість оцінити продуктивність праці у період виконання роботи та перевірити правдивість наданих даних виконавцем що-до ціни роботи. З цього маємо, що обидві сторони у виграші при використанні цього алгоритму. Виконавець отримує справедливу оплату, а замовник прозорість ціни за виконання замовлення.

UML діаграму роботи алгоритму зображено на рисунку 1.

При старті роботи алгоритм очікує надання інформації, для початкових розрахунків, де C – кількість часу витраченого на виконання W роботи у годинах , M – кількість отриманих коштів зв виконану W роботу за заданого часу у гривні. Проводиться розрахунок вартості однієї години праці P, P = C/M. Надалі йде перевірка чи не надійшли нові данні за час розрахунку, якщо «ні» то отримуймо результат та очікуймо нові данні, якщо «так» оновлюймо масив даних C та M. Робимо розрахунок усіх наявних на цей момент даних : $P = (C_{n-k}+\ldots+ C_n)/(M_{n-k}+\ldots+ M_n)$. Перевіряймо чи не надійшли нові данні, якщо «ні» отримаймо результат, якщо «так» повертаємось до кроку з оновленням даних та повторно виконуймо дії у циклі.

Таким чином маємо точне значення вартості години праці P у конкретний момент часу, що дозволить більш практично їм користуватись. За допомогою алгоритму буде вирішена головна проблема грошових відносин

між замовником та виконавцем, а саме справедлива оцінка виконаної роботи відносно двох сторін. Також алгоритм може зможе вплинути на оптимізацію дня для будь-якої любини, завдяки йому стане можливим оцінити свій час на виконання завдання та порівняти його з пропозиціями на ринку, слідкуючи з результатів оцінки та порівняння зробити вибір в ту чи іншу сторону, що буде більш раціональним, виконати роботу власними зусиллями або передоручити іншому виконавцю, саме у вашому випадку.

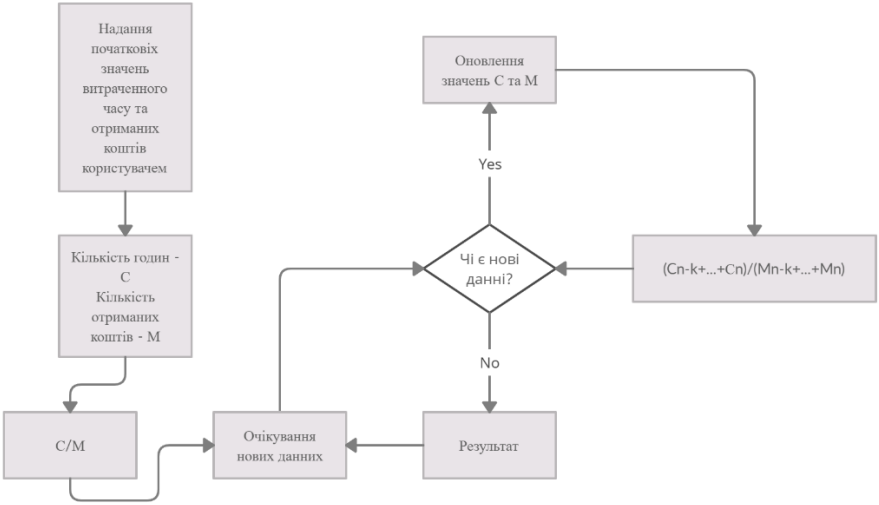


Рисунок 1 – UML діаграма алгоритму розрахунку вартості робочого часу

СПИСОК ВИКОРИСТАННОЇ ЛІТЕРАТУРИ

1. Сайт Cashgo [Електроний ресурс]. – Режим доступу: http://cashgo.pp.ua/publ/19-1-0-62;

2. Сайт Smarttender [Електроний ресурс]. – Режим доступу: https://smarttender.biz/terminy/view/zamovnyk/;

3. Сайт Sum [Електроний ресурс]. – Режим доступу: http://sum.in.ua/s/vykonavecj#:~:text=%D0%A2%D0%BB%D1%83%D0%BC%D0%B0%D1%87%D0%B5%D0%BD%D0%BD%D1%8F%2C%20%D0%B7%D0%BD%D0%B0%D1%87%D0%B5%D0%BD%D0%BD%D1%8F%20%D1%81.

УДК 004.4:004.93
Неруш Б.С.[1], Паромова Т. О.[2], Хохлов М.М.[2]
[1] студ. гр. КНТ-528 НУ «Запорізька політехніка»
[2] старш. викл. НУ «Запорізька політехніка»

СИСТЕМИ КОНТРОЛЮ ВЕРСІЙ ПРИ РОЗРОБЦІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Сьогодні процес створення програмного забезпечення стає все складнішим. Тому виникає проблема з обміну версій файлів проекту між розробниками. В процесі розробки можуть з'являтися нові ідеї, як слід нові версії програми. Для рішення цих проблем використовують системи контролю версій (СКВ), що забезпечують систематизацію версій, які об'єднуються при редагуванні і спільній роботі. Система управління версіями дозволяє зберігати кілька версій одного і того ж документа (початкового коду програми, скрипту, веб-сторінки, вебсайту, 3D-моделі, текстового документу тощо), при необхідності повертатися до попередніх версій, визначати, хто і коли зробив ту чи іншу зміну, і багато іншого. Цей інструмент дозволяє одночасно, не заважаючи один одному, проводити роботу над груповими проектами.

Сьогодні існує чи мало таких систем з відкритим вихідним кодом, наприклад клієнт- серверні системи (Concurrent Versions System (CVS), OpenCVS, Subversion (SVN)), розподілені системи (Git, Mercurial).

Рисунок 1- Клієнт- серверна СКВ

Такі системи мають єдиний сервер, який містить всі версії файлів, і деяку кількість клієнтів, які отримують файли з цього централізованого сховища і там же їх зберігають. Але вихід сервера з ладу обернеться втратою всіх даних. Крім того, в таких системах може бути ускладнена одночасна робота декількох розробників над одним файлом.

До систем типу Клієнт- сервер відноситься Subversion («SVN»). Це вільна централізована система управління версіями. SVN має загальний сервер, де зберігаються всі зміни, і розробники оновлюють зміни там. Тому SVN вимагає постійного підключення до інтернету для збереження змін.

Однією з широко розповсюджених є система Git, яка є однією з найефективніших, надійних і високопродуктивних систем керування версіями. Крім того, це повністю безкоштовне програмне забезпечення. Git підтримує багато ОС, таких як Mac, Linux, Windows і Solaris. Завантажити відповідні дистрибутиви можна з офіційного сайту Git.

Рисунок 2- Розподілена СКВ

Система контролю версій Git надає гнучкі засоби розробки, які базуються на відгалуженні та злитті гілок. Git дозволяє зберігати попередні версії файлів та завантажувати їх за потребою, зберігати повну інформацію про версію кожного з файлів, а також повну структуру проекту на всіх стадіях розробки. Це забезпечує декільком розробникам можливість працювати над своїм файлом, не заважаючи іншим і не псуючи основну гілку. Всі файли зберігаються у репозиторії, в середині якого можуть бути

створені паралельні лінії розробки - гілки. Гілка - це напрям розробки, який існує незалежно від іншого напряму, але має з ним спільну історію. Гілка починається як копія чогось і, рухаючись з цієї точки, створює свою власну історію. Git зберігає дані набором знімків. Кожного разу при фіксації поточної версії проекту Git зберігає знімок того, як виглядають всі файли проекту. Для кожного файлу, якій відстежується, Git зберігає розмір, час створення і останньої зміни. Більшість дій можна виконувати на локальній файловій системі без використання інтернет підключення. Вся історія змін зберігається спочатку локально і при необхідності вивантажується у окремий репозиторій.

В своїй базі Git зберігає все по хешам файлів. Як хеш-функція використовується SHA-1. Перед кожним збереженням файлів обчислюється значення хеш-функції SHA-1 для файлу і отриманий хеш стає індексом файлу. Використовуючи хеш Git легко відслідковує зміни в файлах. Існує кілька підходів до використання системи Git, серед яких, найпоширенішою є так звана концепція gitflow — центральний репозитарій працює з двома паралельними між собою гілками: master і develop. Гілка Master містить

версію коду, яка найімовірніше буде запроваджена у виробництво. Паралельно до основної гілки (master) існує ще робоча гілка Develop, яка використовується для розробки нової версії програмного забезпечення та тестування його стабільності.

Система Git дозволяє використовувати віддалений репозиторій. Для цього потрібно створити Git-сервер або скористатися хостингом репозиторіїв. Одним з найпопулярніших є github.com. Віддалені репозиторії дозволяють спростити контроль версій над спільним проектом. Кожен розробник може вести свою гілку і при необхідності додавати свої зміни в основну гілку.

Крім цього багато компаній використовують git як інструмент клонування проекту. Тобто, розробники додають свої зміни на проміжний репозиторій, після чого тестують код і після знаходження багів і їх виправлення перевірений код клонується на робочий сервер.

До переваг Git слід віднести:

1. можливість обміну оновлень файлів між учасниками розробки;
2. не потребує постійного інтернет з'єднання, у кожного своя копія сховища;
3. всі зміни у файлах відбуваються локально (всі програми можуть переглядатись іншими учасниками процесу) і тільки при необхідності завантажуються в загальний репозиторій;
4. висока швидкість роботи (Git не зберігає копії файлів, а зберігає історію змін.)

До недоліків системи Git можна віднести складність вивчення системи.

На основі проведеного аналізу слід зробити висновки, що використання систем контролю версій значно прискорює процес розробки програмного забезпечення, при цьому слід вибирати розподілену систему Git.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Гриценко В. Г. Використання системи управління версіями GIT для організації командної роботи над IT проектом / В. Г. Гриценко, О. М. Подолян // Інформаційні технології і засоби навчання. - 2014. - Т. 39, вип. 1. - С. 250-263.

2. Переклад українською книги «Pro Git» Скотта Чакона (Scott Chacon) та Бена Страуба (Ben Straub) [Електроний ресурс]. – Режим доступу: http://git-scm.com/book/uk/v2

3. Apache Subversion Community Guide (aka "HACKING") [Електроний ресурс]. – Режим доступу: http://subversion.apache.org/docs/community-guide/

4. Bryan O'Sullivan. Mercurial: The Definitive Guide. — O'Reilly Media, Inc., 2009. — 288 с. — ISBN 9780596800673.

5. Ресурс Git [Електроний ресурс]. – Режим доступу: https://ru.hexlet.io/courses/intro_to_git

УДК 004.658, 004.3, 519.688
Мороз А.Р. [1], Сологубов І.Я. [1], Зеленьова І.Я. [2]
[1]студ. гр. КНТ-528, НУ "Запорізька політехніка"
[2] к.т.н., доц. НУ "Запорізька політехніка"

ВИКОРИСТАННЯ FPGA ДЛЯ ПРОЕКТУВАННЯ ГІБРИДНИХ СИСТЕМ ПРИСКОРЕНОЇ ОБРОБКИ БАЗ ДАНИХ

Ми живемо в еру технологій та інформації і саме інформація відіграє ключову роль в житті будь-якої людини. Інформацією є будь-які дані того чи іншого роду, які несуть смислове навантаження. Для успішного використання і розвитку інформації, її необхідно коректно зберігати і каталогізувати.

Наразі актуальною проблемою в роботі з базами даних скорочення часу виконання операції вибірки відповідно сучасним вимогам до швидкості. Наріжним каменем цієї проблеми є саме процесор і його засоби роботи з базою даних.

Наприклад, розглянемо алгоритм виконання запиту процесором в MySQL. При отриманні запиту процесор розпізнає критерії вибірки і починає через функцію підрядника порівнювати дані критерії з кожним рядком в базі даних [1]. Сучасні процесори мають високу продуктивність, однак при великій кількості рядків цей показник часто не задовольняє вимогам до швидкодії системи.

Одним із рішень задачі прискорення обробки великої кількості даних є спосіб розпаралелювання обчислень. Суть даного способу полягає в розбитті всього обсягу інформації на незалежні блоки, які можуть оброблятися одночасно. Так як обробка кожного рядка відбувається незалежно по константній умові, такий спосіб є прийнятним рішенням даної проблеми.

Паралельні обчислення можна реалізувати як програмно, так і апаратно, однак апаратний спосіб має значні переваги в швидкості виконання і ефективності в цілому. Реалізацією може служити апаратний прискорювач (БД-прискорювач) в сучасному елементному базисі FPGA.

Для апаратної реалізації БД-прискорювача ідеально підходять саме FPGA [2], які, на відміну від універсального процесора, можна перепрограмувати відповідно до особливостей розв'язуваної обчислювальної задачі [3]. Вибірка бази даних як раз і є однією з таких задач. Важливими відмінностями FPGA також є знижене енергоспоживання в розрахунку на одиницю обчислювальної потужності, а також архітектура з можливістю паралельного виконання множини векторних операцій одночасно – так звана масивно-паралельна дрібнозерниста архітектура. Число логічних вентилів в чіпі FPGA може досягати декількох мільйонів, що підходить для вирішення задач з великим обсягом даних.

В сучасному проектуванні цифрових систем широко використовуються гібридні рішення з використанням універсальних CPU та FPGA для виконання спеціалізованих функцій. При цьому гібридізація може бути виконана як на рівні зовнішніх функціональних зв'язків між CPU та FPGA, так і внутрішня – процесорні ядра у складі FPGA.

Пропонується гібридна система «процесор+прискорювач», в якій процесор отримує запит, а також розбиває його на ключові складові:

1. Об'єкти вибірки (поля, атрибути)
2. Джерело вибірки (таблиця)
3. Умова вибірки

БД-прискорювач отримує надіслані йому від процесора інструкції і вже безпосередньо починає роботу з базою даних без участі центрального процесора для управління передачею даних. Проте на відміну від центрального процесора, БД-прискорювач може використовувати кілька ліній одночасно. Наприклад, будемо вважати, що у БД-прискорювача 200 доступних ліній, а це означає, що він може опитувати по 200 рядків бази даних за такт, при умові, що порядок вибірки не має значення.

При порівнянні алгоритмів роботи з базою даних процесора і апаратного БД-прискорювача слід зазначити, що хоч алгоритм роботи прискорювача і вимагає деяких додаткових операцій для початку вибірки, сама вибірка виконується, в даному випадку, в 200 разів швидше.

Обсяг каталогізованих даних, в тому числі і бази даних різних корпорацій та інших формувань всякого роду, за найскромнішими підрахунками до 2025 року збільшиться в 50 разів у порівнянні з 2010 роком. При такому прирості даних оптимізація роботи з базами даних за допомогою паралельних обчислень і апаратних прискорювачів, які підвищують продуктивність багаторазово, роблять цю технологію вкрай перспективною для подальшого вивчення і розвитку.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Хомоненко А.Д., Цыганков В.М., Мальцев М.Г. Базы данных. Учебник для высших учебных заведений - [6-е изд.]. - СПб.:КОРОНА-Век, 2009. – 734 с.

2. Woods R. FPGA-based Implementation of Signal Processing Systems / Woods R., John McAllister, Gaye Lightbody, Ying Yi // 2nd Edition - John Wiley & Sons, 2017.- 356 с.

3. Пример программирования FPGA-ускорителя [Електроний ресурс]. – Режим доступу: https://habr.com/ru/company/selectel/blog/418403/

УДК 004.912
Тягунова М.Ю.[1], Бережко Я.А. [2]
[1] доц. НУ «Запорізька політехніка»
[2] студ. гр. КНТ-527 НУ «Запорізька політехніка»

ОРГАНІЗАЦІЯ ЕЛЕКТРОННОЇ КОМЕРЦІЇ НА БАЗІ ІНСТАГРАМ

Соціальні мережі зробили велику революцію в тому, як ми приймаємо інформацію, як проводимо спілкування, як купуємо товари і багато іншого.

Взаємодія електронної комерції та соціальних мереж дало великий розвиток соціальної комерції.

У той же час, соціальна комерція - це різновид електронної комерції, продаж різних послуг або товарів через соціальні мережі.

Популярними соціальними мережами для електронної комерції є Фейсбук, Твітер, Телеграм та Інстаграм.

Кожна з цих систем має свої переваги та недоліки. Та, як показують останні дослідження у даному напрямі, саме Інстаграм є найперспективнішою платформою для продажу товарів та послуг малого бізнесу. І вагомим для досягнення великого об'єму продажу є правильно розроблений інтерфес сторінки профілю.

Інстаграм - це соціальна мережа для обміну фотографіями, відеозаписами, повідомленнями. Instagram має великий асортимент можливостей, тому він такий популярний [1]. Завдяки своїй популярності та масштабності, Інстаграм дає змогу заробляти різним брендам, компаніям та окремим особистостям. Зараз є декілька видів аккаунтів, звичайний аккаунт користувача, бізнес-аккаунт, пабліки, блогери, боти [2]. І тому дуже багато компаній і брендів долучилися до Інстаграму, бо там не тільки зручно вести бізнес, але й можна побачити статистику - охват, огляди кожного посту тощо.

Інстаграм підійде для багатьох починаючих бізнесменів, продавати можно все що завгодно. Але більше за все він підійде для салонів краси, арт-студій, ресторанів/кафе, магазинів одягу, магазинів аксесуарів тощо [2].

Але мати бізнес-акаунт та фотографії/відео товара або послуг замало. Треба якось зацікавити користувача, щоб він перейшов до вашого аккаунту та придбав якийсь товар або послугу. Потрібен свій стиль для подачі інформації - привабливий дизайн аккаунта. Це дуже важливо у наш час, ніхто не захоче переходити на аккаунт, який ніяк не чіпляє. Немає красивого оформлення - немає клієнтів. Тому дуже важливо мати не тільки потрібні товари або послуги, а й привабливий дизайн. Саме Інстаграм дає змогу це зробити.

Для початку необхідно визначитися з настроєм сторінки. Це може бути як похмуро темне настрій, так і веселий яскравий настрій. Тому вибрати

настрій потрібно відразу, бо від нього будуть залежати подальші дії: вибір шрифтів, фотографій, шаблонів, фільтрів і так далі.

Все повинно бути в єдиній кольоровій гамі, яка буде добре поєднуватися між собою. Візуально стрічка повинна бути плавною і одноманітною. Людське око прагне до пошуку закономірностей, тому ми пов'язуємо воєдино різні фотографії в стрічці [3].

Профіль можна оформляти по різному, є пару популярних видів оформлення сторінки Інстаграм:

— шаховий порядок (в цьому виді графічні публікації йдуть через одну з фотографіями);

— стовпчик (в цьому виді пости візуально будуються в стовпчик);

— обрізані фотографії або фото з великою білою рамкою зверху та знизу;

— нескінченна стрічка (увесь профіль виглядає, як одна велика картинка).

Окрім красивого профілю потрібно мати якісні та презентабельні фотографії, які також повинні підходити під загальний вид оформлення та мати однаковий стиль. Та треба притримуватись сталості, Інстаграм профіль виглядає набагато краще, якщо там повторюються шаблони, фільтри, шрифти тощо [3].

Особливості Інстаграм:

— є хештеги для пошуку;

— є бізнес-аккаунт;

— місце для збереження чужих фотографій;

— історії;

— можливість брендам перенаправляти користувачів на свій сайт;

— видно статистику охоплення аккаунта тощо;

— можливість красиво оформити аккаунт для підвищення охвату.

Інстаграм має дуже багато зручних функцій для бізнесу, дає можливість красиво і правильно оформити сторінку. Та має велику аудиторію різного віку. Тому Інстаграм краще за все підійде для малого бізнесу.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Тенденции social commerce в 2019 году [Електронный ресурс]. – Режим доступа: https://spark.ru/startup/angift/blog/46404/tendentsii-social-commerce-v-2019-godu

2. Как выбрать социальную сеть для бизнеса [Електронный ресурс]. – Режим доступа: https://netology.ru/blog/vybor-smm;

3. 11 советов для бизнеса по оформлению Инстаграм [Електронный ресурс]. – Режим доступа: https://zen.yandex.ru/media/canva/11-sovetov-dlia-biznesa-po-oformleniiu-instagram-5d570eccf8ea6700adec9974.

УДК 004.05
Шумченко С.Г[1], Киричек Г.Г.[2]
[1] студ. гр. КНТ-527, НУ «Запорізька політехніка»
[2] канд. техн. наук, доц., НУ «Запорізька політехніка»

АНАЛІЗ ТЕХНОЛОГІЙ ДЛЯ СТВОРЕННЯ ДИНАМІЧНИХ ВЕБДОДАТКІВ

У сучасному світі веброзробок найбільшу роль грає швидкість, простота і можливості масштабування додатку. На даний час існує багато технологій, які використовуються для розробки веб-додатків та мають готові рішення: бібліотеки та фреймворки [1].

Метою роботи є проведення досліджень і визначення технології, яку найдоцільніше використовувати для розробки вебсистем різного розміру та призначення. Об’єктом дослідження є бібліотеки та фреймворки. Предметом – архітектура, методи, властивості та особливості їх застосування.

Для досягнення поставленої мети проведені порівняння за наступними характеристиками: бібліотека/фреймворк; вбудована архітектура; Single Page Application/Multi Page Application; переваги для інтерфейсів різного розміру; функціональність для роботи з DOM (Document Object Model), тобто швидкість розробки, представлення кінцевого результату, зворотня сумісність, поріг входження, тощо.

Бібліотека є комплексом підпрограм, які використовуються при розробці програмного забезпечення [2]. А фреймворк - програмна платформа, яка створює структуру вебдодатку та полегшує розробку різних компонентів системи [3]. Розглянемо дві популярні бібліотеки: React, jQuery та фреймворк Angular, визначимо їх переваги та недоліки, враховуючи те, що кожен з них створений для вирішення конкретних проблем.

JQuery є швидкою та багатофункціональною бібліотекою JavaScript. Вона спрощує роботу з документами HTML, обробку подій, анімацію та робить Ajax запити на сервер простішими за допомогою API. Завдяки поєднанню універсальності та розширюваності, jQuery змінив спосіб написання JavaScript. React є більш сучасною JavaScript бібліотекою, яка дозволяє створювати інтерактивні інтерфейси та односторінкові додатки (Single Page Applications). Дія у вкладці браузера не потребує перезавантаження сторінки, яка просто динамічно перезаписується, не завантажуючи нові сторінки. React одразу завантажує увесь скрипт, запитує у

сервера дані, які змінилися та вносить зміни в ті місця, які повинні оновитись. У випадку з jQuery завжди при взаємодії користувача відбувається перезавантаження сторінки [4].

Для маленьких додатків, які складаються з невеликої кількості компонентів, логіки та сторінок, краще використовувати jQuery, через його простоту та швидкість. JQuery дозволяє швидко створювати анімації та працювати з елементами DOM. Для landing page (спеціальний односторінковий сайт) jQuery також є найкращім вибором. React майже завжди може замінити jQuery, тому сучасні розробники використовують React. Виходячи з цього більш доцільним є порівняння React з іншою сучасною технологією Angular. Але React є бібліотекою, а Angular – фреймворком для створення динамічних SPA (Single Page Application) веб-додатків. Він використовує TypeScript та не має вбудованої типізації. Це на даний час є його перевагою над React [5]. Та щоб зрозуміти в яких випадках краще використовувати Angular, а в яких React, порівняємо їх за основними характеристиками.

Angular побудований на архітектурному патерні MVC (Model-View-Controller). Він відповідає: за дані та визначає структуру додатку; за зовнішній вигляд додатку та способи його використання; за зв'язок між моделлю та представленням, тобто як реагує сайт на дії користувача. А React має лише вбудовані функції для роботи з View. Тому в React потрібно створювати свою архітектуру додатку або використовувати Flux-архітектуру [6].

Flux-архітектура потрібна для керування станами на рівні усього додатку. Це часто потрібно для багаторівневих складних додатків. React-додаток який має деревоподібну структуру компонентів та їх вкладеність. При занадто великій вкладеності компонентів, важко керувати даними: потрібно створювати контексти для їх передачі по одній великій гілці. Flux-архітектура допомагає зберігати всі стани додатку, обробляти та отримувати до них доступ з різних частині додатку (рис. 1).

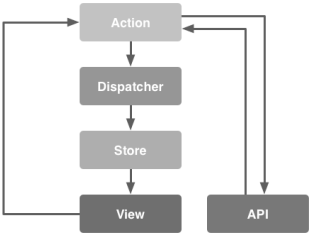


Рисунок 1 - Схема роботи Flux-архітектури.

У випадку з Angular, доступ до даних можна отримати за допомогою Angular сервісів, які є постачальниками даних, або за допомогою Angular Output декоратору, що дозволяє у батьківському компоненті відслідковувати зміни, які вносить дочірній. Тобто, завдяки впровадженій архітектурі, дочірній компонент маніпулює даними батьківських компонентів.

Що стосується відображення кінцевого результату: React створює копію DOM, обробляє її, а потім порівнює результат з початковою версією та відображає тільки ті частини сторінки, які відрізняються від результатів обробки. Це прискорює завантаження та відображення сторінки. React і Angular підходять по-різному до обробки DOM, що і впливає на відображення кінцевої сторінки. За рендерінг DOM сторінки в Angular відповідає браузер, а за створення директив, завантаження коду та сервісів – сервер [7]. Тобто, React відображає сторінки швидше, що є критичним для великих систем.

Оновлення архітектури для додавання нових модулів та бібліотек є важливою. Різниця між Angular та React у неможливості перейти з однієї версії Angular на іншу без наслідків. Модуль, який застосовувався у старій версії Angular можуть просто замінити на інший, що потребує використання нового коду. Якщо ви плануєте вдосконалити проект, додавши нові функції, React є найкращим вибором, тому що він має повну зворотну сумісність, яка дозволяє додавати різні версії бібліотек до програм та оновлювати старі [8]. Окрім того, обидві технології на початку розробки потребують високого рівня знань JavaScript, але увійти у React легше, бо він частіше використовує чистий JavaScript синтаксис. Angular використовує двосторонній підхід прив’язки, що автоматично змінює стан моделі при змінах в елементі інтерфейсу, а React використовує односторонній підхід, тому в Angular для досягнення потрібного результату необхідно менше дій.

При веброзробках використовується багато вебтехнологій, які часто заміняють одна одну. JQuery є застарілою бібліотекою, яка має перевагу у швидкості розробки маленьких, неодносторінкових додатків. React і Angular є чудовими варіантами для односторінкових додатків, мають невеликий поріг входження, багато переваг та небагато недоліків, тому вибір серед них потрібно робити, виходячи з потреб, функціональності та зручності використання.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Wohlgethan, E. SupportingWeb Development Decisions by Comparing Three Major JavaScript Frameworks: Angular, React and Vue.js /E. Wohlgethan //дис. – Hochschule für Angewandte Wissenschaften Hamburg, 2018. – 84 p.

2. jQuery, [Електронный ресурс]. – Режим доступу: https://jquery.com/.

3. Le, T. Implementation of React-Redux in web application /T. Le, N. Tran. – 2020. – 46 p.

4. React, [Електронный ресурс]. – Режим доступу: https://reactjs.org/.
5. Fedosejev, A. React. js essentials /A. Fedosejev. – Packt Publishing Ltd, 2015. – 179 p.
6. Freeman, A. Understanding React /A. Freeman //Pro React 16. – Apress, Berkeley, CA, 2019. – P. 31-36.
7. Saks, E. JavaScript Frameworks: Angular vs React vs Vue /E. Saks. – 2019. – 47 p.
8. Angular, [Електронный ресурс]. – Режим доступу: https://angular.io/.

УДК 004.07
Щетінін М.О.[1], Киричек Г.Г.[2]
[1] студ. гр. КНТ-510м, НУ «Запорізька політехніка»
[2] канд. техн. наук, доц., НУ «Запорізька політехніка»

СИСТЕМА УПРАВЛІННЯ КОНФІГУРАЦІЄЮ СЕРВЕРІВ НА ОСНОВІ ANSIBLE

Розвиток вебтехнологій протягом останніх двох століть та всеосяжний світовий карантин 2020-2021 років призвів до значної популяризації вебхостингу, адже все більше приватних осіб та, особливо, підприємців зацікавилися можливістю займатися комерційною діяльністю у мережі Інтернет. Такий сплеск популярності робить розгортання нових серверів вручну дуже неефективним через значні затрати часу, тому більшість середніх, великих та навіть малих хостинг-компаній переходять або вже перейшли на розгортання серверів за допомогою систем управління конфігурацією.

Ці системи автоматизують і спрощують складні, виснажливі операції, які постійно повторюються та які необхідно виконувати при черговому розгортанні серверу. Це значно економить час при налаштуванні або внесенні змін до конфігурації великої кількості серверів.

Метою дослідження є проведення аналізу систем управління конфігурацією серверів та вибір методів проектування і конфігурування віртуальних серверів на основі операційної системи (ОС) CentOS Stream 8. Об’єктом є процес вибору методів розгортання серверів для дослідження систем управління конфігурацією. Предметом – моделі, методи, інструментальні та програмні засоби забезпечення дослідження систем управління конфігурацією. Кінцевою метою є створення практичного прикладу застосування систем управління конфігурацією серверів для розгортання одного з віртуальних серверів на основі ОС CentOS Stream 8. Для цього були придбані два віртуальні сервери у одного з хостинг-провайдерів.

При проведенні аналізу та задля порівняння, обрано наступні системи: Ansible, Puppet та Chef [1, 2].

Ansible – це простий механізм, який автоматизує надання хмарних послуг, управління конфігурацією, розгортання додатків, внутрішньо-сервісне оркестрування та багато іншого. Він розроблений для багаторівневого розгортання з першого дня та моделює інфраструктуру, описуючи те, як системи взаємопов'язані, а не просто керує однією системою одночасно. У ньому не використовуються агенти та додаткова власна інфраструктура безпеки, тому його легко розгорнути. І найголовніше, він використовує дуже просту мову (YAML, у формі Ansible Playbooks), що дозволяє описувати завдання з автоматизації мовою, схожу на просту англійську [3, 4].

Puppet допомагає керувати серверною інфраструктурою. При його використанні, визначається бажаний стан систем у керованій інфраструктурі, що відбувається за допомогою написання коду інфраструктури доменною мовою Puppet (DSL), яка сумісна з широким спектром пристроїв та операційних систем. Код Puppet є декларативним, а це означає, що він описує бажаний стан систем, а не кроки, необхідні для досягнення цього стану. Puppet автоматизує процес приведення систем у бажаний стан та фіксує, що вони залишаються у цьому стані. Puppet робить це, використовуючи основний сервер Puppet та агента Puppet. Основний сервер Puppet - це сервер, на якому зберігається код, що визначає бажаний стан систем, а агент Puppet переводить цей код у команди, які виконуються на вказаних системах. Цей процес називається запуском маріонеток [5].

Chef є системою управління конфігурацією, яка використовується для автоматизації розгортання інфраструктури. Вона розроблена на основі мови Ruby DSL та має можливість інтегрування з будь-якою хмарною технологією. Ці два показника вважаються найголовнішими плюсами системи. До мінусів відносять складність впровадження та підтримки через недостатньо якісну документацію та орієнтованість на розробників, які мають досвід з мовою Ruby, а не на системних адміністраторів [6].

Після порівняння популярних систем управління конфігурацією, обрано Ansible за більш зручну технологію роботи, що не потребує агента, повну сумісність з ОС CentOS Stream 8, наявність якісної документації та використання легких для розуміння мов YAML та Python. Ansible фокусується на оптимізації і швидкості, і не вимагає установки агентів на керовані вузли - всі функції здійснюються за SSH. Ansible написаний на Python, на відміну від Puppet і Chef, заснованих на Ruby.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. How Ansible Works. [Електронный ресурс]. – Режим доступу: https://www.ansible.com/overview/how-ansible-works/.

2. Обзор: Puppet, Chef, Ansible, Salt. [Електронный ресурс]. – Режим доступу до ресурсу: https://habr.com/ru/post/211306/.

3. Anufrijevs V. Virtual Private Server Configuration Management Tools Research in the Network Infrastructure of JSC. - Balticom, 2020. – 124 p.

4. Palma S.D., Nucci D.Di., Palomba F., Tamburri D.A. Towards a catalogue of software quality metrics for infrastructure code // Journal of Systems and Software. – 2020. – pp. 110726.

5. Introduction to Puppet. [Електронный ресурс]. – Режим доступу: https://puppet.com/docs/puppet/7.4/puppet_overview.html.

6. Chef Overview. [Електронный ресурс]. – Режим доступу: https://www.tutorialspoint.com/chef/chef_overview.htm.

УДК 004.4

Куцак С.О.[1], Киричек Г.Г.[2]
[1] студ. гр. КНТ-517, НУ «Запорізька політехніка»
[2] канд. техн. наук, доц. НУ «Запорізька політехніка»

ПОРІВНЯЛЬНИЙ АНАЛІЗ ІНСТРУМЕНТАЛЬНИХ ЗАСОБІВ МОБІЛЬНОЇ РОЗРОБКИ

Останні кілька років у світі спостерігається все більше поширення пристроїв на базі мобільних платформ. З'являються нові мобільні платформи. У цей час основна частина ринку поділена між декількома мобільними платформами. З погляду різних аналітичних агенств, найбільше активно серед них у найближчі кілька років будуть розвиватися iphone, Android і Blackberry.

Однієї з перших тенденцій, що супроводжують швидкий розвиток мобільних платформ, став перенос на них практично всіх технологій, розвинених для настільних комп'ютерів. У розробці додатків намітилися кілька напрямків. Серед них варто відзначити ігрові додатки, додатки для роботи із соціальними мережами й бізнес додатки, орієнтовані на документообіг у середніх і великих компаніях. По способу розробки виділилися два основні виду додатків: "native" додатка й web-додатка. "Native" додатки створюються повністю на базі власного API мобільної платформи. Web-додатки включають клієнтський додаток, на основі невеликої кількості "native" коду, що одержує контент від вилученого серверного додатка, що й відображає його з використанням стандартних web-технологій, у тому числі HTML, Java Script. Вилучений серверний додаток може бути створене заново або може бути задіяний наявний серверний додаток з відомим API.

Нижче наведена таблиця 1 порівняння середовищ для розробки мобільних додатків. Основними критеріями, якими є: різноманітність мов програмування, зручність для користувача інтерфейсу, різноманітність для розробки мобільних платформ, монетизація середовища розробки.

Таблиця 1 - Порівняння середовищ для розробки мобільних додатків.

Середовище розробки	Мова програмування	Зручність інтерфейсу	Мобільні платформи для розробки	Плата
Android Studio	Java, C/C++, Delphi	+	Android	-
Eclipse IDE	Java, C/C++, PHP, Ruby, Python, Cobol	+	Android, IOS, windows phone, symbian	-
Intel Beacon Mountain	Java, C, C++	-	Android	-
XCode	C, C++, Objective-C, ObjectiveC, Swift, Java, AppleScript, Python, Ruby	+	IOS, OS X,	-

УДК 004.4
Куликовська Н.А.[1], Шмигля К.В.[2]
[1] асист. НУ «Запорізька політехніка»
[2] студ. гр. КНТ-519сп, НУ «Запорізька політехніка»

ПОРІВНЯННЯ СУЧАСНИХ ТЕХНОЛОГІЙ РОЗРОБКИ ВЕБ-САЙТІВ

Сучасні технології створення та підтримки веб-сайтів орієнтовані на платформи, що дозволяють ефективно керувати інформаційним наповненням і даними, які надходять від відвідувачів сайту. Як правило, такі рішення базуються на серверних технологіях типу ASP, ASP.NET, JSP, PHP або використовують готові потужні засоби для створення корпоративних сайтів, що орієнтовані на впровадження вказаних технологій.

Розглянемо детальніше популярні інформаційні технології.

Створення веб-сторінок за фрагментами серверного коду є технологією ASP, ASP.NET (Active Server Pages). Це розроблена Microsoft комерційно доступна технологія, за допомогою якої веб-майстер може самостійно формувати динамічно поновлювальні веб-сторінки. Характерною особливістю такої технології є можливість відокремлення функціональної частини розробки від процесів створення дизайну.

Технологія JSP (Java Server Pages) − це технологія створення серверних сторінок Java. Специфікація JSP є розширенням Java Servlet API для генерації динамічних веб-сторінок на веб-сервері. Така крос-платформа є

альтернативою технології ASP корпорації Microsoft. Специфікація Sun за назвою JSF (Java Server Faces) реалізує технологію JSP, що описує правила створення веб-додатків зі зручним для користувача інтерфейсом та орієнтована на розробку серверних компонентів створення інтерфейсу.

Серед інших популярних технологій, які реалізують створення веб-сторінок із фрагментами коду, виконуваного на сервері, виділимо некомерційну, вільно розповсюджувану технологію PHP (Personal Home Pages). Ця технологія заснована на використанні CGI-застосувань, що інтерпретують впроваджений у HTML-сторінку код на скриптовій мові. Головною особливістю мови PHP є її практичність. PHP надає програмісту інструмент для швидкого й ефективного вирішення поставлених завдань. Вона вирізняється винятковою гнучкістю до потреб розробника.

Результати проведеного аналізу дозволяють зробити висновок про наявність переваг експлуатаційних характеристик PHP-технології. Хоча PHP є досить молодою мовою, вона користується значною популярністю серед web-програмістів і сьогодні вважається мало чи не найпопулярнішою мовою для створення web-додатків (скриптів). Головними перевагами PHP вбачаємо практичність, легкість у застосуванні, ефективність, продуктивність та гнучкість.

УДК 004.4
Куликовська Н.А.[1], Грабовецька А.Е.[2]
[1] асист. НУ «Запорізька політехніка»
[2] студ. гр. КНТ-517, НУ «Запорізька політехніка»

АНАЛІЗ ЯКОСТІ ІНФОРМАЦІЙНОГО НАПОВНЕННЯ САЙТУ

На сьогоднішній день, перехід ринкових відносин на онлайн-простір, сприяв появі нового методу – контент-маркетингу. Одним із завдань якого є аналіз інформаційного наповнення сайту. Виходячи з цього, дослідження маркетингових компаній в даному напрямку показали, що для аналізу якості маркетингових комунікацій важливо оцінити, перш за все, ступінь сприйняття цільовою аудиторією інформаційного контенту. У зв'язку з цим слід здійснювати оцінку вмісту сайтів і зручності роботи користувачів з інформацією, розміщеною на інформаційному ресурсі. Треба зазначити, що, розглянуті джерела, надають тільки загальні принципи та методи здійснення аналізу контенту.

У проведених дослідженнях оцінюються такі параметри, як функціональність і дизайн. Деякі методики, використані в розглянутих дослідженнях, представляють простий експрес-аналіз, в основі інших лежать більш ємкі і складні алгоритми отримання підсумкових кількісних

характеристик і формування на їх основі рейтингів сайтів з однаковою тематикою. Рейтинговий підхід дозволяє проводити порівняльний аналіз сайтів, причому підсумкова кількісна міра рейтингу показує, наскільки краще або гірше показники відповідного сайту в порівнянні з лідером рейтингу.

Порівняння сайту з іншими не надає повної оцінки якості інформаційного наповнення. Тому проблема визначення основних критеріїв, за якими можна дати більш адекватну оцінку веб-сайту залишається не вирішеною.

Таким чином можна виділи такі п’ять основних напрямків аналізу якості інформаційного наповнення сайту:

1. Відповідність семантичному ядру сайту.

2. Розгляд сайту з точки зору відвідувачів – це визначає цінність для користувача контенту сторінок сайту, доступність і корисність поданих на сайті текстів.

3. Визначення кількості необхідного контенту. В даному випадку визначається оптимальна кількість сторінок сайту і аналізуються об’єми тексту на сторінках, як з точки зору пошукових систем, так і – відвідувачів.

4. Змістовний та лінгвістичний аспект ресурсу. Дуже важливим фактором є перевірка унікальності тексту на сторінках сайту, а також його орфографічних та стилістичних помилок.

5. Графічний аналіз включає в себе перевірку правильності оформлення текстів на сторінках, шрифти, заголовки, виділення, посилання тощо.

УДК 004.42
Точилін С.Д.[1], Тронькіна О.Е.[2]
[1]канд. фіз.-мат. наук, доц. НУ «Запорізька політехніка»
[2]студ. гр. КНТ-527, НУ «Запорізька політехніка»

WEB-ДОДАТОК - ЛАБОРАТОРНА РОБОТА «ВИЗНАЧЕННЯ ПРИСКОРЕННЯ ВІЛЬНОГО ПАДІННЯ ЗА ДОПОМОГОЮ МАТЕМАТИЧНОГО МАЯТНИКА»

У наш час у навчальний процес освітніх установ активно впроваджується дистанційне навчання. У цьому зв'язку розробка навчальних Web-додатків є актуальною задачею.

У даній роботі був розроблений Web-додаток (програма ST_Pendulum) - лабораторна робота «Визначення прискорення вільного падіння за допомогою математичного маятника». Програма ST_Pendulum (Java-аплет) вбудована в html-сторінку Pendulum.html, що розміщається на Web-сервері.

Вікно браузера зі сторінкою Pendulum.html зображено на рис.1.

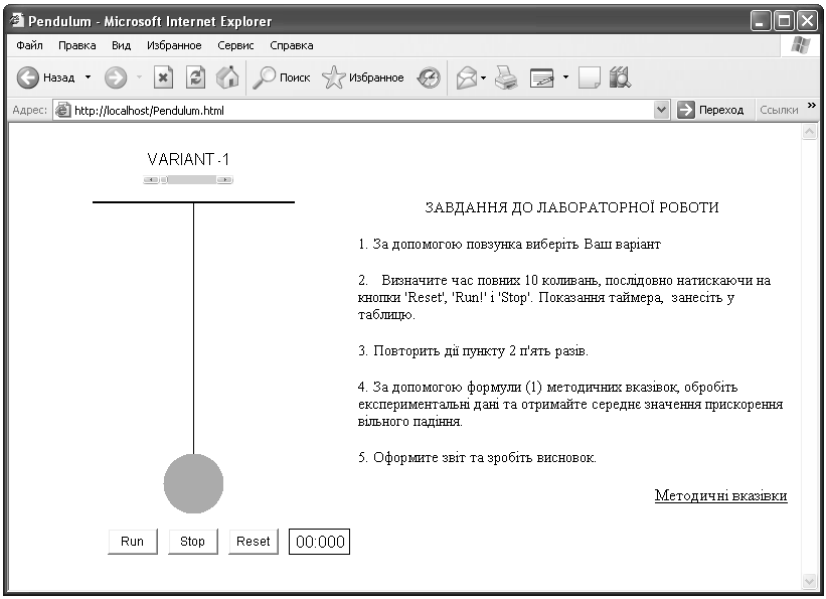


Рисунок 1 - Вікно браузера з html-сторінкою Pendulum.html.

Програма ST_Pendulum має графічний інтерфейс користувача. На початку виконання роботи студент за допомогою повзунка вибирає свій варіант. Після натискання на кнопку «Run» починається виконання експериментальної частини роботи - кулька починає коливатися. Час її коливального руху визначається таймером. Кнопка «Stop» зупиняє коливання кульки та фіксує час їх виконання. Кнопка «Reset» приводить додаток у вихідний стан.

На рис. 2 зображене вікно браузера при виконанні лабораторної роботи (обраний варіант № 10).

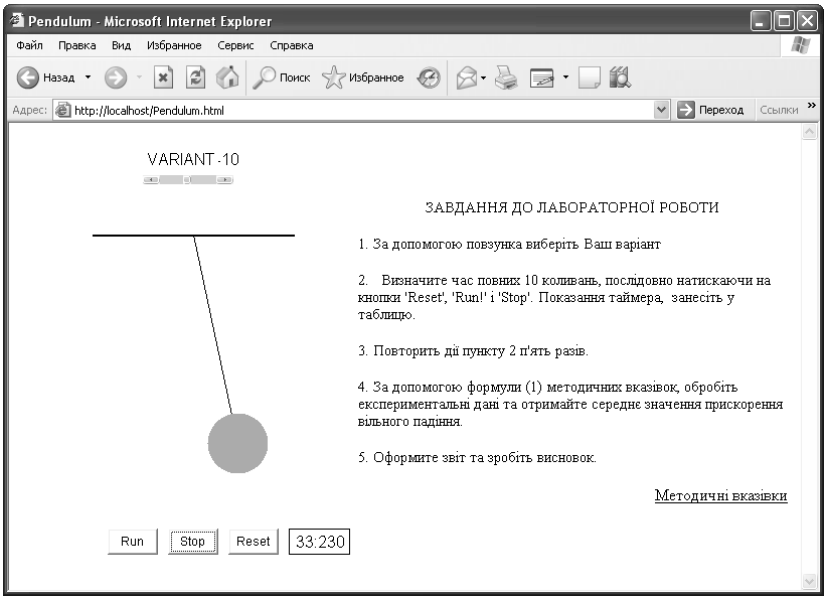


Рисунок 2 - Вікно браузера при виконанні лабораторної роботи

Розроблений Web-додаток може працювати, як у мережі Internet, так і в мережі intranet, а також на локальному комп'ютері.

Надалі передбачається модернізувати програму ST_Pendulum, зокрема, забезпечити можливість збереження результатів виконання лабораторної роботи у файлах формату *.csv.

УДК 004.42
Точилін С.Д.[1], Тронькіна О.Е.[2]
[1]канд. фіз.-мат. наук, доц. НУ «Запорізька політехніка»
[2]студ. гр. КНТ-527 НУ «Запорізька політехніка»

КОМП'ЮТЕРНА ПРОГРАМА ДЛЯ ОЦІНКИ ВАРТОСТІ ОБ'ЄКТІВ АВТОМОБІЛЬНОГО РИНКУ НА ОСНОВІ НЕЙРОННОЇ МЕРЕЖІ

У наш час нейромережеве програмування знаходить широке застосування в різних областях людської діяльності. При цьому розробка прикладних програм на основі нейронних мереж є актуальною задачею.

У даній роботі розроблена кросплатформна комп'ютерна програма ST_Valuation для оцінки вартості об'єктів автомобільного ринку на основі нейронної мережі.

Програма ST_Valuation створена за допомогою мови програмування Java™. При функціонуванні вона використовує алгоритм зворотного поширення помилки і має графічний інтерфейс користувача Він дозволяє вводити та зберігати дані для аналізу, а також одержувати до них доступ. Крім того, проводити навчання нейронної мережі і здійснювати оцінку вартості об'єктів автомобільного ринку з її допомогою.

Для зберігання даних програма використовує СКБД PostgreSQL, яка вільно розповсюджується.

При цьому нейронна мережа додатка ST_Valuation має три шари та використовує при своїй роботі чотири параметри для оцінки вартості об'єкта автомобільного ринку.

Верифікація роботи програми проводилося при зіставленні її оцінок вартості з аналогічними, які раніше були отримані для подібної нейронної мережі розробленої за допомогою аналітичної платформи Deductor Studio Academic.

Як з'ясувалося у цьому випадку, при роботі з повним обсягом тієї ж інформації та пройшовши подібне навчання, нейронні мережі додатка ST_Valuation та аналітичної платформи Deductor Studio Academic показували близькі значення вартості окремих об'єктів автомобільного ринку.

Надалі передбачається модернізувати розроблений Java-додаток, зокрема, забезпечити можливість оцінки об'єктів ринку автомобілів із числом параметрів більшим чим чотири.

УДК 004.42:303.724.32
Точилін С.Д.[1], Резця Д.А.[2]
[1]канд. фіз.-мат. наук, доц. НУ «Запорізька політехніка»
[2]студ. гр. КНТ-527 НУ «Запорізька політехніка»

ПРОГРАМА ДЛЯ АНАЛІЗУ ДАНИХ НА ОСНОВІ РІВНЯННЯ РЕГРЕСІЇ У ВИГЛЯДІ РІВНОСТОРОННЬОЇ ГІПЕРБОЛИ

Парний регресійний аналіз (ПРА) широко використовується прикладною статистикою. Парна регресія - модель, де середнє значення Y_r залежної змінної Y розглядається як функція однієї незалежної змінної X, тобто це рівняння вигляду: $Y_r = f(X)$.

При цьому у багатьох випадках при ПРА статистичних даних як рівняння регресії застосовується рівностороння гіпербола (РГ). Вона може

бути використана для характеристики зв'язку питомих витрат сировини, матеріалів, палива з об’ємом готової продукції, часом обігу товарів з величиною товарообігу не тільки на макрорівні, але й на макрорівні [1]. Відомі й інші випадки її застосування при регресійному аналізі (РА) даних (див., наприклад, [1]).

РА виконується, як правило, за допомогою комерційних комп'ютерних програм, які мають графічний інтерфейс користувача (Graphical User Interface - GUI) і в багатьох випадках жорстко прив'язані до певної платформи. При цьому розробка кросплатформих комп'ютерних програм з GUI для аналізу даних, зокрема, з застосуванням рівняння регресії у вигляді рівносторонньої гіперболи є актуальною проблемою.

У даній роботі, за допомогою мови програмування Java™, була розроблена кросплатформна програма ST_Hyperbola для РА з GUI, яка при роботі використовувала модель регресії у вигляді РГ. Знаходження її параметрів здійснювалося за допомогою метода найменших квадратів, а також метода Гаусса (при рішенні систем лінійних рівнянь).

На початку роботи із програмою за допомогою меню «File», дані для аналізу поміщалися в таблицю «Data». Рядки цієї таблиці також могли заповнюватися даними для X та Y, які розміщалися в полях вводу «X» та «Y or PV», за допомогою кнопки «Input» (поля «X» та «Y or PV» використовувалися і при прогнозуванні значень Y_r для відповідних аргументів X). Для видалення рядка, всіх рядків таблиці «Data» використовувалися кнопки «CRow» та «Clear», відповідно.

Запуск обробки даних здійснювався при натисканні на кнопку «Calc». При цьому програма розраховувала коефіцієнти рівняння регресії , середню помилку апроксимації, а також індекс детермінації, їх значення з'являлися в полях вводу «Coefficient A», «Coefficient B», «ME, %» та «R^2», відповідно. Також програма для аргументу X, який цікавив користувача, визначала прогнозовану величину PV моделі, і поміщала її в поле вводу «Y or PV». Крім того, розраховувалися і зіставлялися фактичне значення та критичне (для рівня значимості 0,05) F-критерію Фішера. Результат зіставлення з'являвся в полі вводу «F ? $F_{0.05}$».

У той час вихідні дані і розрахована залежність $Y_r = f(X)$ відображалися в графічному вигляді, а значення Y_r і залишків регресії, для всіх X з набору даних, розміщалися в таблиці «Data».

Вікно програми при роботі в тестовому режимі зображене на рис. 1.

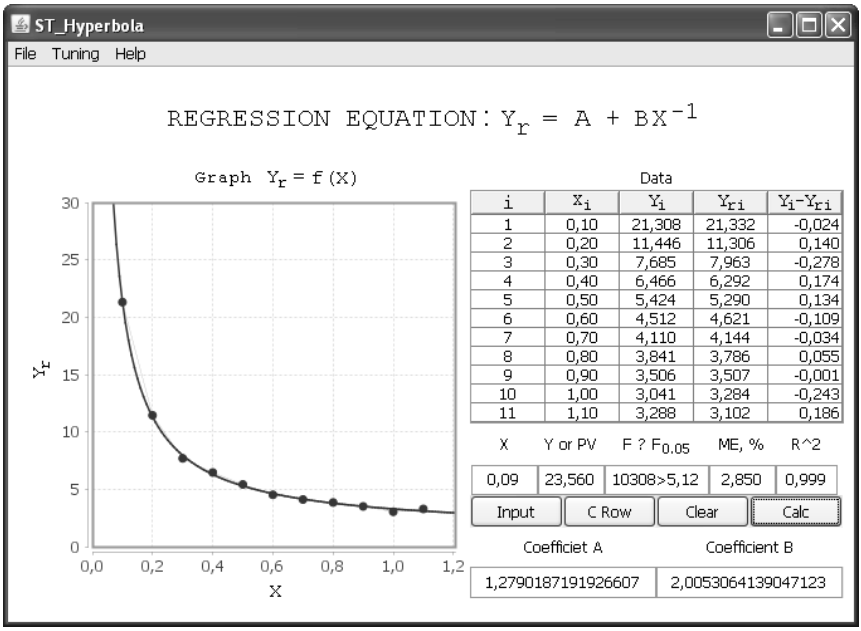


Рисунок 1 - Вікно програми ST_Hyperbola при регресійному аналізі даних.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Елисеева И.И. Эконометрика: учебник / И.И. Елисеева, С.В. Курышева, Т.В. Костеева и др. – М.: Финансы и статистика, 2007. – 576 с.

УДК 004.42
Точилін С.Д.[1], Волкова К.С.[2]
[1]канд. фіз.-мат. наук, доц. НУ «Запорізька політехніка»
[2]студ. гр. КНТ-527 НУ «Запорізька політехніка»

КРОСПЛАТФОРМНИЙ ВІРТУАЛЬНИЙ АВОМЕТР

Розробка кросплатформних віртуальних приладів (КВП) для електричних вимірів (ЕВ) є актуальною задачею.

На сучасному ринку електронних компонентів доступні мікроконтролерні плати Arduino. Їх можна використовувати як апаратну частину КВП для ЕВ. Із цією метою плати програмують, зокрема, за допомогою скетчів Arduino IDE, які зберігаються у файлах формату *.ino.

У той час для створення кросплатформних КВП із платами Arduino можна використовувати комп'ютерні додатки, які написані на мові програмування Java™.

У даній роботі був розроблений авометр для ЕВ на основі Arduino UNO та програм Avometer_DAS.ino і Avometer_GUI.class.

Основні характеристики авометра:

— Діапазон виміру напруги: 0-25 В.

— Діапазон виміру струму: 0-3 А (для виміру струму використовувався датчик струму GY-MAX471).

— Діапазон виміру опору: 0-10 кОм.

Програма Avometer_DAS.ino дозволяла використовувати Arduino UNO як систему збору даних ЕВ. Для відображення результатів вимірів і керування роботою авометра використовувався додаток Avometer_GUI.class із графічним інтерфейсом користувача, розроблений на Java. Вікно цієї програми, перед проведенням ЕВ, зображене на рис. 1.

Для вимірювання напруги, току або опору потрібно активувати перемикач з назвою «V», «A», «R», відповідно, та натиснути на кнопку «START». При цьому відбувається відображення їх розмірності та періодичне відображення їх значень у відповідних полях вводу. Натискання на кнопку «STOP» приводить до зупинки ЕВ, а активація кнопки «RESET» переводить прилад в вихідний стан.

За допомогою меню «Help» користувач має можливість отримати інформацію про розроблену програму та особливостях її функціонування.

На рис. 2, як приклад, зображене вікно програми Avometer_GUI.class у процесі ЕВ електричного струму.

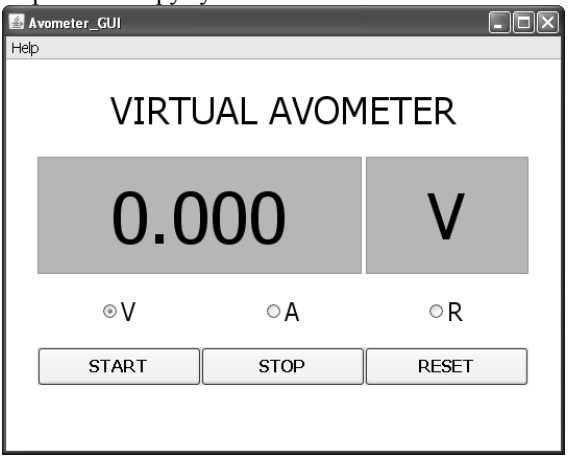


Рисунок 1 - Вікно програми Avometer_GUI.class перед проведенням ЕВ.

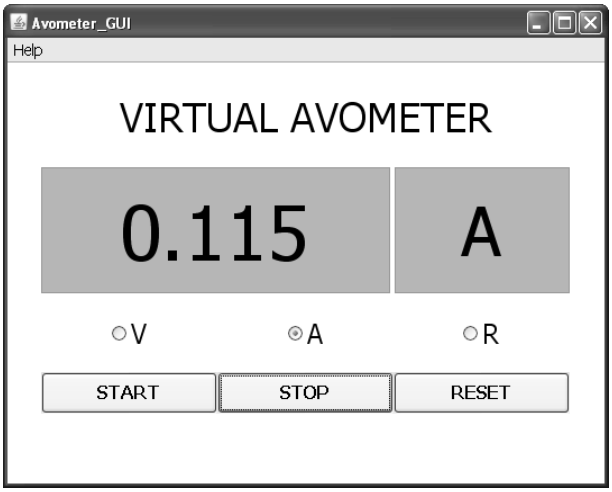


Рисунок 2 - Вікно програми Avometer_GUI.class при ЕВ струму.

Надалі передбачається модернізувати розроблений віртуальний авометр, зокрема, розширити діапазони ЕВ напруги та опору.

СЕКЦІЯ «ПРОГРАМНІ ЗАСОБИ»

УДК 004.93
Субботін С.О.[1], Гофман Є.О.[2], Корнієнко О.В.[3]
[1] зав. каф. НУ «Запорізька політехніка»
[2] с.н.с. НДЧ НУ «Запорізька політехніка»
[3] асп. НУ «Запорізька політехніка»

ПОБУДОВА НЕЙРОМЕРЕЖІ LVQ НА ОСНОВІ ДЕРЕВА РІШЕНЬ

Нейронна мережа (НМ) LVQ [1] складається з двох шарів: перший шар відображає вхідний вектор у кластери, які виділяються мережею під час навчання, а другий шар об'єднує групи кластерів першого рівня в класи, задані користувачем. Традиційний підхід до навчання НС LVQ [1] передбачає встановлення кількості кластерів випадковим чином або користувачем, а ініціалізацію ваг – випадковим чином, що призводить на практиці до значних витрат часу на побудову моделі.

Метою даної роботи було створення методу, який дозволить прискорити процес побудови нейромоделей на основі мереж LVQ.

Формально запропонований метод може бути поданий таким чином.

1. Ініціалізація: задати навчальну вибірку <*x*, *y*> і побудувати по ній дерево рішень *t* на основі методів [2].

2. Виділення кластерів. Для заданого дерева *t* визначимо у кожному з листових вузлів усереднені екземпляри вузлів – центри відповідних кластерів. При цьому доцільно використовувати тільки ті ознаки, які використовуються у дереві рішень. Після виділення набору інформативних ознак на основі дерева рішень необхідно видалити з вихідного набору неінформативні ознаки, Перенумерувати їх і скорегувавши значення кількості ознак *N*. Кожному з центрів кластерів співставимо найбільш частий номер класу $Y^{t,q}$ серед екземлпярів даного *q*-го кластера дерева *t*.

3. Структурний синтез НМ LVQ: на першому шарі розмістимо стільки нейронів, скільки виділено кластерів *Q*, а на другому шарі – стільки нейронів, скільки задано класів – *K*. Нейрони першого шару використовують вагову функцію відстані та активаційну функцію використовують функцію "переможець отримує все". Нейрони другого шару використовують вагову функцію зважену суму та активаційну функцію – лінійну.

4. Параметричний синтез НМ: занесемо у ваги нейронів першого шару мережі координати центрів відповідних нейронам кластерів-листів дерева рішень, далі встановимо ваги нейронів другого шару $w_j^{(2,i)} = \{1 | Y^{t,j} = i\}$, де *i* = 1, 2, ..., *K*, $w_j^{(2,i)}$ – вага *j*-го входу *i*-го нейрона другого шару.

Після чого підлаштуємо значення вагових коефіцієнтів НМ на основі навчальної вибірки, використовуючи один з методів навчання мереж LVQ [1].

У порівнянні з традиційним підходом до побудови моделей на основі мереж LVQ, де кількість кластерів задається користувачем і може виявитися надлишковою або недостатньою для отримання моделі, у запропонованому методі кількість кластерів визначається на основі попередньо отриманого дерева рішень. При цьому, на відміну від методів кластер-аналізу, що визначають кількість кластерів тільки за близькістю екземплярів у просторі ознак без урахування їх фактичної належності до класів, дерево рішень здійснює класифікацію, групуючи екземпляри у кластери за ступенем близькості, але з урахуванням приналежності до класів. Таким чином, запропонований метод забезпечує оптимальний вибір кількості кластерів у порівнянні з традиційним підходом.

На відміну від традиційного підходу [1], де при навчанні мереж LVQ початкові значення ваг мережі задаються випадковим чином, у запропонованому методі початкові значення ваг нейронів встановлюються на основі параметрів центрів сформованих кластерів, що забезпечує краще початкове наближення до оптимального набору значень ваг, і, отже, скорочує витрати часу навчання.

Можливість виділення набору інформативних ознак з заданого вихідного набору на основі побудованого дерева рішень, дозволяє у запропонованому методі істотно скоротити кількість входів нейронів першого шару мережі, а, отже, зменшити число ваг нейронів першого шару, що робить модель більш простою, скорочує час навчання, а також підвищує узагальнювальні властивості моделі та її інтерпретабельність.

Недоліком запропонованого методу є використання однакової метрики (способу визначення відстані) для усіх кластерів. Перспективи подальших досліджень полягають у тому, щоб розробити методи побудови мереж LVQ, що використовують адаптивні метрики, які визначаються в результаті побудови дерева рішень.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Kohonen, T. Self-Organizing Maps / T. Kohonen. – Berlin: Springer-Verlag, 2001. – 502 p.

2. Субботин С.А. Построение деревьев решений для случая малоинформативных признаков // Радіоелектроніка, інформатика, управління. – 2019. – № 1. – С. 122-131.

УДК 004.021 (004.89)
Леощенко С.Д.[1], Олійник А.О.[2], Субботін С.О.[3]
[1]асп. НУ «Запорізька політехніка»
[2]канд. техн. наук, доц. НУ «Запорізька політехніка»
[3]д-р техн. наук, проф. НУ «Запорізька політехніка»

ВИКОРИСТАННЯ ХЕШУВАННЯ ДЛЯ ПРИСКОРЕННЯ НЕЙРОННИХ МЕРЕЖ

В останні роки величезна кількість сил було спрямоване на прискорення штучних нейронних мереж (ШНМ) – як для навчання, так і для отримання висновків її роботи – так були представлені апаратні платформи, на яких можна виконувати обчислення: Google розробили Tensor Processing Unit; пізніше Nvidia представила V100 Graphics Processing Unit, описуючи його, як чіп, спеціально розроблений для навчання і використання ШНМ, а також для інших високопродуктивних обчислювальних потреб.

Однак, вже в 2019 році автори Бейді Чен, Тарун Медіні і Аншумалі Шрівастава з Університету Райса, у своїй роботі довели, що, можливо, спеціальне обладнання, що розробляється для роботи ШНМ, оптимізується не під той алгоритм. Дослідники зрозуміли, що активація багатьох нейронів в певному шарі ШНМ виявляється занадто малою, і не впливає на вихідне значення, що підраховується наступними шарами. Тому, якщо знати, що це за нейрони, їх можна просто ігнорувати.

Так найпростішим способом дізнатися, які нейрони в шарі не активуються, буде спочатку зробити всі операції матричного перемноження для цього шару. Але дослідники зрозуміли, що насправді можна визначитися з цим більш ефективним способом, якщо подивитися на проблему під іншим кутом і запропонували підходити до цього питання, як до вирішення завдання пошуку. Тобто, замість того, щоб обчислювати множення матриць і дивитися, які нейрони активувалися для заданих вхідних даних, можна просто подивитися, які це нейрони в базі даних. Перевага такого підходу в задачі полягає в тому, що можна використовувати узагальнену стратегію, давно вже вдосконалену фахівцями з інформатики з метою прискорення пошуку даних в базі: хешування.

Хешування дозволяє швидко перевірити, чи є потрібне значення в таблиці бази даних, без необхідності послідовно проходити кожну її рядок. Використовуючи хеш, легко розраховується через застосування хеш-функції до потрібної величини, який вказує, де це значення має зберігатися в базі. Потім можна перевірити тільки одне це місце, щоб дізнатися, чи зберігається там це значення.

Але що якщо просунутися у використанні хешування від використання ШНМ до її синтезу. Так, з огляду на широке використання хеш-функцій при

кодуванні інформації в імовірнісних структурах даних звернемося до них. Bloom Filter – це бітовий масив з *M* бітів, ініціалізованих в 0. Щоб додати елемент, він обробляється *k* хеш-функціям, щоб отримати *k* позицій масиву, і встановіть біти в цих позиціях рівними 1. Щоб запросити елемент, він передається в *k* хеш-функцій, щоб отримати *k* позицію масиву. Тоді логічним буде використання такої структури даних при пересилці інформації про ШНМ між ядрами паралельної системи. MinHash – це метод швидкої оцінки того, наскільки схожі два набори. Ця схема була винайдена Andrei Broder (1997) і спочатку використовувалася в пошуковій системі AltaVista для виявлення дублікатів веб-сторінок і виключення їх з результатів пошуку. Він також був застосований у великомасштабних завданнях кластеризації, таких як кластеризація документів за подібністю їх наборів слів. Тоді, MinHash використаємо при послідовному виконанні нейроеволюційного синтезу для кодування генетичної інформації про особини, для оцінки прискорення на етапі селекції. Нарешті, Count–min sketch – це імовірнісна структура даних, яка представляє собою таблицю частот подій в потоці даних. Вона використовує хеш-функції для зіставлення подій з частотами, але на відміну від хеш-таблиці використовує тільки сублінійний простір, за рахунок перерахунку деяких подій через колізій. Саме тому, Count-Min Sketch також будемо використовувати для кодування генетичної інформації про особини під час нейроеволюційного синтезу.

Безумовно, такий підхід має бути випробуваний і лише на основі результатів тестування можна буде говорити про його імплементацію. Але вже можна сказати про те, що це безумовно підкреслює небезпеку захоплення певного типу апаратним забезпеченням у випадках, коли існує можливість появи нового і кращого алгоритму для роботи ШНМ.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. SLIDE: In Defense of Smart Algorithms over Hardware Acceleration for Large-Scale Deep Learning Systems / B. Chen, T. Medini, J. Farwell, S. Gobriel, C. Tai, A. Shrivastava // arXiv [Electronic resource]. – Access mode: https://arxiv.org/abs/1903.03129

2. Gakhov A. Probabilistic Data Structures and Algorithms for Big Data Applications / A. Gakhov. – Madison : Books on Demand, 2019. – 220 p.

УДК 004.852
Leoshchenko S.[1], Oliinyk A.[2], Subbotin S.[3]
[1]PhD student Zaporizhzhia Polytechnic National University
[2]PhD, associate professor Zaporizhzhia Polytechnic National University
[3]Doctor of Technical Science, prof. Zaporizhzhia Polytechnic National University

USING REFLECTION MECHANISMS TO INCREASE THE LEVEL OF COGNITIVE PERFORMANCE OF NEURAL NETWORKS

In the last decade, cognitive methods of analysis, diagnostics, forecasting and modeling for the research of large-scale systems operating under various types of uncertainty, as well as for managing the development of situations in these systems, have become increasingly widespread. The purpose of cognitive analysis of such systems is either to find out the mechanism of functioning of the system (the mechanism of phenomena and processes occurring in the system), or to predict the development of the system, or to control the system, or to determine the possibilities of adaptation of the subject to the existing system, or to solve all these problems. At the same time, the issues of cognitive synthesis have not yet been considered separately, although the results of the cognitive analysis of complex systems actually lead the researcher to results that can be interpreted as the synthesis of systems with the desired properties. The process of cognition (cognition) of a large system with the help of the developed tools is included in the metamodel of object research. All these features of the cognitive methodology allow us to attribute it to intelligent technologies, on the basis of which it is possible to develop intelligent decision support systems. Thus, the development of cognitive methodology is seen in the direction of the development of methods of cognitive synthesis and the development of intelligent information management systems, which contain intelligent modules built on the basis of this methodology and using the principles of distributed multiprocessor computing systems. When designing such systems, it is proposed to use the advantages of distributed information and control systems.

Currently, cognitive modeling is understood as the solution of interrelated system problems of cognitive analysis and synthesis. Cognitive analysis: development of a cognitive model; analysis of the paths and cycles of the cognitive model; analysis of the stability and possibility of disasters, etc.. Cognitive synthesis: based on the data of cognitive analysis, the solution of problems of decomposition and composition of the model, the solution of inverse problems, optimization, synthesis of a system with specified properties: from simple cognitive structures, by decomposition or completion of the original structure; decision-making is performed.

The most promising basis for model-based cognitive analysis is artificial neural networks (ANNs). However, it should be noted that today it is difficult to

fully determine the level of their cognition, as well as the internal processes and features that affect it.

Until researchers have enough knowledge to reproduce at least fragments of consciousness in the ANNs, such as reflection, neural network systems will remain just an apparatus for approximating complex functions. In this case, we will consider reflection

– in a broad sense, as a reflection of the external world, available for mental manipulation;

– in a narrow sense, as an internal representation of yourself and your actions.

However, part of the reflection can be provided by the use of self-learning mechanisms (unsupervised learning or reinforcement learning) and recurrent properties. Then, such states are arranged in complex scenarios (plots), reflecting in some way the previous experience, but do not repeat it directly. The next stage may be the use of training methods with a generative component. Thus, by competing and transmitting the current state of another neuromodel, networks can achieve better results at the stage of parametric synthesis based on the use of a system of rewards and penalties.

Further, based on the results of the study, it is planned to create a multi-layer perceptron, as one of the simple models of neural networks with high computational abilities, which is capable of learning and extracting important features from the input data. Its role will be the compression of visual information and orientation in the virtual space. It is assumed that when combining a multilayer perceptron with a recurrent neural network, it will help the latter to form an internal representation of the external world within its structure as a necessary condition for the emergence of reflection.

REFERENCES

1. F. Wang Visualizing deep neural network by alternately image blurring and deblurring / W. Feng, H. Liu, J. Cheng // Neural Networks. – 2018. – P. 162-172.

2. Pramanik P.K.D. Beyond Automation: The Cognitive IoT. Artificial Intelligence Brings Sense to the Internet of Things / P.K.D. Pramanik, S. Pal, P. Choudhury // Cognitive Computing for Big Data Systems Over IoT. Lecture Notes on Data Engineering and Communications Technologies. – Vol. 14. – 2018 – P. 1-7.

УДК 004.91: 004.67
Дейнега Л.Ю.[1]
[1]старш. викл. НУ «Запорізька політехніка»

ВИКОРИСТАННЯ ПРОДУКТІВ GOOGLE В НАВЧАЛЬНОМУ ПРОЦЕСІ

Через карантинні обмеження багато процесів людської діяльності в світі перейшли в режим онлайн. Сфера освіти не стала виключенням, навчання зараз відбувається в дистанційній або змішаній формі. Для проведення занять та заходів контролю рівня знань існує різне програмне забезпечення. Для ефективної роботи педагога варто використовувати комплекс продуктів, які доповнюють один одного. Таким комплексом є набір рішень Google Workspace for Education.

Багато педагогів переконалось в багатофункціональності та зручності використання продукту Google Клас. В ньому можна створювати класи з окремих дисциплін, керувати завданнями та оцінками, додавати до своїх завдань відео на YouTube, опитування Google Форми та інші елементи з Google Диску, залишити відгук у реальному часі, використовувати потік занять, щоб публікувати оголошення та залучати студентів (учнів) до дискусій. При створенні класу в Google Календар додається окремий календар з термінами виконання завдань даного потоку, що спрощує відстеження їх виконання. Кожен користувач має унікальний вхід, щоб захистити індивідуальні облікові записи, діяльність у класі обмежена лише учасниками класу.

В Google Класі педагоги можуть створити унікальне посилання на відео-конференцію в Google Meet для кожного класу. Конференції не є обмеженими в часі. За потреби можна зробити відеозапис зустрічі, який збережеться на Google Диску. Функція демонстрації екрана учасника відеозустрічі зробить подачу матеріалу педагогом більш цікавою.

Спільну інтерактивну роботу під час конференції можна організувати через інтегрований продукт Google Jamboard. В ньому можна писати і малювати стилусом, вставляти зображення або вебсторінки, додавати стікери, ділитись своїми Jam-сторінками зі співавторами. Також для спільної роботи можна використати інший документ з наданим всім учасникам доступом. Багато можливостей для такої роботи надає продукт Google Презентації. Крім звичних тексту і зображень до вмісту такого документа можна додати аудіо, відео, таблиці, діаграми, фігури. Можливість додавання настільки різних об’єктів робить презентацію більш цікавою і заохочує до вивчення поданого в ній матеріалу.

Повний перелік можливостей комплексу Google Workspace for Education є набагато ширшим. Але навіть скорочений огляд його продуктів доводить

доцільність використання цього комплексу в дистанційному і змішаному навчанні.

УДК 004.42
Вічева Г.Ю.[1], Каплієнко Т.І.[2]
[1]студ. гр. КНТ-128сп НУ «Запорізька політехніка»
[2]канд. техн. наук, доц. НУ «Запорізька політехніка»

ПРОГРАМНА РЕАЛІЗАЦІЯ ТЕЛЕГРАМ-БОТА СЛУЖБИ ТЕХНІЧНОЇ ПІДТРИМКИ

На даний момент на підприємствах найчастіше функціонують IT-служби або відділи технічної підтримки. З ростом кількості співробітників підприємства збільшується кількість звернень до технічної підтримки.

Існує множина питань, які є приводом приблизно семидесяти відсотків звернень до технічної підтримки. Відповідь на будь-яке питання з цієї множини у співробітника IT-служби в середньому займає від 5 до 15 хвилин, якщо відповідь на питання заздалегідь відома. В цілому тривалість відповіді не велика, але зі зростанням кількості співробітників підприємства зростає кількість заявок, що надходять до IT-служби, відповідно, збільшується черга з питань, які необхідно з'ясувати. Чим більше часу заявка знаходиться без уваги, тим більший ризик простоїв підприємства та робочого часу інших співробітників. Отже для технічної підтримки вкрай важливо скоротити чергу із заявок.

Існує декілька варіантів вирішення даної проблеми. Підприємство або розширює штат операторів, які відповідають на заявки до технічної підтримки, або автоматизує обробку певної частини заявок. Економічно доцільніше одноразове створення програмного продукту, ніж постійне утримання додаткової робочої сили.

Перед обранням технологій та засобів розроблення важливо розуміти перелік необхідного функціоналу майбутнього програмного продукту. Коли сформовано мету розроблення, варто також врахувати технічні особливості підприємства (стабільність доступу до мережі Інтернет, тощо). Найчастіше автоматизується інформаційна функція (отримання інформації за запитом користувача або інформаційна розсилка) та частково функція спілкування.

Через деякий проміжок часу роботи IT-служби накопичується перелік питань, на які заздалегідь відомі відповіді, що несуть виключно інформаційний характер. Обробку саме таких заявок було вирішено автоматизувати, оскільки якщо питання повторюються, надання стандартних відповідей на такі питання не є ефективно використаним часом оператора.

Також було вирішено реалізувати для користувача можливість залишити питання в певній категорії.

Для реалізації програмного продукту доцільно використовувати звичні пересічному користувачу засоби взаємодії, також варто забезпечити кросплатформність. Ідеально підходить варіант месенджера, оскільки важко знайти співробітника, який не користується останнім, також таким чином буде дотримано вимогу про кросплатформність, бо для більшості месенджерів реалізовано як мобільну, так і десктопну версії.

З дня свого заснування месенджер Telegram стабільно піднімається в рейтингах найпопулярніших застосунків. Про це в понеділок, 8 лютого 2021 року, повідомив засновник месенджера Павло Дуров. Він зазначив, що Telegram став самим завантажуваним мобільним застосунком в світі в січні 2021 року та що з моменту запуску в 2013 році база користувачів Telegram щорічно зростала більш ніж на 40%. З огляду на статистичні дані про поточну популярність месенджерів прийнято рішення про використання Telegram в якості платформи для розроблення програмного продукту.

В середині червня 2015 року, творець месенджера Telegram Павло Дуров оголосив про запуск платформи для створення співрозмовників-ботів, здатних реагувати на команди користувачів, взаємодіючі при цьому з зовнішніми сервісами. За лічені місяці платформа набула широкого успіху. У квітні 2016 було анонсовано перше масштабне оновлення – Bot Platform 2.0, підготовлене з урахуванням побажань розробників і потреб користувачів, яке значно розширює можливості роботів. Додана можливість обміну музикою і відео, варіанти швидких відповідей, ідентифікація місця розташування користувача, глибока інтеграція з іншими службами на основі телефонних номерів (при авторизації) і ряд інших корисних можливостей.

Таким чином знайдено зручний засіб розроблення програмного продукту для спілкування служби технічної підтримки з кінцевим співробітником – Telegram-бот. Прийнято рішення про програмну реалізацію бота мовою програмування Python, оскільки існує широкий вибір бібліотек для зручної реалізації необхідних функцій. Прийнято рішення про реалізацію інформаційної функції бота наступним чином: надання інформації про корпоративне програмне забезпечення за запитом користувача, інформаційна розсилка на всіх зареєстрованих користувачів. Функція часткового спілкування передбачує можливість залишення заявки у відповідному проблемі розділі, можливість спілкування з оператором засобами бота у разі необхідності.

Висновки. В роботі було розглянуто основні проблеми, що можуть виникнути при веденні технічної підтримки підприємства. Було запропоновано рішення щодо автоматизації даного процесу. На даний

момент відповідний програмний застосунок знаходиться в процесі розроблення та найближчим часом може бути впроваджений в експлуатацію.

УДК 004.42
Зирянов Ю.О.[1], Каплієнко Т.І.[2]
[1]студ. гр. КНТ-128сп НУ «Запорізька політехніка»
[2]канд. техн. наук, доц. НУ «Запорізька політехніка»

ОСОБЛИВОСТІ РОЗРОБЛЕННЯ СУЧАСНИХ ІГОР

На сьогоднішній день розроблення ігор є досить великою і затребуваною частиною цифрової індустрії. Відеоігри є одним з популярних видів розваг для людей будь-якого віку, вони можуть надавати як розважальні так і розвиваючі або оздоровчі ефекти. Все залежить від інструментів, які надаються самою грою, а також людини, яка використовує ці інструменти. Для корпорацій відеоігри є бізнесом, який приносить великі гроші, як за самі ігри, так і за комп'ютерні компоненти, які постійно оновлюються.

Створенням комп'ютерних ігор може займатися як окрема студія з великим штатом співробітників, або навіть декілька студій, так і окрема людина або група людей. Зазвичай ігри поділяються на високобюджетні, середньобюджетні та низькобюджетні.

Високобюджетні ігри – це ігри, які розроблюються однією або декількома студіями з великим бюджетом і виходять із-під крила видавця, який бере на себе маркетинг та розповсюдження гри, а також може частково чи повністю фінансувати саме розроблення.

Середньобюджетні ігри – це ігри, які розробляються невеликою студією з невеликим бюджетом, яка має видавця, що розповсюджує їх продукт та може частково фінансувати розроблення.

Низькобюджетні ігри – це ігри, які створюються однією людиною або невеликими командами людей, мають дуже обмежений бюджет, зазвичай вони не мають видавця, а фінансування надходить з різних джерел, у тому числі й пожертвувань.

Нерідко перед початком основного розроблення створюється прототип майбутньої гри, який демонструється керівництву для отримання дозволу на розроблення, а також гравцям, щоб оцінити зацікавленість у проєкті.

Прототип гри – це модель майбутньої гри, в яку можна пограти і відчути ігровий процес. На підставі прототипу можна сказати чи є майбутнє у проєкта. Прототип дає можливість дизайнеру гри відразу почати маніпулювати з ігровим світом, фізикою всередині гри, економікою і

процесами. Основний сенс прототипу – це перевірити ідею на практиці і, зменшивши ризики, продовжити розроблення проєкту.

Створення гри досить тривалий і складний процес, який може займати від кількох років до майже десятка. На розроблення впливають багато різних факторів, як творчих, так фінансових і ринкових. Чим довше триває розроблення, тим більше виникає ризиків: бюджети зростають, важливі учасники йдуть з проєкту, ринок постійно змінюється, технології розвиваються. Тому нерідко буває, що розроблення можуть зовсім припинити чи віддати іншій студії, або випустити у тому вигляді як є. Також буває, що проєкт можуть почати розроблювати з самого початку, викинувши більшу частину готового проєкту, зазвичай це трапляється, якщо проєкт дуже довго знаходиться у розробленні і втратив свою актуальність.

Дуже важливу роль у проєкті грає творчість, від цього залежить весь проєкт і його подальша доля. В студіях, які займаються створенням ігор, більша частина вакансій так чи інакше пов'язана з творчими професіями, навіть програмісти виконують свою роботу з більш творчим підходом ніж математичним. Це пов'язано з тим, що програміст не може заздалегідь знати, що гравець буде робити в той чи інший момент, багато нюансів потрібно передбачити, а деякі елементі замаскувати. Нерідко при розробленні прибігають до використання «заглушок», і це звичайна практика, яка допомагає вирішувати деякі складні проблеми, а більшість гравців цього навіть не помітять. Головною задачею розроблення є в першу чергу створення віртуального світу, у який гравець повинен повірити, гра повинна викликати емоції і дарувати унікальний досвід.

Створення відеоігор відрізняється від створення програмного забезпечення, в першу чергу своїм підходом. У програми є чіткий план розроблення, список функцій, які вона повинна в собі містити, та проблеми, які повинні вирішуватися за допомогою програми, а при розробленні завжди можна оцінити вже готовий функціонал на різних етапах проєкту. В іграх хоч план по розробленню є, але багато чого в ході розроблення може доповнюватися, змінюватися або зовсім видалятися, все це залежить від того, як усі заплановані функції будуть працювати разом. І весь цей процес триває до самих пізніх стадій розроблення, тому що ігри занадто залежать від безлічі різних механік, і нерідко лише на фінальній стадії розроблення починають здобувати виразну форму, яку можна нормально оцінити. Тому при плануванні та створенні відеоігор неможна орієнтуватися на план зіставлений як для створення ПЗ, бо це лише загальмує розроблення проєкту та викличе зайві складнощі.

Висновки. В роботі було розглянуто основні особливості створення відеоігор, а також головна відмінність від створення програмного забезпечення. Цей напрямок дуже швидко розвивається та має багато

перспектив, за останні двадцять років ігри розвинулися від примітивної псевдо 3D геометрії до повноцінних 3D ігор з реалістичною графікою, освітленням, звуками, і навіть тактильної передачею, а також освоїли віртуальну реальність.

УДК 004.42
Каплієнко Т.І.[1], Українцева Є.С.[2], Дяченко М.О.[2]
[1]канд. техн. наук, доц. НУ «Запорізька політехніка»
[2]студ. гр. КНТ-110сп НУ «Запорізька політехніка»

ЗАСТОСУНОК ДЛЯ ЗАМОВЛЕННЯ АВІАКВИТКІВ «ПОМІЧНИК МАНДРІВНИКА»

У сучасному світі людині відкрита величезна кількість можливостей для пізнання себе, навчання чомусь новому, отримання незабутніх вражень, здобуття досвіду, формування знайомств. Одним з найпоширеніших методів реалізації цього є подорожі. Проте більшість мандрівників зустрічаються з певними складнощами під час підготовки до мандрівки. Іноді це причини, які від них не залежать, а іноді основна складність полягає у так званому «людському факторі».

Отже актуальність даної роботи полягає у створенні програмного забезпечення – помічника мандрівника, що спрямовано автоматизувати замовлення авіаквитків та підготовку користувача до подорожі. Область використання такого застосунку включає в себе різні сфери людської діяльності, такі як трудова та особистий відпочинок.

Питання автоматизованого придбання авіаквитків вирішує значну проблему сучасного суспільства – нестачу часу, обсяг якого за звичайних ситуацій обмежений, а під час екстрених ситуацій – критичний. Вирішення питання «людського фактору» – полегшить процес підготовки, знову таки, економлячи час, за рахунок логічної концентрації однотипних факторів в одному місці, візуалізуючи необхідне.

Отже, мета даного дослідження – створення застосунку для автоматизованого замовлення авіаквитків з урахуванням погодних умов для формування списку необхідних речей – є актуальною та необхідною задачею сьогодення.

Функції, що виконує програма наведені нижче:

– пошук необхідного рейсу за параметрами – місто відправлення, місто прибуття, дата відправлення;

– формування квитка, придатного до друку;

– створення та модифікація облікового запису користувача;

– формування списків майбутніх та минулих рейсів;

– генерація списку рекомендованих речей на основі даних про температуру повітря та опису погоди;

– збереження, зміна та видалення речей користувача, які він планує взяти з собою у подорож.

Програму реалізовано за допомогою використання мови програмування високого рівня C#. Середовище розроблення Visual Studio 2019. База даних створена в системі керування базою даних SQLite. Для збереження даних у базі даних було створено алгоритми шифрування. Один алгоритм заснований на основі RC4, змінений відповідно до вимог розроблюваного програмного забезпечення та вдосконалений. Другий – ґрунтується на генерації токену доступу користувача до свого облікового запису у застосунку.

Результат, що отримає користувач у процесі роботи з програмою – це маршрутна квитанція (рис. 1). Створення квитка здійснюється за допомогою пошуку особистих даних користувача, введених при реєстрації, а також даних майбутнього рейсу. Після чого результат виводиться до форми Crystal Report, що містить розширення .rpt.


Електронний квиток

Маршрутна квитанція

Пасажири

Прізвище	ім'я	Паспорт	Тип пасажира
Ivanov	Ivan	FE123456	Дорослий

Дані польоту

Рейс	Подробиці перельоту		Ціна
3	Запоріжжя, OZH Україна, UA 2021-06-18 21:45:00	Париж, PAR Франція, FR 2021-06-19 09:35:00	4688 грн.

Дата друку : 08.02.2021

Рисунок 1 – Маршрутна квитанція

Висновки. У результаті виконання роботи було створено програмне забезпечення, призначене автоматизувати замовлення авіаквитків та підготовку користувача до подорожі. Було проаналізовано предметну область та виявлено, які критерії та вимоги має кожна її складова. Було розроблено інструментарій, що має зручний та зрозумілий користувачеві інтерфейс та вирішує головну задачу – автоматизовані дистанційні пошук та замовлення авіаквитка, а також другорядну, проте важливу задачу – збереження списку речей користувача та надання рекомендацій щодо його наповнення на основі даних погоди.

Розроблений продукт, окрім головного алгоритму – пошуку та замовлення авіаквитка за введеними параметрами, має власні алгоритми шифрування. Ці алгоритму були використанні для створення токену, а також

для збереження конфіденційних даних (паролю) у базі даних. Одна з головних переваг розробленого програмного забезпечення є можливість його подальшого розширення та доопрацювання. Якісне проєктування бази даних, розроблені алгоритми, структура класів, зрозуміла структура інтерфейсу – усе це дає змогу реалізувати подальше розроблення.

УДК 004.032.26
Олійник А.О.[1], Басанець М.І.[2]
[1]канд. техн. наук, доц. НУ «Запорізька політехніка»
[2]студ. гр. КНТ-117 НУ «Запорізька політехніка»

ПРОГРАМНА РЕАЛІЗАЦІЯ ШАХОВОГО РУШІЯ З ВИКОРИСТАННЯМ ТЕХНОГОЛІЇ ГЛИБОКОГО НАВЧАННЯ НЕЙРОМЕРЕЖЕВИХ МОДЕЛЕЙ

Шахове програмування — вид програмування, що вивчає та реалізує засоби програмного забезпечення для оцінки переваги певної сторони. Також до цієї області належать користувацькі інтерфейси (Chess GUI), що надають можливість грати в шахи, які зазвичай в свою чергу передають інформацію про дії користувача до рушія [1].

Шаховий рушій (Chess Engine) — програмне забезпечення, що підтримує наступні базові операції:

– можливість вводу будь-якої позиції та отримання інформації про неї від рушія;

– пошук декількох найкращих ходів у позиції;

– оцінка переваги для певної сторони у позиції (оцінка позиції);

– пошук та підрахунок усіх можливих ходів у позиції.

Зазвичай, шахові рушії розповсюджуються у вигляді консольної утиліти, яка підтримує універсальний шаховий інтерфейс (Universal Chess Interface — UCI). Це дозволяє шаховим користувацьким інтерфейсам з’єднуватися з процесом рушія та надавати команди через стандартний ввід та отримувати відповіді через стандартний вивід.

Основні складності у реалізації класичного шахового двигуна полягають у наступному:

– реалізація представлення дошки, що має уявлення про всі правила шахів. Основна складність даного аспекту полягає в підтримці усієї інформації про позицію використовуючи щонайменше пам’яті для побудови більшого графу пошуку, та використання ефективних структур даних для підтримки місцезнаходження фігур для пришвидшення генерації ходів;

– пошук позицій, що представляє собою побудову орієнтованого графу, де кожна вершина — можлива позиція, а ребра графу — можливі ходи. За

правильної генерації можливих ходів побудова такого графу не є важкою задачею. Основною складністю в даному напрямі є відсічення “неінформативних” гілок, що можна реалізувати як класичними алгоритмами, так і з використанням глибокого навчання для класифікації вартості гілки;

– реалізація генерації можливих ходів в даній позиції. Головні складності в даному напрямку полягають у відрізненні легальних ходів від псевдо-легальних (наприклад, ті ходи, що дозволені для фігури, проте залишають короля під атакою), та класифікація ходів для їх сортування. Класифікація та сортування потрібні для оптимізації обходу графа позицій та відсічення “неінформативних” гілок графу;

– оцінка позиції, що представляє собою визначення переваги деякої сторони. Оскільки шахи — гра з нульовою сумою, перевага у однієї сторони означає відставання протилежної сторони. Для оцінки позиції використовується дуже багато складних правил, підтримувати та визначати пріоритети для яких стає дуже важко.

Для використання технології глибокого навчання наразі у шахах є два основні напрямки: створення агенту з використанням нейронних мереж з підкріпленням, що є новим та некласичним підходом до реалізації шахових рушіїв, та використання нейромережевих моделей у класичних шахових рушіях для оцінки позиції або визначення типу ходу. Для реалізації було обрано класичний шаховий рушій, що використовує нейронну мережу для оцінки позиції.

Для тренування нейромережі було знайдено декілька мільйонів позицій у відкритих ресурсах, та знайдено оцінку для кожної за допомогою відомого рушія з відкритим вихідним кодом Stockfish [2] на глибині 15 ходів.

За архітектуру для нейромережі було обрано тришаровий персептрон. Для отримання найкращих результатів, було створено кілька нейронних мереж з різними функціями активації та кількістю нейронів, а також з різними форматами вхідних шарів.

Основний формат вхідного шару: вектор з 768 вхідних ознак, в якому для кожного з 64 полів та 12 типів фігур встановлене значення 0 або 1 в залежності від того, чи займає фігура дане поле.

Як результат, було отримано нейромережу, що в об’єднанні з класичним шаховим двигуном збільшило точність оцінки позиції. Приріст точності відбувся за рахунок фактичного збільшення глибини аналізу, оскільки класична оцінка позиції дозволяє це зробити на поточній глибині, а нейромережа на тренованій глибині.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Chess Programming Wiki [Електронний ресурс]. – Режим доступу : https://www.chessprogramming.org/Main_Page.

2. Stockfish — шаховий рушій з відкритим вихідним кодом [Електронний ресурс]. — Режим доступу: https://stockfishchess.org/blog/

УДК 004.032.26

Козлов В.В.[1], Льовкін В.М.[2], Олійник А.О.[3]
[1]студ. гр. КНТ-137 НУ «Запорізька Політехніка»
[2]канд. техн. наук, доц. НУ «Запорізька Політехніка»
[3]канд. техн. наук, доц. НУ «Запорізька Політехніка»

ВИКОРИСТАННЯ НЕЙРОМЕРЕЖЕВИХ МОДЕЛЕЙ VGGNET В ЗАДАЧІ КЛАСИФІКАЦІЇ ЗОБРАЖЕНЬ

На даний момент застосування та використання нейронних мереж для вирішення задачі класифікації зображень є одним з передових напрямків розвитку технологій машинного навчання. Для вирішення такої задачі прийнято використовувати підвиди згорткових нейронних мереж.

Згорткова нейронна мережа – це алгоритм глибокого навчання, який може приймати вхідне зображення, призначати важливість різним аспектам або об'єктам на зображенні та здатний диференціювати один об’єкт від іншого за рахунок механізмів, подібних зоровій корі [1].

Головною особливістю мережі є наявність операції згортки [2]. В такій архітектурі кожен мережевий рівень виступає в якості фільтра виявлення на наявність певних ознак або шаблонів, присутніх у вхідних даних.

Згорткова нейронна мережа є багатошаровою мережею без зворотних зв'язків. В загальному випадку її структура складається з таких шарів (рис. 1): вхідний (input), згортковий (convolutional), агрегувальний (pooling), згладжувальний (flatten), повнозв'язний (fully connected), вихідний (ouput).

Рисунок 1 – Загальна архітектура згорткової нейронної мережі

В якості функцій активації нейронів застосовується ReLU та Softmax (для вихідного шару). Для навчання найчастіше використовується метод зворотного поширення помилки.

Підвидом згорткової нейронної мережі є сімейство моделей VGGNet, запропонованих вченими Оксфордського університету Кареном Сімоняном та Ендрю Зіссерманом в 2014 році [3].

Усі конфігурації VGG моделей мають загальну архітектуру і розрізняються лише кількістю шарів з вагами (рис. 2).

VGG-11	VGG-11 (LRN)	VGG-13	VGG-16-1	VGG-16	VGG-19
11 вагових шарів	11 вагових шарів	13 вагових шарів	16 вагових шарів	16 вагових шарів	19 вагових шарів
Вхідний шар (224 × 224 RGB зображення)					
Згортка 3×3×64	Згортка 3×3×64 LRN	Згортка 3×3×64 Згортка 3×3×64	Згортка 3×3×64 Згортка 3×3×64	Згортка 3×3×64 Згортка 3×3×64	Згортка 3×3×64 Згортка 3×3×64
Максимізаційне агрегування					
Згортка 3×3×128	Згортка 3×3×128	Згортка 3×3×128 Згортка 3×3×128	Згортка 3×3×128 Згортка 3×3×128	Згортка 3×3×128 Згортка 3×3×128	Згортка 3×3×128 Згортка 3×3×128
Максимізаційне агрегування					
Згортка 3×3×256 Згортка 3×3×256	Згортка 3×3×256 Згортка 3×3×256	Згортка 3×3×256 Згортка 3×3×256	Згортка 3×3×256 Згортка 3×3×256 Згортка 1×1×256	Згортка 3×3×256 Згортка 3×3×256 Згортка 3×3×256	Згортка 3×3×256 Згортка 3×3×256 Згортка 3×3×256 Згортка 3×3×256
Максимізаційне агрегування					
Згортка 3×3×512 Згортка 3×3×512	Згортка 3×3×512 Згортка 3×3×512	Згортка 3×3×512 Згортка 3×3×512	Згортка 3×3×512 Згортка 3×3×512 Згортка 1×1×512	Згортка 3×3×512 Згортка 3×3×512 Згортка 3×3×512	Згортка 3×3×512 Згортка 3×3×512 Згортка 3×3×512 Згортка 3×3×512
Максимізаційне агрегування					
Згортка 3×3×512 Згортка 3×3×512	Згортка 3×3×512 Згортка 3×3×512	Згортка 3×3×512 Згортка 3×3×512	Згортка 3×3×512 Згортка 3×3×512 Згортка 1×1×512	Згортка 3×3×512 Згортка 3×3×512 Згортка 3×3×512	Згортка 3×3×512 Згортка 3×3×512 Згортка 3×3×512 Згортка 3×3×512
Максимізаційне агрегування					
Повнозʼєднаний шар ×4096					
Повнозʼєднаний шар ×4096					
Повнозʼєднаний шар ×1000					
Нормалізована експоненціальна функція (Softmax)					

Рисунок 2 – Архітектура моделей VGGNet

В якості вхідних даних модель приймає RGB зображення розміру 224x224 пікселів. Зображення проходить через послідовність згортальних шарів, в яких використовуються фільтри з дуже маленьким рецептивним полем розміру 3x3. Просторова агрегація здійснюється за допомогою п'яти max pooling шарів на вікні розміру 2x2 з кроком 2, які розташовано за одним із згортальних шарів. Після згортальних шарів розміщено три повнозв'язних шари та softmax шар. В якості функцій активації нейронів застосовується ReLU.

Ефективність та перспективність використання VGG архітектури для вирішення задачі класифікації зображень доведено в ILSVRC [4], де моделі сімейства VGGNet досягли одного з найкращих результатів точності розпізнавання, випереджаючи навіть мережу GoogleNet.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Saha S. A Comprehensive Guide to Convolutional Neural Networks – the ELI5 way [Electronic resource] / S. Saha. – Access mode: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

2. Субботін С.О. Нейронні мережі: теорія та практика: навч. посіб. / С.О. Субботін. – Житомир: Вид. О.О. Євенюк, 2020. – 184 с.

3. Very Deep Convolutional Networks [Electronic resource]. – Access mode: https://arxiv.org/abs/1409.1556/

4. ImageNet [Electronic resource]. – Access mode: http://www.image-net.org/challenges/LSVRC/

УДК 004.056.5

Зайко Т.А.[1], Косенков.І.С[2]
[1]канд. техн. наук, доц. НУ «Запорізька Політехніка»
[2]студ. гр. КНТ-118 НУ «Запорізька Політехніка»

ЗАХИСТ ІНТЕРНЕТУ РЕЧЕЙ

Концепції IoT, такі як розумні пристрої, розумні машини, розумні міста та розумні будинки, охоплюють як статичні, так і динамічні об'єкти у фізичному світі та світі освіти, які можна ідентифікувати та інтегрувати в комунікаційні мережі. Важливо зазначити, що надані дані часто є конфіденційними. Вони можуть включати стан навколишнього середовища, стан наших будинків і міст або стан нашого особистого здоров'я та діяльності. З цієї причини механізми забезпечення та гарантування безпеки та конфіденційності даних в Інтернеті речей мають вирішальне значення. За своєю природою захист Інтернету речей є комплексним і складним завданням.

Завдяки своїй природі як різнорідної мережі, потенційні загрози для даних IoT мають майже нескінченні можливі вектори атак [1]. Ці вектори можна приблизно розділити за початковою ціллю атаки:

– атаки проти IoT-пристроїв: в першу чергу експлуатуються вразливості обладнання;

– атаки проти комунікацій: в основному експлуатується вразливості, пов'язані з перевіркою цілісності даних;

– атаки на рівень сприйняття: в основному використання вразливості безпеки в сенсорних мережах, як цілісність служб та доступність мережі;

– атаки на фізичний рівень: експлуатуються вразливості, пов'язані з фізичними каналами;

– атаки на мережевий рівень: експлуатуються вразливості каналів передачі.

Складні системи, такі як IoT та Cloud, не можуть бути захищені єдиним загальним протоколом. Кожен рівень мережі має свої вимоги [2]. Один із підходів полягає в тому, що взаємодія між користувачем та пристроєм

повинна обмежуватися підключенням до віртуального об’єкта за допомогою відповідного захисту. Кожен користувач у хмарі може навіть мати особисте уявлення про послуги та налаштування пристрою, що обмежує ймовірність витоку даних та крадіжки дозволів.

Часто дані IoT не є надійно засвідченими, а послуги репутації даних не доступні [4]. Застосування Blockchain вирішує проблему довіри, надаючи вузлам можливість перевіряти дані, розподілені мережею, щоб переконатись, що вони жодним чином не підроблені. Однак Blockchain сам по собі не гарантує, що дані не будуть підроблені до того, як дані з нього потраплять у мережу. Ці завдання повинна вирішувати система моніторингу. Необхідність збереження історії транзакцій може бути вирішена принаймні частково за допомогою частково розподіленим блок-ланцюгом, коли кожен вузол містить лише відповідні дані.

IoT інтегрує передові технології комунікацій, мереж, хмарних обчислень, зондування та спрацьовування, а також прокладе шлях для новаторських додатків у різних областях, що вплине на багато аспектів життя людей та принесе багато зручностей. Тим не менше, враховуючи величезну кількість підключених пристроїв, у питаннях безпеки, конфіденційності та управління в IoT виникають дуже значні ризики. Незважаючи на те, що жодне рішення "для всіх" неможливо створити найближчим часом, все ж є можливість значно підвищити безпеку мережі IoT за допомогою комбінації запропонованих способів та рішень.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Choudhury T. Privacy and Security of Cloud-Based Internet of Things (IoT) 2017 3rd International Conference on Computational Intelligence and Networks (CINE), 2017 – pp. 40-45

2. Kakanakov N. Adaptive models for security and data protection in IoT with Cloud technologies / N. Kakanakov, M. Shopov // 2017 40th International Convention on Information and Communication Technology 2017. – pp.1001-1004.

3. Oh S. Development of IoT security component for interoperability / S. Oh and Y. Kim // 2017 13th International Computer Engineering Conference (ICENCO), 2017. – pp. 41-44

4. Rodrigo R. On the features and challenges of security and privacy in distributed internet of things / R.Rodrigo, Z. Jianying, J. Lopez // Computer Networks 57.10, 2013 – pp. 2266- 2279

5. Liu C. A Novel Approach to IoT Security Based on Immunology / C. Liu, Y. Zhang and H. Zhang // 2013 Ninth International Conference on Computational Intelligence and Security, 2013, – pp. 771-775

6. Соколов М.Н., Смолянинова К.А., Якушина Н.А. Проблемы безопасности интернета вещей: обзор. – Вопросы кибербезопасности : журнал. – 2015. – № 5(13). – 34с.

7. Исаков В.Б., Сарьян В.А., Фокина А.А. Правовые аспекты внедрения Интернета вещей // ИТ-Стандарт. – 2015. – № 4-1 (5). – С. 9-16.

УДК 004.056

Зайко Т.А.[1], Дем'янова Д.К.[2]
[1]канд. техн. наук, доц. НУ «Запорізька політехніка»
[2]студ. гр. КНТ-128 НУ «Запорізька політехніка»

КІБЕРБЕЗПЕКА, КІБЕРЗАХИСТ ТА ІНФОРМАЦІЙНА БЕЗПЕКА БАНКІВСЬКОЇ СФЕРИ В УКРАЇНІ

Сучасна банківська система значно залежить від надійності інформаційних технологій, які вона використовує. Тому кіберпезпека і кіберзахист стали фундаментальними засадами існування банківської системи взагалі. Стрімка інформатизація та розвиток глобальних інформаційно-комунікаційних мереж окрім автоматизації звичних банківських процесів ще й постійно надають можливості створення нових банківських продуктів (послуг) (сьогодні це «SMS-банкінг», «Інтернет-банкінг», «WebMoney Banking» тощо).

Найбільш суттєвими загрозами безпеці інформаційних ресурсів є витік або втрата ресурсів (зокрема відомостей, що становлять банківську таємницю). Загрози інформаційним ресурсам можуть бути реалізовані шляхом підкупу осіб, необережного, недбалого поводження з банківською таємницею, недотримання вимог збереження інформації з обмеженим доступом.

Задля збереження інформаційної безпеки банківської системи телекомунікаційне середовище повинно за вимогами ISO (Міжнародна організація стандартизації) забезпечити:

– захист інформації при її зберіганні, обробці та передачі по мережах;

– підтвердження справжності даних і користувачів;

– виявлення та попередження порушення цілісності даних;

– захист програмних продуктів від впровадження програмних закладок і "вірусів";

– захист від несанкціонованого доступу до інформаційних ресурсів тощо.

У зв'язку з цим банк використовують системи передачі повідомлень та здійснення міжбанківських операцій, як один із методів захисту інформації.

Наприклад, у США використовуються Fedwire – мережа федеральної резервної банківської системи, CHIPS – міжбанківська платіжна мережа, Bankwire. У Франції міжбанківські розрахунки засновані на системі SIT. У Великобританії застосовуються системи CHAPS (C1earing Houses Payment System) і BAC (Bankers Automated Clearing Services).

Спеціальна мережа передачі банківських повідомлень на міжнародному рівні S.W.I.F.T. (Society for Worldwide Interbank Financial Telecommunications) – одна з найбільших у світі систем банківських повідомлень. Вона створена задля забезпечення всім учасникам доступу до цілодобової високошвидкісної мережі передачі банківської інформації в стандартній формі за високого ступеня контролю і захисту від несанкціонованого доступу. У цілому система SWIFT являє собою глобальну всесвітню мережу на основі комп'ютерних центрів, з'єднаних різними каналами зв'язку.

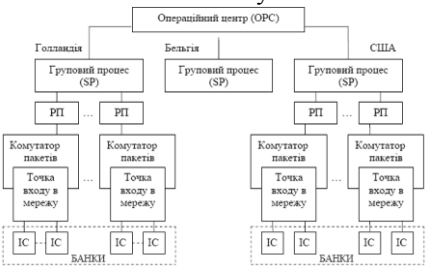


Рисунок 1 – Архітектура SWIFT

Усі питання, пов’язані з безпекою в системі S.W.I.F.T., умовно можна поділити на такі розділи:

– фізична безпека;

– безпека логічного доступу до системи S.W.I.F.T.;

– забезпечення безпеки повідомлень, переданих і збережених у системі;

– безпека обміну повідомленнями “користувач – користувач”.

Говорячи про програмно-апаратну реалізацію системи SWIFT, слід зазначити той факт, що всі можливі варіанти такої реалізації теж чітко стандартизовані. Як інтерфейси різних рівнів для підключення до мережі SWIFT використовуються інтерфейси ST200, ST400 і ST500, які мають різну продуктивність і можуть бути реалізовані на основі різних комп'ютерних платформ.

Банківська сфера з часом вдосконалює методи захисту інформації, такі як системи передачі повідомлень та здійснення міжбанківських операцій. Існує декілька платіжних мереж по всьому світу, але найпоширенішою наразі

є система SWIFT, до якої підключено багато українських банків. Вона має чіткі вимоги до безпеки, які базуються на міжнародних стандартах.

УДК 681.51
Малюков М. К.[1], Зайко Т. А.[2]
[1]студ. гр. КНТ-120м НУ «Запорізька політехніка»
[2]канд. техн. наук, доц. НУ «Запорізька політехніка»

ВРАЗЛИВОСТІ ПРОМИСЛОВИХ SCADA-СИСТЕМ

Промислові системи за останні десятиліття стали використовувати звичайні персональні комп'ютери, операційні системи Windows та Linux, мережеві протоколи TCP/IP, доступ до мережі Інтернет, а також веб-браузери. Це стало причиною появи нових вразливостей, недоліків та розширення можливих загроз для таких систем. Існує велика кількість вразливостей, характерних для промислових систем в АСУ ТП, які можуть бути використані зловмисником. До них належить у першу чергу людський фактор. Дослідження показали, що ті, хто відповідає за безпеку промислових систем, часто мають або недостатній рівень компетентності, або навмисно нехтують безпекою таких систем, щоб забезпечити надійність, підвищити ефективність, доступність, вирішити технологічні проблеми, що виникають під час роботи системи і мінімізувати витрати системи в цілому.

Якщо розглядати вразливості операційної системи, виправлення операційної системи часто не робляться операторами свідомо, оскільки вони не можуть вносити корективи без належної сертифікації оновлень. Тому операційні системи з добре відомими вразливими місцями часто використовуються для забезпечення безперебійної роботи SCADA-систем.

Також існує недолік у автентифікації. Використання загальних паролів є звичайною практикою для промислових систем. Багатофакторні системи автентифікації використовуються рідко, і інформація часто передається по мережі відкрито.

Для управління SCADA-системами співробітники часто встановлюють віддалений доступ через комутацію каналів або через VPN через інтернет. Це може призвести до серйозних інцидентів з інформаційною безпекою.

Незважаючи на те, що зовнішні з'єднання вкрай не рекомендуються для SCADA-систем, в інформаційному звіті CyberX зазначено, що близько 40% АСУ ТП мають пряме підключення до інтернету. Це робить такі системи легкою мішенню для зловмисників.

В АСУ ТП часто використовуються різні типи бездротового зв'язку, включаючи протоколи 802.11, які, як відомо, не забезпечують достатніх можливостей захисту.

Деякі класи пристроїв дистанційного керування, що використовуються в промислових системах для управління технологічними процесами, мають такі характеристики, які не завжди дозволяють виконувати функції безпеки.

Програмне забезпечення для промислових систем зазвичай не містить достатньо функцій безпеки. Крім того, він часто містить значну кількість архітектурних дефектів.

Також, слід враховувати, що якщо обладнання знаходиться поза контрольованою зоною, то за таких умов він не може фізично контролюватися персоналом, і єдиним механізмом фізичного захисту є використання залізних замків та дверей, що, очевидно, не є серйозною перешкодою для зловмисників.

Успішна атака може призвести до несправності SCADA-системи, пошкодження промислового обладнання, порушення виробничого процесу, зниження його якості, шкоди здоров'ю людей, флорі та фауні, порушенню екологічної безпеки та охорони праці.

Більшість SCADA-систем, які використовуються на даний момент, були розроблені більше ніж двадцять п'ять років тому і не захищені як слід, оскільки тоді не розраховували на виникнення корпоративних мереж. Оскільки вони не призначені для роботи в мережі, більшість SCADA-систем, що використовуються у критичній інфраструктурі, не захищені належним чином, саме тому при проєктуванні нових необхідно враховувати усі вищезазначені недоліки задля створення якісного програмного продукту.

УДК 004.5
Зіненко Д.І.[1], Зайко Т.А.[2]
[1]студ. гр. КНТ-120м НУ «Запорізька політехніка»
[2]канд. техн. наук, доц. НУ «Запорізька політехніка»

ЛЮДИНО-МАШИННИЙ ІНТЕРФЕЙС В SCADA-СИСТЕМАХ

Напевно багато хто чув термін «автоматизована система управління», АСК. Слова «автоматизована система» говорять нам про те що в управлінні бере участь людина, на відміну від автоматичної системи управління, де людина не бере участь в процесі управління системою. Сучасні реалії такі, що на даний момент немає можливості повністю замінити людську працю на машинну. Наскільки розумними не були б наші виробництва, все одно вони вимагають нагляду і контролю.

Саме тому при розробці АСК використовують два рівні управління.

Нижній рівень.

Безперервне регулювання технологічних параметрів (температури, тиску, і т.д.) і програмно-логічне керування різними механізмами (засувками,

клапанами, двигунами, конвеєрами і т.п.). Сьогодні базою нижнього рівня є програмовані логічні контролери (Programmable Logical controller)

Верхній рівень АСУ.

Супервізорний рівень, або SCADA (Supervisory Control And Data Acquisition), супервізорний контроль і збір даних.

Так само його можуть називати HMI (Human-machine interface), людино-машинний інтерфейс або HMI / SCADA. Супервізорне управління полягає в ідентифікації ситуації на об'єкті і видачі завдань на нижній рівень. HMI / SCADA - промислове програмне забезпечення, яке реалізує обмін даними з контролером, а так само взаємодіє між особою, яка приймає рішення і безпосередньо системою управління. HMI (Human-machine interface) -візуалізація об'єкта управління і всіх поточних параметрів (мнемосхема).

Людино-машинний інтерфейс (HMI).

Людино-машинний інтерфейс в SCADA-системах реалізується у вигляді мнемосхем. На мнемосхемах відображається основне обладнання, сигнали, стан регулюючих органів та інші частини системи. Мнемосхеми можуть відображати як загальну картину стану системи, технологічного процесу, так і стан окремих агрегатів, пристроїв, значення параметрів і т.п. HMI система може мати декілька десятків вікон з мнемосхемами, трендами, алярмами і т.п. Оператор може перемикатися між ними і працювати з тим чи іншим елементом технологічного процесу, тобто з тієї чи іншої мнемосхемою. Мнемосхема - основний посередник при передачі інформації від системи оператору або інженеру.

Якісно розроблена мнемосхема може сильно вплинути на ефективність роботи виробництва. Грамотна візуалізація виробничого процесу зменшує час, необхідний користувачу для реакції на події, що відбуваються в системі і дозволяє отримати інструмент для зручного і своєчасного управління всією установкою, лінією, цехом. Так само дивлячись на якісну мнемосхему навіть незнайома з виробництвом людина зможе швидко зрозуміти і усвідомити суть процесів що відбуваються. Таким чином можна з легкістю навчати новий персонал і знизити навантаження на операторів технологічних процесів.

Збір і зберігання даних.

Одна з найважливіших функцій SCADA. Обмін даними здійснюється через польову шину. Польова шина складається з інтерфейсу (апаратна, фізична частина шини) і протоколу (програмна частина, логіка обміну інформацією). Таким чином комп'ютер зі встановленою SCADA повинен мати відповідні порти і підтримувати протоколи обміну даних за допомогою яких «спілкується» контролер, встановлений на нижньому рівні системи.

SCADA збирає дані з усіх контролерів і пристроїв нижнього рівня і зберігає їх в одному місці стільки часу, скільки необхідно. Такий підхід забезпечує зручний доступ до інформації про процеси в системі і можливість її подальшого аналізу для поліпшення ефективності роботи виробництва в цілому. Доступ до інформації, яку збирає SCADA можуть мати тільки користувачі, яким такий доступ дозволений. Тобто керівник може бачити всю картину виробництва, а його підлеглі тільки те, що необхідно для виконання їхніх безпосередніх обов'язків.

УДК 004.056.5
Денисенко С.В.[1], Зайко Т.А.[2]
[1]студ. гр. КНТ-138 НУ «Запорізька Політехніка»
[2]канд. техн. наук, доц. НУ «Запорізька політехніка»

АНАЛІЗ ПОКАЗНИКІВ ЕФЕКТИВНОСТІ ЗАСТОСУВАННЯ СИСТЕМ ЗАХИСТУ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Системи захисту програмного забезпечення (СЗПЗ) широко поширені і знаходяться в постійному розвитку, завдяки розширенню ринку програмного забезпечення і телекомунікаційних технологій. Необхідність використання систем захисту СЗПО обумовлена поруч проблем, серед яких варто виділити: незаконне використання алгоритмів, що є інтелектуальною власністю автора, при написанні аналогів продукту (промислове шпигунство); несанкціоноване використання ПО (крадіжка і копіювання); несанкціонована модифікація ПО з метою введення програмних зловживань; незаконне поширення і збут ПО (піратство). Орієнтовна структура наслідків неефективного забезпечення інформаційної безпеки в американських організаціях така : крадіжка конфіденційної інформації – 20-25% від загального річного збитку; фальсифікація фінансової інформації - 21-25%; зараження шкідливими програмами – 11-12%; порушення доступу до Web-сайтів – 1-11%; зрив роботи інформаційної системи – 4-10%; незаконний доступ співробітників до інформації – 4-9%; інші види шкоди – 14-33%.

Для захисту ПО використовується ряд методів, таких як:

Алгоритми заплутування – використовуються хаотичні переходи в різні частини коду, впровадження помилкових процедур – "пустушок", неодружені цикли, перекручування кількості реальних параметрів процедур ПО, розкид ділянок коду по різних областях ОЗУ і т.п.

Алгоритми мутації – створюються таблиці відповідності операндів – синонімів і заміна їх один на одного при кожному запуску програми за певною схемою або випадковим чином, випадкові зміни структури програми.

Алгоритми компресії даних – програма упаковується, а потім розпаковується в міру виконання.

Алгоритми шифрування даних – програма шифрується, а потім розшифровується у міру виконання.

Обчислення складних математичних виразів в процесі відпрацювання механізму захисту – елементи логіки захисту залежать від результату обчислення значення якої-небудь формули або групи формул.

Методи утруднення дизассемблирования – використовуються різні прийоми, спрямовані на запобігання дизассемблирования в пакетному режимі.

Методи утруднення налагодження – використовуються різні прийоми, спрямовані на ускладнення налагодження програми.

Емуляція процесорів і операційних систем – створюється віртуальний процесор і / або операційна система (не обов'язково реально існуючі) і програма-перекладач із системи команд IBM в систему команд створеного процесора або ОС, після такого перекладу ПО може виконуватися тільки за допомогою емулятора, що різко ускладнює дослідження алгоритму ПО.

Нестандартні методи роботи з апаратним забезпеченням – модулі системи захисту звертаються до апаратури ЕОМ, минаючи процедури ОС, і використовують маловідомі або недокументовані її можливості.

Немає сумнівів, що захист критично важливих для власників інформаційних систем відповідає численним міжнародним, національним, корпоративним, нормативним і методичним документам. Застосовуються досить дорогі технічні засоби і впроваджуються строго регламентовані організаційні заходи. Однак немає відповіді на найважливіше питання – наскільки пропоноване або вже реалізоване рішення добре, яка його планована або реальна ефективність.

Поява міжнародного стандарту ОК є якісно новим етапом у розвитку нормативної бази оцінки безпеки ІТ. Порівняння оцінок здійснюється за допомогою загального переліку (набору) вимог для функцій захисту продуктів і систем, а також методів точних вимірювань, які проводяться під час отримання оцінок захисту. Грунтуючись на цих вимогах, в процесі вироблення оцінки рівня захисту встановлюється рівень довіри.

Ефективність захисних заходів (ЗМ) повинна оцінюватися на стадії проектування, для отримання найкращих показників працездатності системи в цілому.

У загальному випадку ефективність ЗМ оцінюється як на етапі розробки, так і в процесі експлуатації системи захисту. В оцінці ефективності ЗМ, в залежності від використовуваних показників і способів їх отримання, можна виділити три підходи:

– класичний;

– офіційний;

– експериментальний.

При розробці і аналізі захисту програмного забезпечення необхідно враховувати існуючу законодавчу базу, при цьому потрібно проводити детальний економічний аналіз ситуації, застосовуючи різні критерії оцінки, а потім створювати стратегію захисту, що включає застосування технічних і організаційних заходів захисту програмного забезпечення.

УДК 004.056

Заікіна К.Є.[1], Зайко Т.А.[2]
[1]студ. гр. КНТ-128 НУ «Запорізька політехніка»
[2]канд. техн. наук, доц. НУ «Запорізька політехніка»

ТЕХНІЧНИЙ РЕГЛАМЕНТ ЗАСОБІВ КРИПТОГРАФІЧНОГО ЗАХИСТУ ІНФОРМАЦІЇ В УКРАЇНІ

У XXI столітті вже майже неможливо здійснювати комерційну діяльність без використання комп'ютера, факсу, мобільного зв'язку і таке інше. Але ефективне використання високих технологій можливе лише за умови належного нормативно-правовому регулювання. Не є винятком і відносини, що виникають у зв'язку з використанням засобів криптографічного захисту інформації (засіб криптографічного захисту інформації). Раніше криптографічні системи з огляду на їх високу вартість і складності використовувалися лише правоохоронними органами. З поширенням же потужних і відносно дешевих персональних комп'ютерів з'явилася реальна можливість використання криптографії як громадянами, так і суб'єктами господарювання.

Для врегулювання використання засобів криптографічного захисту інформації в Україні протягом останніх п'яти-семи років був прийнятий ряд нормативно-правових актів, найважливішими з яких є Положення про порядок здійснення криптографічного захисту інформації в Україні, затверджене Указом Президента України від 22 травня 1998 року № 505, Положення про порядок розроблення, виробництва та експлуатації засобів криптографічного захисту конфіденційної інформації та відкритої інформації з використанням електронного цифрового підпису (ЕЦП), затверджене наказом Департаменту спеціальних телекомунікаційних технологій та захисту інформації Служби безпеки України (Департамент) № 53 від 30 листопада 1999 року (Положення про засоби криптографічного захисту інформації) та Закон України «Про електронний цифровий підпис» від 22 травня 2003 року.

Зазначені нормативно-правові акти визначають порядок розробки, виробництва і експлуатації як звичайних засобів криптографічного захисту інформації, так і використовують засоби ЕЦП. Проте неоднозначність певних правових норм Положення про засоби криптографічного захисту інформації ускладнює або робить неможливим його застосування до певних правовідносин, які виникають у зв'язку з експлуатацією засобів криптографічного захисту інформації та ЕЦП суб'єктами господарювання на Україні.

Кабінет Міністрів України затвердив Технічний регламент засобів криптографічного захисту інформації. Про це повідомила Держслужба спеціального зв'язку та захисту інформації України.

Відповідну постанову «Про затвердження Технічного регламенту засобів криптографічного захисту інформації» прийнято на засіданні Уряду 21 жовтня 2020 року.

Технічний регламент засобів криптографічного захисту інформації (КЗІ) встановлює вимоги до кожного етапу життєвого циклу засобів КЗІ, призначених для захисту відкритої та/або конфіденційної інформації, у відповідності до міжнародних стандартів, що дозволить підвищити рівень довіри до результатів таких процедур.

«Ухвалення цього акту є важливою частиною виконання Угоди про асоціацію між Україною та ЄС і гармонізує національне законодавство з міжнародними та європейськими нормативними документами», – йдеться у повідомленні.

Мова йде, зокрема, про імплементацію положень Регламенту Європейського Парламенту та Ради від 23 липня 2014 року про електронну ідентифікацію та довірчі послуги для електронних транзакцій в межах внутрішнього ринку та про скасування Директиви 1999/93/ЄС та відповідно до положень Закону України «Про технічні регламенти та оцінку відповідності».

Згідно з основними положеннями Технічного регламенту засоби КЗІ вважатимуться такими, що відповідають його суттєвим вимогам, якщо в них реалізовано норми ДСТУ ISO/IEC 19790 «Інформаційні технології. Методи захисту. Вимоги безпеки до криптографічних модулів».

Також оцінка відповідності засобів КЗІ має виконуватись відповідно до вимог ДСТУ ISO/IEC 24759 «Інформаційні технології. Методи захисту. Вимоги до тестування криптографічних модулів».

Для введення в обіг засобів КЗІ виробники мають забезпечити відповідність процесів їх розроблення та виготовлення суттєвим вимогам регламенту.

Адміністрація Держспецзв'язку, у межах сфери своєї відповідальності, перевірятиме відповідність засобів КЗІ вимогам Технічного регламенту.

Постанова набирає чинності з 01 січня 2022 року. Водночас до 1 січня 2027 року засоби КЗІ, які введено в обіг до дня набрання чинності постанови, можуть бути присутні на ринку та введені в експлуатацію.

Підводячи підсумки можна сказати, що акт для забезпечення захисту інформації та електронних систем дуже важливий, так як зараз розробка засобів криптографічного захисту інформації майже не врегульована законодавством та не відповідає сучасному технічному рівню та міжнародним стандартам. Кабінет Міністрів робить все для того, щоб запровадити обов'язкові вимоги національних стандартів у сфері безпеки інформації до засобів КЗІ, процедури оцінки їх відповідності вимогам законодавства.

УДК 004.056.55
Клоков А.О.[1], Зайко Т.А.[2]
[1]студ. гр. КНТ-118 НУ «Запорізька Політехніка»
[2]канд. техн. наук, доц. НУ «Запорізька Політехніка»

ПРОБЛЕМИ ТА МЕТОДИ ШИФРУВАННЯ ДІЛОВОГО ДОКУМЕНТООБІГУ

Для кожного підприємства, без сумніву, пропорційно його розвитку збільшується обсяг документації, що використовується. Фізичне поширення, розмноження та пошук документів значно знижує швидкість та якість прийняття рішень у цілому. Через це все більше підприємств впроваджують системи електронного документообігу (СЕД). Такі системи мають забезпечувати цілісність, доступність та конфіденційність корпоративної інформації, але в силу того, що СЕД набули широкого застосування відносно недавно, їх робота недостатньо налагоджена, і незважаючи на переваги від використання таких систем, залишаються певні невирішені проблеми.

Одна з най масштабніших проблем СЕД – захист інформації від несанкціонованого доступу. Адже є багато бажаючих перехопити конфіденційну інформацію, задля використання у своїх інтересах, або пошкодження її, для зменшення конкурентоспроможності підприємства.

Рішенням таких проблем є криптографічне шифрування. Загалом шифрування – це оборотне перетворення даних, з метою приховування інформації. Шифрування відбувається із застосуванням криптографічного ключа. Ключ – це певна кількість символів, сформованих вільним чином з символів, що доступні у системі шифрування.

Першою відомою пам'яткою шифрування прийнято вважати єгипетський текст, який було створено напевно десь в 1900 році до нашої

ери., у якому використовувались другі символи замість відомих єгипетських ієрогліфів.

До нашого часу, криптографія займалася виключно забезпеченням конфіденційності повідомлень (тобто шифруванням) – перетворенням повідомлень із зрозумілої форми в незрозумілу і зворотнє відновлення на стороні одержувача, роблячи його неможливим для прочитання для того, хто перехопив або підслухав без секретного знання (а саме ключа, необхідного для дешифровки повідомлення). В останні десятиліття сфера застосування криптографії розширилася і включає не лише таємну передачу повідомлень, але і методи перевірки цілісності повідомлень, ідентифікування відправника/одержувача (аутентифікація), цифрові підписи, інтерактивні підтвердження, та технології безпечного спілкування, тощо.

Виділяють два методи криптографічного шифрування: симетричне та асиметричне.

У симетричному шифруванні один ключ, який зберігається в секреті, служить і для шифрування, і для дешифрування. Переваги симетричних методів шифрування:

– велика пропускна здатність;

– ключі мають невеликий розмір;

– застосовуються, як основа для різних криптографічних механізмів.

Серед недоліків даного шифрування:

– необхідно використовувати значну кількість ключів;

– необхідно досить часто змінювати ключі;

– при використанні кількома клієнтами ключ слід засекречувати на двох кінцях.

К симетричним алгоритмам шифрування відносять: Twofish, Serpent, AES, Blowfish, CAST5, RC4, TDES, та IDEA.

Асиметричне шифрування, або метод відкритого ключа, передбачає застосування в парі двох відмінних ключів, а саме секретний та відкритий. В асиметричному шифруванні інформація шифрується відкритим ключем, а розшифровування відбувається тільки відповідним секретним ключем та навпаки. Неможливим є використовування відкритого ключа із однієї пари та секретного ключа із іншої пари. Найбільшою перевагою є вищий рівень захисту цілісності даних. Але при цьому вагомими є вади швидкості дії асиметричних методів. До асиметричних алгоритмів шифрування належать RCA та ECC.

Оскільки існує запит не тільки на максимальний захист інформації, а й на швидкість її поширення та обробки, у новітніх крипто системах, застосовуються комбінації симетричних та асиметричних алгоритмів, з метою отримання переваг обох схем.

Захищений документообіг не тільки допомагає у боротьбі зі зловмисниками та «брудною» конкуренцією, а й лежить в основі успішного розвитку підприємств, дозволяє більш ефективно використовувати часові та фінансові ресурси, забезпечує стабільність і надійність в очах партнерів. У цьому важливу роль має галузь кібербезпеки, що вважається перспективною та затребуваною як для підприємств, що розвиваються, так і для цілих держав.

УДК 303.7
Медведєв С.Р.[1], Зайко Т.А.[2]
[1]студ. гр. КНТ-127 НУ «Запорізька Політехніка»
[2]канд. техн. наук, доц. НУ «Запорізька Політехніка»

УСКЛАДНЕННЯ ЛОГІКИ ПРОГРАМИ ЯК РЕАЛІЗАЦІЯ ЗАХИСТУ ВІД ДИЗАСЕМБЛЮВАННЯ

На сьогоднішній день, взлом і піратство програмного забезпечення є проблемою для багатьох початківців і навіть для досвідчених розробників і програмістів. Зловмисники в різних технологічних сферах діяльності всіляко намагаються викрасти інформацію як у "сирих" або нових програмних продуктів, так і у тих продуктів, які вже давно на ринку. Як приклад, можу навести багаточисленні взломи відеоігор від різних хакерських угруповань. Взлом і піратство становлять загрозу для розробників як у фінансовому плані, так і в плані довіри. Одним з видів взлому є процес дизасемблювання.

Більшість програм поширюються в вигляді виконуваного файлу. Для взлому програми з метою піратства необхідно перш за все перевести машинний код в асемблерне уявлення. Такий процес носить назву дизасемблювання. Якщо вдасться зробити нерозв'язною задачу зловмисника вже на етапі дослідження програми, шанси на збереження інтелектуальної власності стануть стовідсотковими.

Процес дизасемблювання дуже ризиковий процес з більшістю втрат, тому що він однонаправлений. При повторному асемблюванні відновленого тексту немає гарантії отримати той же самий код, тобто отримана програма, швидше за все, відмовиться працювати. Будь-яка спроба модифікації дизасемблювання тексту розвалить програму остаточно. Справа в тому, що асемблер замінює всі мітки на реальні зміщення, інакше кажучи - на константи. При внесенні змін до програми необхідно скорегувати всі посилання на мітки.

Першим і універсальним методом протидії дизасемблювання програми є шифрування. Вочевидь, що дизасемблювання зашифрованого коду марно. Рекомендується використовувати шифрування з відкритим ключем (алгоритм

RSA, шифр Єль-Гамаля і ін.). В цьому випадку можлива вдала спроба розшифрувати код і зрозуміти логіку роботи захисного механізму не дозволить вносити зміни в захищений код, так як для повноцінної подальшої роботи програми ці зміни необхідно впровадити в код в зашифрованому вигляді. А зловмиснику доступний лише ключ для розшифровки. Можлива атака в даному випадку - знаходження «секретного» ключа за допомогою трудомістких математичних обчислень в залежності від використовуваного алгоритму шифрування. Для захисту програм від дизасемблювання, не рекомендується використання симетричних криптографічних алгоритмів.

Другі методи, які непогано зарекомендували себе на практиці, використовують разом з шифруванням архівування програмного коду. До переваг даних методів відносять і зменшення розміру виконуваного файлу. Найбільш відомими архіваторами на сьогоднішній день є WinRAR та 7-Zip. Однак слід враховувати, що алгоритми роботи цих архіваторів відомі багатьом зловмисникам.

Треті методи, які засновані на динамічній зміні коду програми в процесі виконання, теж широко поширені на практиці,. Суть цих методів зводиться до отримання істинних здійсненних команд на етапі виконання програми шляхом деякого перетворення початкових кодів. Найчастіше цей спосіб захисту називають самостійно-згенерованим кодом.

Четвертий метод, більш простий, ніж перераховані вище, адже його реалізацію можна почати ще на етапі розробки коду програми. Називається він "обманом дизасемблера". Метод передбачає такий стиль програмування, який викликає порушення правильної роботи стандартного дизасемблера за рахунок нестандартних прийомів використання окремих команд або ж порушення загальноприйнятих угод.

На сьогоднішній день, існує чимало програм дизасемблерів. Наприклад: PE-Explorer, FlexHex, Radare2, PE-bear, Hiew, IDA тощо. Але найкращими представниками даної функціональності є IDA Pro Disassembler та вільний багатоплатформений фреймворк Radare2.

IDA є комерційним продуктом, ціни якого коливаються в межах від 400 до 4000 тисяч доларів. Існує також безкоштовна версія, але в ній вирізаний майже весь контент та немає підтримки модулів.

Дизасемблер IDA Pro має консольну і графічну версії. Підтримує велику кількість форматів виконуваних файлів. Однією з відмінних рис IDA Pro є можливість дизасемблювання байт-коду віртуальних машин Java і .NET. Також підтримує макроси, плагіни і скрипти, а останні версії містять інтегрований відладчик.

Фреймворк Radare2 є безкоштовним кросплатформним продуктом, який спочатку розроблявся одною людиною, а після - цілою командою.

Radare2 доступний у вигляді бібліотек і утиліт, в склад яких входять: власний асемблер/дизасемблер, утиліта для роботи з різними виконуваними файлами, утиліта для отримання хеш-значень даних, декомпілятор тощо. Також, на його архітектурі був оснований інтерактивний дизасемблер Cutter.

Підсумовуючи наведене, можна сказати, що за захист від дизасемблювання відповідають методи ускладнення логіки програм, а вся відповідальність при їх використанні чи невикористанні, при розробці програм, відповідально лежить на плечах розробників програмного забезпечення.

УДК 004.056.5
Зайко Т.А.[1], Лизя Є.С.[2]
[1]канд. техн. наук, доц. НУ «Запорізька Політехніка»
[2]студ. гр. КНТ-217 НУ «Запорізька політехніка»

ЗАХИСТ ВІД SQL-ІН'ЄКЦІЙ В PHP І MYSQL

Вебсайти стали частиною нашого життя. Більшість сьогоденних справ ми виконуємо за допомогою функціоналу тих чи інших веб сервісів, а це говорить про те, що веб-сайти мають доступ до баз даних, де зберігається безліч особистої інформації користувачів. Хакери та інші злочинники використовують, так звану, SQL-ін'єкцію для того, щоб вкрасти та змінити інформацію, яка зберігається в БД (базі даних). Саме тому виникла необхідність створити методи для захисту інформації від SQL-ін’єкцій.

SQL-ін'єкція [1] – це атака, спрямована на веб-додаток, в ході якої конструюється SQL-вираз з введених користувачем даних, шляхом простої конкатенації (наприклад, $ query = "SELECT * FROM users WHERE id =". $ _ REQUEST ["id"]) . У разі успіху хакер може змінити логіку виконання SQL-запиту так, як це йому потрібно. Найчастіше він виконує простий fingerprinting СУБД (визначення типу системи управління базами даних), а також витягує таблиці з найбільш "цікавими" іменами (наприклад "users"). Після цього, в залежності від привілеїв, з якими запущено вразливий додаток, він може звернутися до захищених частин бекенду (серверна сторона) веб-додатку (наприклад, прочитати файли на стороні адміністратора або виконати довільні команди).

Для захисту вебсайтів розробники винайшли багато рішень. Нище наведені два методи [2], які є найбільш популярними та ефективними серед вебпрограмістів.

Перший метод полягає в тому, що дані підставляються в запит тільки через плейсхолдери (заповнювачі). Будь-які дані повинні потрапляти в запит не безпосередньо, а через якогось представника, а саме – підставний вираз.

Найкращим варіантом буде форматувати дані безпосередньо перед виконанням запиту – таким чином ми завжди будемо впевнені в тому, що дані форматуються правильно, це робиться тільки один раз, і відформатовані дані потраплять строго за призначенням – в БД і нікуди більше. Існує два варіанти реалізації плейсхолдерів – серверний і клієнтський. У першому випадку запит так і йде на сервер з плейсхолдерами, а дані відправляються окремо від нього. Англійською має назву native prepared statements – «рідні» підготовлені вирази – тобто, обробка плейсхолдерів здійснюється самою СУБД (системою управління базами даних) на сервері. У другому випадку дані форматуються і підставляються в рядок запиту на місце плейсхолдерів прямо на клієнті, формуючи класичний SQL запит, який потім йде в базу звичайним порядком.

Другий метод полягає в тому, щоб ідентифікатори і ключові слова підставлялись тільки з білого списку, прописаного в нашому коді. Ми зіштовхуємося з необхідністю підставляти в запит не тільки дані, але і інші елементи – ідентифікатори (імена полів і таблиць) і навіть елементи синтаксису, ключові слова. Нехай навіть такі незначні, як DESC або AND, але вимоги до безпеки таких підстановок все одно повинні бути не менш суворими. Суть методу полягає в тому, що всі можливі варіанти вибору повинні бути жорстко прописані в коді, і в запит повинні потрапляти тільки вони, на підставі призначеного для користувача введення.

Я вважаю, що для повноцінного захисту вебсайту від SQL-ін’єкції потрібно використовувати обидва методи, оскільки лише одного з них може бути не достатньо. Оскільки, якщо підставляти імена полів без попередньої фільтрації (тобто використовувати тільки метод з використанням плейсхолдерів), можна отримати ін'єкцію іншого роду – адже користувач тоді може вписати в ті імена полів, які йому змінювати не можна. Скажімо, якщо ми формуємо SQL (Structured Query Language) запит автоматично на базі масиву $ _POST, то хакер при реєстрації додає в форму поле admin зі значенням «1» і стає адміністратором. Отже, на мою думку, спочатку потрібно отримувати ідентифікатор з білого списку, а потім додавати його через плейсхолдер.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. SQL инъекции [Електрон. ресурс]. – Режим доступу: https://xakep.ru/2011/12/06/57950/.

2. Защита от SQL инъекций [Електрон. ресурс]. – Режим доступу: https://habr.com/ru/post/148701.

УДК 004.056.5
Пархоменко В.В.[1], Зайко Т.А.[2]
[1]студ. гр. КНТ-138 НУ «Запорізька Політехніка»
[2]канд. техн. наук, доц. НУ «Запорізька Політехніка»

АНАЛІЗ ЕФЕКТИВНОСТІ МЕТОДІВ ПІДВИЩЕННЯ СТІЙКОСТІ ЕЛЕКТРОННИХ КЛЮЧІВ ДЛЯ ЗАХИСТУ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Проблеми інформаційної безпеки стали актуальними одночасно з виникненням перших електронно-обчислювальних машин і удосконалювались разом з розвитком технічної бази. Вдосконалення технічної бази систем обробки інформації забезпечило подальший розвиток концепції безпеки, в рамках якої стали виділяти захист технічних засобів, програмного забезпечення і інформації.

Програмне забезпечення – сукупність програм системи обробки інформації і програмних документів, необхідних для експлуатації цих програм.

Безпека програмного забезпечення в широкому змісті є властивістю даного програмного забезпечення функціонувати без прояву різноманітних негативних наслідків для конкретної комп'ютерної системи. Під рівнем безпеки програмного забезпечення (ПЗ) розуміється ймовірність того, що при заданих умовах у процесі його експлуатації буде отриманий функціонально придатний результат.

На сьогоднішній день постали такі проблеми, як незаконне використання алгоритмів, несанкціоноване використання, модифікація, поширення і збут програмних продуктів. Однією з головних проблем є несанкціоноване використання програмного забезпечення. Існують різні типи захисту від несанкціонованого використання, такі як : захист за допомогою компакт-дисків, захист за допомогою електронних ключів, локальний програмний захист, мережевий захист програмного забезпечення, захист програмного забезпечення з прив’язкою до параметрів комп’ютера і активація та інші.

Найпоширенішим захистом програмного забезпечення від несанкціонованого використання є захист за допомогою електронних ключів.

Електронний ключ – це апаратна частина системи захисту, що представляє собою плату з мікросхемами пам'яті і, у деяких випадках, мікропроцесором, поміщену в корпус і призначену для установки в один з стандартних портів ПК (COM, LPT, PCMCIA, USB) або слот розширення материнської плати. Так само як такий пристрій можуть використовуватися смарт-карти (SmartCard).

Електронні ключі по архітектурі можна підрозділити на ключі з пам'яттю (без мікропроцесора) і ключі з мікропроцесором (і пам'яттю).

Найменш стійкими є системи з апаратною частиною першого типу. У таких системах критична інформація (ключ дешифрування, таблиця переходів) зберігається в пам'яті електронного ключа. Для дезактивації таких захистів у більшості випадків необхідно наявність у зловмисника апаратної частини системи захисту (перехоплення діалогу між програмною й апаратною частинами для доступу до критичної інформації).

Найбільш стійкими є системи з апаратною частиною другого типу. Такі комплекси містять в апаратній частині не тільки ключ дешифрування, але і блоки шифрування/дешифрування даних, у такий спосіб при роботі захисту в електронний ключ передаються блоки зашифрованої інформації, а приймаються відтіля розшифровані дані. У системах цього типу досить складно перехопити ключ дешифрування тому, що усі процедури виконуються апаратною частиною, але залишається можливість примусового збереження захищеної програми у відкритому виді після відпрацьовування системи захисту. Крім того, до них застосовні методи криптоаналізу.

Позитивні фактори захисту за допомогою електронних ключів: значне утруднення нелегального поширення і використання ПЗ, рятування виробника ПЗ від розробки власної системи захисту, висока автоматизація процесу захисту ПЗ, наявність API системи для більш глибокого захисту, можливість легкого створення демо-версій, досить великий вибір таких систем на ринку.

До негативних факторів відносяться: утруднення розробки і налагодження ПЗ через обмеження з боку систем захисту, додаткові витрати на придбання системи захисту і навчання персоналу, уповільнення продажів через необхідність фізичної передачі апаратної частини, підвищення системних вимог через захист (сумісність, драйвери), зниження відмовостійкості ПЗ, несумісність систем захисту і системного або прикладного ПЗ користувача, несумісність захисту й апаратури користувача, обмеження через несумісність електронних ключів різних фірм, зниження розширюваності комп'ютерної системи, утруднення або неможливість використання захищеного ПЗ в переносних і блокнотних ПК, наявність в апаратної частини розмірів і погроза крадіжки апаратного ключа.

Одним із найбільш перспективних методів захисту електронних ключів є використання динамічних кодів та пропрієтарних протоколів. Це дозволяє забезпечити унікальність переданих даних при кожній автентифікації користувача. Завдяки цьому, зчитавши дані при обміні (ці дії мають назву сніфінг), передавши ідентичні дані до зчитувача системи контролю і управління доступом, зловмисник не отримує результату у вигляді дозволу на прохід.

Можна зробити висновок, що зараз існує досить багато проблем із захистом програмного забезпечення, і тому розробляється велика кількість систем захисту програмного забезпечення, для їх подолання, без яких неможлива безпечна робота, як з програмним забезпеченням, так і з комп'ютером. Було розглянуто захист програмного забезпечення за допомогою електронних ключів та за результатами проведеного аналізу, виявлено найбільш перспективні методи для підвищення стійкості електронних ключів.

УДК 004.056.5
Зайко Т.А.[1], Мироненко Н.В.[2]
[1]канд. техн. наук, доц. НУ «Запорізька Політехніка»
[2]студ. гр. КНТ-120м НУ «Запорізька політехніка»

АНАЛІЗ ПРОБЛЕМ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ АСК ТП

В останні роки питання інформаційної безпеки систем керування технологічними процесами постало дуже гостро. Особливий інтерес виник після інцедентів з комп'ютерними вірусами, які атакували ядерні об'єкти, держустанови і промислові об'єкти у різних країнах світу.

Також великого резонансу набув випадок в Україні, який трапився у 2015 році. Тоді невідомі хакери зламали систему керування телемеханікою компанії «Прикарпаттяобленерго», яка спеціалізується на передачі та забезпеченні електроенергією споживачів Івано-Франківської області. У результаті, протягом декількох годин більша частина області і місто залишалося без електроенергії [1].

До цих випадків вважалося, що в роботу АСК ТП досить важко втрутитися, адже програмне забезпечення кожної АСК ТП є унікальним і закритим; локальна мережа АСК ТП вирішує проблеми обмеження доступу; проникнення в АСК ТП пов'язано з великими інтелектуальними витратами [2]. Однак, усіх цих обмежень виявилося недостатньо, щоб забезпечити повний захист і тому на сьогодні важливою задачею є підвищення інформаційної безпеки таких систем.

Для підвищення безпеки необхідно спочатку розглянути проблеми, які призводять до виникнення ризиків:

– слабкий захист від несанкціонованого доступу;
– недекларовані можливості SCADA;
– використання слабко захищених бездротових комунікацій;
– відсутність чітких границь між різними сегментами мережі;
– несвоєчасне чи некоректне оновлення програмного забезпечення;
– дистанційні методи керування;

– відмова або недостатня кількість засобів захисту (наприклад, відсутність антивірусу чи паролю);

– створення систем без урахування кращих практик розробки безпечного коду;

– людський фактор [3].

Для зменшення та уникнення розглянутих ризиків необхідно відслідкувати тенденції розвитку кіберзагроз при проєктуванні і розробці нових АСК ТП. Однак, оскільки змінити склад і якість АСК ТП, які введено в експлуатацію, для підвищення інформаційної безпеки є досить складне і дороге завдання, то потрібно проводити регулярну роботу з контролю та попередження потенційних загроз.

Для нейтралізації потенційних загроз використовується дві групи заходів: адміністративно-організаційні та програмно-технічні.

Перша група пов’язана з формуванням програми робіт із надання інформаційної безпеки (ІБ) АСК ТП і розробкою набору документів, які регламентують високорівневий підхід до забезпечення ІБ, а також описують політику розвитку АСК ТП [4]. Основна мета ІБ – підтримувати необхідний рівень безпеки на підприємстві, а також підтримувати неперервність виробництва

Основний набір засобів забезпечення ІБ утворюють програмно-технічні заходи. До них відносяться наступні дії:

– керування доступом;

– забезпечення цілісності;

– створення безпечної міжмережевої взаємодії;

– антивірусний захист;

– аналіз захищеності;

– виявлення вторгнень;

– неперервний моніторинг стану системи ІБ, виявлення інцедентів, реагування [5].

Перелічені дії дозволять знизити ризики та попередити потенційні загрози. Це є особливо актуальною задачею у теперішній час інформаційної війни, коли атака на небезпечний виробничий об’єкт або критично важливий об’єкт може призвести до невиправних дій.

Підводячи підсумки, варто сказати, що проблема інформаційної безпеки АСК ТП є дуже великою проблемою на сьогодні. Вона потребує негайних дій, бо незахищеність у даній сфері призводить не тільки до значної втрати грошових та часових ресурсів, а і до спричинення дискомфорту або навіть створення загрози людині.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. В Украине зафиксирована первая в истории государства успешная хакерская атака на АСУ ТП [Электрон. ресурс] – Режим доступа: http://digitalsubstation.com/blog/2015/12/25/v-ukraine-zafiksirovana-pervaya-v-istorii-gosudarstva-uspeshnaya-hakerskaya-ataka-na-asu-tp/.

2. Пищик, Б. Н. Безопасность АСУ ТП [Текст] / Б. Н. Пищик // Вычислительные технологии. 2013. – №18. – С. 170–175.

3. Стандарты безопасности АСУ ТП [Электрон. ресурс] – Режим доступа: https://www.slideshare.net/CiscoRu/ss-8690963.

4. Демидович Д. И. Современые технологии защиты информации в АСУ ТП [Текст] / Д. И. Демидович. // Сборник 56-ой научной конференции аспирантов, магистрантов и студентов БГУИР. – 2020. – С. 15–17.

5. Роль АСУ ТП на нефтеперерабатывающем заводе [Электрон. ресурс] – Режим доступа: https://cccp-online.ru/rol-asu-tp-na-neftepererabatyvayushhem-zavode/.

УДК 303.7

Петелін Д.Д.[1], Зайко Т.А.[2]
[1]студ. гр. КНТ-118 НУ «Запорізька Політехніка»
[2]канд. техн. наук, доц. НУ «Запорізька Політехніка»

ФІЗИЧНІ ЗАСОБИ ПІДВИЩЕННЯ ЦИФРОВОЇ БЕЗПЕКИ

В останні роки багато компаній добре усвідомили необхідність управління цифровою безпекою підприємства. Ефективне управління питаннями цифрової безпеки набуває все більшого значення для компаній по мірі їх зростання і просування на нові ринки товарів і послуг. Клієнтам важливо знати, що дотримується конфіденційність їхніх персональних і ділових даних. Інвесторам необхідна впевненість в тому, що бізнес та інформаційні активи компанії захищені. Ділові партнери очікують, що компанія буде функціонувати без збоїв, які можуть бути викликані помилками в роботі інформаційних систем, навмисними або ненавмисними діями персоналу, шкідливим програмним забезпеченням та іншими факторами.

Управління цифровою безпекою – це комплекс заходів, які ґрунтуються на підході, що враховує бізнес-ризики, призначені для розроблення, впровадження, функціонування, моніторингу, перегляду, підтримування та вдосконалення інформаційної безпеки.

Побудова ефективної системи управління цифровою безпекою – це не разовий проект, а комплексний процес, спрямований на мінімізацію зовнішніх і внутрішніх загроз при обліку обмежень на ресурси та час.

Для побудови ефективної системи цифрової безпеки необхідно спочатку описати процеси діяльності. Потім слід визначити поріг ризику – рівень загрози, при якому вона потрапляє в процес управління ризиками. Потрібно побудувати таку систему цифрової безпеки, яка забезпечить досягнення заданого рівня ризику.

Фізичні види порушень включають фізичне пошкодження апаратних засобів автоматизованих систем, ліній зв'язку і комунікаційного устаткування, крадіжки або несанкціоноване ознайомлення з вмістом носіїв інформації, що зберігаються в неналежних місцях, розкрадання носіїв інформації, відмови апаратних засобів та ін.

Як же запобігти фізичним видам порушень? Фізичні засоби захисту – це засоби, необхідні для зовнішнього захисту засобів обчислювальної техніки, територіїта об'єктів на базі ПК, які спеціально призначені для створення фізичних перешкод на можливих шляхах проникнення і доступу потенційних порушників до компонентів інформаційних систем та інформації, що захищаються.

До законодавчих засобів захисту відносяться законодавчі акти, які регламентують правила використання і обробки інформації обмеженого доступу і встановлюють заходи відповідальності за порушення цих правил.

Найпростіший і надійний спосіб захисту інформації від загроз несанкціонованого доступу (НСД) - режим автономного використання ПК одним користувачем у спеціально виділеному приміщенні при відсутності сторонніх осіб. У цьому випадку роль замкненого контуру захисту виконує виділене приміщення, а фізичний захист- вікна, стіни, підлога, стеля, двері. Якщо стіни, стеля, підлога і двері міцні, підлога не має люків, які з'єднуються з іншими приміщеннями, вікна і двері обладнані охоронною сигналізацією, то стійкість захисту буде визначатись технічними характеристиками охоронної сигналізації при відсутності користувача в неробочий час.

У робочий час, коли ПК працює, можливий витік інформації каналами побічного електромагнітного випромінювання. Для усунення такої загрози здійснюються спеціальні дослідження щодо апаратих засобів та їх випромінювання, основним змістом яких є атестування та категорування засобів і об'єктів електронно-обчислювальної техніки (ЕОТ) з видачею відповідного дозволу на експлуатацію. Крім того, двері приміщення повинні бути обладнані механічним або електромеханічним замком. У деяких випадках, коли відсутня охоронна сигналізація, на період тривалої відсутності користувача ПК для підвищення безпеки доцільно системний блок і машинні носії інформації зберігати в сейфі. Використання в деяких

ЕОМ у системі вводу-виводу BIOS апаратного паролю, що блокує завантаження і роботу ПК, не зовсім надійно забезпечує захист від загроз НСД, оскільки при відсутності на корпусі системного блоку механічного замка та самого власника-користувача апаратна частка BIOS-носія паролю може бути замінена на іншу таку ж, оскільки вузли BIOS уніфіковані, але вже з відомим значенням паролю. Саме тому механічний замок, що блокує вмикання і завантаження ПК є найбільш ефективним заходом у цьому випадку.

Спектр сучасних фізичних засобів захисту дуже широкий. До цієї групи засобів захисту належать також різні засоби екранування робочих приміщень та каналів передачі даних.

Можна зробити висновок, що інформацію слід захищати не тільки від внутрішніх загроз, а також і від зовнішніх.

УДК 004.056.57
Зайко Т.А.[1], Сокол Р.В.[2]
[1] канд. техн. наук, доц. НУ «Запорізька Політехніка»
[2]студ. гр. КНТ-128 НУ «Запорізька Політехніка»

ЗАХИСТ ВІД ПРОГРАМ-ВИМАГАЧІВ

У часи повної автоматизації систем, як банківських, так і державних, окрім ергономічності та автономності, важливо приділяти не меншу увагу і до питань безпеки й надійності цих самих систем, бо через це дуже збільшився процент програм-вимагачів та шахраїв.

Згідно зі статистикою ресурсу Purplesec шкода завдана програмами-вимагачами росте дуже швидко, протягом часу. Якщо у 2018 році збитки від такого ПЗ були близько 8 мільйонів доларів, то у 2020 році ця цифра збільшилась вже до 20 мільйонів доларів на рік.

Програма-вимагач – це шкідливе програмне забезпечення, яке шифрує важливі файли на вашому ПК, щоб зробити їх недоступними, і погрожує опублікувати або видалити їх, якщо не буде виплачена певна грошова сума (викуп). Але навіть після виплати викупу немає ніяких гарантій, що ви зможете відновити свої файли або знов отримати доступ до вашого комп'ютера [1].

Найчастіше таке ПЗ потрапляє на комп’ютер користувача через e-mail. Зазвичай він починається з класичного фішингового електронного листа, який служить приманкою для завантаження зараженого файлу. У більшості випадків зараження програмою-вимагачем відбувається через відкриття прикріпленого PDF, DOC або XLS файлу. Відображення розширень файлів за замовчуванням деактивовано в більшості поштових клієнтів, тому

користувач, як правило, не може розпізнати формат файлу, на перший погляд [2].

Також популярним способом потрапляння програм-вимагачів на девайси є завантаження зараженого файлу з мережі інтернет або встановлення неліцензійного програмного забезпечення, що насправді являє собою вірус.

Відкриття шкідливого файлу, є безповоротним моментом, бо після цього проводиться установка на відповідну систему програми-вимагача. Як тільки активується програма-вимагач, починається фактична шкода: розпочинається процес шифрування. Окремі файли в одній системі або навіть декількох системах в межах мережі локальної можуть бути зашифровані. Відтепер користувач більше не має доступу до певних файлів або до всього свого комп'ютера. Він повністю втратив свої права адміністратора. Контроль знаходиться в руках хакера.

Як тільки все зашифровано, на екрані жертви з'являється повідомлення. Тут хакер вимагає викуп, щоб повернути користувачу доступ до файлів. Після чого зловмисникам залишається лише чекати, поки жертва заплатить викуп.

Для захисту від програм-вимагачів необхідно розробляти план кібербезпеки від шкідливих програм. Оскільки таке ПЗ дуже важко виявити та боротися з ними, для боротьби з ним слід використовувати різні механізми захисту. Найважливіший захист - це навчання та сенсибілізація працівників [3]. Але, звісно, вивчення проблеми є актуальним не лише для працівників компаній та і для пересічних користувачів ПК та мережі інтернет, бо обізнаність дозволяє утримувати проблему від себе як можна далі.

Якщо ж комп'ютер був вражений програмою-вимагачем, то найпростішим рішенням буде резервна копія вашої системи. Таким чином можна завантажити стару версію без зараження. Це зменшує втрату даних якомога менше. Резервне копіювання можна зробити вручну або автоматично. Хмарне рішення для компаній стане чудовою можливістю для резервного копіювання даних[2].

На мою думку, найкращим захистом від програм-вимагачів – є комплексний підхід до цієї проблеми, а саме: використовувати антивіруси, що мають функції відстеження таких програм, вивчати та бути у курсі того, які програми вимагачі існують та що вони собою уявляють і у разі отримання підозрілих листів поводитися логічно та раціонально, не піддаючись на хитрощі шахраїв. Також не малу частину захисту від такого роду ПЗ буде раціональне використання інтернет-ресурсів, які ви відвідуєте та дуже ретельна фільтрація ресурсів з яких будуть завантажені будь-які файли.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. AVAST “Защита от программ-вымогателей: часто задаваемые вопросы [Электрон. ресурс]. – Режим доступа : https://support.avast.com/ru-ua/article/Antivirus-Ransomware-Shield-FAQ/

2. ESET “Програми-вимагачі”. [Електрон. ресурс]. – Режим доступу : https://www.eset.com/ua-ru/support/information/entsiklopediya-ugroz/programma-vymogatel/

3. Hornetsecurity “Ransomware” [Electronic resource]. – Access mode : https://www.hornetsecurity.com/en/knowledge-base/ransomware/

УДК 004.056.5

Зайко Т.А.[1], Фоміна М.О.[2]
[1]канд. техн. наук, доц. НУ «Запорізька Політехніка»
[2]студ. гр. КНТ-118 НУ «Запорізька Політехніка»

РЕФОРМА КІБЕРБЕЗПЕКИ В УКРАЇНІ

У часи повної автоматизації систем, як банківських так і державних (наприклад, додаток «Дія»), окрім ергономічності та автономності, важливо приділяти не меншу увагу і до питань безпеки цих самих систем.

Згідно рейтингу дослідницької компанії Compratech, яка порівнювала безпеку комп’ютерних систем 76 країн світу, в особливості країни Європи, Азії та Південої Америки, у 2020 Україна зайняла 39 місце, що на 28 пунктів вище ніж у 2019 [1].

Як і у всіх країнах, в Україні питання кібербезпеки регламентуються урядом та відповідними законами. Ще у 2016 році указом президента був створений Національний координаційний центр кібербезпеки (НКЦК) – робочий орган Ради національної безпеки і оборони України. Він координує та контролює діяльність суб'єктів сектору безпеки і оборони, які забезпечують кібербезпеку у відповідних сферах, – Держспецзв'язку, СБУ, Національної поліції, Національного банку, Генерального штабу ЗСУ, Служби зовнішньої розвідки та Головного управління розвідки Міністерства оборони України.

Окрім створення НКЦК, у 2016 була розроблена стратегія кібербезпеки України, на наступні 4 роки. Саме тому, 24.09.2020 РНБО почали займатися розробкою нової стратегії кібербезпеки України. На відміну від розробки стратегії у 2016, у 2020 році в перший раз до розробки були залучені фахівці з приватного сектору.

Після подій 2017 року, а саме після масштабної хакерської атаки сімейством вірусів Petya, країна втратила 0,5% ВВП. Це змусило уряд

займатися детальною розробкою закону про забезпечення регламентованих норм запровадження кібербезпеки для компаній і установ. І вже 05.10.2017 перша версія закону «Про основні засади забезпечення кібербезпеки України» була затверджена урядом. Протягом всього часу, у закон були внесені правки 3 рази: 08.07.2018, 03.07.2020, 24.10.2020.Після другої правки, (08.07.2018) закон набув чинності.

Закон передбачає створення Національної телекомунікаційної мережі та Державного центру кіберзахисту. Крім того, це перший закон в Україні, який офіційно визначає для держави такі терміни: кіберзагроза, кібершпигунство, кіберзлочинність та кіберзахист.

Останні дві правки у 2020 році не внесли категоричних змін у закон, окрім уточнення деякої термінології та посилань на інші закони.

За словами Олександра Кардакова [3] через рік після створення закону, у 2018, чверть корпорацій серйозно почали займатися кібербезпекою, а ще треть обговорюють створення таких систем.

У часи пандемії, питання кібербезпеки особливо важливі, адже карантин змінив формат роботи багатьох фірм та компаній. Через дистанційну роботу, багато придприємств надають доступ з мережі інтернет до внутрішних інформаційних ресурсів, що наражає небезпеку на ці дані. Саме тому, за словами Сергія Демедюка, на базі НКЦК фахівці вже працюють над розробкою унікального програмного забезпечення для виявлення кібератак ще на початковому етапі[4].

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Comparitech. Which countries have the worst (and best) cybersecutiry : [Electronic resource]. – Access mode : https://www.comparitech.com/blog/vpn-privacy/cybersecurity-by-country/

2. Верховна Рада України [Електрон. ресурс]. – Режим доступу : https://zakon.rada.gov.ua/laws/show/2163-19/ed20171005#Text

3. Радіо Свобода [Електрон. ресурс]. – Режим доступу : https://www.radiosvoboda.org/a/29336511.html

4. Українська правда. [Електрон. ресурс]. – Режим доступу : https://www.pravda.com.ua/columns/2020/08/7/7262150/

УДК 004.7
Шутко А.С.[1], Зайко Т.А.[2]
[1]студ. гр. КНТ-128 НУ «Запорізька Політехніка»
[2]канд. техн. наук, доц. НУ «Запорізька Політехніка»

ІНФОРМАЦІЙНА БЕЗПЕКА-2021. ОСНОВНІ ЗАГРОЗИ, ТРЕНДИ

З упевненістю можна сказати, що Інтернет заволодів усіма аспектами життєдіяльності людини. Існує безліч загроз кібербезпеки, основні з них: DDoS-атаки, шкідливе програмне забезпечення в десктоп і мобільних додатках, уразливості операційних систем, фішингові атаки, парольні атаки, атаки на пристрої Інтернету речей, витоки даних та ін. Пандемія COVID-19 привела до суттєвого зростання кіберзлочинності, примусивши співробітників працювати з дому, де вони більш уразливі. З'явилися нові загрози, нові методи і засоби захисту і, звичайно ж, серйозно змінилися тренди в розвитку галузі.

До актуальних загроз в інформаційній безпеці 2021 року можна віднести: атаки підприємств через віддалених співробітників, витік даних, атаки на 5G-мережі та атаки з використанням штучного інтелекту. Відбувається зростання діяльності шифрувальників та фішингу.

Ще до пандемії багато компаній переходили на віддалену роботу. Немає сумнівів в тому, що дистанційна робота має ряд переваг як для співробітників, так і для їх організацій, але це також проблема для кібербезпеки. Самий поширений варіант атаки – коли зловмисники отримують доступ до службового облікового запису жертви. Тепер це простіше, тому що віддалені співробітники залишилися без нагляду фахівців з інформаційної безпеки; деякі компанії досі не оновили політику безпеки; до того ж працівники часто підключаються до корпоративної мережі з особистих пристроїв. На цьому тлі прийоми соціальної інженерії і теж вірусне ПЗ працюють ефективніше, ніж будь-коли.

Говорячи про витік даних, важливо зауважити, що зловмисники отримують доступ до корпоративних мереж найчастіше шляхом крадіжки облікових даних співробітників. Бо вони використовують пристрої, що не оновлюються, не керуються і не захищені корпоративним IT-відділом.

5G – благодатне середовище для кіберзлочинців. Прогалини в захисті доведеться усувати по ходу справи, і на те, щоб забезпечити прийнятний рівень безпеки технології, піде, певно, не один рік. Стек технологій в 5G потенційно залишає злочинцям можливість здійснювати атаки на абонентів і мережу оператора. Такі атаки можуть бути виконані з мережі міжнародного роумінгу, мережі оператора або партнерських мереж, які надають доступ до послуг.

Крім цього, штучний інтелект тепер теж на озброєнні кіберзлочинців. Зловмисники все активніше використовують можливості ШІ і машинного навчання. Просунуті кіберзлочинці застосовують ШІ, щоб складати профіль жертв і знаходити уразливості в застарілих інформаційних системах. Також все більше з'являється доступних фреймворків ШІ, які можна використовувати за передплатою, тобто розробляти ефективні інструменти для атак без серйозних вкладень.

Майже третина атак була пов'язана з методами соціальної інженерії, де кіберзлочинці намагаються використовувати страх, пов'язаний з пандемією COVID-19, як частину своєї тактики. На жаль, фішингові та спуфінгові атаки, пов'язані з пандемією, стали серйозною проблемою в 2021 році. Атаки часто збігаються з великими подіями, такими як спалах нових випадків або оголошення про нові ліки або вакцини. Їх мета – змусити нічого непідозрюючих жертв натиснути шкідливе посилання, щоб передати конфіденційну інформацію.

Не менш важливим фактом є те, що в цьому році зросла кількість атак шифрувальників – шкідливого ПЗ, яке шифрує дані, блокує роботу і нерідко вимагає викуп. Також популярність здобув шантаж викраденими приватними даними. Це перетворилося на повноцінну індустрію: зловмисники навіть створили власні сайти і аукціони для продажу викраденої інформації. Таким чином загальна сума вимог про викуп в 2020 році досягла 1,4 мільярда доларів, а середня сума відшкодування збитку – 1,45 мільйона доларів. Вимагачі є третім за популярністю типом шкідливих програм, що використовуються при витоку даних.

Крім вищенаведених загроз інформаційної безпеки, зросли DDoS-атаки. DDoS-атака містить в собі перевантаження сервера великими обсягами трафіку, щоб вивести з ладу бізнес-сайт. Такі напади останнім часом стали досить частими. Наприклад, у 2020 році було зроблено 4,83 мільйони DDoS-атак та порушення роботи сервісу може коштувати бізнесу в середньому до 100 тисяч доларів кожної години.

Поки триває пандемія, ринок хмарних обчислень збільшився на 50%. Хоча хмарна інфраструктура дуже безпечна, клієнти несуть відповідальність за реалізацію функцій кібербезпеки та їх правильне налаштування. Неправильна конфігурація хмари – поширене джерело витоку даних, яке зростає в міру того, як все більше компаній впроваджують хмарні сервіси для підтримки віддалених співробітників. Таким чином було здійснено 7,5 мільйона атак на хмарні акаунти в 2021 році. З початку року кількість спроб зросла на 250% в порівнянні з 2020 роком.

Підводячи підсумки можна сказати, що пандемія COVID-19 вплинула на зростання кіберзлочинності і на весь IT-світ. Компанії почали працювати віддалено і працювати з хмарними платформами. Витонченість загроз зросла

через застосування нових технологій, таких як машинне навчання, штучний інтелект і впровадженого 5G. Все це говорить про те, що індустрія кібербезпеки як ніколи важлива.

УДК 004.056.55
Зайко Т.А.[1], Івахненко Б.О.[2]
[1]канд. техн. наук, доц. НУ «Запорізька Політехніка»
[2]студ. гр. КНТ-118 НУ «Запорізька Політехніка»

КВАНТОВА КРИПТОГРАФІЯ

Для вирішення завдань збереження даних нам необхідний найсучасніший метод криптографічного шифрування. Таким методом є квантова криптографія.

Технологія квантового розподілу криптографічних ключів вирішує одну з основних завдань криптографії – гарантоване на рівні фундаментальних законів природи розподіл ключів між віддаленими користувачами по відкритих каналах зв'язку

Вперше ідея захисту інформації за допомогою квантових об'єктів була запропонована Стівеном Візнер в 1970 році. Через десятиліття Чарльз Беннет і Жиль Брассар запропонували передавати особистий ключ з використанням квантових об'єктів, ними була запропонована схема BB84.

До початку чергового раунду генерації сеансового ключа передбачається, що у Аліси і Боба, як учасників протоколу, є:

– квантовий канал зв'язку;

– класичний канал зв'язку.

Протокол гарантує, що втручання зловмисника в протокол можна помітити аж до тих пір, поки зловмисник не зможе контролювати і на читання, і на запис всі канали спілкування відразу.

Протокол складається з трьох частин:

– передача і прийом фотона по квантовому каналу зв'язку від Аліси до Боба;

– передача Бобом інформації про використані анализаторах;

– передача Алісою інформації про збіг обраних аналізаторів і вихідних поляризаций.

Далі Беннет запропоновал для реєстрації змін в переданих за допомогою квантових перетворень даних використовувати наступний алгоритм:

– відправник і одержувач домовляються про довільній перестановці бітів в рядках, щоб зробити положення помилок випадковими;

– рядки діляться на блоки розміру k (k вибирається так, щоб ймовірність помилки в блоці була мала);

– для кожного блоку відправник і одержувач обчислюють і відкрито сповіщають один одного про отримані результати. Останній біт кожного блоку видаляється;

– для кожного блоку, де парність виявилася різною, одержувач і відправник виробляють ітераційний пошук і виправлення невірних бітів.

– щоб виключити кратні помилки, які можуть бути не помічені, операції попередніх пунктів повторюються для більшого значення k;

– якщо відмінностей немає, після m ітерацій одержувач і відправник отримують ідентичні рядки з ймовірністю помилки 2-m.

Схема реалізація односпрямованого каналу з квантовим шифруванням показана на рис. 1. Передавальна сторона знаходиться зліва, а приймаюча - справа. Осередки Покеля служать для імпульсної варіації поляризації потоку квантів передавачем і для аналізу імпульсів поляризації приймачем. Передавач може формувати одне з чотирьох станів поляризації (0, 45, 90 і 135 градусів). Власне передані дані надходять у вигляді керуючих сигналів на ці осередки. В якості носія даних може використовуватися оптичне волокно. В якості первинного джерела світла можна використовувати і лазер.

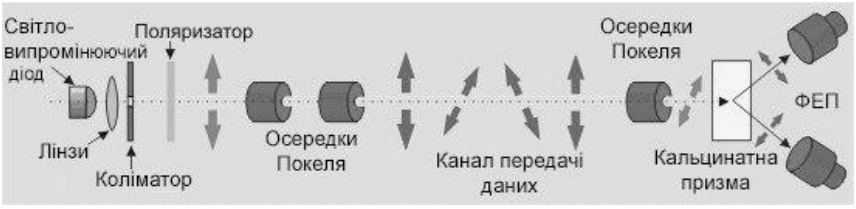


Рисунок 1 – Реалізація односпрямованого каналу з квантовим шифруванням

На приймаючій стороні після осередки Покеля ставиться каліцтва призма, яка розщеплює пучок на два фотодетектора (ФЕУ), що вимірюють дві ортогональні складові поляризації. При формуванні переданих імпульсів квантів доводиться вирішувати проблему їх інтенсивності. Якщо квантів в імпульсі 1000, є ймовірність того, що 100 квантів по шляху буде відведено зловмисником на свій приймач. Аналізуючи пізніше відкриті переговори проміжній стороною, він може отримати потрібну йому інформацію. В ідеалі число квантів в імпульсі має бути близько одного. Тут будь-яка спроба відводу частини квантів зловмисником призведе до істотного зростання числа помилок у приймаючої сторони. В цьому випадку прийняті дані повинні бути відкинуті і спроба передачі повторена. Але, роблячи канал більш стійким до перехоплення, ми в цьому випадку стикаємося з проблемою "темнового" шуму (видача сигналу в відсутності фотонів на вході) приймача (адже ми змушені підвищувати його чутливість). Для того щоб забезпечити

надійне транспортування даних логічного нуля і одиниці можуть відповідати певні послідовності станів, що допускають корекцію одинарних і навіть кратних помилок.

Подальшого поліпшення надійності криптосистеми можна досягти, використовуючи ефект EPR (Einstein-Podolsky-Rosen).

На даному етапі квантова криптографія тільки наближається до практичного рівня використання.

УДК 004.056.53
Зайко Т.А.[1], Колесникова М.В.[2]
[1]канд. техн. наук, доц. НУ «Запорізька Політехніка»
[2]студ. гр. КНТ-128 НУ «Запорізька Політехніка»

МІЖНАРОДНІ СТАНДАРТИ КІБЕРБЕЗПЕКИ

У сьогоденні, як ніколи швидко розвиваються всі сфери життя людини. Основними напрямками такої еволюції побуту є автоматизація та діджиталізація. Як результат – виникнення нової сучасної проблеми – кіберзлочинності. Уряд, бізнес, та пересічні громадяни потребують захисту пристроїв, програм, особистої та комерційної інформації від шкідливого втручання ззовні. З цього постає необхідність прийняття заходів, щодо захисту мереж, систем, пристроїв та даних, попередження та відбиття кібер-атак. Цим і займається кібербезпека.

Зважаючи на еволюцію кіберзагроз, що можуть катастрофічно вплинути на важливу інфраструктуру держави, спеціалісти з питань безпеки були змушені звернутися до стандартів заснованих на ризиках. Достовірність оцінки ризиків залежить від кількісного аналізу загроз, вразливостей і їх наслідків, що передбачає збір і вивчення величезних обсягів інформації. Цей процес є вартісним, трудомістким і тривалим. Через це, організації не займаються власною розробкою, а спираються у своїй стратегії та політиці безпеки на існуючі міжнародні стандарти кібербезпеки.

Стандарти кібербезпеки - це твердження, які описують, що має бути досягнуто з точки зору результатів безпеки, щоб виконати заявлені цілі безпеки підприємства. Зараз у світі існує безліч стандартів, якими користуються різні провідні країни, наступні з них є найбільш поширеними на міжнародному рівні:

Сімейство стандартів ISO 27000. Набір стандартів безпеки, випущених Міжнародною організацією зі стандартизації (ISO), які набули широкого поширення у всьому світі. Вони включають у себе документи: ISO/IEC 27001 – вимоги до систем менеджменту інформаційної безпеки (СМІБ); ISO/IEC 27000 – огляд та словник термінів, що стосуються СМІБ; ISO/IEC 27003 та

ISO/IEC 27004 – керівництво та вимірювання ефективності СМІБ; ISO/IEC 27006 – вимоги до органів, що здійснюють аудитів та сертифікацію, ISO/IEC 27007 – керівництво до проведення аудиту СМІБ.

Існують також інші стандарти ISO: ISO/IEC 27032 до: 2012 – інформаційні технології та методи безпеки, настанови щодо кібербезпеки; ISO/IEC 15408 - інформаційні технології та методи безпеки, критерії оцінки IT-безпеки; ISO 22301 – вимоги до соціальної безпеки та системи менеджменту безперервності бізнесу; ISO/IEC 27035 – інформаційні технології та методи безпеки, управління інцидентами інформаційної безпеки та ін.

NIST SP 800 розроблено для задоволення і підтримки вимог безпеки і конфіденційності інформації та інформаційних систем уряду США та Канади. Ці стандарти налічують 800 публікацій, що містять керівництва та довідкову інформацію на безліч різних тем, наприклад: SP 800-184 – посібник з відновлення подій кібербезпеки, SP 800-53 Rev. 5 контроль безпеки та конфіденційності для інформаційних систем та організацій, SP 800-40 – керування вразливостями, SP 800-81-2 посібник із розгортання системи безпечних доменних імен (DNS). Всі статті є у вільному доступі на сайті організації NIST.

Британський стандарт BS 7799 регламентує управління інформаційною безпекою організації незалежно від сфери її діяльності. На його основі було розроблено міжнародний стандарт ISO/IEC 17799 проте його початковим виданням і досі керуються у 27 країнах світу.

ISA/IEC 62443 стандарт з кібербезпеки, що розроблений як національний стандарт США, проте поширений і у країнах Європейського союзу. Визначає вимоги до захисту промислових автоматизованих систем керування.

General Data Protection Regulation (GDPR) - обов'язкові законодавчі норми, що базуються на конфіденційності для підприємств, які обробляють або контролюють приватні персональні дані, що належать громадянам ЄС.

PCI DSS – це стандарт безпеки даних платіжних карток. Він є обов’язковим для більшості підприємств, які збирають, обробляють та зберігають дані платіжних карток, таких як Visa чи Mastercard.

Деякі цих стандартів регулюються законодавчими органами (GDPR, PCI DSS), а деякі носять рекомендативний характер. Проте, якщо організація розгортає діяльність на міжнародному рівні, дотримання стандартів кібербезпеки є обов’язковою умовою співпраці багатьох провідних країн світу.

Отже, використання стандартів кібербезпеки є невід’ємною частиною політики безпеки підприємств критичної інфраструктури. Але незважаючи на різноманіття стандартів, необхідно вибирати найостанніші, ті, що

повністю задовольняють вимогам до безпеки і конфіденційності конкретної організації. Основними мірами підтримки захищеності є дотримання, регулярне вимірювання ефективності прийнятих стандартів та впровадження нових. Така стратегія призведе не лише до зменшення ризиків зовнішнього втручання, а й до збільшення довіри зацікавлених сторін, а отже й принесе нові інвестиції.

УДК 681.51
Яремченко Є.В.[1], Зайко Т.А.[2]
[1]студ. гр. КНТ-120м НУ «Запорізька політехніка»
[2]канд. техн. наук, доц. НУ «Запорізька політехніка»

ВИКОРИСТАННЯ SCADA-СИСТЕМИ TRACE MODE 6 ЯК ІНСТРУМЕНТУ АВТОМАТИЗАЦІЇ УПРАВЛІННЯ ІНЖЕНЕРНОЮ ІНФРАСТРУКТУРОЮ БУДІВЕЛЬ

Відомо, що витрати на проєктування і будівництво становлять не більше 20-40% вартості володіння будівлею протягом її життєвого циклу. Відповідно саме на експлуатацію будівлі припадає основна частина витрат власників. Причому, чим сучасніше будівля, і чим більше вона насичена інженерними системами, тим ця цифра вище. Витрати на експлуатацію сучасного будинку протягом життєвого циклу можуть досягати 80% від загальної вартості володіння! У той же час відомо, що диспетчеризація і автоматичне керування дозволяє скоротити витрату ресурсів до 20%, і продовжити термін служби обладнання на 10-15 років. Це означає, що впровадження системи автоматизації будівлі може заощадити гроші, які можна порівняти з вартістю нового будівництва [1].

Розробка автоматизованих систем управління будівлею здійснюється за допомогою спеціалізованого інструментального програмного забезпечення з використанням технологій промислової автоматизації (АСУ ТП). Одним з найбільш популярних інструментальних пакетів такого роду є SCADA і Softlogic-система TRACE MODE 6 виробництва компанії AdAstrA. Система TRACE MODE добре відома у багатьох галузях промисловості. На її основі створені системи управління, диспетчеризації та обліку ресурсів для тисяч підприємств. Все більшої популярності TRACE MODE отримує і в автоматизації будівель [1].

Архітектуру даної SCADA-системи можна умовно розділити на два керуючих рівня. На нижньому рівні розташовуються датчики, виконавчі механізми, контролери і пристрої зв'язку з об'єктом. Для інформаційного зв'язку контролерів та виконавчих пристроїв використовується мережева технологія Profibus. До верхнього рівня управління можна віднести

високопродуктивний сервер з вбудованою базою даних, автоматизоване робоче місце оператора-технолога, а також різні клієнтські і Web-додатки. Мережева взаємодія між рівнями також може здійснюватися за допомогою технології Ethernet. Основа TRACE MODE – це потужний сервер і база даних, яка працює в реальному часі для своєчасного внесення змін до технологічного процесу. Всі програмні інтерфейси максимально стандартизовані для зручного зв'язку з різними внутрішніми та зовнішніми компонентами, а також для отримання можливості розширювати і доповнювати дану систему. Обмін інформацією між сервером TRACE MODE, клієнтами та незалежними додатками проводиться через інтерфейси OPC, DDE і DCOM [2].

TRACE MODE дозволяє розробити автоматизовану систему диспетчерського управління (АСДУ) інженерним обладнанням будівлі в графічних редакторах і представити інформацію на анімованих графічних мнемосхемах, не написавши при цьому жодного рядку коду [1].

В системі передбачена велика бібліотека стандартних технологічних об'єктів: різних ємностей, теплообмінників, регуляторів, приладів і т.д. Для більшої зручності і точності відображення технологічного процесу є можливість створення власних операторських форм ActiveX [2].

Крім цього редактор дозволяє створювати багатошарові зображення і динамічно управляти шарами. Це дуже зручно для автоматизації будівель, так як на один поверховий план приміщення можна вивести шар з електричною мережею, на інший – з охоронною сигналізацією, на третій з кондиціонуванням і вентиляцією і т.д. [1].

Інформаційна база проєкту структурується за категоріями, звичними для інженерів будинків. Можна виділити корпуси, поверхи, зали будівлі. І одночасно розділити проєкт по автоматизуємих інженерних об'єктах: ліфти, котельні, насоси, кондиціонери, генератори, фанкойли і т.д. [1].

Величезна перевага системи TRACE MODE – наявність великої бібліотеки безкоштовних драйверів практично для будь-яких лічильників, контролерів, кондиціонерів і т.д. [1].

Отже, використання TRACE MODE як інструменту розробки АСДУ інженерними системами будівлі дозволить швидко вирішити завдання автоматизації та обліку ресурсів будівлі.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Кушнарева, Е. А. SCADA TRACE MODE как инструмент автоматизации управления инженерной инфраструктурой зданий [Текст] / Е. А. Кушнарева // Энергия интеллекта: Оборудование. – 2008. – № 23. – С. 13.

2. Благовещенская, М. М. Внедрение SCADA-системы Trace Mode в производственные процессы на примере бестарного хранения муки [Текст] /

М. М. Благовещенская, Е. Б. Карелина, Д. Ю. Клехо // Вестник Воронежского государственного университета инженерных технологий. – 2015. – № 2. – С. 82–85.

УДК 004.056.5
Зайко Т.А.[1], Мінібаєв Д.О.[2]
[1] канд. техн. наук, доц. НУ «Запорізька Політехніка»
[2]студ. гр. КНТ-118 НУ «Запорізька Політехніка»

КІБЕРПРОСТІР УКРАЇНИ 2021

Кібепростір у 21 сторіччі - один з найцікавіших середовищ у нашому житті, де люди проводять досить багато часу, але як простих людей, так й великі компанії можуть страдати від того, який він є у їх країні[1].

Згідно українським законодавством, кіберпростір – середовище, яке надає можливості для здійснення комунікації та реалізації суспільних відносин, утворене в результаті функціонування сумісних комунікаційних систем.

В ході звіту про глобальні загрози останнього зібрання у Давосі в кінці 2020 року, кібератаки увійшли до десятки найнебезпечніших загроз поруч із глобальним потеплінням, зброєю масового ураження й епідемією COVID-19[2].

Серед найбільших кібератак відносять ті, що спрямовуються проти держ.органів, компаній, або ж ті, що задають збитків в мільйони доларів. Згідно аналізу фахівців з Specops Software, Україна посідає сьоме місце серед країн, що найбільш постраждали.

Тому, задля покращення інформаційного середовища в Україні діє кіберполітика. Кібербезпековими питаннями в Україні займаються як державні органи, так і неурядові організації, наприклад, «Український кіберальянс», що протидіє російським кібератакам. Також в Україні існує Національний кординаційний центр кібербезпеки як перший робочий орган РНБО. Наразі, на кінець 2020 року була затверджена «Стратегія кібербезпеки України (2021-2025 роки)» [3], що говорить про повищену інформаційну безпеку в питанні COVID-19, поширенню міжнародного співробітництва у сфері кібербезпеки, а також формування нової моделі відносин у сфері кібербезпеки для оптимізації праці.

У чинній воєнній доктрині України йдеться про саме «…поглиблення кооперації та співробітництва з НАТО і ЄС у сфері розвідки щодо … боротьби с кіберзлочинністю», крім того зазначено, що підрозділи з кіберзахисту вже відповідають вимогам нормативно-правових актів України,

а також з урахуванням стандартів НАТО і ISO/IEC, що вже підтверджує розвиток нашої країни у здатності до свого захисту.

Серед речей, що є досить важливими саме для пересічних громадян, можна назвати додаток «Дія», що на початок 2021 року стає основною платформою взаємодії людини і держави. Людина може використовувати додаток для верифікації себе, запису на отримання вакцини, сплати штрафів, або просто для підтримання інфомаційної безпеки себе. Також серед законопроектів наразі є «Дія. City» [4] для позначення особого статусу IT-підприємств в Україні, а також запропонований ВРУ закон про розмитнення авто через додаток «Дія» [5].

Тобто, слід зазначити, що на початок 2021 року, український кіберпростір продовжує свою трансформацію й розвиток, покращуючи життя пересічним громадянам, за допомогою держ.додатку «Дія», а також розвиваючи кібербезпеку нашої країни за допомогою спеціальних органів.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Вікіпедія [Електрон. ресурс]. – Режим доступу : https://uk.wikipedia.org/wiki/%D0%9A%D1%96%D0%B1%D0%B5%D1%80%D0%BF%D1%80%D0%BE%D1%81%D1%82%D1%96%D1%80
2. Український кіберпростір [Електрон. ресурс]. – Режим доступу : https://adastra.org.ua/blog/ukrayinskij-kiberprostir-bezpekovi-zagrozi-vikliki-ta-perspektivi-rozvitku
3. РНБО Україна [Електрон. ресурс]. – Режим доступу : https://www.rnbo.gov.ua/files/2021/STRATEGIYA%20KYBERBEZPEKI/proekt%20strategii_kyberbezpeki_Ukr.pdf
4. Diia City [Електрон. ресурс]. – Режим доступу : https://city.diia.gov.ua/news/parlamentu-rekomendovano-priynyati-za-osnovu-novu-redaktsiyu-zakonoproektu-pro-diya-city
4. Українська правда [Електрон. ресурс]. – Режим доступу : https://www.pravda.com.ua/news/2021/03/20/7287283/

УДК 004.415.25
Львовкін В.М.[1], Руснак Н.А.[2]
[1]канд. техн. наук, доц. НУ «Запорізька політехніка»
[2]студ. гр. КНТ-217 НУ «Запорізька політехніка»

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ВІДСЛІДКОВУВАННЯ ФІНАНСОВИХ ВИТРАТ ЗА ГЕОЛОКАЦІЄЮ

Відслідковування фінансових витрат та надходжень є важливою складовою повсякденного життя сучасної людини – добре задокументований грошовий потік може впливати на велику кількість прийнятих рішень щодо фінансових можливостей та тенденцій. Документування та зберігання цих даних на папері є дуже часозатратним та неефективним методом, що не дозволяє без додаткових ручних підрахунків побачити загальну картину фінансових змін.

В свою чергу більшість існуючих програмних засобів автоматизації відслідковування фінансових витрат мають примітивну функціональність, наприклад, не використовуються переваги платформ на яких працюють (у випадку сучасних мобільних платформ – датчики геолокації та підключення до мережі Інтернет), або мають складний користувацький інтерфейс, що перешкоджає їх широкому використанню.

Для розробки програмного застосунку було обрано платформу Android через її велику розповсюдженність на ринку мобільних пристроїв. Розроблений програмний застосунок надає доступ зокрема до наступних функціональних можливостей:

– додавання грошових рахунків користувача;
– додавання записів про грошові операції;
– додавання геоінформації до записів про грошові операції;
– додавання категорій грошових операцій;
– відображення сумарних поточних фінансів за рахунками;
– відображення грошових операцій переліком з фільтрацією;
– відображення місць, де виконувались операції, на мапі;
– відображення статистики грошових операцій за обраний проміжок часу у вигляді графіків та кругових діаграм;
– синхронізація рахунків та грошових операцій користувача з Monobank.

Процес розробки було обрано вести за допомогою мови програмування Kotlin – Android-сумісної мови, яка є лаконічною, виразною та розроблена для забезпечення безпеки при роботі з типами та посиланнями об’єктів. Kotlin також має сумісність з мовою програмування Java, що дозволяє використовувати бібліотеки та класи Java без їх зміни [1].

При розробці було використано наступні технології:

– робота з базою даних SQLite – Room;

– Maps SDK for Android – інструменти роботи з мапою Google Maps;

– Places SDK for Android – інформація про місця на Google Maps;

– HTTP-клієнт – Retrofit;

– ін'єкція залежностей – Hilt;

– бібліотеки AndroidX – допоміжні архітектурні інструменти архітектурного шаблону MVVM, сумісність з застарілими версіями Android, підтримка пагінації;

– корутини Kotlin для забезпечення асинхронності при роботі застосунку;

– обробка Json – Moshi;

– відображення графіків та діаграм – MPAndroidChart.

На рисунку 1 зображений головний екран застосунку.

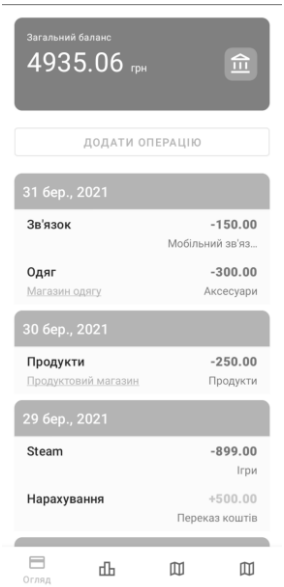


Рисунок 1 – Головний екран застосунку

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Kotlin on Android FAQ | Android Developers [Electronic resource]. – Access mode : https://developer.android.com/kotlin/faq.

УДК 004.942
Льовкін В.М.[1], Лизя Є.С.[2]
[1] канд. техн. наук, доц. НУ «Запорізька політехніка»
[2] студ. гр. КНТ-217 НУ «Запорізька політехніка»

МЕТОДИ ПРОГНОЗУВАННЯ В МЕДИЧНІЙ ДІАГНОСТИЦІ

Актуальність проблеми медичної діагности загалом і завдань прогнозування, які виникають під час її реалізації, в сучасному світі не викликає запитань. Серед завдань, які відносяться до прогнозування відносно цієї проблеми, можна виділити ті, що пов'язані з прогнозуванням плину епідеміологічної ситуації, що є стратегічним завданням, пов'язаним з діагностуванням, та безпосередньо індивідуальне прогнозування стану пацієнта відносно захворювання певною інфекційною хворобою. У даній роботі виділено найбільш значущі методи розв'язання описаної проблеми і окреслити область їх застосування.

Класичний регресійний аналіз – найбільш базовий метод прогнозування захворюваності. Завдання регресії полягає в формуванні функціональної залежності між захворюваністю людей і факторами, які її зумовлюють. Прикладом використання даного підходу є оцінка частоти виникнення менінгококової інфекції, що виконується на основі рівняння лінійної регресії загальної сукупності, яке визначає, на скільки зросте частота інвазійних менінгококових захворювань при збільшенні кількості носіїв менінгококової інфекції на заданий відсоток [1].

Динамічні байєсовські мережі – підхід до моделювання часових рядів, що дозволяє оцінити ймовірність настання деякої події при спостереженні послідовності явищ, на даний момент застосовуються до задачі прогнозування захворюваності переважно в формі прихованих марківських моделей [2]. Прикладом застосування є оцінка ризику смерті та тяжкого випадку етеровірусної хвороби протягом місяця в різних експериментальних сценаріях, які були проведені у містах із високим рівнем захворюваності в піковий період, в непіковий період, з низьким рівнем захворюваності в піковий період та з низьким рівнем захворюваності в непіковий період, у кожному експериментальному сценарії 70% даних спостереження було використано для побудови моделі для передчасного попередження спалаху важкого випадку та смерті в наступному місяці [3].

Одним із методів машинного навчання є дерева рішень, які також можуть використовуватися для розв'язання проблеми, що розглядається. У роботі [4] запропоновано модель, яка використовувалася для діагностування COVID-19, використовуючи вісім бінарних ознак: стать, вікова категорія (більше 60 років або ні), чи було встановлено контакт із зараженою особою та появу п'яти початкових клінічних симптомів (кашель, підвищена

температура, біль у горлі, задишка, головний біль). Модель була створена на основі градієнтного бустинга з навчанням на основі дерев рішень. Експериментальне дослідження проводилось на основі 51 831 перевірених осіб, з яких було виділено тих, у кого захворювання не було виявлено, а далі для них виконувалося тестування через тиждень, яке мало підтвердити або спростувати результати прогнозування наявності інфекції.

Важливими в цій галузі є моделі на основі штучних нейронних мереж, що добре підходять для вирішення задач, аналітичне дослідження яких важко виконати. Такі моделі застосовуються як для прогнозування поширення захворювань, так і для прогнозування стану пацієнта. Зокрема в роботі [5] було запропонованого використовувати глибоку згорткову нейронну мережу для визначення COVID-19 на основі рентгенівських знімків грудної клітини. Прогнозування виконувалось для виявлення випадків, коли COVID-19 виявлено, не виявлено або не виявлено ніякої інфекції.

На даний момент слід констатувати, що універсального способу прогнозування інфекційної захворюваності не існує – оптимальний підхід слід вибирати, порівнюючи результати, отримані за допомогою різних технік на основі емпіричних даних. Для того, щоб сформувати раціональний прогноз з множини конкуруючих методів, доцільно скористатися методами теорії прийняття рішень.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Mokhort, H. Multiple Linear Regression Model of Meningococcal Disease in Ukraine [Electronic resources] : 1992–2015 / H. Mokhort. Computational and Mathematical Methods in Medicine. – 2020. – Access mode : https://www.hindawi.com/journals/cmmm/2020/5105120/.

2. Динамические байесовские сети [Электронный ресурс]. – Режим доступа : http://www.rriai.org.ru/dinamicheskie-bayesovskie-seti.html.

3. Using a Bayesian belief network model for early warning of death and severe risk of HFMD in Hunan province, China [Text] / Y. Liao, B. Xu, X. Liu et al. // Stochastic Environmental Research and Risk Assessment. – 2018. – № 32. – Pp. 1531-1544.

4. Machine learning-based prediction of COVID-19 diagnosis based on symptoms [Electronic resources] / Y. Zoabi, S. Deri-Rozov, N. Shomron // Digital Medicine. – 2021. – № 4. – Access mode : https://www.nature.com/articles/s41746-020-00372-6.

5. Wang, L. COVID-Net [Electronic resources] : a tailored deep convolutional neural network design for detection of COVID-19 cases from chest X-ray images / L. Wang, Z. Qiu Lin, A. Wong // Scientific Reports. – 2020. – № 10. – Access mode : https://doi.org/10.1038/s41598-020-76550-z.

УДК 004.031.42
Льовкін В.М.[1], Камєнський Д.В.[2]
[1]канд. техн. наук, доц. НУ «Запорізька політехніка»
[2]студ. гр. КНТ-127 НУ «Запорізька політехніка»

ПРОЄКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ СПІЛЬНОГО ВІДТВОРЕННЯ МЕДІАРЕСУРСІВ

Все більше аспектів соціально життя переходить в онлайн формат: планування та проведення зустрічей, побудова та контроль власної справи та, що не менш важливо, організація дозвілля. Враховуючи запропоновану велику кількість опцій щодо останнього, не усі користувацькі потреби задоволено.

Серед таких незайнятих ніш можна виокремити спільне відтворення відео та аудіо контенту, для чого ще не було представлено застосунку, який в повній мірі вирішує проблеми користувача та реалізує потенціал ідеї. Наявні на ринку пропозиції або першочергово спрямовані на вирішення іншої специфічної задачі і тому реалізують спільне відтворення медіаресурсів лише побічним чином, або виконують поставлену задачу лише частково.

Однією з центральних задач при розробці застосунку для відтворення медіаресурсів є інтеграція інформаційних джерел, з яких система буде відтворювати контент. Зважаючи на серйозну інфраструктурну логіку для гнучкого механізму інтеграції, насиченість предметної області та потенціальне розширення функціональності системи, з'являється вимога розділення відповідальності між частинами програми за видами логіки. Тому було вирішено будувати систему на основі багаторівневої архітектури із наступними шарами:

– шар представлення, який надає інтерфейс для взаємодії с програмою;

– шар прикладної логіки, який описує сценарії використання системи;

– шар предметної області, в якому сконцентровані поняття і правила предметної області;

– шар інфраструктури, в якому виконується взаємодія або використання зовнішніх залежностей.

Такий варіант розшарування системи чітко формує межі та відповідальність кожного із рівнів, завдяки чому отримано ізольовані шари предметної області та інфраструктури, що значно спрощує підтримку та розвиток системи у перспективі.

Через насиченість предметної області та наявність підобластей з пересічними поняттями рівень домену було спроєктовано за допомогою основних принципів предметно-орієнтовного проєктування (Domain Driven Design [1]). Цей підхід допомагає контролювати складність предметної

області завдяки використанню спільної мови із бізнесом (Ubiquitous Language [1]), розділенню відповідного шару на контексти (Bounded Contexts [1]) та шаблонам їх взаємодії.

Предметну область розроблюваної системи було розділено наступним чином:

– контекст авторизації;

– контекст пошуку контенту;

– контекст спільного відтворення контенту;

– контекст інформаційного джерела (є представленням сторонніх систем, які надають медіаресурси).

Щодо прикладного рівня, було вирішено будувати його за базовими принципами розділення відповідальності «команд» та «запитів» (CQRS [2]). Основними перевагами цього підходу у порівнянні із використанням служб (сервісів) прикладного рівня є:

– більш структурований прикладний рівень, завдяки виділенню сценаріїв використання системи у відповідні модулі;

– можливість розділення моделей «запису» та «читання», яка дозволяє винести бізнес-логіку до частини, що модифікує стан застосунку, що значно сприяє спрощенню отримання даних і позитивно впливає на продуктивність системи.

Для першої версії продукту повноцінне розділення моделей «запису» та «читання» є надмірним, тому його не було впроваджено, але завдяки слідуванню базовим принципам CQRS це буде просто реалізувати у разі зміни рішення у майбутньому.

Із урахуванням вимог до технічного аспекту архітектури та компетенції розробника було обрано наступні технології для реалізації системи:

– .NET Core 3.1;

– C# 8 для серверної частини;

– HTML/CSS та JavaScript для клієнтської частини;

– MS SQL Server у якості бази даних та EF Core для взаємодії із нею.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Еванс, Е. Предметно-ориентированное проектирование (DDD) [Текст] : структуризция сложных программных систем. : Пер. с англ. / Е. Еванс.— М. : Вильямс, 2011. – 448 с.

2. CQRS [Електронний ресурс]. – Режим доступу : https://martinfowler.com/bliki/CQRS.html.

УДК 004.93
Лизя Є.С.[1], Субботін С.О.[2]
[1]студ. гр. КНТ-217 НУ «Запорізька політехніка»
[2]зав. каф. НУ «Запорізька політехніка»

ПОРІВНЯННЯ НЕЙРОННОГО КОМП'ЮТЕРА З КОМП'ЮТЕРОМ ФОН НЕЙМАНОВСЬКОЇ АРХІТЕКТУРИ

Ця робота представляє собою порівняння нейронного комп'ютера з комп'ютером фон Неймановської архітектури, з ціллю доведення та обґрунтування, що нейронний комп'ютер може повністю замінити звичайний комп'ютер для повсякденного використання в найближчому майбутньому.

В таблиці 1 приведено порівняння нейронного комп'ютера з комп'ютером фон Неймана.

Таблиця 1 – Порівняння фон Неймановської та нейронної архітектури

	Комп'ютер з архітектурою фон Неймана	Нейронний комп'ютер
Процесор	Складний	Простий
	Високошвидкісний	Повільний
	Один або декілька	Велика кількість
Пам'ять	Відділена від процесора	Інтегрована в процесор
	Локалізована	Розподілена
	Адресація не по зберіганню	Адресація по зберіганню
Обчислення	Централізовані	Розподілені
	Послідовні	Паралельні
	Зберігаються програми	Самонавчання
Спеціалізація	Символьні та чисельні операції	Сприйняття та обробка образів
Середовище функціонування	Строго визначена	Погано визначена
	Строго визначена	Без обмежень
Надійність	Висока вразливість	Більша живучість

Недоліки комп'ютерів фон Неймановської архітектури – є їх нездатність до самонавчання і нестійкість до апаратних збоїв, коли при поломці одного компонента вся система виходить з ладу. До того ж, вони не мають ряд можливостей, які з легкістю виконуються мозком людини, таких як асоціативна пам'ять, здатність вирішувати завдання класифікації, розпізнавання об'єктів, кластеризація або прогнозування [1]. Іншою проблемою – є наближення до фізичної межі збільшення потужності

мікропроцесорів, а саме: збільшення кількості транзисторів не приведе до значного збільшення швидкості процесора.

Переваги нейрокомп'ютерів. Основне завдання нейрокомп'ютерів – обробка образів. При цьому у них, як і в мозку, відсутні загальні шини, немає поділу на активний процесор і пасивну пам'ять, а обчислення і навчання розподілені по всім елементарним процесорам – нейронам, які функціонують паралельно. За рахунок цього, нейрокомп'ютери дозволяють домогтися фантастичної продуктивності, яка може в мільйони разів перевищувати продуктивність традиційних комп'ютерів з послідовною архітектурою [2].

У 1996 році фірмою Accurate Automation Corp був розроблений експериментальний гіперзвуковий літак-розвідник LoFLYTE. Літак мав довжину всього 2,5 м і вагу 32 кг. LoFLYTE використовував нейронні мережі, що дозволяють автопілоту навчатися, копіюючи прийоми пілотування льотчика. Оскільки літак був призначений для польотів зі швидкістю 4-5 махів, то швидкість реакції пілота-людини могла бути недостатньою для адекватного відгуку на зміну режиму польоту. У цьому випадку на допомогу приходили нейронні мережі, які переймали досвід управління у льотчика і за рахунок високої швидкості обробки інформації дозволяли швидко знаходити вихід в аварійних і екстремальних ситуаціях [3].

Згідно вищезазначеному огляду функціональних та апаратних можливостей двох типів комп'ютерів, нейрокомп'ютери є більш продуктивними та менш вразливими по зрівнянню з сучасними комп'ютерами фон Неймана, а отже можуть повністю замінити їх в повсякденному використанні.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Архитектура фон Неймана: [Электронный ресурс]. – Режим доступа: http://ru.wikipedia.org/wiki/Архитектура_фон_Неймана

2. Нейрокомпьютер: [Электронный ресурс]. – Режим доступа: http://ru.wikipedia.org/wiki/Нейрокомпьютер

3. Accurate Automation LoFLYTE: [Electronic resource]. – Access mode: http://www.designation-systems.net/dusrm/app4/loflyte.html

УДК 004.514
Сердюк С.М.[1], Камінська Ж.К.[2], Бровченко І.Д.[3], Гвоздилко Д.О.[3], Рудь О.К.[3]
[1]канд. техн. наук, доц. НУ «Запорізька політехніка»
[2]асист. НУ «Запорізька політехніка»
[3]студ. гр. КНТ-139 НУ «Запорізька політехніка»

РОЗРОБКА ПРОБЛЕМНО – ОРІЄНТОВАНОГО ІНТЕРФЕЙСУ ПОШТОМАТА

Ми живемо в час розвитку інформаційних технологій і можемо використовувати безліч різних сервісних послуг. Один з популярних сервісів є поштові служби. Поштомати «Нова пошта», «Meest Express» користуються великою популярністю. Таким чином розробка інтерфейсу поштомата є актуальною.

Об'єкт дослідження – процес обслуговування клієнтів для поштових служб.

Предмет дослідження – методи розробки програмного інтерфейсу для систем обслуговування клієнтів у сфері поштових послуг.

Мета – підвищення якості обслуговування клієнтів поштомата.

Розробка взаємодії клієнта з поштоматом базується на рекомендаціях стандарту ISO 9241-210:2010 «Ергономіка взаємодії людина-система. частина 210. Людино-орієнтоване проектування інтерактивних систем». В даній роботі розглянуто реалізацію перших двох етапів розробки, наведених у стандарті ISO 9241-210:2010.

Перший етап розробки – розуміння та визначення умов використання, тобто аналіз характеристик користувачів і завдань, а також організаційного, технічного та фізичного середовищ. В результаті аналіза поштомата-аналога «Нова Пошта» було виявлено недолік, а саме відсутність функції відправлення посилки.

Другий етап – визначення вимог користувачів. Для його виконання, проведено системний аналіз системи «користувач – поштомат» у ході якого, для формалізації знань щодо апаратних і програмних компонентів системи, було розроблено компонентно-функціональну та функціонально-об'єктну структури.

Компонентно-функціональна структура являє собою семантичний граф у вигляді ієрархічного дерева, вершинами якого є імена функціональних одиниць (F_i) різного масштабу, а дуги відображають відношення R_{FiFj} - "для реалізації функціональної одиниці F_i (мети, задачі) необхідно виконати функціональну одиницю F_j (задачу, процедуру, функцію)". Корінь дерева – поняття "Процес функціонування системи N)". Перший рівень вершин дерева – назва цілей - T_{Ri} , для яких створена система N, шо досліджується. Другий рівень – назва додаткових цілей T_{DRij}, які треба вирішити для досягнення

цілей T_{Ri}. Третій рівень – назва задач T_{Sijk}, які треба вирішити для досягнення додаткових цілей. Четвертий рівень – назва процедур T_{PRijkl}, які необхідні для рішення задачі T_{sijk} [1].

Структури FOS являють собою набір графів, число яких дорівнює числу функцій, що досліджується. Кожний граф складається з вершин трьох типів: перший – ім'я функції F_i ; другий - узагальнене ім'я комплексу компонентів K_i, що реалізують виконання цієї функції; третій-імена елементів, що входять у цей комплекс: активний елемент (людина) K_{i1}, предмети праці K_{i2}, знаряддя праці K_{i3}. Дуги відображають відношення: між F_i і K_{ij} – R_k: «функція F_i виконується комплексом компонентів *K»;* між K_i і K_j – R_{ij}: «у комплексі K_i виконувати функції *j*: *j*=1 - активного елемента, *j*=2 - предмета праці, *j*=3 - знаряддя праці». Перераховані вище компоненти беруться із числа компонентів, що були отримані при компонентно-елементному аналізі [1].

Третій етап – розробка проєктних вирішень відповідних вимог користувачів. Розроблений програмний продукт, має функції: отримання; відправлення; код безпеки та оплата карткою (рис.1).

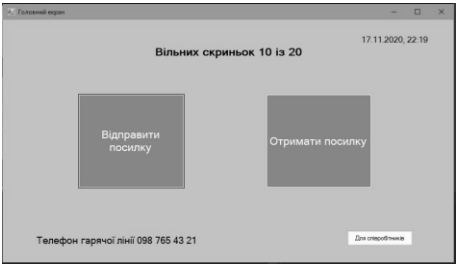


Рисунок 1 – Головне меню поштомата

Інтерфейс був розроблений за допомогою мови C# та середовища розробки Visual Studio.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Камінська Ж. К., Аналіз проблем автоматизації процесу ергономічного проектування інформаційних моделей технологічних процесів / Ж.К. Камінська // Вісник ЖДТУ. Серія "Технічні науки". – 2010. – № 1(52). – С. 103–108.

УДК 004.514
Сердюк С.М.[1], Камінська Ж.К.[2], Зуєв Б.В.[3]
[1]канд. техн. наук, доц. НУ «Запорізька політехніка»
[2]асист. НУ «Запорізька політехніка»
[3]студ. гр. КНТ-139 НУ «Запорізька політехніка»

РОЗРОБКА ЛЮДИНО-ОРІЄНТОВАНОГО ІНТЕРФЕЙСУ ТЕРМІНАЛУ «МОРОЗИВО»

Впровадження самообслуговування в ресторанах та кав'ярнях є світовим трендом останніх років. Термінали самообслуговування – це, перш за все, зручність гостя. Адже сучасні тенденції говорять про те, що все більшій кількості людей вже зараз зручніше вибирати замовлення через електронні девайси. Наприклад, в McDonald's в інших країнах, через посередництво терміналів оформляється до 85% замовлень [1]. Таким чином, розробка українського аналогу терміналу для замовлення морозива є актуальною задачею.

Метою даної роботи є підвищення якості взаємодії користувачів з терміналами у сфері громадського харчування.

Найбільшою проблемою терміналів для вибору й замовлення їжі є перевантаження їх інтерфейсу зайвими функціями та послугами. Це призводить до використання зайвих ресурсів і підвищення ціни на виробництво і обслуговування терміналів.

Було використано досвід аналогів найближчої Польщі (Aruba), Канади (Framec Easy) та Італії (Dolce Italia). Окрім послуг щодо замовлення морозива, пропонується також функція доступу до мережі Інтернет та фото, відео послуги. Проведено системний аналіз системи «клієнт – термінал», у ході якого було формалізовано знання щодо клієнтів та функціональних вимог до апаратних засобів у вигляді компонентно-функціональної та функціонально-об'єктної структур.

Інтерфейс української версії терміналу "Морозиво" створювався з урахуванням вимог мережі ринкових запитів та законодавства України. Інтерфейс є простим та інтуїтивно-зрозумілим для користувача, без зайвих надписів, елементів або зображень. В головному меню інтерфейсу терміналу (рис. 1) користувач створює власну комбінацію з рожку, смаків морозива та топінгу. Покупці мають інтерактивний доступ до меню та вибору типу замовлення (з собою або на місці). Термінал передає інформацію про замовлення співробітникам, які відразу ж починають його збірку. Отримання замовлення відбувається в спеціально обладнаній зоні видачі. Також термінали можуть зробити більш комфортним оформлення замовлення для людей з особливими потребами. Людям з проблемами слуху та мовлення

може бути простіше оформити покупку самостійно, не вдаючись до спілкування з касиром.

Аналіз досвіду використання аналогічних терміналів, показав, що більшість з них містить три варіанти продукції для вибору. В подальшому розвитку проєкту, можлива реалізація зміни різновиду смаків щодня, без повторів впродовж кожних трьох днів.

Також користувачу надається можливість переглянути вартість, оцінити бюджет й зібрати ще одне морозиво, за бажанням анулювати попередню комбінацію. Усі форми наповнені елементами управління, однакової форми та кольорової гами, приємної для перегляду.

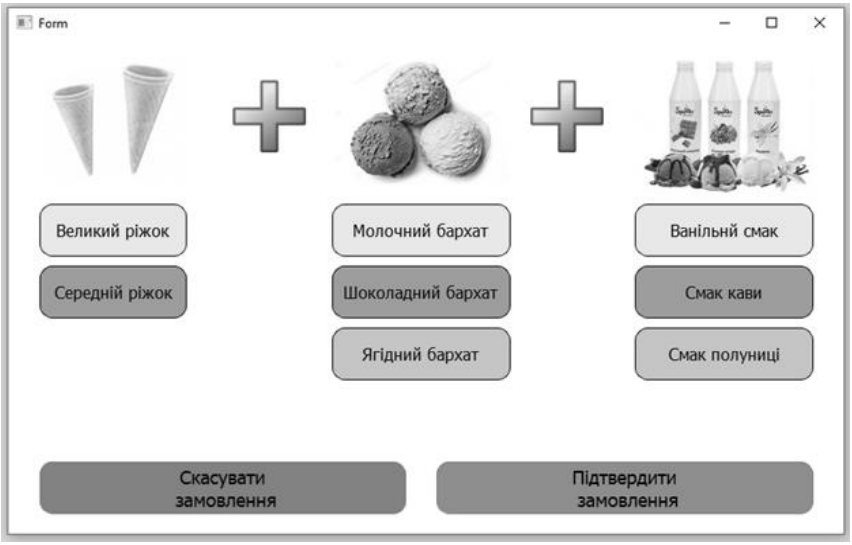


Рисунок 1 – Головне меню терміналу «Морозиво»

Інтерфейс було розроблено за допомогою мови C++ та середовища розробки Qt Creator.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Термінали самообслуговування в McDonald's [Електронний ресурс]. – Режим доступу : https://systemgroup.com.ua/uk/terminaly-samoobslugovuvannya-v-mcdonalds-svitovi-tehnologiyi-z-ukrayinskoyu-specyfikoyu

УДК 004.514
Сердюк С.М.[1], Камінська Ж.К.[2],Карнаух В.Д.[3], Яценко А.К.[3]
[1]канд. техн. наук, доц. НУ «Запорізька політехніка»
[2]асист. НУ «Запорізька політехніка»
[2]студ. гр. КНТ-129 НУ «Запорізька політехніка»

РОЗРОБКА ІНТЕРФЕЙСУ ЗАСТОСУНКУ ДЛЯ ВИКЛИКУ ТАКСІ

Об'єктом дослідження даної роботи є програмні засоби, що надають різноманітні сервісні послуги користувачам. Предметом дослідження є людинно-машинний інтерфейс програмного застосунку для виклику таксі. Мобільні застосунки виклику таксі полегшують якість життя та ефективність взаємодії між клієнтами та таксистами. Застосунок дозволяє таксисту безпосередньо взаємодіяти з клієнтом, а клієнтові – викликати машину за потрібною адресою всього за декілька кроків.

Метою роботи є розробка людино-машинного інтерфейсу мобільного застосунку для виклику таксі в місті Запоріжжя.

Було проаналізовано аналогічні інтерфейси таких застосунків, як Uber, Opti та Bolt. Інтерфейси вказаних застосунків дуже вдалі, але в них бракує деяких важливих функцій: швидкий виклик спеціальних служб та розширена інформація про водіїв, що дозволяє підвищити рівень безпеки як замовників, так і водіїв.

Для проєктування людино-орієнтованого інтерфейсу було проведено системний аналіз системи «клієнт – застосунок» та «водій – застосунок» у ході якого було розроблено компонентно-функціональні та функціонально-об’єктні структури, які представляють формальний опис знань про види користувачів (клієнт та водій таксі), вимоги користувачів до застосунку та вимоги до системи.

Зважаючи на вимоги користувачів, розроблений програмний продукт дозволяє користувачам використовувати наступні функції.

Для клієнта:

– замовлення таксі на вказану адресу зі вказаним часом, класом авто, кількістю пасажирів та наявністю великого вантажу;

– вибір водія зі списку вільних, представленого у вигляді мапи;

– можливість подивитися відгуки інших клієнтів про водія та випадки порушення правил дорожнього руху;

– залишення відгуку про роботу застосунку (текстовий відгук);

– залишення відгуку про водія (рейтинг від 1 до 5 та текстовий відгук);

– можливість виклику спеціальних служб.

Для водія:

– можливість подивитися відгуки про клієнта та підтвердити або скасувати замовлення;

– використання вбудованої мапи для навігації;

– залишення відгуку про клієнта (рейтинг від 1 до 5 та текстовий відгук);

– можливість виклику спеціальних служб.

На рис. 1 наведено фрагменти інтерфейсу програми.

Рисунок 1 – Фрагменти інтерфейсу програми

Інтерфейс характеризується зручністю і простотою. Можливий подальший ровиток застосунку "Служба виклику таксі" та його використання для міста Запоріжжя.

Інтерфейс був розроблений за допомогою мови C++ та середовища розробки Qt Creator.

На основі програм-аналогів Uber, Opti та Bolt та системного аналізу був розроблений ергономічний дизайн інтерфейсу застосунку виклику таксі.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Qt Documentation [Електронний ресурс] – Режим доступу : https://doc.qt.io/

УДК 004.514
Сердюк С.М.[1], Камінська Ж.К.[2], Чернявський К.Є.[3], Коростильов Д.Ю.[3]
[1]канд. техн. наук, доц. НУ «Запорізька політехніка»
[2]асист. НУ «Запорізька політехніка»
[3]студ. гр. КНТ-139 НУ «Запорізька політехніка»

РОЗРОБКА ЛЮДИНО – ОРІЄНТОВАНОГО ІНТЕРФЕЙСУ АВТОМОБІЛЬНОЇ ЕЛЕКТРОЗАПРАВКИ

В теперішній час в Україні активно розвивається ринок електрокарів. Станом на 2019 рік, в Україні нараховувалось близько 23000 електрокарів. Ще декілька років тому електрокари не були популярними, тому розвиток та доопрацювання електрозаправок відбувається кожен день. До переваг електрозаправок можна віднести: повний контроль постачань та витрат, зменшення паперової роботи, самообслуговування, менша потреба у персоналу.

Проте, головна проблема розумної техніки - непотрібні функції, котрі потребують використання зайвих ресурсів, що позначається на ціні та комплектації.

Об’єктом дослідження у роботі є процес обслуговування акумуляторних батарей електроавтомобілів.

Предметом дослідження – моделі, методи і засоби розробки автомобільної електрозаправки, системного меню наладчика.

Мета роботи – підвищення ефективності людино-машинної системи «користувач – електрозаправка», шляхом розроблення людино–орієнтованого інтерфейсу користувача, та системного меню.

У якості аналога було проаналізовано електрозаправку AutoEnterprise[1], яка має сенсорну панель за допомогою якої можна вибирати мову, теми та потужність зарядки.

Для проєктування інтерфейсу була проаналізована інформація щодо об’єкта, що проєктується. Проведено системний аналіз системи «користувач – електрозаправка», було розроблено компонентно-функціональну та функціонально-об’єктну структури. Результатом є вимоги та функції до елктрозаправки, що розробляється.

У розробленому інтерфейсі були реалізовані такі основні функції:

– функція вибору мови користування, яка надає можливість користувачу вибрати зручнішу для себе мову (було використано українську та англійську мови) (рис. 1);

– функція вибору потужності зарядки, яка дозволяє заряджати акумуляторну батарею (рис. 1);

– функція отримання статистики у системному меню наладчика, котра дозволяє системному наладчику отримати статистику про: наявність електромережі, тариф, кількість паперу для друку чеку (рис. 2);

– функція зміни та установки тарифу у системному меню наладчика (грн/кВт) (рис. 2);

– функція перегляду помилок у системному меню наладчика, яка дозволяє наладчику дивитися поточні помилки у системі електрозаправки (рис.2).

Рисунок 1 – Головна сторінка електрозаправки, вибір потужності

Рисунок 2 – Системне меню наладчика

Інтерфейс електрозаправки був розроблений за допомогою мови C# та середовища розробки Visual Studio.

На основі аналізу аналогу AutoEnterprise та структурного аналізу було розроблений ергономічний, людино-орієнтований інтерфейс автомобільної електрозаправки.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. AutoEnterprise [Електронний ресурс]. – Режим доступу : https://www.autoenterprise.com.ua/contacts/o_nas/

УДК 004.514
Сердюк С.М.[1], Камінська Ж.К.[2], Горобець В.І.[3]
[1]канд. техн. наук, доц. НУ «Запорізька політехніка»
[2]асист. НУ «Запорізька політехніка»
[3]студ. гр. КНТ-119 НУ «Запорізька політехніка»

АНАЛІЗ ЛЮДИНО-ОРІЄНТОВАНОГО ІНТЕРФЕЙСУ ТЕКСТОВОГО РЕДАКТОРА

Кожен користувач комп'ютера зустрічається з необхідністю підготовки різних документів, листів, статей тощо. Для роботи з текстами будь-якої складності існують спеціальні програми – текстові редактори.

Метою даної роботи є розробка та аналіз людино-машинного інтерфейсу текстового редактора. Для його проєктування була зібрана та проаналізована інформація щодо об'єкта, що проєктується. Згідно зі стандартом ISO 9241-210: 2010 «Ергономіка взаємодії людина-система. Частина 210. Людино-орієнтоване проектування інтерактивних систем» були проведені такі етапи: планування процесу проєктування; розуміння і визначення умов використання; визначення вимог користувачів; розробка проектних рішень; оцінка відповідності проекту. З метою реалізації людино-орієнтованого підходу до проєктування проведено системний аналіз системи «користувач - текстовий редактор - середовище», у ході якого було розроблено компонентно-функціональну та функціонально-об'єктну структури. Результат – створення початкового макету дизайну та програмного функціоналу відповідно до вимог цільової аудиторії.

Прототипом текстового редактора що розроблявся, можна назвати Notepad++. Інтерфейс даного редактора інформативний, має великий функціонал, але нагромаджено багато функцій, які важкі в розумінні непідготовленим користувачам.

На етапі розробки варіанту графічного інтерфейсу текстового редактора було застосовано основи дизайну форм і елементів керування. Перевірка моделі, на якість функціонування проведена за наступними показниками [1]: ймовірність безпомилкового виконання; математичне очікування часу виконання; дисперсія часу виконання.

Була створена функціонально-семантична мережа (ФСМ) для текстового редактора (рис.1), яка складається з типових функціональних одиниць (ТФО). Поступово йде "згортання" початкової мережі, а саме, проводиться аналіз ФСМ на предмет виявлення в ній типових функціональних структур (ТФС), що представляють собою деякі типові сукупності ТФО, які повторюються. Структура ТФС і вміст ТФО, що до неї входять, визначають надійністні та часові характеристики цієї системи. Для усіх ТФС функціональної мережі розраховуються відповідні показники

якості. Етапи повторюються доки ФСМ не буде приведена до однієї узагальненої ТФО Pe9 (рис.1), показники якої і будуть представляти собою узагальнену характеристику показників ефективності, якості та надійності процесу функціонування систем «людина-техніка-середовище».

З отриманих результатів було зроблено висновок, що показники процесу взаємодії користувача з текстовим редактором, згідно розробленої ФСМ, вкладаються в директивні показники якості (директивний час виконання даного процесу T_0 = 35,3 с та дисперсія часу виконання - 4,62 $с^2$), а саме – ймовірність безпомилкового виконання – 94,5%, математичне очікування часу виконання – 29,96 с.

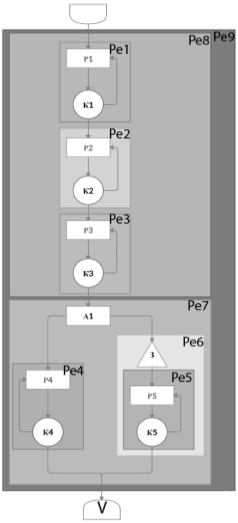


Рисунок 1 – Функціонально-семантична мережа

Інтерфейс: C++/фреймворку Qt та середовища розробки Qt Creator.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Інформаційно-керуючі людино-машинні системи: Дослідження, проектування, випробування: Довідник/ Адаменко А.Н., Ашеров А.Т., Лавров Е.А. і ін.. під загальн. ред. Губинського А.І. і Евграфова Е.Г. – М., Машинобудування, 1993. – 528с.

УДК 004.9
Степаненко О.О.[1], Федорченко Є. М.[2], Грабчук А.А.[3]
[1]канд. техн. наук, доц. НУ«Запорізька політехніка»
[2]старш. викл. НУ«Запорізька політехніка»
[3]студ. гр. КНТ-226, НУ«Запорізька політехніка»

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СПІЛЬНОГО ФІНАНСУВАННЯ ПРОЄКТІВ

У сучасному світі часто трапляється, що для реалізації ідей немає коштів: з цієї причини сотні стартапів завершують своє існування, винаходи не реалізуються, не знімаються фільми, однак, якщо ідея дійсно цікава, то в світі знайдеться багато людей, готових підтримати її і вкласти гроші в її реалізацію, саме для цього існують краудфандінгові майданчики. Вони дозволяють розповсюдити інформацію про проєкт серед людей, які можуть інвестувати власні кошти в реалізацію цікавих для них ідей [1].

Розвиток суб'єктів економічної діяльності залежить від багатьох чинників, де важливе місце займає залучення фінансових ресурсів для розвитку інноваційних сфер економіки, внаслідок розвитку інформаційних технологій створюються нові форми фінансування, які базуються на громадській участі: потенційними інвесторами можуть бути пересічні громадяни, проте через складну економічну ситуацію, брак коштів та ризикованість, вони поки що утримуються від інвестування. Важливе при цьому розуміння того, що самофінансування є актуальним способом пошуку ресурсів для власного розвитку компанії, це зумовлює розвиток краудфандингу, який на сьогодні потребує активізації свого розвитку в нашій країні. Одним зі шляхів розвитку даного напрямку є створення нових платформ, на яких можлива комунікація авторів ідей з потенційними інвесторами. Тому розроблений застосунок присвячений актуальній проблемі розроблення програмного забезпечення для залучення коштів на фінансування проєктів для поліпшення зручності взаємодії авторів проєктів зі спонсорами [2].

Перед розробкою програмного забезпечення було проаналізовано принципи роботи, моделі організації залучення коштів на фінансування проєктів і функціональність програмних аналогів (Kickstarter, Спільнокошт) для того, щоб раціонально виділити функціональні вимоги до програми з представленням діаграми прецедентів [3].

Вибір мови програмування випав на Java для реалізації, а для системи керування базами даних було обрано MySQL. Також, для реалізації використано вебінфраструктуру для швидкої розробки – мікро-фреймворк Spark [4].

Розроблене програмне забезпечення є кросплатформним і представляє собою клієнт-серверну систему. Рішення, за необхідності, може бути перетворене на вебзастосунок. Програма дозволяє взаємодіяти авторам проєктів з потенційними спонсорами. Програма надає можливість публікації власного проєкту, перегляду проєктів, перегляду найкращих проєктів, перегляду нових проєктів, перегляду успішних проєктів, перегляду невдалих проєктів, коментування, можливість вкладати гроші в проєкт, поповнювати рахунок, виконувати реєстрацію та авторизацію акаунту. Програма має актуальні за функціональними можливостями характеристики та будову. Вона може працювати на персональному комп'ютері або ноутбуці зі встановленою операційною системою Windows 10, Mac OS X, Linux, або Solaris. Додатково повинно бути встановлене таке програмне забезпечення, як система керування базами даних MySQL, а також комплект Java SE Development Kit 13. Інтерфейс розробленого програмного забезпечення є інтуїтивно зрозумілий та легкий у використанні. Наприклад, головна сторінка програми містить такі елементи: основне меню, відображене в лівій частині вікна, та меню керування відображенням проєктів, розташоване правіше, посередині вікна в верхній частині.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Попович, Д.В. Краудфандинг як новітній спосіб фінансування в Україні [Текст] / Д.В. Попович, Н.В. Назар, Н.В. Савчин // Young Scientist. – 2018. – № 10 (62). – С. 873-875.
2. Єлісєєва, Л.В. Краудфандинг в Україні [Текст]: проблеми та перспективи в контексті зарубіжного досвіду / Л.В. Єлісєєва // Науковий вісник Міжнародного гуманітарного університету. Серія : Економіка і менеджмент. – 2017. – Вип. 23(1). – С. 8-11.
3. Васильєв, О. В. Краудфандинг – сучасний інструмент фінансування розвитку інтелектуального капіталу [Текст] / О. В. Васильєв, С. С. Матющенко // Інвестиції : практика та досвід. – 2016. – № 18. – С. 36-40.
4. Салтыков, Б.Г. Краудфандинг как инструмент венчурного инвестирования в России и за рубежом [Текст] / Б.Г. Салтыков, М.Н. Гордеев // Инновации. – 2016. – № 4 (210). – С. 40-46.

УДК 004.9
Степаненко О.О.[1], Федорченко Є. М.[2], Ніколаєвський Д.О.[3]
[1]канд. техн. наук, доц. НУ«Запорізька політехніка»
[2]старш. викл. НУ«Запорізька політехніка»
[3] студ. гр. КНТ-216 НУ«Запорізька політехніка»

ПРОГРАМНА РЕАЛІЗАЦІЯ ОНЛАЙН ГІДУ ПО ТУРИСТИЧНИМ ПАМ'ЯТКАМ МІСТ УКРАЇНИ ДЛЯ ОС ANDROID

На сьогодення, складно уявити щось інше у руках людей, аніж смартфон. Отже, одна й таж сама модель одного девайса у руках різних людей повинна бути однакова і працювати так само, але насправді вони дуже сильно різняться між собою. Якщо його власник любить слухати музику, то телефон містить десятки гігабайтів різноманітних аудіо та відео файлів з новими додатками для їх прослуховування. Коли хазяїн пристрою любить подорожувати, то в ньому обов'язково є фотографії, туристичні гіди та мапи. Телефон є відображенням внутрішнього світу людини. Логічно припустити, що застосунки, котрі більш за всіх будуть використовувати – це мобільні застосунки. За статистикою, українці в середньому за день проводять більш ніж 5 годин свого часу в смартфоні. Але що можливо придумати на сьогодні, якщо велика кількість ідей вже реалізована? Така проблема, дійсно, має своє місце і потребує її вирішення [1]. Треба вивчити потреби користувачів, подивитись на готові варіанти, проаналізувати їх, зрозуміти як це все можливо поліпшити чи зробити по новому і внести щось цікаве, яке змусить зацікавитися потенційного користувача. Або можна зробити простий, але робочий аналог вже існуючих застосунків [2].

У звичайних людей, коли вони виїжджають кудись, завжди виникають питання:

– що можливо подивитись там;

– як туди дістатись;

– яка історія цього монументу чи пам'ятки;

– як він взагалі виглядає.

Без відповіді на ці запитання складно отримувати задоволення від поїздки, що є життєво-необхідною складовою. Відповіді на них можливо знайти у туристичному гіді для внутрішніх і зовнішніх туристів. Також, застосунок змалює чіткий план, який допоможе контролювати свій час під час мандрування [3].

В наш час сучасних технологій не треба шукати та брати з собою великі гіди та атласи, щоб спланувати свій маршрут. Все що треба – це знайти та встановити потрібний для вас застосунок, який буде містити в собі всю необхідну інформацію. Не дивлячи на те, що мандрувати люди могли і

можуть з давніх часів, більшість з них помилково витрачають свій обмеженій час на непотрібні речі. Проте, оптимізувати свій час насправді не є великою проблемою, це всього лиш потребує деякої мислячої роботи, котра у фіналі може принести безліч найкращих емоцій. Розроблений мобільний застосунок призначений для зовнішніх та внутрішніх туристів України. Застосунок надає користувачу вибір міста із запропонованих, потім за допомогою мапи або списку користувач знайомиться із існуючими пам'ятками, парками. Після цього, турист обирає один із них та вивчає надану інформацію, обрамлену фотографіями [4].

Також, застосунок має можливість визначити місцерозташування користувача. Завдяки цьому можна зорієнтуватися у просторі і знайти необхідний маршрут. Найголовніша функція застосунку – це перегляд повного списку існуючих пам'яток. Окрім цього, застосунок буде вимагати постійно увімкненого інтернету та GPS для визначення місцезнаходження. Для того, щоб користуватися застосунком, необхідно мати пристрій на базі Android версії 6.0 Marshmallow і новіше. Пристрій мусить мати вільного місця не менше 3.05 МБ [5]. Обов'язково потрібен акаунт у Google Play, аби скачати застосунок. Для поліпшення користування готової системи було розроблено керівництво користувача, у якому користувач зможе в доступній формі ознайомитися з функціями програмного забезпечення для максимально ефективної роботи із ним.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Альфред, В. Ахо Компиляторы. Принципы, технологии и инструментарий / Альфред В. Ахо и др. – М.: Вильямс, 2015. – **689** с

2. Гонсалвес, Энтони Изучаем Java EE 7 / Энтони Гонсалвес. – М.: Питер, 2016. – 640 с.

3. Савитч, Уолтер Язык Java. Курс программирования / Уолтер Савитч. – М.: Вильямс, **2015**. – 928 с.

4. Хабибуллин, Ильдар Самоучитель Java / Ильдар Хабибуллин. – М.: БХВ-Петербург, **2014**. – 768 с.

5. Нотон Java. Справочное руководство. Все, что необходимо для программирования на Java / Нотон, Патрик. – М.: Бином, **2015.** – 448 с.

УДК 004.9
Степаненко О.О.[1], Федорченко Є. М.[2], Терещенко Р.М.[3]
[1]канд. техн. наук, доц. НУ«Запорізька політехніка»
[2]старш. викл. НУ«Запорізька політехніка»
[3]студ. гр. КНТ-116 НУ«Запорізька політехніка»

ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ КОНТРОЛЮ РОБОТИ ПЕРСОНАЛУ НА ПІДПРИЄМСТВІ

На даний момент на підприємствах з великою кількістю робітників, особливо в разі, якщо їх система оплати праці є погодинною, виникає необхідність контролювати час, який робітник провів на робочому місці.

Програмна реалізація інструменту для вирішення такої проблеми має багато переваг. При наявності системи автоматизації керівник підприємства може запобігти злочинного змінення даних про відпрацьований час та звільняється від необхідності утримання в штаті окремого робітника для контролю запізнень іншого персоналу [1]. Дані у електронному вигляді можуть допомогти керівникам в подальшому аналізі для виявлення проблем на підприємстві, які сприяють порушенням графіку роботи, та для виявлення окремих працівників, які регулярно порушують графік роботи, для застосування до них штрафних санкцій. Також за наявності додаткових відділень підприємства, наприклад інших цехів, офісів у інших містах тощо, керівник підприємства може виявити проблеми з керівниками відділень, при виявленні у цих відділеннях масових порушень трудової дисципліни [2].

Є декілька варіантів вирішення проблеми фіксації часу:

– фіксація часу вручну записами на паперових носіях (окремий співробітник робить відмітки в журналі відвідувань про час роботи персоналу). Такий варіант має такі проблеми: можливість фальсифікації даних, проблеми з подальшим аналізом, потреба утримувати на підприємстві окремого співробітника;

– фіксація часу за допомогою засобів контролю входу до приміщення з індивідуальними ключами, наприклад турнікети, електронні замки або замки зі сканером відбитків пальців. Проблема такого способу – необхідність встановлення фізичних пристроїв, що вимагає фінансових витрат на покупку, монтаж та демонтаж у разі переїзду [3] – [5].

В обох випадках виникають додаткові витрати, при чому в будь-якому разі спосіб фіксації буде недостатньо гнучким при масштабуванні підприємства.

Тому було вирішено розробити програмну реалізацію системи контролю часу для працівників підприємства задля контролю фіксації запізнень та фактично відпрацьованого часу кожним працівником, яка не

буде прив'язуватися до фізичних пристроїв та буде використовувати дані з персонального мобільного телефону працівника.

Експлуатація розробленого програмного засобу показала доцільність використання подібних систем з метою зменшення кількості запізнень та зменшення обов'язків керуючого персоналу.

Система може бути використана на будь-яких підприємствах, де важливо фіксувати час, проведений працівниками на роботі, наприклад: заклади харчування, лікарні, крамниці, офіси тощо.

Інструмент було розроблено на мобільні платформи iOS та Android з використанням API. Також було розроблено панель керування для спрощеного аналізу та обробки отриманих даних керуючим персоналом.

Для використання програмного засобу потрібна така мінімальна конфігурація технічних засобів:

– Wi-Fi мережа на підприємстві зі статичною ip-адресою;

– смартфони з Android 5.0 і вище або iOS 10 та вище у кожного працівника.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Ташков, Петро Веб-мастеринг HTML, CSS, JavaScript, PHP, CMS, AJAX, розкрутка / Петро Ташков. – М .: Книга по Требованию, 2014. – 512 с.

2. Дакетт, Джон HTML і CSS. Розробка і дизайн веб-сайтів / Джон Дакетт. – М .: Ексмо, 2013. – 480 с.

3. Дакетт, Джон Основи веб-програмування з використанням HTML, XHTML і CSS / Джон Дакетт. – М .: Ексмо, 2013. – 768 с.

4. Дебольт HTML і CSS. Спільне використання / Дебольт, Вірджинія. – М .: НТ Пресс, 2013. – 512 с.

5. Чебикін, Ростислав Самовчитель HTML і CSS. Сучасні технології / Ростислав Чебикін. – Москва: Вогні, 2012. – 624 с.

УДК 004.9

Степаненко О.О.[1], Федорченко Є. М.[2], Трошина Н.Є.[3]
[1]канд. техн. наук, доц. НУ«Запорізька політехніка»
[2]старш. викл. НУ«Запорізька політехніка»
[3]студ. гр. КНТ-116 НУ«Запорізька політехніка»

ПРОГРАМНА РЕАЛІЗАЦІЯ ІНТЕРНЕТ-МАГАЗИНУ З ПРОДАЖУ ГАМАНЦІВ

У наш час досить швидко розвиваються інтернет-технології. Постійно виходять нові, більш вдосконалені версії програм і різних проектів, реалізація

кожного з них дозволяє швидко знаходити нових користувачів. Якщо ще 10 років тому не кожна компанія могла дозволити собі власний інтернет сайт, то на сьогодні практично не залишилося жодного приватного підприємця, який би не реалізував свої послуги або продукцію за допомогою інтернет ресурсу [1] – [4].

Людина, яка хоче придбати будь-який товар, в першу чергу звертається до інтернету, щоб визначитися з моделлю і її вартістю, і тільки після того вирішує, відправитися йому в магазин або замовити товар через інтернет з доставкою додому. Більшість користувачів обирає другий варіант, оскільки різниця у вартості досить незначна, але таким чином вони можуть заощадити свій час. Компанії, які йдуть в ногу з часом, і можуть передбачити, що буде потрібно в майбутньому, мають набагато більше шансів на успіх, ніж ті, хто діє виключно з позицій консерватизму. Тому, чим раніше організація усвідомлює необхідність створення сайту, тим міцніший фундамент вона закладає для себе на майбутнє. Таким чином, актуальність створення сайту очевидна. Для кожної компанії створення сайту - це величезна перевага, оскільки за досить короткий термін клієнтська база збільшиться в рази, і це найефективніша реклама, таким чином, будуть заощаджені кошти. Власний сайт дозволяє отримувати додатковий прибуток шляхом розміщення на своїх сторінках реклами інших компаній. Згодом завдяки сайту продукція стає впізнаваною. Розробка Інтернет-магазину та його використання є актуальним питанням на сьогоднішній день, оскільки мільйони людей щодня, не виходячи з дому, купують різні товари в електронних магазинах. В світі, а зокрема в Україні, величезними темпами росте кількість користувачів Інтернету і, як наслідок, кількість «електронних» покупців [4] – [5].

Зважаючи на вищесказане, було розроблено інтернет-магазин з продажу гаманців. Це програмне рішення автоматизує процес вибору й покупки гаманців, дозволяє користувачеві здійснювати покупки, не виходячи з будинку, дозволяє аналізувати та прогнозувати продажі гаманців, розширює клієнтську базу. Після того, як Інтернет-магазин було налагоджено і протестовано, виявилося, що:

– інтернет-магазин стійко й автономно функціонує;

– інтернет-магазин має зручний інтерфейс і діалог з користувачем;

– інтернет-магазин забезпечує правильність вхідних й вихідних даних (наявна обробка виняткових ситуацій);

– інтернет-магазин має достатній рівень безпеки функціонування;

– інтернет-магазин коректно відображується у всіх сучасних браузерах;

– в інтернет-магазині є наявність необхідної інформації щодо процесів функціонування системи (допомога користувачу).

Галузь використання розробленого програмного засобу дуже широка – це інтернет-торгівля на ринках в Україні. Для використання системи

потрібен персональний комп'ютер або ноутбук зі встановленою операційною системою Microsoft Windows XP/7/8/10 та браузером, у якому буде відбуватися робота користувача або адміністратора інтернет-магазину. Також інтернет-магазин повинен бути встановлений на ПК або серверну станцію із працюючим вебсервером. Використання розробленого інтернет-магазина дозволить залучити нових клієнтів та автоматизувати процес продажу гаманців.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Швачич, Г.Г. Сучасні інформаційно-комунікаційні технології : навчальний посібник / Г.Г.Швачич, В.В.Толстой, Л.М.Петречук. – Дніпро: НМетАУ, 2017. –230 с.

2. Иванова, Г.С. Основы программирования: учебник для вузов / Г.С. Иванова – М. : Изд-во МГТУ им. Н.Э. Баумана, 2002. – 416 с.

3. Кормен, Т. Алгоритмы: построение и анализ / Т.Кормен, Ч. Лейзерсон, Р. Ривест. – М. : МЦНМО, 2009. – 960 с.

4. Левитин, В. Алгоритмы: введение в разработку и анализ / В. Левитин. – М. : Вильяме, 2006. – 576 с.

5. Сырецкий, Г. Информатика. Фундаментальный курс. Основы информационной и вычислительной техники / Г. Сырецкий. – СПб. : БХВ-Петербург, 2005. – 832 с.

УДК 004.891.3

Судаков В.Р.[1], Терещенко Е. В.[2], Лук'янова Т.В.[3], Миронова Н.О.[4]
[1]студ. гр. КНТ-137 НУ«Запорізька політехніка»
[2]канд. фіз.-мат. наук, доц. НУ«Запорізька політехніка»
[3]асп. KU Leuven
[4]наук. співробітник FH Dortmund

ПРОГРАМНА РЕАЛІЗАЦІЯ МЕТОДІВ РОЗПІЗНАВАННЯ ТРІЩИН БУДІВЕЛЬ

На сьогоднішній день серед будівель, які перебувають в експлуатації, достатньо велика частка має пошкоджені конструкції. Одним із найбільш розповсюджених типів пошкоджень будівель є тріщини. Проблема забезпечення максимально можливого строку служби, продовження термінів експлуатації будівель в умовах реальних обмежень є однією з найактуальніших проблем для вчених та технічних спеціалістів у всьому світі. Наслідки виникнення відмов, несправностей або дефектів можуть призвести до техногенних катастроф із втратами людей, фатальними для

оточуючого середовища та із значними матеріальними збитками. А як відомо, тріщини є однією із найбільших загроз та причиною виникнення інших дефектів [1]. Тому дослідження виявлення тріщин є актуальною задачею.

Програмно-апаратна система нагляду за станом технічних систем «VultureEye» фактично фіксує або появу дефекту конструкції, який може привести до аварійної ситуації. Об'єкти будівництва, як складні інженерні конструкції, повинні надійно безаварійно функціонувати значний період часу експлуатації незважаючи на широкий спектр природних та техногенних впливів, які зазвичай є джерелами утворень нових дефектів. Наявні промислові рішення мають надто високу вартість виробництва та підтримки. Тому актуальною задачею є створення бюджетного програмно-апаратного комплексу для забезпечення виявлення, фіксації та аналізу стану будівель певних районів в режимі реального часу.

Метою даної роботи є розробка методів та моделей в рамках інформаційної технології моніторингу, діагностики та розпізнання дефектів типу «тріщина» будівельних об'єктів у процесі адаптивної фільтрації на основі цифрових зображень в оптичному діапазоні як програмної частини комплексу «VultureEye».

Для реалізації програмної частини проекту досліджено створення та навчання нейронних мереж за допомогою засобів для Python. Для досягнення поставленої мети в роботі пропонується вирішення наступних задач:

а) дослідження особливості візуалізації дефектів об'єктів будівництва на основі зображень в оптичному діапазоні;

б) визначення класифікуючих ознак фільтрації цифрових зображень тріщин на об'єктах будівництва;

в) визначення інформаційної моделі дефекту типу «тріщина» за цифровим зображенням в оптичному діапазоні;

г) визначення методу розпізнання та класифікації дефектів типу «тріщина» об'єктів будівництва на основі поточного аналізу їх зображень;

д) побудова та дослідження алгоритму адаптивної фільтрації для пошуку і класифікації зображень дефектів типу магістральної та корозійної тріщини;

е) розробка та дослідження інформаційної технології для адаптивної фільтрації, пошуку і класифікації зображень з дефектами.

Використання розробки дозволить за малі кошти проводити аналіз місцевості на наявність пошкоджених конструкцій та активно спостерігати за процесом сканування та збору інформації. В подальшому можна вдосконалити систему повністю функціональною апаратною частиною у вигляді енергоефективних дронів з камерами та вдосконаленою моделлю нейронної мережі для аналізу отриманих зображень.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Case Study on Cracks in Public Buildings and their Remedies, [Electronic resource]. URL: https://www.ijsr.net/archive/v6i5/ART20173175.pdf

УДК 004.9
Шемберко Ю.А.[1], Терещенко Е.В.[2], Миронова Н.О.[3]
[1]студ. гр. КНТ-129м НУ«Запорізька політехніка»
[2]канд. фіз.-мат. наук, доц. НУ«Запорізька політехніка»
[3]наук. співробітник FH Dortmund

ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ МЕТОДІВ РОЗПІЗНАВАННЯ ОБ'ЄКТІВ НА ВИРОБНИЦТВІ

Пандемія COVID-19 вплипнула на існування та розвиток суспільства, на умови соціального життя, навчання, на умови виробництва.

Карантині заходи вимагають використання систем розпізнавання в місцях великого скупчення людей: в торгових центрах, вокзалах, аеропортах, громадському транспорті, виробництві. Насьогодні найбільш популярними є такі системи розпізнавання обличчя в масках та без масок: Xeoma Mask Detector PRO, Aerialtronics Face Mask Detection Software, Hikvision Mask Detection. Ці системи мають певні недоліки: високу ціну, складність в використанні, необхідність потужних апаратних ресурсів, недостатню продуктивність та достовірність результатів в умовах браку освітленості, й, головне, неможливість використання цих систем в умовах виробництва.

Для умов виробництва було поставлено задачу по розробці програмного забезпечення, що дозволяє розпізнавати наявність об’єктів певного заданого класу: маски, каски та захисних окулярів у людині, що працює на виробництві в реальному часі у відеопотоках.

Метою роботи є розробка системи машинного зору, що дозволяє виконувати розпізнавання об’єктів заданого класу: маски, каски та захисних окулярів у людині, що працює на виробництві в реальному часі у відеопотоках з високим ступенем достовірності розпізнавання.

В роботі побудовано технологію розпізнавання об’єктів заданого класу: маски, каски та захисних окулярів у людині, що працює на виробництві у реальному часі у відеопотоках, яка складається з шістьох етапів.

Також виконано дослідження програмних засобів розробки системи: була обрана мова програмування Python та бібліотека комп'ютерного зору з відкритим вихідним кодом OpenCV.

Наведено опис файлів розробленої системи, процесу тренування або навчання моделі та розпізнавання об’єктів заданого класу: маски, каски та

захисних окулярів у людині, що працює на виробництві у реальному часі у відеопотоках.

Було описано та досліджено функціональні можливості розробленої системи. Ці функціональні можливості були порівняні з іншими відомими системами, що працюють у сферах визначення об'єктів заданого класу на виробництві або підприємстві за допомогою алгоритмів комп'ютерного зору або машинного навчання. Для оцінки показника достовірності розпізнавання об'єктів заданого класу на виробництві було проведено ряд експериментів. В ході цих експериментів встановлено залежність показника достовірності від умов, у яких цей показник розраховується. На нього впливають такі параметри як якість освітленість робочої зони, кут ведення відеофіксації та фізична відстань від відео пристрою до потенційного об'єкта розпізнавання. Результати експериментів показали достатньо добрий показник достовірності розпізнавання у розробленій технології (76,8 %).

Також запропоновано можливе поліпшення технології за рахунок додавання протоколу Modbus TCP/IP для сигналізації результатів розпізнавання на сервер.

УДК 004.021
Чемерис К.М.[1], Дейнега Л.Ю.[2]
[1]студент гр. КНТ-137 НУ «Запорізька політехніка»
[2]старш. викл. НУ «Запорізька політехніка»

ВИЯВЛЕННЯ НЕСАНКЦІОНОВАНИХ ДІЙ І АТАК В МЕРЕЖАХ МЕТОДОМ ВЕЙВЛЕТ-АНАЛІЗУ

В даний час Інтернет є місцем високої ділової активності. Тому в даний час особливо актуальні проблеми, що стосуються виявлення мережних атак і аномалій мережевого трафіку з метою попередження подальшого вторгнення і зниження ризиків від подібних порушень. Далі представлено аналіз існуючих основних методів вирішення задачі виявлення мережних атак, вибір методів та їх застосування для виявлення вторгнень.

Мережні аномалії можна поділити на дві основні групи: програмно-апаратні відхилення і проблеми безпеки. До програмно-апаратних відхилень належать: апаратні несправності, помилки конфігурації, помилки програмного забезпечення та проблеми продуктивності обладнання. Порушення мережевої безпеки містять: сканування, атаки з метою відмови від обслуговування, вірусну активність, поширення програмних "черв'яків", експлуатацію вразливостей, аналізатори трафіку і мережні модифікатори. Найбільший економічний збиток операторам зв'язку наносять атаки з метою перевантаження мереж або сервісів і мережна вірусна активність.

Мережна атака - деякий набір дій (активність), що має на меті зробити з комп'ютером (сервером) якісь дії віддалено. Дії, зазвичай, є небажаними, і дуже важливо вчасно застосовувати алгоритми виявлення мережних атак.

Застосування алгоритму на основі вейвлет-перетворення є однією з найбільш перспективних технологій пошуку потрібної інформації в базах даних мережної безпеки. Вейвлет - це математична функція, яка дозволяє аналізувати різні частотні компоненти даних [1]. Оскільки вейвлети мають хорошу частотно-часову адаптацію, вони можуть служити зручним інструментом для дослідження частотних характеристик нестаціонарного сигналу. Перевага такого підходу - характерні деталі, які можуть залишатися непоміченими при одному масштабі, легко можуть бути виявлені на іншому.

Можна використовувати неортогональні вейвлети для безперервного вейвлет-перетворення і ортогональні вейвлети для дискретного. При безперервному вейвлет-перетворенні можливе більш детальне вивчення поведінки трафіку. Дискретне перетворення більш корисне для автоматичного виявлення аномалій в мережному трафіку, тому що може працювати швидше і можна встигнути виявляти аномалії мережного трафіку[2].

У загальному випадку, зі збільшенням числа коефіцієнтів вейвлета, функції стають більш гладкими, що може полегшити виявлення мережної атаки. Для виявлення аномалій мережного трафіку необхідно проводити деталізацію інформації з масиву цифрових даних. Тому для вибору системи базисних вейвлетів були проаналізовані вейвлети з компактним носієм: Хаара, Добеши, вейвлети Койфлети, а також Морлі і меськиканський капелюх, які найбільш якісно виділяють локальні особливості сигналів.

Для виконання виявлення мережних аномалій мережного трафіку та шумозаглушення сигналу з використанням різних вейвлет-функцій був використаний пакет Wavelet Tools в програмі Matlab. який має засоби для побудови вейвлет-спектрів сигналів з поліпшеною візуалізацією. Був згенерований мережний трафік з аномаліями і застосовані безперервне і дискретне вейвлет-перетворення з використанням різних вейвлет-функцій для виявлення мережних аномалій. При практичних дослідженнях в розділі Wavelet 1-D пакету Wavelet Tools використано вейвлет Хаара до 10 рівня. Аномалії легко виявлено тільки на 5-9 рівні, краще на 7-9. При використанні вейвлета Добеші другого порядку аномалії теж добре помітні на 7-10 рівнях. В розділі Wavelet Packet 1-D використовувався вейвлет Хаара з максимальним рівнем 8, виявлено аномалії на обраних вузлах кращого дерева після реконструкції. В розділі Continuous Wavelet 1-D результат з використанням Хаара можна бачити в кращому для людського сприйняття вигляді.

Згідно з дослідженнями для моніторингу мережного трафіку доцільно використовувати вейвлет Хаара, і алгоритм Малла для отримання найкращого результату на основі використання вейвлет-перетворення.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Введение в вейвлет-преобразование [Text] / А.Н. Яковлев. – Новосибирск, 2003. – 104 с.

2. Дубровін, В.І. Виявлення DOS-атак в мережевому трафіку методом Вейвлет-перетворення [Текст] / В.І. Дубровін, Б.В. Петрик, Г.В. Неласа // Сучасний захист інформації. – 2020. – № 2. – С. 37 – 46.

УДК 004.5

Архипов А.С.[1], Пархоменко А.В.[2]
[1]студ. гр. КНТ-210м НУ «Запорізька політехніка»
[2]канд. техн. наук, доц. НУ«Запорізька політехніка»

СТВОРЕННЯ ТАБЛИЦЬ КОНФІГУРАЦІЇ ДЛЯ АНАЛІЗУ, ОБРОБКИ ТА ВІЗУАЛІЗАЦІЇ ДАНИХ З ВИПРОБУВАЛЬНОГО СТЕНДУ ДЛЯ ГАЗОТУРБІННИХ ДВИГУНІВ

Як показали проведені дослідження, для ефективного управління випробуваннями авіаційних двигунів необхідна надійна автоматизована інформаційно-вимірювальна система (АІВС).

Основною метою випробувань є перевірка працездатності двигуна шляхом порівняння отриманих з випробувального стенду даних з даними зі стандартів. Під час випробування датчики на двигуні збирають величезну кількість даних, які потребують обробки та аналізу. Це завдання вирішує підсистема аналізу даних, що створена на основі платформи LabVIEW. Створені в Microsoft Excel таблиці за декілька кліків вбудовуються у проєкт АІВС завдяки технології віртуальних приладів LabVIEW. Таблиці Excel мають змогу отримувати цифрові дані за запитом напряму від плат стенду. Аналіз відбувається завдяки перерахунку первинних даних з плат за певними алгоритмами. Тому, актуальною задачею є удосконалення таблиць конфігурації для випробувального стенду газотурбінних двигунів.

Метою роботи є отримання та передача даних зі стенда, розробка алгоритмів аналізу даних у таблицях Microsoft Excel, візуалізація оброблених даних з використанням інструментів LabVIEW для підвищення ефективності випробувань газотурбінних двигунів.

В роботі пропонується модифікувати розроблену АІВС шляхом створення таблиць конфігурації Microsoft Excel для різних типів двигунів,

реалізувати розроблені алгоритми аналізу даних в таблицях, розробити екрани для візуалізації оброблених даних, використовуючи інструменти LabVIEW. На основі реалізації групи алгоритмів можливо задати діапазони значень контрольованого параметра, а для кожного діапазону можливо використовувати різні формули перерахунку даних; отримати попередні значення параметрів у вигляді масиву даних та обробити його за допомогою вбудованої бібліотеки LabVIEW - Pcalc.dll; оцінити нормальне поводження параметра.

Подальший аналіз таблиць конфігурації дозволить сформувати рекомендації щодо оптимальної роботи АІВС.

УДК 004.5
Бережний О.Ю.[1], Туленков А.В. [2], Пархоменко А.В.[3]
[1]студ. гр. КНТ-127 НУ «Запорізька політехніка»
[2]асп. НУ «Запорізька політехніка»
[3]канд. техн. наук, доц. НУ«Запорізька політехніка»

РОЗРОБКА ПРОГРАМНО-АПАРАТНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ РЕАЛІЗАЦІЇ ТЕХНОЛОГІЇ ІНТЕРНЕТУ РЕЧЕЙ В НАВЧАЛЬНІЙ ЛАБОРАТОРІЇ

Перелік пристроїв, які використовують технології Інтернету речей (Internet of Things, IoT) постійно розширюється. Вже розроблено та пропонується на ринку велика кількість пристроїв, які надають можливість комунікації та налаштування через мережу Інтернет. Згідно з даними аналітичних центрів, у 2020 кількість пристроїв, які застосовано в системі IoT, сягнула 8,7 мільярдів, хоча у 2017 році було під’єднано лише 7,7 мільярдів одиниць [1]. Показники зростання свідчать про перспективність напрямку IoT і, відповідно, про важливість навчання студентів даним технологіям. Тому, тема роботи є актуальною.

Навчальна лабораторія REIoT представляє собою комплекс, який об'єднує дві підсистеми: RELDES (віддалена лабораторія розробки вбудованих систем) та Smart House&IoT (віддалена лабораторія Розумний будинок та Інтернет речей). Комплекс призначено для віддаленого користування: на сайті лабораторії доступна можливість передачі команд на пристрої комплексу й спостереження за їх виконанням через онлайн відеотрансляцію. Наявний стенд домашньої автоматизації MyHome від компанії Legrand здатен розширити функціонал існуючого комплексу, оскільки дозволяє ознайомитись з промисловими рішеннями для реалізації IoT. Але, через відсутність можливості відеотрансляції, стенд не може бути

під'єднаний до REIoT. Тому, важливою задачею є реалізація дистанційного спостереження за функціонуванням стенду MyHome.

Враховуючі високу мобільність стенду, найкращим способом забезпечити спостереження є впровадження пристроїв відеофіксації, які базуються на передачі зображення бездротовим способом. Існуючі промислові рішення мають надто високу вартість. Тому, пріоритетною є реалізація бюджетного програмно-апаратного комплексу (ПАК) для відеотрансляції роботи пристроїв стенду в режимі реального часу на сайті лабораторії.

Метою роботи є створення ПАК для реалізації безперебійної трансляції відеопотоку на сервер віддаленої лабораторії для забезпечення можливості спостерігати за проведенням експериментів системи домашньої автоматизації онлайн.

Проведений аналіз компонентів, які мають бути застосовані при реалізації ПАК, виявив, що важливою складовою є мікроконтролерний модуль, який повинен забезпечити отримання й обробку зображення з камери, та подальшу передачу його на сервер. Розглянувши існуючі апаратні рішення (модулі на базі мікроконтролерів EPS, AVR, та ін.) було обрано ESP-32-CAM через те, що він відрізняється компактністю, відносно великою обчислювальною потужністю і широким спектром вбудованого функціоналу (вбудований Wi-Fi адаптер, ліхтар, підтримка MicroSD карти та ін.). Щоб збільшити дальність передачі даних, модуль було модифіковано під застосування зовнішньої антени.

Для реалізації програмної частини проекту досліджено стандарти передачі відеопотоку та обрано Motion JPEG (MJPEG), в якому кожен відеокадр передається як окреме JPEG зображення. Застосування даного стандарту, на відміну від інших, забезпечує стійкість передачі даних до можливих шумів і перешкод, оскільки втрата певної частини інформації (кількох відеокадрів) не впливатиме на потік даних, який передається далі. Враховуючі наявні архітектурні обмеження апаратного комплексу на максимальну кількість користувачів і зростаючий попит на використання онлайн лабораторії, доцільним є впровадження опосередкованої відеотрансляції через сервер лабораторії. Комплекс було налаштовано на передачу відеопотоку через Wi-Fi мережу у OpenHAB - систему керування автоматизацією стенду MyHome, яку встановлено на сервері. Наступним кроком роботи є інтеграція та налаштування відеотрансляції з системи OpenHAB на сайт навчальної лабораторії.

В майбутньому компоненти комплексу можна замінити іншими для підвищення дальності бездротової передачі даних і зменшення енергоспоживання з метою підвищення мобільності та автономності стенду.

Використання розробки в навчальній лабораторії дозволить в умовах дистанційного режиму навчання отримати студентам повноцінний доступ до матеріально-технічного забезпечення лабораторії, підвищить наочність досліджень, забезпечивши можливість в режимі онлайн спостерігати за процесом проведення різноманітних експериментів, що сприятиме покращенню якості освітнього процесу.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Number of IoT connected devices worldwide 2019-2030 [Electronic resource]. – Access mode: http://surl.li/qgzd

УДК 004.8
Горіченко Ю.Є.[1], Туленков А.В.[2], Пархоменко А.В.[3]
[1]студ. гр. КНТ-227 НУ «Запорізька політехніка»
[2]асп. НУ «Запорізька політехніка»
[3]канд. техн. наук, доц. НУ«Запорізька політехніка»

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ АНАЛІЗУ ДАНИХ СИСТЕМИ ДОМАШНЬОЇ АВТОМАТИЗАЦІЇ

Головною метою впровадження системи домашньої автоматизації (СДА) є забезпечення енергоефективності, безпеки, низьких експлуатаційних витрат та підвищення рівня комфорту мешканців.

СДА складається з різноманітних пристроїв, які виробляють велику кількість даних. Методи інтелектуального аналізу (ІА) даних та машинного навчання можуть відігравати важливу роль, тому що накопичені дані можна перетворювати на корисну інформацію для користувача у реальному часі. Таким чином, це дозволяє спрогнозувати та швидко відреагувати на виникаючі ситуації, дати корисні поради та зробити систему більш інтелектуальною. Тому, тема роботи є актуальною.

Метою роботи є дослідження методів та засобів інтелектуального аналізу даних СДА та розробка програмного забезпечення (ПЗ) для аналізу даних і контролю показників СДА для надання рекомендацій користувачу у реальному часі щодо безпечного користування електроприладами.

Було детально проаналізовано досвід використання інтелектуального аналізу даних для СДА. Автори у своїх дослідженнях найчастіше використовували метод пошуку асоціативних правил, який допомагає проаналізувати великі обсяги даних та встановити залежності між діями, які виконуються в певний момент часу. Отримані результати дають можливість побачити звички мешканців, їх розпорядок дня, використання побутових

приладів та багато іншої корисної інформації. Автори виявили, що при прогнозуванні ситуацій, пов'язаних з використанням електроенергії, найкращі результати дає використання наступних методів: мережа Байєса, алгоритм «випадковий ліс» та штучні нейронні мережі. Ці методи мають великий потенціал, бо можуть з «сирих» даних витягати корисну інформацію, але для того, щоб мати змогу використовувати ці знання для СДА, потрібно провести ще багато досліджень та розробити стратегію для досягнення поставленої мети.

Прикладом готового рішення є іспанський стартап Smappee. Розробники пропонують ПЗ, яке дозволяє зрозуміти, які пристрої та в якому обсязі споживають електроенергію та яким чином можна зменшити споживання електроенергії. До цієї системи можна підключити не тільки споживачів енергії, але й генератори – вітряки, сонячні панелі, тощо. Але ця технологія ще знаходиться на стадії розробки та потребує багато допрацювань.

Існує багато програмних засобів та бібліотек, які дозволяють використовувати алгоритми машинного навчання та ІА даних: ELKI, KNIME, Orange, Weka, Pandas, PAW, R, ROOT, SciPy, Julia.

Для проведення власного експерименту було використано накопичені дані з СДА реальної однокімнатної квартири, в якій проживає дві людини. З досліджуваної СДА були отримані такі дані: температура і вологість в кімнаті та у ванній; статус роботи кухонної витяжки, підсвітки на кухні, штор, телевізора; датчик дверей; температура автомата в електричному щиті; сила струму та потужність струму. Для аналізу даних було обрано відкрите ПЗ Orange, яке постійно оновлюється та дозволяє застосовувати різні алгоритми ІА даних. В ході досліджень було проведено ряд експериментів з різними алгоритмами ІА для різних наборів даних.

За результатами проведеного дослідження з використанням алгоритму пошуку асоціативних правил можна зробити висновок, що більшість правил не мають сенсу або є тривіальними. Наприклад, було зроблено припущення, що мешканець рідко користується приладами або рідко буває вдома; він полюбляє дивитися телевізор вранці або в обід під час прийому їжі; коли ввечері ввімкнено телевізор, або у ванній кімнаті підвищується вологість та це може означати, що він в цей час приймає ванну або душ. В даному випадку отримані правила можуть дати інформацію лише про те, які звички та розпорядок дня має мешканець будинку. Таким чином, було перевірено багато алгоритмів для прогнозування. Підводячи підсумки експерименту з алгоритмами машинного навчання, можна відмітити, що найкращим виявився алгоритм AdaBoost, а його середня абсолютна помилка становила 60%, що є незадовільним результатом для прогнозування. Дослідження показали, що всі використані методи інтелектуального аналізу даних не дозволять вирішити поставлену мету, тому що платформа OpenHAB

накопичує недостатньо даних, які можна перетворити на корисні знання та це не дозволить сформувати рекомендації щодо безпечного та економічного користування електричними приладами.

Таким чином прийняте рішення про розробку власного ПЗ, яке буде взаємодіяти з платформою OpenHAB за допомогою API. Розроблений застосунок буде обчислювати поточну силу струму та якщо вона буде перевищувати норму, сповіщати користувач та надавати певні рекомендації. Таким чином можливо підвищити рівень безпеки при користуванні електроприладами.

УДК 004.8
Задоян М.К.[1], Туленков А.В.[2], Пархоменко А.В.[3]
[1]студ. гр. КНТ-120м НУ «Запорізька політехніка»
[2]асп. НУ «Запорізька політехніка»
[3]канд. техн. наук, доц. НУ«Запорізька політехніка»

РОЗРОБКА СИСТЕМИ ІНТЕЛЕКТУАЛЬНОГО АНАЛІЗУ ДАНИХ СИСТЕМИ ДОМАШНЬОЇ АВТОМАТИЗАЦІЇ

Сьогодні інтелектуальні системи моніторингу енергоспоживання успішно впроваджуються по всьому світу великими підприємствами, офісами та адміністративними будівлями з метою реалізації політик енергозбереження та зміни поведінки з ресурсами на локальному та глобальному рівні.

В системі домашньої автоматизації (СДА) велика кількість пристроїв збирають, обробляють і передають дані про параметри датчиків та стани актуаторів, що пов'язані з діями користувачів або самої СДА. Ці дані можуть бути зібрані у великій кількості та збережені для подальшої візуалізації та інтелектуального аналізу (ІА).

Аналіз показав, що є кілька типів аналітики даних: аналітика в реальному часі, офлайн-аналітика, аналітика на рівні пам'яті, масова аналітика. Для СДА є сенс вибрати аналітику в режимі реального часу, для якої представлені методи аналізу великих даних в категоріях класифікації, кластеризації, аналізу асоціативних правил і прогнозування.

Для реалізації задачі ІА даних СДА можливо використати алгоритми пошуку асоціативних правил (Association Rule Mining, ARM): Apriori, AprioriTID, SETM, Apriori hybrid, AIS та FP-Growth. В результаті дослідження було виділено два з них - це Apriori Hybryd та FP-growth. Apriori – це найпопулярніший алгоритм, у якого є велика спільнота практиків. В той же час, FP-growth – це більш сучасний алгоритм, в нього найкращі показники за використання дерева FP.

Сьогодні є програми, в яких реалізовані алгоритми ARM (наприклад, Healthcare, Market Analysis, Industry, Bioinformatics, e-governance та ін.). Але їх складно інтегрувати з СДА та не в усіх реалізовано алгоритм Apriori Hybryd. Тому, було прийнято рішення розробити власний сервіс - інтелектуальний помічник на основі алгоритму Apriori Hybryd, який надасть рекомендації у вигляді списку повідомлень користувачеві щодо використання електричних приладів та споживання електроенергії, які допоможуть попередити можливі проблеми в СДА з урахуванням можливих дій користувачів. Надалі буде виконана інтеграція розробленого сервісу у вже існуючу СДА.

УДК 004.5
Малюков М.К.[1], Пархоменко А.В.[2]
[1]студ. гр. КНТ-126 НУ «Запорізька політехніка»
[2]канд. техн. наук, доц. НУ«Запорізька політехніка» розробка

РОЗРОБКА ВІРТУАЛЬНОГО СЕРЕДОВИЩА ДЛЯ ЛІКУВАННЯ НЕРВОВИХ ТА ПСИХІЧНИХ РОЗЛАДІВ У ЛЮДЕЙ

Сучасна підготовка студентів в галузі медицини включає проблемно-орієнтоване навчання, комунікативні навички та навчання на основі використання технологій віртуальної реальності.

Як відомо, за допомогою віртуальної реальності (Virtual Reality, VR) можна змоделювати будь-яку медичну ситуацію, щоб студенти могли впоратися з нею, як в реальному житті. Наприклад, VR можна використовувати, щоб візуалізувати внутрішню частину людського тіла для вивчення анатомії людини та проведення віртуального розтину. VR корисна при попередньому плануванні складних втручань, таких як нейрохірургічні операції, оскільки вона допомагає хірургічній бригаді пройти етапи операції та прорепетирувати майбутні дії [1].

Медична віртуальна реальність також може бути корисна для лікування від алкогольної та наркотичної залежності, використовуючи методи поступового впливу поряд з навчанням того, як реагувати на спонукання до залежності.

Дані про стан людей в країнах ЄС свідчать про те, що проблеми з психічним здоров'ям щорічно зачіпають десятки мільйонів європейців. За останніми оцінками IHME, майже 84 мільйони чоловік мали проблеми з психічним здоров'ям у 2016 році. 25 мільйонів чоловік (або 5,4% населення) живуть з тривожними розладами, за якими йдуть депресивні розлади, від яких страждають більше 21 мільйона чоловік (або 4,5% від чисельності

населення). Важкі психічні захворювання, такі як біполярні розлади, зачіпають майже 5 мільйонів чоловік [2].

Саме тому актуальною є розробка віртуального середовища для лікування певних нервових та психічних розладів з використанням окулярів VR Oculus Pro.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Application of virtual reality technology in clinical medicine [Electronic resource]. – Access mode : https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5622235/

2. IHME, Global Health Data Exchange [Electronic resource]. – Access mode : www.healthdata.org/.

УДК 004.5

Терещенко О.В.[1], Соколянський О.В.[2], Пархоменко А.В.[3]
[1]студ. гр. КНТ-227 НУ «Запорізька політехніка»
[2]асп. НУ «Запорізька політехніка»
[3]канд. техн. наук, доц. НУ «Запорізька політехніка»

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ УПРАВЛІННЯ ВІДДАЛЕНОЮ ЛАБОРАТОРІЄЮ RELDES

На сьогоднішній день віддалені лабораторії (ВЛ) набувають дуже великої популярності через світову пандемію та тотальний перехід на дистанційне навчання, коли відсутня можливість роботи в реальних лабораторіях навчального закладу. ВЛ представляють собою програмно-апаратний комплекс, який дозволяє проводити певні експерименти без безпосереднього контакту з реальним обладнанням, шляхом віддаленого управління та спостереження за результатами за допомогою вебкамери.

Треба відзначити, що важливою перевагою є те, що при неспроможності придбати коштовне обладнання для створення або оновлення власних лабораторій, навчальний заклад має чудову альтернативу використовувати ВЛ інших розробників по всьому світу.

Для організації коректного використання обладнання ВЛ необхідно передбачити множинний доступ, тому на стороні сервера повинна бути реалізована система черги, яка дозволила б користувачу у потрібний час отримати доступ до обладнання. Під час очікування доступу до обладнання, користувач повинен мати можливість стежити за станом черги.

У першій версії лабораторії RELDES черга була реалізована на мові PHP з використанням фреймворку Codeigniter, потім використовувався

фреймворк Yii2. На сьогоднішній день використовується фреймворк Laravel, базові функції якого дозволяють суттєво прискорити процес розробки.

Метою роботи є розробка модулю відображення черги на експериментах віддаленої лабораторії RELDES для підвищення наочності та зручності використання. Модуль повинен відображати всі доступні експерименти та показувати користувачу які на даний момент вільні, а які слід зачекати. Також він повинен сповіщати користувача про час очікування в черзі та вести зворотній відлік.

Впровадження удосконаленого модулю черги для ВЛ дозволить студентам та викладачам раціонально використовувати час, відведений на виконання експериментів та уникнути проблем в роботі.

УДК 004.5
Яремченко Є.В.[1], Залюбовський Я.І.[2], Пархоменко А.В.[3]
[1]студ. гр. КНТ-120м НУ «Запорізька політехніка»
[2]аспірант НУ «Запорізька політехніка»
[3]канд. техн. наук, доц. НУ «Запорізька політехніка»

ВІРТУАЛЬНЕ СЕРЕДОВИЩЕ РОЗУМНИЙ БУДИНОК ДЛЯ ГІБРИДНОЇ ЛАБОРАТОРІЇ GOLDI

Необхідність інтеграції віртуальних лабораторій у процес навчання стає все більш актуальним питанням, особливо в умовах пандемії. Віртуальне навчання розглядається як ефективна підтримка процесу викладання та навчання в різних сферах. Дослідження показало, що результати навчання студентів залишаються однаковими або навіть стають кращими при використанні віртуальних лабораторій замість традиційних реальних лабораторій. Стандартні аргументи для використання віртуальних лабораторій включають:

– доступність - така лабораторія доступна через Інтернет і до неї можна отримати доступ з будь-якого місця в будь-який час;

– економія ресурсів - відсутність фізичних умов та фізичних приладів;

– демонстраційні можливості - вона може показувати речі, які зазвичай складно продемонструвати або важко пояснити.

Однак, як показали проведені дослідження, віддалені та віртуальні лабораторії для вивчення технологій Інтернету речей, систем типу Розумне місто та Розумний будинок (РБ) ще недостатньо поширені, хоча актуальність цих технологій постійно зростає.

Однією з таких лабораторій є гібридна лабораторія Розумний будинок та Інтернет речей, створена на кафедрі програмних засобів [1]. Вона надає користувачам набір обладнання для проведення віддалених експериментів,

що демонструють роботу різних підсистем системи РБ: підсистема контролю освітлення, підсистема контролю доступу, підсистема генерації сонячної енергії, підсистема контролю периметру, підсистема контролю мікроклімату, підсистема кібербезпеки, підсистема контролю присутності, підсистема контролю безпеки. Крім того, лабораторія надає віртуальну модель Розумного будинку, що власне і дозволяє віднести її до класу гібридних лабораторій. Ця модель є веборієнтованою та дозволяє наочно дослідити склат системи РБ та особливості взаємодії з різними її підсистемами.

Гібридна лабораторія GOLDi, розроблена в Technische Universität Ilmenau, активно використовує можливості дистанційних та віртуальних експериментів. Однак реалізація експерименту для навчання студентів у галузі систем РБ розширить функціональні можливості лабораторії та зробить її більш привабливою для студентів.

Метою роботи є розробка віртуального середовища (ВС) РБ на основі стандартного набору обладнання та популярного протоколу передачі даних для навчання основам систем домашньої автоматизації.

Впровадження віртуальної лабораторії в навчальний процес є складним завданням, що включає різні технічні та педагогічні аспекти, такі як: організація ефективної взаємодії, якісна візуалізація та активне використання. Для формалізації вимог до розроблюваного ВСРБ було зібрано та проаналізовано експертні думки. Після цього була розроблена специфікація на розробку ВСРБ, визначені функціональні та нефункціональні вимоги. Основні з них – доступність через Інтернет та використання надійного протоколу передачі даних для зв'язку між ВС та програмним забезпеченням або пристроями студентів. Розроблено UML-діаграму взаємодії користувачів із ВСРБ. Дослідження можливих шляхів реалізації проєкту показали, що сучасні ігрові рушії можуть бути використані для рендерингу та моделювання реалістичної поведінки компонентів РБ. Такі ігрові рушії мають достатній набір інструментів і вже реалізовані об'єктні моделі, що може скоротити час розробки. Таким чином, ігровий рушій Unreal Engine 4 був використаний як інструментарій розробки.

На сьогоднішній день розроблено архітектуру системи, яка включає сервер та графічне представлення ВСРБ. Сервер виконує підключення клієнтів користувача до його екземпляра ВС. Графічне представлення ВС виконує всі функції, пов'язані з візуалізацією. Протокол MQTT був обраний як системний протокол передачі даних.

Використання віртуальних середовищ відкриває широкі можливості для експериментів у різних сферах навчання та покращує результати навчального процесу. Реалізація віртуальних експериментів у галузі систем домашньої автоматизації забезпечує інтерактивне навчальне середовище, що дозволяє

залучити студентів до активного навчального процесу та підвищити їх мотивацію до вивчення сучасних інформаційних технологій та процесів.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Створення та використання віддаленої лабораторії як відкритого онлайн ресурсу / А.В. Туленков, А.В. Пархоменко, О.В. Соколянський, О.М. Гладкова, Я.І. Залюбовський // Вісник НУ «Львівська політехніка» - Інформатизація вищого навчального закладу. – №903. – 2018. – С. 88-97.

УДК 004.5

Троц Є.С.[1], Туленков А.В.[2], Пархоменко А.В.[3]
[1]студ. гр. КНТ-217 НУ «Запорізька політехніка»
[2]асп. НУ «Запорізька політехніка»
[3] канд. техн. наук, доцент НУ «Запорізька політехніка»

МОДЕРНІЗАЦІЯ АПАРАТНОГО ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ВІДДАЛЕНОЇ ЛАБОРАТОРІЇ

В ході досліджень було проаналізовано структурні та функціональні особливості віддаленої лабораторії (ВЛ) REIoT та виявлено ряд як програмних, так і апаратних проблем. Вони перешкоджали зручному використанню лабораторії, а деякі з них повністю унеможливлювали виконання деяких експериментів. Однією з вагомих апаратних помилок ВЛ було неякісне живлення системи, внаслідок чого дисплеї працювали з низькою яскравістю та виникали артефакти при повороті валу серводвигуна.

Метою роботи є дослідження та удосконалення апаратного та програмного забезпечення ВЛ, що дозволить користувачам зручно проводити експерименти.

В результаті модернізації у проблемних місцях було додано додаткові лінії живлення та електролітичні конденсатори великої ємності. На сервер ВЛ завантажено нову більш зручну у використанні версію бібліотеки для дисплею, а також під неї переписано приклади коду, що надаються на сайті для демонстрації роботи.

Також в ході модернізації було перепідключено датчики струму ACS712 та датчики напруги на основі дільника опору, а також удосконалено програмну частину шляхом реалізації декількох алгоритмів фільтрації. В результаті стало можливим коректне зчитування показників напруги 5 та 12 В та сили струму до 30 Ампер.

Іншим важливим рішенням стало перевстановлення платформи OpenHAB з поточної на версію 3.0, а також заміна самої плати Raspberry Pi з

2 до 3 версії. Нова версія openHAB відрізняється від попередньої значно зручнішим використанням, більшою швидкодією, а особливо – можливостями візуального інтерфейсу, що покращує сприйняття та поліпшує наочність інформації.

Даний комплекс заходів покращить функціонування ВЛ RELDES шляхом усунення наявних апаратних та програмних проблем, що дозволить ефективно використовувати це навчальне середовище та отримувати корисні навички і знання, не виходячи з дому.

СЕКЦІЯ «СИСТЕМНИЙ АНАЛІЗ ТА ОБЧИСЛЮВАЛЬНА МАТЕМАТИКА»

УДК 536.24
Денисенко О.І.[1]
[1] канд. техн. наук, доц. НУ «Запорізька політехніка»

3-D МОДЕЛЮВАННЯ ЕФЕКТИВНОСТІ СИСТЕМ ОПАЛЕННЯ ТА ЕНЕРГОЗБЕРІГАЮЧИХ ТЕХНОЛОГІЙ

Проблема енергозбереження при опалюванні житлових приміщень є досить актуальною у зв'язку зі значним подорожчанням енергоресурсів Використання спрощених балансових моделей для розрахунку ефективності опалювальних систем не дають змоги дослідити складні ефекти конвективного перемішування повітряних мас та визначити ділянки втрат теплової енергії.

Сучасні обчислювальні системи та пакети прикладних програм дозволяють в тривимірному наближенні моделювати складні ефекти процесів опалення житлових приміщень, досліджувати різні конфігурації розташування опалювальних елементів, аналізувати використання енергозберігаючих технологій, теплоізоляційних матеріалів та ін. Для видачі достовірних рекомендацій по економії енергоресурсів необхідно мати детальний розв'язок сполученої задачі конвективного теплопереносу в середині приміщення та теплопровідності в стінових конструкціях.

В роботі розглядається тривимірна модель конвективного опалення житлового приміщення. Математична модель включає рівняння гідродинаміки на основі к-ε моделі турбулентності та рівняння теплопровідності. Чисельна реалізація моделі виконувалась методом скінчених елементів за допомогою програмного комплексу COMSOL Multiphysics. Проведено серію чисельних експериментів для моделювання конвекції в приміщеннях з джерелами тепла. Визначено найбільш вразливі зони, які потребують додаткової теплоізоляції. Проведена оцінка ефективності заходів по додатковій теплоізоляції зовнішніх поверхонь житлових приміщень. Досліджено розподіл температур та характер конвективних потоків в приміщеннях в залежності від розташування систем опалення, наявності вентиляції, геометрії приміщень, зовнішніх умов, фізичних параметрів стінових панелей та ін. Застосування чисельних експериментів дозволяє видати рекомендації стосовно розташування опалювальних елементів, їх кількості та потужності, доцільності використання теплоізоляційних матеріалів та їх характеристик.

На рис. 1, 2 представлено приклад розрахунку теплового стану типового приміщення з одним вікном і одним опалювальним елементом. Розрахунки дозволили візуалізувати складний характер конвективних потоків та

визначити температуру в різних зонах в залежності від зовнішніх факторів та геометрії приміщення.

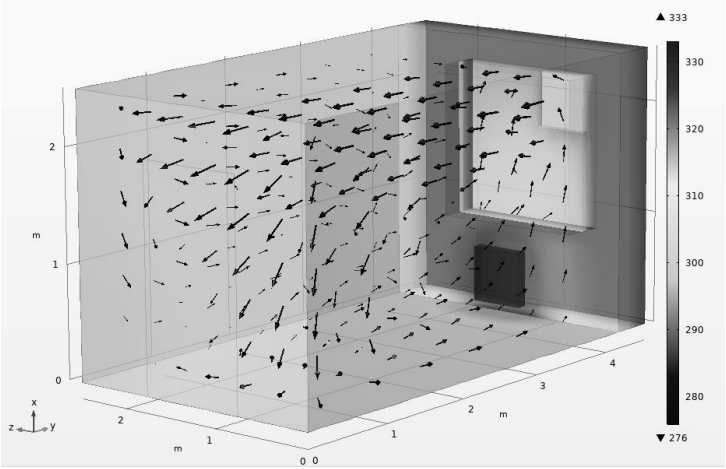


Рисунок 1– Розподіл температури

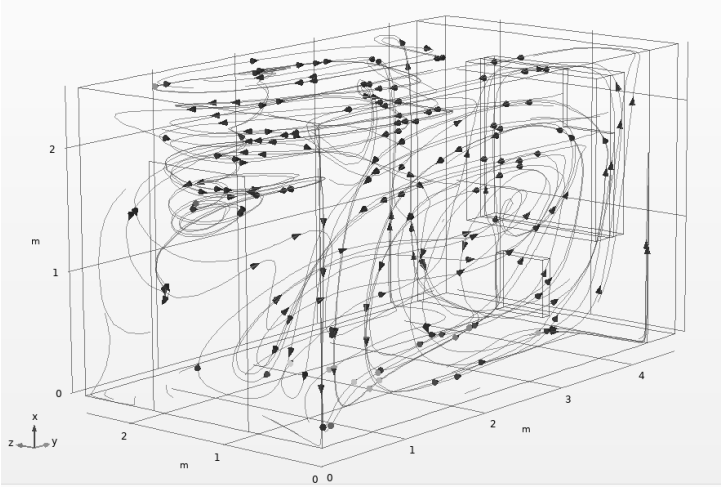


Рисунок 2 – Лінії току конвективного перемішування

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Коваленко, А.В. Математическое моделирование физико-химических процессов в среде COMSOL Multiphysics 5.2 [Текст]: учебное пособие / А.В. Коваленко, А.М. Узденова, М.Х. Уртенов, В.В. Никоненко – С-Петербург. : Лань, Планета музіки, 2017 – 238 с.

УДК 548.53:536.4
Пархоменко Л.А.
канд. физ.-мат. наук, доц. НУ «Запорожская политехника»

АНАЛИЗ ЗАКОНОМЕРНОСТИ ФОРМИРОВАНИЯ ПОРИСТОЙ СТРУКТУРЫ В ИОННО-ОБЛУЧЕННОМ МЕТАЛЛЕ

Исследование заключается в сопоставлении и анализе экспериментальных данных о специфических особенностях образования, пространственного распределения, эволюции и миграции на различные стоки радиационно-индуцированных нанопор в ионно-облученных ГЦК-металлах. Это металлы, имеющие гранецентрированную кубическую решетку (ГЦК). Подобную решетку имеют железо, алюминий, медь, никель, свинец и др. металлы. В процессе изучения нанопористой структуры, образующейся в поверхностных слоях металлов при обработке их различными видами радиационного воздействия, сотрудниками ИЭФ УрО РАН получен богатый экспериментальный материал, установлен ряд эмпирических закономерностей. Вместе с тем, физический смысл наблюдаемых закономерностей на данный момент не ясен, отсутствует представление о динамике возникновения и эволюции пор при облучении. По этой причине возникла необходимость создать математическую модель физических процессов, происходящих на атомном уровне в кристаллических телах. Экспериментальные данные о конфигурации и пространственном распределении радиационных повреждений на поверхности и в объеме облученных материалов являются важными исходными данными для формирования представлений о реальной структуре металлов, подвергнутых радиационному воздействию. Экспериментальные данные получаются методом полевой ионной микроскопии, которая позволяет идентифицировать радиационно-индуцированные поры сколь угодно малых размеров, установить их геометрию, размеры, и их распределение в объеме облученного металла. Полевая ионная микроскопия является одной из самых структурно-чувствительных методик, обладающей не только высокой разрешающей способностью (0,2-0,3 нм), но и возможностью анализировать объем объекта исследования путем контролируемого удаления поверхностных атомов электрическим полем.

Основной характеристикой пористости в материале является функция плотности распределения пор по размерам, представляемая в виде гистограммы – образа. Подверженная трансформации в зависимости от предыстории и способа получения массива пор в материале, характера их саморазвития такая функция содержит важную информацию о физической природе процессов в системе.

Создана математическая модель для исследования объемного распределения пор в ионно-облученном материале. Процессы функционирования элементов исследуемой сложной системы записываются в виде интегро-дифференциальных уравнений, в качестве численных параметров которых закладываются физически значимые характеристики исследуемого вещества и взаимодействующего с ним излучения. Решается система интегро-дифференциальных уравнений: уравнение непрерывности, уравнение движения размеров и закон сохранения общей массы вещества. Получена функция распределения пор по размерам. Определяются численные значения параметров исследуемой системы: верхней границы относительных размеров пор, положение максимума и положение точек перегиба на кривых распределения, величины модального радиуса и др. Полученные теоретические расчеты сопоставляются с экспериментом. Результаты сопоставления расчетных и экспериментальных данных позволяют установить механизм формирования пор при радиационном воздействии на кристаллическую решетку. Выявление характерных структурных и кинетических свойств исследуемой системы производится путем идентификационного анализа простых и кумулятивных распределений пор по размерам, их моментов и соотношений между ними с привлечением средств ПК. Выявление природы внутрисистемных процессов в массиве пор путем сопоставления экспериментального распределения – образа с его теоретическим подобием – составляет сущность и предназначение идентификационного анализа.

Предлагаемый анализ изучения газонаполненных пор, образовавшихся после ионного облучения, позволяет выявить особенности протекания процессов образования дефектов кристаллической решетки при радиационном воздействии. Результат данной работы заключается в получении новых знаний о механизме радиационного воздействия на структуру твердых тел. Эти знания будут иметь большое значение для понимания свойств облучаемых материалов, связывая макроскопический ход процессов при облучении с динамикой движения и комбинаторикой единичных точечных дефектов.

Совпадение характеристик экспериментального распределения (гистограммы) с теоретическим может служить подтверждением реальности подобранного механизма, а обнаруженные при их сравнении различия в сходстве и сходство в различиях – источником научной информации о процессах в системе.

UDC 538.971
Shyrokorad D.V.[1], Kornich G.V.[2]
[1] Snr lecturer, NU “Zaporizhzhia Polytechnics”
[2] head of department, NU “Zaporizhzhia Polytechnics”

KINETICS OF JANUS-LIKE ATOMIC CLUSTERS UNDER LOW ENERGY BOMBARDMENT

The evolution of free 390-atom Janus-like Ni-Al clusters along 100 and 500 ps under the bombardment by Ar_1 and Ar_{13} particles with energies up to 1.0 keV is simulated by the classical method of molecular dynamics. The initial metastable Janus-like clusters have two monocomponent parts, equal in number of atoms, with a small spatial overlap. After the impact of projectiles and rapid expansion, accompanied by spraying, the observed kinetics of Ar_1 and Ar_{13} cases are critically different; Ni-Al clusters gradually evolve due to exothermic mixing of components, with a tendency to surface segregation of Al atoms.

The melting points of metal nanoclusters are usually lower than the melting points of the same macroscopic materials, moreover, they can change nonmonotonically with decreasing of cluster size, especially below two hundred atoms [1]. The values of AEI (atomic equivalence indices), which are the sums of the modulus of difference of atomic radii-vectors, were calculated in our study using the Berendsen method depending on the temperature of the cluster. As expected, the results show that the melting process, that destroys the atomic structure of the cluster, begins at its surface at lower temperatures and reaches the inner regions at about 870 and 550 K for monocomponent Ni and Al clusters consisting of 195 atoms each.

The difference in the melting points of the surface and inner layers in our calculations reaches more than 200 K, especially for the Ni cluster. Simulation of Ar_1 particle bombardment showed that the temperatures of the Ni-Al cluster after 100 ps of evolution are higher than the melting points of the corresponding monocomponent Al and Ni clusters. Moreover, the temperature of the cluster is higher than the melting points of the clusters Al and Ni after 5 and 15 ps at all impact energies. Therefore, in many cases of bombardment, the aluminum part of the Janus-like cluster demonstrates faster transition to a disordered state. In these cases, over a period of time, part of the nickel is partially covered by more mobile aluminum atoms. Thus, for some period of time there is an unstable for this atomic system form of distribution of components "ball-and-cup" [2].

In addition, the interaction of the target cluster with the Ar projectile leads to the generation of radiation defects and the destruction of the ordered atomic structure in the cluster under strong nonequilibrium conditions for several initial picoseconds, which also contributes to the formation of the melt. Thus, in these simulations, an unstructured form was predicted, which can be interpreted as an

almost liquid state of the Ni-Al cluster after interaction with the Ar_1 projectile. Note, that the transition to an unstructured form of a cluster also leads to an increase in its potential energy.

The cases with Ar_{13} particles are dramatically different. The effects caused by the atomization of most of the atoms lead to a significant gradual increase in potential energy over about 10 ps, excluding the case of Ar_{13} with an energy of 100 eV. Also, a pronounced effect of increasing the temperature of the cluster with the size of the bombardment particle was observed throughout the simulation time. It also helps to increase potential energy. In the case of Ar_{13} bombardment, the cluster temperatures after 15 ps evolution are higher than the boiling points of the Al cluster, and comparable to the boiling point of the Ni cluster (~ 2520 K and 3200 K, respectively), except for low energies.

This is the main reason for the long high sputtering yield after the end of the collision stage, especially for the component Al. This can be interpreted as intense evaporation, which slightly lowers the temperature of the cluster for long periods and is accompanied by a slow increase in potential energy. It is obvious, that under such conditions the substance of the clusters is in a liquid state with an intense transition to the gaseous state. In the case of low energy effects, the sputtering yield and cluster temperature are minimal. As a result, in this case, the potential energy of the cluster after the evolution of 100 ps is lower than the initial one, similarly to the cases of Ar_1 bombardment.

REFERENCES

1. Hamid, I. Molecular dynamical simulations of melting behaviors of metal clusters [Текст] / I. Hamid, M. Fang, H. Duan // AIP Adv. – 2015. – Vol. 5. – p. 047129. doi:10.1063/1.4918770.

2. Ferrando, R. Symmetry breaking and morphological instabilities in core-shell metallic nanoparticles [Текст] / R. Ferrando // J. Phys. Condens. Matter. – 2015. – Vol. 27. – p. 13003. doi:10.1088/0953-8984/27/1/013003.

УДК 004.42
Супрун О.С.[1], Кондратов Д.О.[2], Сластніков О.П.[3], Бакурова А.В.[4], Широкорад Д.В.[5]
[1] студ. гр. КНТ-810м НУ«Запорізька політехніка»
[2] студ. гр. КНТ-818 СП НУ«Запорізька політехніка»
[3] студ. гр. КНТ-817 НУ«Запорізька політехніка»
[4] проф. НУ «Запорізька політехніка»
[5] ст. викл. НУ «Запорізька політехніка»

ОРГАНІЗАЦІЯ ДОКУМЕНТООБІГУ СТРУКТУРОВАНОГО ПІДПРИЄМСТВА

В наш час є необхідним скорочення часу, що витрачається на вирішення задач документообігу. Саме електронний тип документообігу відкриває можливості для вдосконалення, довготривалого збереження, управління електронною базою документів, враховуючи процедури списання та знищення документів. Розробки програм для поліпшення документообігу активно здійснюються передовими компаніями світу наприклад Microsoft та їх «Microsoft SharePoint Portal Server» або HBS з їх «Docs Open», що безперечно доводить актуальність розробки[1].

При виконанні роботи були проаналізовані джерела з теми документообігу, в основу була взята інформація з посібника [2] та книги [3]. У посібнику у логічній послідовності базові поняття основи організації електронного документообігу, викладено загальні принципи організації документообігу в організаціях, основні підходи до впровадження систем електронного документообігу, подано принципи побудови систем електронного документообігу, сформульовано вимоги до їх функціональних можливостей, розглянуто проблеми, які при цьому виникають. У книзі же комплексно розглянуто теоретичні та практичні питання застосування електронного документообігу в сфері управління, включаючи інформаційно-технологічні, комунікаційні, документоведческой, правові, організаційно-управлінські аспекти.

В нашому випадку підприємство зіткнулось з проблемою – з причини поточної завантаженості майстри цехів або матеріально-відповідальні особи не оформлюють в строк необхідні документи, що вказані в порядках обробки первісних документів. Необхідно було знайти шляхи вирішення цієї проблеми. Як один з підходів було запропоновано створити юзер-френдлі програму по типу майстер формування звітів. Основними критеріями оптимізації при створенні програми були максимальна простота та зручність інтерфейсу для користувача. Підприємством було надано документацію, необхідну для розуміння його структури роботи з документами. А саме: основні типи відомостей, порядок обліку та оформлення первісних

документів за необоротними активами, порядок обліку та оформлення первісних документів по ремонтам та порядок обліку закуповуваних від постачальників сировини, інших товарно-матеріальних цінностей і послуг.

При брейнштормі командою був визначений функціонал програмного забезпечення (ПЗ). Програма повинна вміти: створювати відомості у форматі pdf, редагувати вже створенні відомості, видаляти не потрібні відомості та обмежувати доступ до деяких функцій некомпетентних працівників. Для зберігання потрібної інформації та відомостей було вирішено створити окрему базу даних (БД) на основі СУБД Oracle Database, так як вона вже поширено використовувалась на підприємстві.

На основну сторінку були розміщені основні доступні дії:

«Редагувати відомість»- змінити вже створену та збережену у БД відомість.

«Профіль» - користувач вводить свій Id та пароль та отримує доступ до основної інформації про себе та можливість подальшого використання доступного для нього функціоналу.

«Створити відомість» - користувач обирає вже з доступної інформації, збереженої в БД, або вводить свою та по бажанню може додати її в БД. На вкладці «Дати» - користувач встановлює заплановані та фактичні дати робіт. «Перелік робіт та деталей»- користувач обирає тип роботи або деталь, яка використовувалась в процесі, та додає її в створювану відомість. При виборі деталі деякі з полей блокуються. Є можливість додати декілька робіт/деталей. «Корисні залишки»- сторінка створена для випадків, коли в процесі роботи залишаються деталі, які можна використовувати надалі.

Після введення та вибору всієї потрібної інформації при натисканні на «Сформувати» створюється відомість у форматі pdf. Кнопка «Зберегти» вносить створену відомість у БД для подальшого використання.

При впровадженні розробленої ПЗ планується економія часу робітників на підготовку звітності, економія часу на навчання нових робітників роботі з документами, збільшення точності і своєчасності документообігу на підприємстві. Також невід'ємним плюсом програми перед аналогами є те, що вона зроблена спеціально для даного підприємства і при потребі може бути вдосконалена та оновлена під нові стандарти або документацію.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1.Асєєв Г. Управління сучасним документообігом: теорія, структура, методи / [Текст] Г.Асєєв // Вісник Книжкової палати. – 2004. - № 4. - С. 32-36.

2. Матвієнко, О. Основи організації електронного документообігу [Текст] : Навчальний посібник / О. Матвієнко, М. Цивін. – К. : Центр учбової літератури, 2008.-112с.

3. Бобильова, М. П. Управлінський документообіг: від паперового до електронного [Текст] / М. П. Бобильова. – М .: Термика, 2010. – 172 с.

УДК 004.4
Андрюшин А.О.[1], Бурковський Є.О.[1]
[1]студ. гр. КНТ-810сп НУ «Запорізька політехніка»

ЕЛЕКТРОННИЙ СЕРВІС «МЕДИЧНА РЕФОРМА ДЛЯ ЛЮДЕЙ»

З 1 січня 2018 в Україні вступив в дію Закон «Про державні фінансові гарантії надання медичних послуг та лікарських засобів», що ознаменував собою початок медичної реформи. Другий етап впровадження реформи припадає на вторинну ланку медичної допомоги [1]. Не зважаючи на ряд прогресивних положень медичної реформи, вона зустріла неоднозначну реакцію з боку населення та працівників системи охорони здоров'я. Одним з можливих факторів недостатньої успішності медичної реформи та невідповідності її очікуваних та реальних результатів є недосконала діджиталізація. Тим не менш, в процесі реалізації медичної реформи генерується значний об'єм відкритих даних, які можуть бути використані для створення локальних або веб ресурсів, що можуть покращити ефективність впровадження реформи, спростити взаємодію населення із закладами охорони здоров'я або взаємодію цих закладів між собою.

В роботі розглядаються перші етапи розробки такого сервісу, що являє собою інтерактивну мапу з інформацією про перебіг всеукраїнської медичної реформи в місті Запоріжжя, а також з використанням відкритих даних, що розміщені на Єдиному державному веб-порталі відкритих даних [2]. Загальною задачею проєкту було зробити доступною інформацію про надання спеціалізованої медичної допомоги закладами вторинної ланки медичної допомоги в місті Запоріжжі, а також панель управління до основної частини проєкта.

Програмна реалізація сервісу по відображенню основної інформації про ліки була здійснена за допомогою прогресивного фреймворку Vue.js та відкритої бібліотеки Leaflet, яка має реалізацію на цьому фреймворці (vue-leaflet), що надає потужний інструментарій для створення інтерактивних мап з підтримкою адаптивного відображення на мобільних пристроях [3]. Мапа доступна для перегляду у веб-браузері. На поточному етапі розробки мапа надає актуальну інформацію, у вигляді списку, щодо наявності різних типів препаратів у різних міських лікарнях міста Запоріжжя (рис. 1). У боковій панелі програми користувач може переглянути коли останній раз оновлювалися дані (рис. 2), ця дата береться з серверу, кожен раз коли на ресурсі «data.gov.ua» викладається нова відомість, парсер на сервері скачує

всі нові данні у форматі «.xls» та оновлює інформацію на основній сторінці сайту, також він оновлює файл з інформацією про останню дату оновлення, таким чином коли парсер оновлю дані на сервері користувач бачить уже нові дані станом на певну дату.

Рисунок 1 – Список ліків на певну дату

Після того як користувач натиснув на ліки що його заінтересували, йому відображаються маркери в яких лікарнях цей препарат є.

Розроблений ресурс муже бути використаний як населенням з метою визначення доцільності звернення у ту чи іншу лікарню, так і безпосередньо лікарнями за умови реалізації механізму нагального обміну лікарськими засобами. Також сервіс може стати в нагоді місцевим ЗМІ та фармацевтичним компаніям.

Першочерговий напрям подальшого розвитку веб-сервісу – додавання інформації про весь спектр лікарських засобів, а також про поточний відсоток зайнятих ліжок, що стає особливо актуальним в умовах пандемії. Для ефективної роботи мапи очікується співпраця з Департаментом охорони здоров'я щодо управління регулярним оновленням відповідних даних. Існує можливість додавання аналітичного модулю до порталу Департаменту http://health.zp.ua/, задачею якого буде прогнозування використання лікарських засобів та передчасне попередження про їх можливу нестачу, а також інформація про зайнятість ліжок. В подальшому планується розширювати географію сервісу на територію Запорізької області.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Міністерство охорони здоров'я [Электронный ресурс]. – Режим доступу: https://moz.gov.ua/.

2. Відомості про лікарські засоби, препарати, придбані за бюджетні кошти, відомості про розподілення таких ліків між закладами охорони здоров'я та їх залишки в кожному з них [Электронный ресурс]. – Режим доступу: https://data.gov.ua/dataset/cbda9040-432e-493f-9e17-e44b8c1d73d1.

3. Документація бібліотеки Leaflet [Электронный ресурс]. – Режим доступу: https://github.com/vue-leaflet/vue-leaflet.

УДК 04.94

Москалюк В.В. [1], Терещенко Е.В.[2]
[1] студ. гр. КНТ-818сп НУ «Запорізька політехніка»
[2] доц. НУ «Запорізька політехніка»

ПОБУДОВА НЕЧІТКОЇ ПРОДУКЦІЙНОЇ СИСТЕМИ «ОСОБА ВИННОГО»

Врахування особи винного є обов'язковим для суду в кожному випадку призначення покарання, що зазначено п.3 ч.1 ст.65 Кримінального кодексу України (КК) [1]. Загальна модель СППР у суді розроблено у [2, 3]:

$$(Fine, Years, RF, Public\ Works, Condition) = F(Severity, Lawyer, Personality, Plaintiff, Mitigation, Burden), \quad (1)$$

де F – відповідний алгоритм нечіткого виведення.

Особу винного описано вхідною змінною *Personality*. Опис особи винного є вербальною, якісною, неметричною, слабоструктурованою інформацією. Базуючись на онтологічному підході побудовано модель змінної *Personality* у вигляді орграфу було сформовано колекцію документів з 45 судових вироків за ч.2-3ст.185 КК України [1]. Вироки було отримано з Єдиного реєстру згідно правил користування цим електронним ресурсом [4].

На основі аналізу документів було виокремлено категорії термінів, слів, що відповідають термам концепту онтології «особа винного».

Множина ознак для концепту *Personality* має вигляд: {працевлаштування, сімейний стан, проживання, реєстрація, характеристика, перебування на обліку, наявність зв'язків, кримінальна відповідальність, судимість}, базуючись на яких необхідно визначити оцінку кількісну або якісну як значення координати вхідного вектора СППР у суді. Одним з підходів до визначення особи винного є аналогія за формулою (1) – модель підсистеми СППР для визначення оцінки особи винного:

$$(Позитивна, Негативна, Нейтральна) = F(працевлаштування, сімейний\ стан, проживання, реєстрація, характеристика, перебування\ на\ обліку, наявність\ зв'язків, кримінальна\ відповідальність, судимість), \quad (2)$$

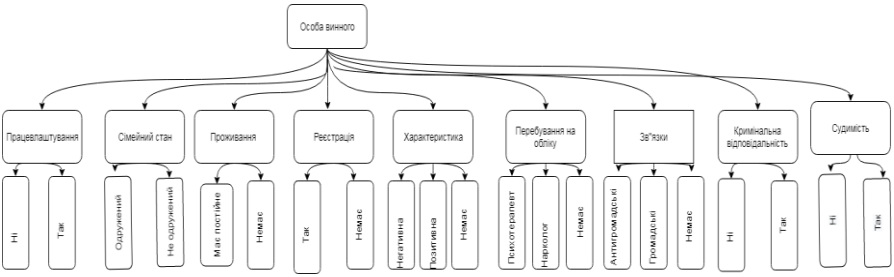


Рисунок 1 - Модель вербальної інформації «Особа винного»

Побудову системи почнемо з визначення відповідних лінгвістичних змінних, що потребує визначення множини носія, функцій належності. Для налаштування такої системи необхідно визначитися з векторами вхідних й вихідних даних, що було зроблено за колекцією відібраних судових вироків відповідно до методу, що запропоновано [3]. Далі було застосовано можливості ANFIS-редактора пакету MATLAB, який дозволяє автоматично синтезувати з експериментальних даних один з різновидів систем нечіткого логічного виведення типу Сугено у вигляді нейро-нечіткої мережі. Найбільш цінним є те, що функції приналежності синтезованих систем побудовано так, щоб мінімізувати відхилення між результатами нечіткого моделювання й даними, які використовували для навчання. Далі було здійснено фазифікацію отриманих даних для застосування оцінки на вхід системи (1), що розширює можливості її налаштування.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Kriminalniy kodeks Ukrayiny. (In Ukrainian) [Электронный ресурс]. – Режим доступу:http://zakon.rada.gov.ua/. Last access 02/28/20.

2. Bakurova, A. Modeling of decision making ontology [Электронный ресурс] / A. Bakurova, E. Tereschenko, Y. Filei, M. Pasichnyk, H. Ropalo // Proceedings of the 3rd international conference on computational linguistics and intelligent systems (colins-2019) – 2019. – Vol.1. – Режим доступу: http://ceur-ws.org/Vol-2362/paper18.pdf

3. Bakurova, A. Fuzzy Modeling of Verbal Information for Production Systems [Текст] / A. Bakurova, M. Pasichnyk, E. Tereschenko // Innovative Technologies and Scientific Solutions for Industries. – 2019. – №4(10). – P.5-13.

DOI: https://doi.org/10.30837/2522-9818.2019.10.005.

4. Court decisions register of Ukraine [Электронный ресурс]. – Режим доступу: http://www.reyestr.court.gov.ua Last access 02/28/20

Наукове електронне видання
Можна використовувати в локальному та
мережному режимах

ТИЖДЕНЬ НАУКИ-2021. Факультет комп'ютерних наук і технологій

Збірник тез доповідей щорічної
науково-практичної конференції викладачів, науковців,
молодих учених, аспірантів та студентів
19–23 квітня 2021 року

Один електронний оптичний диск (DVD-ROM);
супровідна документація.
Тираж 100 прим. Зам. № 304

Видавець і виготовлювач
Національний університет «Запорізька політехніка»
Україна, 69063, м. Запоріжжя, вул. Жуковського, 64 Тел.:
(061) 769–82–96, 220–12–14

Свідоцтво суб'єкта видавничої справи ДК № 6952 від 22.10.2019.