Вінницький національний технічний університет
Факультет інформаційних технологій та комп’ютерної інженерії
Кафедра захисту інформації

Пояснювальна записка

до магістерської кваліфікаційної роботи

на тему «Система виявлення фейкових облікових записів у соціальних мережах»

08-20.МКР.003.00.000 ПЗ

Виконав: студент 2 курсу, групи 1БС-18м
Спеціальність 125 Кібербезпека
ОПП Безпека інформаційних і комунікаційних систем

_____ Головенько В. О.

Керівник: к. т. н., доц. каф. ЗІ

_____ Войтович О. П.

Рецензент: к. т. н., проф., доц. кафедри ОТ

_____ Азарова А. О.

Вінниця - 2019 року

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра захисту інформації
Освітньо-кваліфікаційний рівень магістр
Спеціальність 125 Кібербезпека
ОПП Безпека інформаційних і комунікаційних систем

ЗАТВЕРДЖУЮ

Завідувач кафедри ЗІ, д. т. н., проф.

_____ В. А. Лужецький

_____ _____ 2019 року

З А В Д А Н Н Я

НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Головеньку Віталію Олександровичу

1. Тема роботи: «Система виявлення фейкових облікових записів у соціальних мережах» керівник роботи: Войтович Олеся Петрівна, к. т. н., доц. каф. ЗІ,

затверджена наказом ректора ВНТУ № 254 від 02.10.2019 р.

2. Строк подання студентом роботи _____ 2019 р.

3. Вихідні дані до роботи:

- соціальна мережа – Facebook;
- аналіз за вмістом облікового запису;
- виявлення метрик;
- мова програмування Python.

4. Зміст розрахунково-пояснювальної: Вступ. Аналіз інформаційних джерел. Розробка моделей для виявлення фейкових облікових записів. Експериментальні дослідження. Економічна частина. Висновки. Перелік інформаційних джерел. Додатки.

5. Перелік графічного матеріалу.

Критерії встановлення балів відповідно до значень метрик (плакат, А4). Структура облікового запису (плакат, А4). Структурна модель ознак фейковості у категоріях «Лайки» та «Персональна інформація про користувача» (плакат, А4). Структурна модель ознак фейковості у категоріях «Статуси і пости», «Друзі» та «Фото» (плакат, А4). Архітектура програмного

засобу (плакат, А4). UML-діаграма класів програмного засобу (плакат, А4). Інтерфейс програмного засобу (плакат, А4).

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	Войтович О. П., к. т. н., доц. каф. ЗІ		
2	Войтович О. П., к. т. н., доц. каф. ЗІ		
3	Войтович О. П., к. т. н., доц. каф. ЗІ		
4	Мацкевічус С. С., ст. викл. каф. ЕПВМ		

7. Дата видачі завдання _____ 2019 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз завдання. Вступ	01.09.2019 – 04.09.2019	
2	Аналіз літературних джерел за напрямком магістерської кваліфікаційної роботи	05.09.2019 – 15.09.2019	
3	Науково-технічне обґрунтування	16.09.2019 – 22.09.2019	
4	Розробка технічного завдання	23.09.2019 – 29.09.2019	
5	Розробка рішень	30.09.2019 – 12.10.2019	
6	Практична реалізація, моделювання, експериментування, результати	14.10.2019 – 10.11.2019	
7	Розробка розділу економічного обґрунтування доцільності розробки	11.11.2019 – 17.11.2019	
8	Аналіз виконання ТЗ, висновки	18.11.2019 – 24.11.2019	
9	Оформлення пояснювальної записки	25.11.2019 – 30.11.2019	
10	Попередній захист та доопрацювання МКР	28.11.2019 – 01.12.2019	
11	Перевірка магістерської роботи на наявність плагіату	02.12.2019 – 10.12.2019	
12	Представлення МКР до захисту, рецензування	11.12.2019 – 14.12.2019	
13	Захист МКР	16.12.2019 – 20.12.2019	

Студент _____ Головенько В. О.

Керівник роботи _____ Войтович О. П.

АНОТАЦІЯ

У магістерській кваліфікаційній роботі розглянуто та проаналізовано основні методи аналізу соціальних мереж. Проаналізовано структуру облікових записів у соціальних мережах. Запропоновано метрики ознак фейкових облікових записів у соціальних мережах у категоріях «Лайки», «Друзі», «Пости та статуси», «Персональна інформація про користувача» та «Фото», а також їх можливі параметри та вплив на фейковість облікового запису. Проаналізовано існуючі системи підтримки прийняття рішень. Розроблено систему підтримки прийняття рішень на основі нейронної мережі. Розроблені структурні моделі, що дозволяють виявити фейкові облікові записи у соціальних мережах. На основі запропонованих структурних моделей ознак фейковості та використанні нейронної мережі розроблено засіб підтримки прийняття рішень для виявлення фейкових облікових записів у соціальній мережі «Facebook» мовою програмування Python. Проведено ряд експериментальних досліджень для перевірки роботи програмного засобу шляхом аналізу облікових записів у соціальній мережі «Facebook».

ABSTRACT

In the master's qualification work the basic analysis methods of social networks were considered and analyzed. The structure of accounts on social networks was analyzed. The metrics for fake accounts on social networks in the categories «Likes», «Friends», «Posts and statuses», «Personal information about the user» and «Photos» were proposed, as well as possible parameters and their effect on fake account. Existing decision support systems were analyzed. A neural network decision support system was developed. Structural models to detect fake accounts on social networks were developed. Based on the proposed structural models of fake features and neural network, a decision support tool to detect fake accounts in the social networking site Facebook using Python programming language was developed. A number of experimental researches to test the performance of the software by analyzing accounts on the social network «Facebook» were carried out.

ЗМІСТ

ВСТУП

В теперішній час постійного вдосконалення інформаційних та комунікативних технологій, будь-який конфлікт має своє відображення в мережі Інтернет. Дуже часто таке мережеве відображення впливає на результат протиборства конкуруючих сторін. Активне залучення до віртуальної мережі багатомільйонної аудиторії дозволяє маніпулювати громадською думкою і суттєво впливати на процеси протиборчих сторін [1].

Розуміння значення Інтернету та соціальних мереж змушує передові держави вкладати величезні інвестиції в соціальні мережі, які нині стають не тільки засобом комунікації, а й ефективною політичною зброєю.

Сьогодні соціальні мережі використовуються у якості платформи для проведення спеціалізованих інформаційних дій, таких як інформаційно-психологічні операції, спрямовані на думки суспільства [2, 3]. Величезна кількість людей зі всього світу щодня використовують соціальні мережі як засіб для спілкування, джерело інформаційних новин, перегляд розважального контенту, проте є і інша частина людей, що за допомогою інформаційних вкидів маніпулюють індивідуальною та суспільною свідомістю [2]. Для такого і використовуються зазвичай фейкові облікові записи, на яких розміщена неправдива або взагалі відсутня інформація про користувача. Зазвичай фейкові облікові записи використовуються для прямої або опосередкованої зміни народної думки у різних формах прояву. Оскільки у соціальних мережах немає жорсткої цензури, а іноді вона і зовсім відсутня, це сприяє успішності усіх дій у віртуальному світі. Таким чином, всі ці факти вказують на те, що соціальні мережі перетворюються на своєрідний майданчик інформаційних протиборств.

Проте, формалізованих моделей, на базі яких можна розробити інструментарій для аналізу облікових записів щодо автентичності, не існує, тому дослідження в цьому напрямку є актуальними.

Метою магістерської кваліфікаційної роботи є покращення забезпечення кібербезпеки шляхом розробки метрик та системи підтримки прийняття рішень для визначення фейкових облікових записів у соціальних мережах.

Для досягнення мети необхідно вирішити такі задачі:

– проаналізувати існуючі підходи аналізу облікових записів;

– розробити структурні моделі ознак фейкових облікових записів;

– розробити систему підтримки прийняття рішень;

– розробити програмний засіб для автоматизації виявлення фейкових облікових записів у соціальній мережі «Facebook».

Наукова новизна полягає у вдосконаленні методу виявлення фейкових облікових записів у соціальних мережах шляхом розробки нових метрик ознак фейковості, що дозволяє збільшити достовірність прийняття рішення до 94 %.

Практична цінність полягає у розробці алгоритмів та програмного забезпечення виявлення фейкових облікових записів у соціальній мережі «Facebook», що дозволяє виявляти інформаційних агентів-ботів під час інформаційної війни.

Результати роботи обговорювались на конференціях:

– Шоста Міжнародна науково-практична конференція «Методи та засоби кодування, захисту й ущільнення інформації»;

– Міжнародна науково-практична конференція «Інформаційні технології та комп’ютерне моделювання»;

– III Міжнародна науково-практична конференція «Інформаційна безпека та комп'ютерні технології».

Опубліковано частину у колективній монографії [2], у статті «Вчені записки таврійського національного університету ім. В.І. Вернадського» [4], у матеріалах конференцій [5, 6, 7].

Отримано авторське свідоцтво на комп’ютерну програму «Засіб для виявлення фейкових облікових записів у соціальній мережі «Facebook»» [8].

1 АНАЛІЗ ІНФОРМАЦІЙНИХ ДЖЕРЕЛ

1.1 Науково-технічне обгрунтування розробки системи виявлення фейкових облікових записів у соціальних мережах

Фейковий обліковий запис (фейк) – це обліковий запис у соціальній мережі з неправдивою інформацією про користувача, власника даної сторінки.

Використовувати фейковий профіль можна як з легальних причин (наприклад для продажу товарів у соціальних мережах тощо), так і зі зловмисною метою (шахрайства, маніпуляції, заборонений контент тощо) [4].

Метою підроблених (фейкових) облікових записів в інформаційних війнах є введення в оману інших користувачів, маніпулювання їхньою поведінкою, причому Інтернет дає можливість здійснювати таке маніпулювання суспільством в цілому [5], і фейкові облікові записи, стали використовуватись як джерело інформаційно-психологічних операцій в інформаційній війні [6].

В задачі аналізу контенту в рамках протидії інформаційній війні важливо те, що певний обліковий запис розповсюджує певний контент (генерує репости, лайки, коментарі) за бажанням або за винагороду. Також існують спеціальні біржі облікових записів, де предметом продажу є готовий обліковий запис з потрібними замовнику параметрами. Виконавці створюють фейковий обліковий запис, наповнюють його контентом, заповнюють його, створюючи ілюзію великого числа друзів, високої активності, входження в потрібні групи тощо. Після цього налаштовують обліковий запис під вимоги замовника (соціальні особливості, переваги, стиль поведінки) і продають замовнику, який вже використовує такі фейкові облікові записи на свій розсуд [7].

Для прикладу, під час передвиборчої кампанії Дональда Трампа у США, російськи спецслужби використовували фейкові облікові записи в соціальних мережах для маніпулювання думкою виборців. Обвинувачені росіяни та їхні спільники створювали сотні облікових записів в соціальних мережах, які

представлялися справжніми та вигаданими американцями або американськими організаціями. Багато з таких облікових записів було закрито, але деякі залишилися в архівах. Непоодинокими були пости з критикою мусульман, пости на болючі теми на кшталт імміграції або Black Lives Matter, що закликають меншості не голосувати або розповідають про шахрайство демократів. Деякі облікові записи фокусувалися на географічній самоідентифікації користувачів. Для просування постів такого типу використовувалися різноманітні вірусні хештеги [8].

Серед прикладів розповсюдженого використання фейкових облікових записів є розслідування New York Times, яке довело, що компанії, які продають підписників у соціальних мережах, не тільки створюють ботів, але й копіюють облікові записи реальних користувачів. Іноді такі двійники з'являються у профілів, на які давно не заходив їх власник (особливо для Twitter), а інколи – навіть у активних облікових записів. Зловмисники непомітно змінюють одну-дві літери в імені, накладають різні фільтри на аватар (для того, щоб обманути технологію розпізнавання обличчя) і видають фейкові облікові записи за справжніх користувачів, отримуючи за це кошти від людей, для яких кількість підписників у соціальній мережі є важливою [9].

Продавцем фейкових облікових записів є і компанія Devumi, що знаходиться у США, яка заробляє на тіньовому ринку чималі гроші розповсюджуючи за певну плату сотні тисяч облікових записів. Звичайно, продаж підписників у соціальних мережах згідно з політиками безпеки соціальних мереж вважається шахрайством і карається з боку їх адміністрацій шляхом блокувань таких облікових записів. Проте, Devumi активно співпрацює з різними бізнесменами та знаменитостями, які бажають підвищити свою популярність. Команда газети New York Times з'ясувала, що Devumi налічує у своїй базі 3,5 млн автоматизованих облікових записів, які компанія багаторазово продає своїм клієнтам. Вони надали лідерам думок (так званим інформаційним центрам) понад двісті мільйонів підписників у Twitter.

Проаналізувавши інформацію про діяльність компанії Devumi, що змогла знайти редакція газети, журналісти виявили щонайменше 55 000 фейкових облікових записів Twitter, які дублювали інформацію реальних користувачів [9].

Фейкові облікові записи, а також їх продавці стають все чисельнішими у Facebook та Twitter. Раніше створені для спілкування та перегляду фотографій і відео, сьогодні соціальні мережі використовуються для створення ілюзій про те, що хтось є відомішим, більш значущим та цікавішим, аніж він є насправді. Найцікавіше те, що у якості робочої сили вступають самі боти. За їх допомогою відбувається в плив на міжнародні відносини, політику, розпалюється расова та міжнаціональна ворожнеча, розвивається булінг. За підрахунками (хоча і неофіційними), близько 48 млн. активних облікових записів у Twitter (а це становить близько 15%) — це автоматизовані облікові записи, що були створені спеціально для заробітку на чужих амбіціях. У листопаді 2017 року Facebook зрозуміла, що неправильно підрахувала кількість фейкових облікових записів: насправді їх вдвічі більше, а саме – понад 60 млн. І, не зважаючи на те, що керівники соціальних мереж висловлюють «глибоке занепокоєння» цією ситуацією, бот- та фейк-індустрія розвивається шаленими темпами, комерціалізується та занурюється у тіньову економіку [9].

Команда американської газети New York Times проаналізувала ділову та судову документації, у яких зафіксовано більш як 200 000 покупців облікових записів, серед яких відзначаються зірки телешоу, професійні спортсмени, коміки, лектори TED, моделі та навіть священнослужителі. Найчастіше клієнти купували підписників для себе ж, інколи купували ботів для когось іншого. Бувало, що в гру вступали цілі PR-агенції [10].

Також, закордонні клієнти компанії Devumi використовують їх фейкові облікові записи у своїх цілях. Державна агенція новин Китайської Народної Республіки під назвою Xinhua заплатила за сотні тисяч підписників та перепостів у Twitter для того, щоб поширювати пропаганду офіційної політики комуністичної партії Китаю закордоном [10].

На початку вересня 2017 року Facebook заявила про те, що російська компанія, яка створила величезну кількість ботів і фейкових облікових записів всередині соціальної мережі, купила рекламні оголошення загальною вартістю $ 100 тис, які були використані для втручання Кремля в результат президентських виборів у США. Повідомляється, що покупцем є так звана «фабрика тролів» в Ольгіно, яка є основним інструментом російської пропаганди. За даними Facebook, рекламна кампанія стартувала ще влітку 2015 року. Деякі з оголошень стосувалися безпосередньо кандидатів в президенти Дональда Трампа і Гіларі Клінтон, а інші зачіпали такі теми, як расова дискримінація та права ЛГБТ-меншин. Така тактика також використовувалася і в Twitter - за допомогою спеціальних тем Кремль намагався створити протиборчі табори всередині американського народу, щоб зруйнувати віру в демократію. В цілому Facebook виявив близько 3 тис. підозрілих рекламних оголошень в період з червня 2015 по травень 2017 року. За словами начальника служби безпеки Facebook Алекса Стамос, ця реклама була прив'язана до 470 фейкових облікових записів, які були пов'язані один з одним і, швидше за все, велися з території Росії. Крім Facebook, до такої фейкової кампанії також було залучено і Twitter, на просторах якої також були виявлені сліди російської пропаганди. Завдяки мережі, яка об'єднує сотні тисяч фейкових облікових записів, у Twitter розгорнулася повномасштабна кампанія по автоматичному розсиланню повідомлень, що містять в собі фейкові новини і статті, що виставляють Гіларі Клінтон в негативному світлі. На відміну від Facebook, в Twitter необов'язково реєструватися під своїм справжнім ім'ям. Крім того, автоматична розсилка заохочується керівництвом компанії, яке вважає її ефективним інструментом для спілкування з клієнтами [11].

Фейковому обліковому запису у Facebook на ім'я Беатріс Бустар вдалося зібрати аудиторію з кількох сотень тисяч користувачів. Вона крала дані хворих людей або людей з інвалідністю, щоб домогтися цього. Вона регулярно постила фото облисілих людей, людей з ампутацією, і просила написати «амінь» в

коментарях. Подібні пости неминуче збирали тисячі лайків, коментарів і репостів. Не дивно, що їй вдалося зібрати базу фанатів з 671 000 користувачів - це набагато більше, ніж у провідних канадських ЗМІ на кшталт National Post або Toronto Star, і майже стільки ж, скільки у Globe і Mail. Нескладно зрозуміти, що фото - вкрадені, і що люди на фотографіях не хворі на рак або ж живуть цілком щасливим життям, незважаючи на інвалідність. Одного разу Беатріс перестала постити фото хворих людей, замість цього на сторінці почали з'являтися фото гарних, але дуже юних дівчат. Всі фото супроводжувалися однотипними повідомленнями: «Вона шукає хлопця. Напишіть в коментарі і вона відповість вам в особисті повідомлення. Не соромтеся, додавайте її!». Її друзі також були фейковими обліковими записами, прикрашеними вкраденими фотографіями красивих жінок. Кожен обліковий запис у свою чергу передруковував ці фото. Усі фейкові облікові записи діяли за однією формулою: публікація вкраденої фото гарнох дівчини, заохочення чоловіків написати коментар, тегування інших фейків. Це все робилося для нарощування аудиторії. Було досліджено, що всі фотографії належали різним обліковим записам, і швидше за все, були вкрадені з єдиною метою - залучати чоловіків [12].

Шахраї виманюють у користувачів соціальних мереж криптовалюту, представляючись головою компанії Space X Ілоном Маском або президентом США Дональдом Трампом. Зловмисники використовують фейкові облікові записи на кшталт @Eilon_Musk, @ElonMuski, @EloonMusk, @Elonn_Musk і @Alon_Musk, які з написання легко переплутати зі справжнім мікроблогом глави компанії Tesla. З цих профілів невідомі публікують повідомлення з проханнями перевести незначні суми в обмін на велику винагороду. Найчастіше своє повідомлення вони пишуть прямо в коментарях до публікацій в офіційних облікових записах відомих осіб та чиновників. Так, @DoonaldTrump65 запропонував першим 250 користувачам соціальної мережі Twitter переказати на вказану у твіті адресу 0,2 біткоіна, отримавши натомість 250 одиниць криптовалюти [13].

Розслідування показують, наскільки глобальною є фейкова сторона соціальних мереж. Не лише шахраї, а й люди з хорошою репутацією купують фейкові облікові записи. Всі користувачі, так чи інакше, приймають участь у розповсюдженні та розвиток соціальних мереж, при цьому їх вплив на економіку, життя та розвиток суспільства постійно зростає.

1.2 Аналіз структури соціальних мереж

Для того, щоб почати дослідження фейкових облікових записів, а також способи їх виявлення, необхідно дослідити середовище їх існування, а саме соціальні мережі.

Соціальна мережа – це соціальна структура у мережі, що складається з множини агентів (суб'єктів – індивідів, спільнот, груп індивідів чи організацій), а також множині зв'язкім між агентами (наприклад, спілкування, дружба, знайомства) [7]. Соціальна мережа представляється у вигляді графу G(N, E), де N = {1, 2, …, n} – кінцева множина агентів та E – множина ребер, що відображає взаємодію та зв'язки між агентами.

Серед найвідоміших соціальних мереж виділяють Facebook, WhatsApp, QQ, WeChat, QZone, Tumblr, Instagram, Twitter, Baidu Tieba, Skype та інші. Найбільша кількість активних користувачів знаходиться у соціальній мережі Facebook і становить понад 1.59 млрд осіб (табл. 1.1). Для проведення досліджень соціальних думок населення різних країн на певні події, аналізу вподобань користувачів для формування реклами та навіть проведення інформаційних війн ця соціальна мережа є ідеальною платформою. Facebook є найбільшою у світі соціальною мережею і вона була чи не першою соціальною мережею, що здолала позначку в 1 мільярд облікових записів користувачів. Окрім можливості спілкуватися з друзями та родичами, користувачі також можуть отримати доступ до різних додатків Facebook, для продажів в Інтернеті, і навіть для продажів на ринок чи просування власного бізнесу, бренду та продуктів, використовуючи платні оголошення Facebook [14].

Нещодавно Facebook втратив довіру мільйонів своїх користувачів, дозволивши третім сторонам отримати доступ до особистих даних понад 87 мільйонів користувачів. Це величезна втрата довіри, що створила почуття хвилювання серед аудиторії соціальної мережі. Настільки, що зараз існує кампанія #deletefacebook, де люди повністю видаляють облікові записи на Facebook та використовують інші соціальні мережі. На сьогоднішній день, обсяг інформації, що зберігається на серверах Facebook, дорівнює близько 600 петабайтам. Щоденно, користувачі Facebook роблять більш ніж 5 млрд публікацій, що є чудовою базою для дослідження соціального стану населення будь-якої країни [14].

Таблиця 1.1 – Топ-10 найбільш популярних соціальних мереж у світі

№	Соціальна мережа	Кількість активних користувачів
1	Facebook	1 590 000 000
2	WhatsApp	1 000 000 000
3	QQ	853 000 000
4	WeChat	697 000 000
5	QZone	640 000 000
6	Tumblr	555 000 000
7	Instagram	400 000 000
8	Twitter	320 000 000
9	Baidu Tieba	300 000 000
10	Skype	300 000 000

Тому, зважаючи на такі статистичні дані, можна зробити висновок, що соціальні мережі є чудовою базою для аналітики реакції людей на різноманітні події. Таким чином, необхідно розробити методи для зчитування, структурування, обробки та аналізу отриманої інформації з різних соціальних мереж та для дослідження маніпулювання думкою людей під час інформаційної війни.

1.3 Методи аналізу соціальних мереж

Існують чимало різноманітних методів аналізу соціальних мереж. Кожен з них доцільно використовувати для конкретної мети дослідження соціальної мережі, проте також часто використовують комбінації методів для більш детального аналізу. Основними напрямками дослідження в аналізі є: структурний, ресурсний, нормативний та динамічний [15].

Ресурсний підхід розглядає можливості учасників із залучення персональних та мережевих ресурсів для досягнення певних цілей і диференціює учасників, що перебувають в ідентичних структурних позиціях соціальної мережі, саме за їх ресурсами. У якості персональних ресурсів можуть виступати знання, рейтинг, соціальний статус, раса, національність, стать. Під мережевими ресурсами розуміються вплив, статус, обсяг і характер інформації. Основним показником, що виокремлює відмінності у ресурсах учасників соціальної мережі, є сила зв’язків структурної позиції учасника мережі. Важливим завданням ресурсного підходу є аналіз змісту соціальних мереж. Мережевий контент служить джерелом для великої кількості додатків, що орієнтуються на вилучення та аналіз даних. Використання змісту мережі допомагає значно покращити якість результатів під час аналізу соціальних мереж, таких як задачі класифікації та кластеризації.

Аналіз надання переваги базується на збиранні інформації та створенні персональних анкет, у яких зберігається інформація про порівняння двох учасників спільноти третім учасником за заданим параметром. У результаті порівняння отримується тримісне відношення, якому відповідає бінарна матриця. Після цього дані обробляються за алгоритмом перетворення бінарних матриць до зважених графів. Структура таких графів відображає структуру спільноти та може аналізуватися шар за шаром, залежно від ваги зв’язків.

Ідентифікація користувачів у різних соціальних мережах [9] відбувається за допомогою виявлення усіх облікових записів певного користувача, у

декількох соціальних мережах одночасно. Вихідними даними для пошуку є місце роботи користувача, номер телефону, адреса електронної пошти, місце навчання, місто проживання тощо. Найпростішим способом ідентифікації є пошук за точним співпадінням усіх відомих вищезгаданих метрик.

Одним з ключових методів є аналіз пропаганди SCAME та контрпропаганди [15]. SCAME – це абревіатура від слів: Source (джерело), Content (зміст), Audience (аудиторія), Media (медіа), Effect (ефект).

За допомогою лінгвістичних методів визначається наскільки часто у текстовій інформації, що міститься в обліковому записі, зустрічаються певні терміни, і, при вирахуванні певного рівня їх появи, можна зробити відповідні висновки. Для такого методу необхідно володіти термінологією, дістати яку можна зі словників чи тезаурусів за конкретною предметною областю. Крім того, необхідно розуміти принципи морфології.

Алгоритми машинного навчання розраховані на структуровані та нормалізовані дані, тому перед їх застосуванням текстову інформацію замінюють на набори слів, що зустрічаються в них, або на набір числових значень, що характеризують ці тексти. Для цього використовуються лінгвістичні алгоритми виділення найбільш значущих слів, їх нормалізація, створення лексичного профілю тексту, виявлення теми тексту тощо.

Метод пошуку спільнот користувачів [9] працює на основі соціальних або структурних зв’язків між користувачами. Алгоритм базується на процесі обміну мітками спільнот користувачів між вершинами у відповідності з динамічними правилами взаємодії. Визначення спільнот зі слабким внутрішнім зв’язком та розділення їх на більш зв’язані між собою підспільноти є додатковим кроком алгоритму.

За допомогою методу візуалізації відповідних графів [16] зв’язки між користувачами або спільнотами формують граф. Математичний аналіз графів дозволяє розрахувати цілий ряд параметрів та надати кількісні відповіді на запити. Послідовність вершин та ребер, що з’єднують дві вершини,

називаються шляхом. Кількість кроків, які потрібно зробити, щоб дістатися від однієї вершини до іншої являють собою відстань між вершинами.

Для виконання методу розрахунку індексів [16] використовують елементарні параметри, серед яких число вершин чи ребер, а також щільність, число асиметричних та транзитивних діад, геодезійна відстань, структурні дірки, діаметр соціальної мережі тощо.

Виділення підструктур мережі [16] дозволяє визначити структури, приховані у соціальній мережі. Виокремлення блоків можна зробити за різними метриками агентів соціальної мережі. Еквівалентні агенти також можуть об’єднуватися у блоки.

Застосування скрепінгу [17] можливе за допомогою використання інтерфейсу програмування додатків (API). Під час використання API немає необхдності мати справу з HTML-сторінками, оскільки замість них усі дані будуть надходити у вигляді посилань.

Існують наступні статистичні методи аналізу соціальних мереж: підрахунок кількості репостів, лайків, підписників, згадувань ключових слів з подальшим групуванням а також інших кількісних характеристик агентів соціальних мереж та їх публікацій[18].

Підходи, які застосовуються для аналізу тональності тексту можна поділити на дві основні категорії: інженерно-лінгвістичні та методи на основі машинного навчання. Інженерно-лінгвістичні методи використовують попередньо підготовлені тональні словники та/або лінгвістичні правила, на основі яких відбувається аналіз текстових фрагментів. Методи машинного навчання включають в себе у більшості штучні нейронні мережі, проте також існують і інші методи. Ця група методів використовує математичні моделі, що дозволяють автоматично визначати оптимальний набір метрик для визначення тональності текстової інформації.

У якості вхідних даних метод класифікації використовує тексти повідомлень та поля облікового запису випадкового користувача. Алгоритм

класифікації виконується для зазначеної мови та атрибута. Результатом класифікації є значення атрибута обраного користувача.

Метод кластеризації полягає у розбитті соціальної мережі на підмножини, що не перетинаються між собою. При цьому кожен кластер повинен складатися з подібних об’єктів, а об’єкти різних кластерів мають суттєво відрізнятися. Виділення кластерів можна виконувати за різними атрибутами агентів соціальної мережі, такими як стать. Еквівалентні агенти соціальної мережі можуть об’єднуватися у кластери. Існують декілька методів з виокремлення підструктур у соціальній мережі:

– визначає кліки (підгрупи, у яких агенти пов’язані між собою сильніше, ніж з учасниками інших кліків) у соціальній мережі;

– виділення компонентів (частин графа), що пов’язані всередині спільноти і не пов’язані між собою;

– знаходження блоків і перемичок. У даному випадку вершина називається перемичкою, якщо після її видалення граф розділяєтсья на блоки;

– виділення груп – розбиття на групи еквівалентних агентів соціальної мережі, що мають схожі профілі зв’язків.

У таблиці 1.2 наведено порівняння методів аналізу соціальних мереж, а також виділено їх основні переваги та недоліки для їх подальшого використання.

Таблиця 1.2 – Порівняння методів та підходів аналізу соціальних мереж

Метод	Підхід	Переваги	Недоліки
Аналіз надання переваги	На основі графів	Простота у аналізі даних	Великі апаратні затрати для дослідження
Ідентифікація користувача в різних соціальних мережах	Лінгвістичний	Точність ідентифікації користувача за умови наявності всіх необхідних даних	Не завжди можна чітко ідентифікувати користувача через відсутність необхідних даних
Аналіз пропаганди SCAME та контрпропаганда	Лінгвістичний, нейромережі	Якісна аналітична підготовка, детальна розробленість всіх	Необхідні великі обчислювальні ресурси для точного результату

		етапів породження повідомлення	

Продовження таблиці 1.2

Лінгвістичний	Лінгвістичний	Зручність в оперуванні даними	Низька точність результату за умови більше ніж одної оцінки тональності тексту
Машинне навчання	Лінгвістичний, нейромережі	Концентрація уваги на обчисленнях	Висока обчислювальна складність, нечітка апріорна величина
Пошук спільнот користувачів	На основі графів	Імітація людського спілкування між парами індивідумів	Значна обчислювальна складність, нездатність знаходити зв'язки спільнот
Візуалізація відповідних графів	На основі графів, статистичний	Істотна практична потужність, простота у використанні	Складність побудови графів
Розрахунок індексів для соціальної мережі в цілому	На основі графів, статистичний	Можливість розрахувати цілий ряд параметрів	Невеликий діапазон того, що можна було б досліджувати
Виділення підструктур мережі	Сегментація мережі	Дозволяє виявити структури, приховані у соціальній мережі	Дозволяє проводити лише кількісний аналіз соціальної мережі
Скрепінг	З використанням API	Метод підходить для будь-якої HTML-сторінки, можливість використання методу без застосування API	Перевантаження каналу, часові затрати

Візуалізація допомагає природним чином звести воєдино інформацію про соціальні мережі і зробити її більш доступною для розуміння. Важливим є створення алгоритмів, що поєднують у собі методи аналізу соціальних мереж та методи візуалізації, щоб поліпшити розуміння структури і динаміки стану соціальної мережі.

1.4 Аналіз систем прийняття рішень

DSS (Decision Support Systems) - система підтримки прийняття рішень або СППР - це комп'ютерна система, яка шляхом збору та аналізу великої кількості

інформації може впливати на процес прийняття рішень організаційного плану в будь-якій сфері людської діяльності [19]. Інтерактивні системи дозволяють отримати корисну інформацію з першоджерел, проаналізувати її, а також виявити існуючі моделі для вирішення певних завдань [20].

За взаємодією з користувачем виділяють три види СППР:

– пасивні, що допомагають у процесі прийняття рішень, але не можуть висунути конкретної пропозиції;

– активні, що безпосередньо беруть участь у розробці правильного рішення;

– кооперативні, що припускають взаємодію СППР з користувачем.

Запропоновану системою пропозицію користувач може доопрацювати, вдосконалити, а потім відправити назад у систему для перевірки. Після цього пропозиція знову видається користувачеві, і так до тих пір, поки він не схвалить рішення.

Виділяють чотири основні компоненти, що використовуються у СППР:

– інформаційні сховища даних;

– засоби і методи вилучення, обробки і завантаження даних;

– багатовимірна база даних і засоби аналізу OLAP;

– засоби Data Mining.

Існує безліч підходів для розробки систем підтримки прийняття рішень, що використовують різні підходи: кореляційний і регресійний аналіз, сценарні методи, теорія ігор, нечітка логіка тощо. Але практично всі експертні системи моделюють процес прийняття рішення експертом як дедуктивний процес з використанням виведення, що грунтується на класифікаційних правилах [21]. Це означає, що в систему закладається сукупність правил типу «якщо ... то ...», згідно з якими на підставі вхідних даних генерується те чи інше рішення поставленої проблеми. Останнім часом розвивається «некласичний» підхід у теорії управління і прийняття рішення. Він пов'язаний із застосуванням

алгоритмів на основі нечіткої логіки, нейронних мереж і генетичних алгоритмів, сценарних методів тощо. Крім того, широко використовується ситуаційне управління на основі ієрархічних моделей з нечіткими параметрами; моделі і алгоритми прийняття рішень для захисту інформації на основі методів штучного інтелекту. Використання результатів моделювання і прогнозування ходу випадкових процесів, що описують поведінку - важливий етап в процесі прийняття рішень для підвищення їх ефективності та зниження ймовірності появи невірних рішень. Саме тому актуальним є не тільки дослідження, в якому ступені результатів прогнозування впливають на оцінку альтернативних рішень, а й розробка адаптивної системи підтримки прийняття рішень на основі результатів прогнозування випадкових процесів. Як відомо, прийняття рішень в інформаційних системах і системах управління здійснюється в умовах апріорної невизначеності, обумовленої неточністю або неповнотою вихідних даних, стохастичною природою зовнішніх впливів, відсутністю адекватної математичної моделі, нечіткістю сформульованої мети, людським фактором тощо. Невизначеність системи може призвести до збільшення ризиків прийняття неефективних рішень, в результаті чого можуть спостерігатися негативні економічні, технічні та соціальні наслідки. Невизначеності в системах прийняття рішень компенсуються різними методами штучного інтелекту. Для ефективного прийняття рішень при невизначеності умов функціонування системи застосовують методи на основі правил нечіткої логіки. Такі методи ґрунтуються на нечітких множинах і використовують лінгвістичні величини і вирази для опису стратегій прийняття рішень. Одним з таких є метод нечіткого логічного висновку. Це зручний механізм вирішення задач прийняття рішень, що забезпечує прозорість алгоритму прийняття рішень, легкість його коригування, надає можливість враховувати кількісні значення і якісні характеристики модельованих систем.

Також широко використовуються рейтингові системи, що відіграють важливу роль для прийняття рішення [22]. Такі системи дозволяють отримувати

доступну та своєчасну інформацію у вигляді інтегрального показника, який використовується для прийняття рішення. Якісні показники, як правило, вимірюються за допомогою різних шкал та часто є неспівставними по суті, тому існує проблема отримання рейтингових оцінок на основі традиційних згорток окремих показників. Уникнути таких проблем під час побудови рейтингових систем дозволяє теорія нечітких множин. Представлення окремих показників у вигляді нечітких множин, що визначаються єдиною універсальною множиною, коректність оперування з їх функціями приналежності забезпечує отримання адекватних та стійких рейтингових оцінок. Головною задачею під час побудови рейтингових оцінок з якісними характеристиками є задача формалізації отриманих даних під час їх оцінювання. Вирішення цієї задачі полягає у побудові на єдиному універсальному просторі моделей експертного оцінювання характеристик. З позиції апарату теорії нечітких множин, такими моделями можуть слугувати повні ортогональні семантичні простори.

Для подальшого дослідження доцільно обрати СППР, що вирішує задачу класифікації. Існує велика кількість методів аналізу інформації у соціальних мережах, які вирішують задачі класифікації:

– байесовский класифікатор (метод найближчих сусідів, лінійний дискримінант Фішера);

– нейронна мережа (персептрон);

– лінійний роздільник (логістична регресія, лінійний дискримінант Фішера);

– індукція (дерево рішень, тестовий алгоритм);

– скорочення розмірності (метод головних компонент, метод незалежних компонент);

– вибір моделі (мінімізація емпіричного ризику, генетичний алгоритм, самоорганізація моделей).

Для подальшої розробки програмного засобу обрано нейромережевий метод. Нейронна мережа працює за методом опорних векторів. Метод опорних векторів - це метод аналізу даних для класифікації та регресійного аналізу за допомогою моделей з керованим навчанням з пов'язаними алгоритмами навчання, які називаються опорно-векторними машинами (ОВМ). Алгоритм тренування ОВМ будує модель, яка відносить нові зразки до однієї чи іншої категорії, роблячи це неймовірнісним бінарним лінійним класифікатором. Модель ОВМ є представленням зразків як точок у просторі, відображених так, що елементи з відповідних категорій розділені умовною площиною. Нові елементи, що досліджуються, відображатимуться по той бік від площини, де знаходяться інші елементи з цієї ж категорії. Таким чином відбувається процес передбачення про належність елементів до певної категорії на основі того, на який бік від площини вони потрапляють.

1.5 Відомі аналоги програмних засобів для виявлення фейкових облікових записів

Для виявлення фейкових облікових засобів у соціальних мережах існує достатня кількість як веб-додатків, так і програмних засобів для ПК, проте їх основним недоліком є те, що вони роблять висновок лише на основі декількох параметрів, у той час як для більш точного результату необхідно аналізувати цілий ряд метрик.

1.5.1 FB Checker

FB Checker аналізує фотографії Facebook для виявлення підроблених профілів. Безкоштовна програма FB Checker пропонує користувачам швидкий спосіб допомогти визначити справжність профілів, вивчивши їх фотографії. У додатку є свої обмеження, оскільки він аналізує лише фотографії, а користувачі повинні мати доступ до фотографій відповідних облікових записів. Цей засіб працює за допомогою пошуку в Інтернеті дублікатів обраних фотографій та попереджає користувача, якщо ці фотографії виявлені в декількох місцях [23].

FB Checker перевіряє, чи справді люди, які зустрічаються у Facebook, справжні чи містять у собі підроблені фотографії. Люди створюють фейкові акаунти, щоб рекламувати товари або просто привертати увагу. Ці облікові записи містять фотографії привабливих людей для того, щоб заманити інших користувачів додатися у друзі, поговорити з ними та привернути до них всю увагу користувача. FB Checker повідомляє про те, чи фотографії у обліковому записі, що перевіряється, власними чи їх просто завантажено з інших місць Інтернету.

1.5.2 FAKE FB - Fake profile examiner

FAKE FB - це розумне розширення для Chrome, яке дозволяє перевірити одним натисканням клавіші, чи підроблений будь-який профіль Facebook на предмет фейковості. FAKE FB використовує вдосконалену механіку «Аналіз поведінки» та декілька статистичних показників [24].

FAKE FB v0.9.1 має оновлений інтерфейс, що не блокує спливаючі вікна. Він також надає розширений і більш орієнтований на користувача алгоритм, який пов'язує результати кожної перевірки на основі інформації власного особистого облікового запису. На даний час FAKE FB знаходиться у бета-версії, але рівень достовірності результатів перевірки сягає понад 90%. Проте, серед недоліків є те, що після нещодавніх змін у Facebook, у версії v0.8 виникли проблеми з багатьма обліковими записами, інформація на яких була написана не латиницею. Версія 0.9.1 тепер вирішує цю проблему.

1.5.3 FakeOff

FakeOff - це додаток, який синхронізується з обліковим записом користувача Facebook і може попереджати його про користувачів, які є підозрілими у його списку друзів. Головною метою застосунку FakeOff є попередження користувачів Facebook про шахрайські акаунти [25].

Безкоштовна версія додатка розглядає інформацію про часові шкали користувача за 10 днів, а повна версія аналізує цілий рік. Отримавши дані,

FakeOff надає користувачеві рейтинг 1-10, де 1 - швидше за все фейковий обліковий запис, а 10 - швидше за все, реальний обліковий запис.

Додаток перевіряє не тільки фотографії, але й також часову шкалу, за допомогою якої відстежується складність розмови та видно, чи є розмова логічною. Під час різних пошуків FakeOff збирає велику базу даних справжніх та фейкових профілів, щоб навчити свій алгоритм для майбутніх пошуків.

FakeOff також може працювати і в інших соціальних мережах, але оскільки у Facebook велика кількість користувачів, основний функціонал додатку зосереджено саме на цій соціальній мережі.

1.6 Постановка завдання

Виявлення фейкових облікових записів у соціальних мережах – дуже актуальна задача на сьогоднішній день, що постала як перед звичайними користувачами соціальних мереж, так і перед державними структурами, політиками та комерційними організаціями. Фейкові облікові записи використовують як звичайні користувачі для простого приховування своєї особистості, так і шахраї, що використовують фейки для збагачення незаконними шляхами або при проведенні інформаційно-психологічних операцій під час інформаційної війни. Все частіше фейкові облікові записи використовуються для поширення як правдивих, так і штучно створених фейкових новин та пропаганди у соціальних мережах з метою впливу на думки користувачів та подальшого керування соціальними групами.

Для того, щоб знизити вплив фейкових облікових записів на захищеність соціотехнічної системи, необхідно розробити програмний засіб, що дозволить користувачам виявляти підозру на фейкові облікові записи та чітко розуміти з ким вони мають справу під час як звичайного спілкування, так і під час перегляду новин, приєднання до груп або здійснення торгових операцій у соціальних мережах.

Для подальшої розробки програмного засобу обрано нейромережевий метод. Нейронна мережа працює за методом опорних векторів. Метод опорних векторів - це метод аналізу інформації для її класифікації з використанням моделей з навчанням з пов'язаними алгоритмами для навчання, що називаються опорно-векторними машинами (ОВМ). Алгоритм навчання ОВМ будує модель, що відносить нові елементи до певної категорії, через що такий алгоритм також називають неімовірнісним бінарним лінійним класифікатором.

Дослідження зв’язків користувачів, а також перевірка облікових записів у декількох соціальних мережах одночасно поки не розробляється.

На основі сформованих вимог до програмного засобу розроблено технічне завдання.

Програмний засіб повинен відповідати таким вимогам:

– програмний засіб повинен складатися з двох частин: модуля зчитування та обробки інформації та модуля нейронної мережі;

– модуль зчитування та обробки інформації повинен зчитувати дані про вказаного користувачем облікового запису, обробляти їх та перетворювати у масив чисел;

– нейронна мережа повинна навчатися на навчальній вибірці, що знаходиться у окремому файлі, а також при подачі на вхід оброблених даних про обліковий запис видавати на вихід результат щодо фейковості облікового запису.

Отже, проаналізовано інформаційні джерела, розглянуто основні поняття соціальних мереж, фейкових облікових записів та їх використання під час інформаційних протиборств, а також під час звичайного користування соціальними мережами. Розглянуто основні метрики облікових записів у соціальних мережах, їх типи та способи їх обробки. Розглянуто та проаналізовано методи аналізу соціальних мереж за їх типами та обрано оптимальний метод для подальшого дослідження. Висунуто ряд вимог до розроблюваного програмного засобу.

2 РОЗРОБКА МОДЕЛЕЙ ДЛЯ ВИЯВЛЕННЯ ФЕЙКОВИХ ОБЛІКОВИХ ЗАПИСІВ

2.1 Структура облікових записів

Обліковий запис - збережена в комп'ютерній системі сукупність даних про користувача, необхідна для його розпізнавання (автентифікації) та надання доступу до його особистих даних і налаштувань. Як синоніми також використовується розмовне «аккаунт» (від англ. Account «обліковий запис, особистий рахунок»).

Для використання облікового запису (іншими словами, для входу в систему під ім'ям певного користувача) зазвичай потрібне введення імені (логіну, англ. Login) і пароля (англ. Password). Також може вимагатися інша додаткова інформація.

Користувачі інтернету можуть сприймати обліковий запис як особисту сторінку, профіль, особистий кабінет, місце зберігання особистих та інших відомостей на певному інтернет-ресурсі (соціальній мережі).

Основними складовими облікових записів у соціальних мережах є ім’я користувача, фотографія користувача (аватар), коротка інформація про нього (наприклад, дата народження, країна, місто, інформація про освіту та роботу, інтереси тощо), коло спілкування користувача (друзі, друзі друзів, підписники тощо), альбом для фотографій. Оскільки головною метою соціальних мереж є спілкування з іншими користувачами, кожна соціальна мережа має свою систему обміну повідомленнями, а також іншими файлами (зображення, відеозаписи, текстові документи тощо).

Часто зустрічається можливість об’єднувати користувачів у спільноти за інтересами або іншими ознаками, а також можливість розміщення публічних повідомлень та їх коментування.

Також, деякі соціальні мережі використовують фон облікового запису для унікального оформлення сторінки користувача, а також можливість слухати музику та переглядати відеозаписи. Інколи користувачам надається можливість записувати свою контактну інформацію (номер мобільного телефону, адреса електронної пошти, логін у Skype тощо).

Так, на рисунку 2.1 зображено зовнішній вигляд облікового запису Марка Цукерберга у соціальній мережі «Facebook» на 11.12.2019, де можна побачити його ім’я, фотографії, інформація про нього, список підписників та публічні повідомлення з прикріпленими фотографіями (пости).

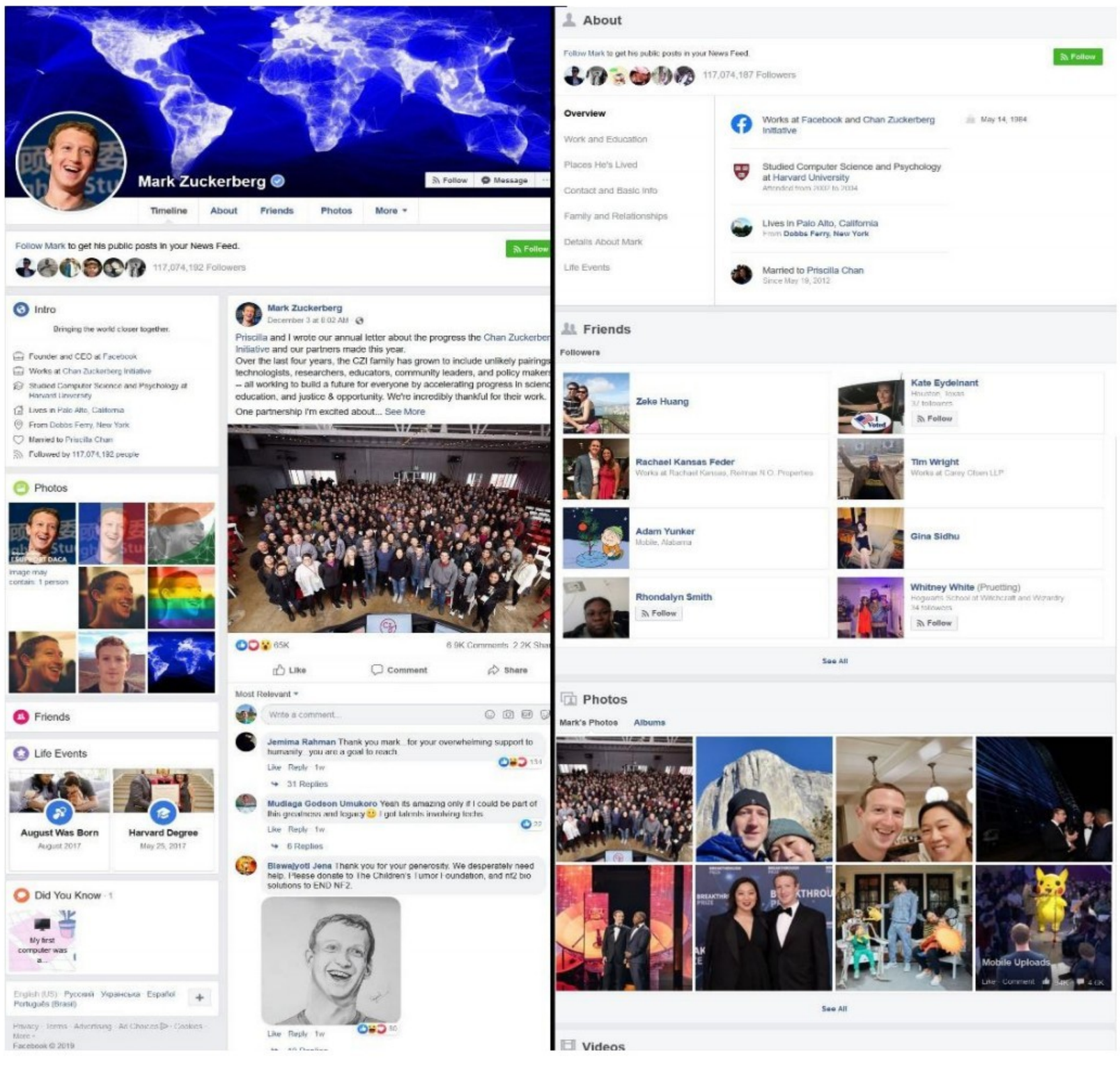


Рисунок 2.1 – Структура облікового запису на прикладі сторінки Марка Цукерберга у соціальній мережі «Facebook»

Обліковий запис, як правило, містить відомості, необхідні для впізнання користувача при підключенні до системи, відомості для авторизації і обліку. Це ідентифікатор користувача (login) і його пароль. Пароль або його аналог, як

правило, зберігається в зашифрованому або загешованому вигляді для забезпечення безпеки користувача. Проте, вже не є таємницею, що «Facebook» зберігає на своїх серверах величезну кількість інформації про обліковий запис, яку не видно звичайному користувачу. До такої інформації належить [26]:

- дані про геолокації користувача;
- дані про коментарі і про пости користувача;
- дані про відмітки користувача на фото тим чи іншим користувачем;
- потужність сигналу мережі;
- операційна система;
- браузер;
- оператор зв'язку / інтернет-провайдер;
- cookies третьої сторони;
- приховані користувачем публікації;
- вже видалена з облікового запису інформація;
- відмітки «подобається»;
- ще багато іншої інформації.

Цю інформацію «Facebook» використовує для аналізу поведінки та побудови загальної статистики щодо користувачів, а також для формування рекламних повідомлень відповідно до вподобань користувача. Наприклад, для кожного окремого поста зберігається декілька мегабайт (від 3 до 8 мегабайт) інформації, у якій міститься інформація: про автора поста; кількість лайків та тих, хто їх залишив; про користувачів, що переглянули пост; текстова частина поста; інформація про зображення поста (якщо таке існує); дата створення поста; кількість користувачів, що поділилися постом; тощо.

Таким чином, загальну структуру облікового запису у соціальних мережах можна зобразити у вигляді схеми (рис. 2.2), яка відображає як видимий вміст облікового запису, який бачить користувач, так і невидиму звичайному користувачу інформацію про нього та його дії, що зберігається на серверах соціальної мережі.

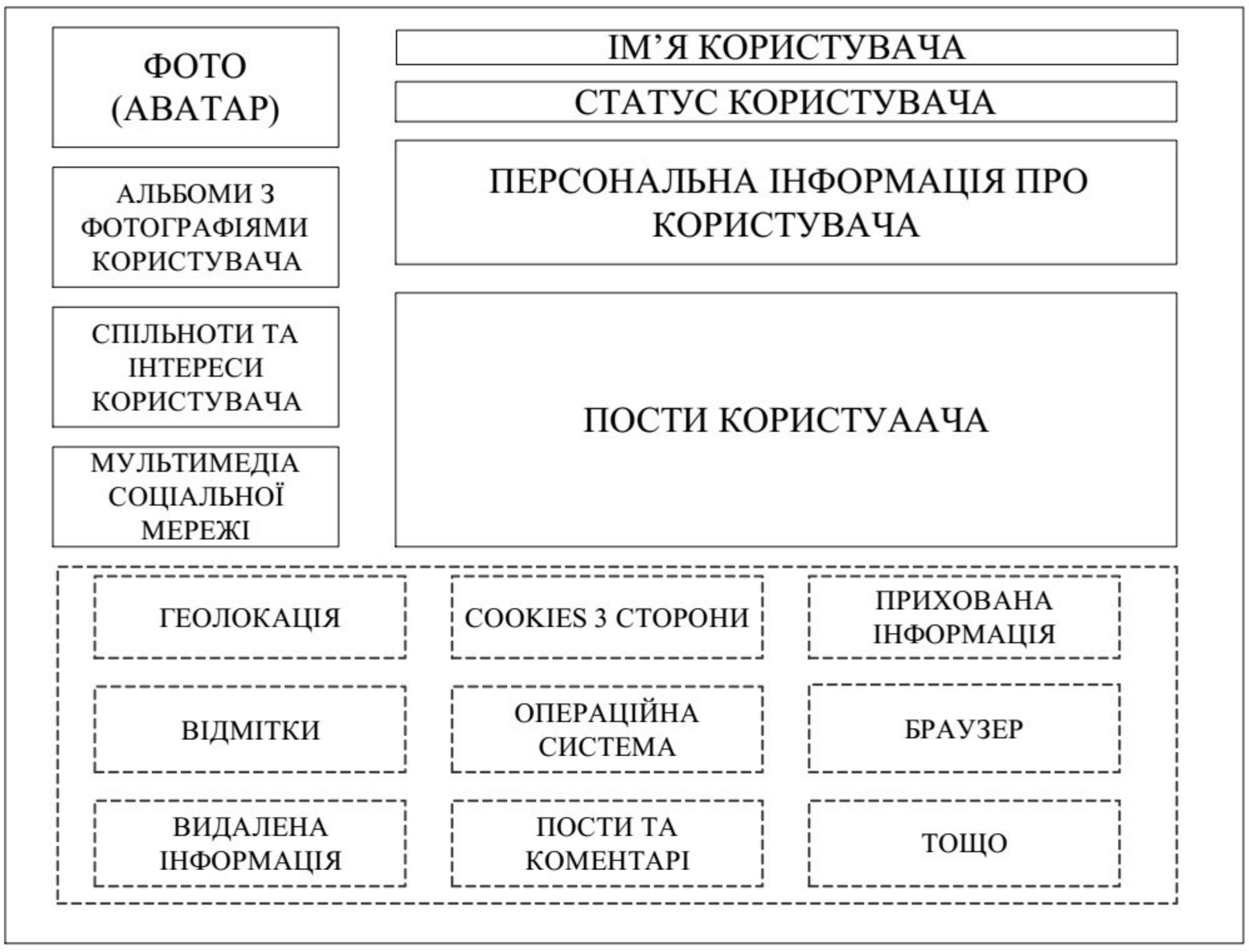


Рисунок 2.2 – Загальна структура даних облікового запису у соціальних мережах

Необхідно пам'ятати, що не вся інформація про обліковий запис може бути доступною для перегляду, оскільки сучасні соціальні мережі мають можливість приховування інформації від сторонніх користувачів, при цьому інформація буде доступною лише друзям або обмеженому колу користувачів. Також, далеко не вся інформація про користувача є правдивою. Марк Цукерберг, засновник «Facebook», що є найбільшою соціальною мережею у світі, заявив, що близько половини облікових записів (а це більше 1 млрд облікових записів) є фейковими. Тому наразі виявлення фейкових облікових записів у соціальних мережах є актуальним завданням.

2.2 Метрики облікових записів у соціальній мережі

Дослідження показали [5, 12, 23, 27-30], що можна виділити такі основні категорії ознак фейкових облікових записів: лайки, персональні дані, статуси та посилання, друзі, фото, дата народження.

Лайки (*LIKES*) за ознаками можна поділити на їх кількість (*QUANTITY*) та хто їх залишив на сторінці користувача (*FROM*). У свою чергу залишити лайки можуть як друзі (*Friends*), так і незнайомці (*Strangers*). Для визначення фейковості профілю також має значення кількість лайків (*NumberOfLikes*). Якщо у користувача під певним постом кількість лайків більша за кількість його друзів (*NumberOfFriends*), це може свідчити про те, що користувач отримав ці лайки незвичним шляхом. Відсутність лайків (*NumberOfLikes = 0*) на сторінці вказує на «ізоляцію» користувача, що також може свідчити про його фейковість. Структурна модель метрик у категорії «Лайки» показана на рисунку 2.3.

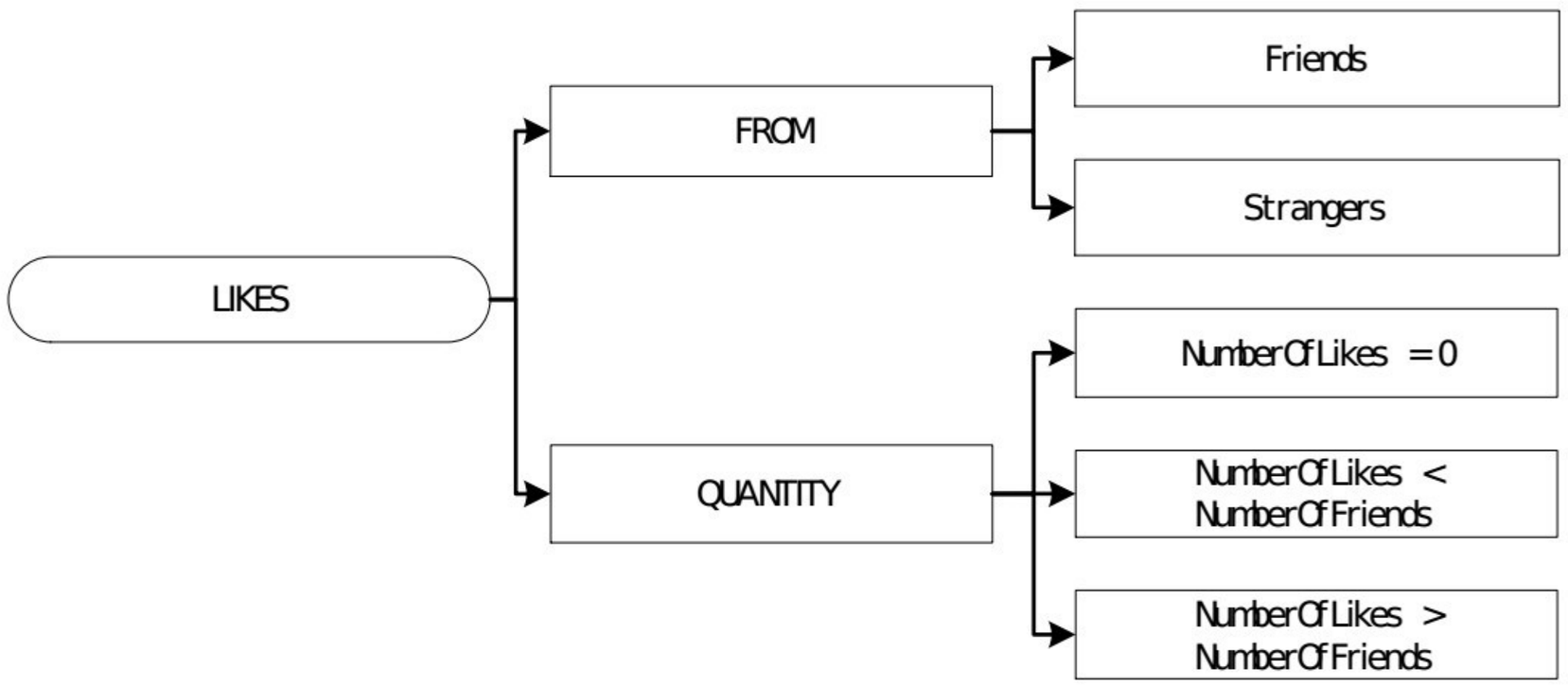


Рисунок 2.3 – Структурна модель ознак фейковості у категорії «Лайки»

Параметри моделі ознак фейковості у категорії «Лайки» можна записати у вигляді кортежів:

LIKES = {FROM; QUANTITY}

FROM = {friends; strangers}

QUANTITY = {NumberOfLikes = 0; NumberOfLikes<NumberOfFriends; NumberOfLikes>NumberOfFriends}

Персональна інформація на сторінці користувача (*PERSONAL INFORMATION ABOUT USER*) [23] може сказати чимало про фейковість або справжність профілю. Для подальшого аналізу персональну інформацію можна поділити на дату народження (*DATE OF BIRTH*) ім’я користувача (*USER`S*

NAME), кількість інформації (*QUANTITY OF INFORMATION*), суперечливу інформацію (*CONTRADICTORY INFORMATION*) та приватну інформацію (*PRIVATE INFORMATION*).

Дата народження має ознаки, які можуть вказувати на фейковість сторінки. Часто користувачі фейкових облікових записів не приділяють уваги детальному заповненню сторінки та залишають дату народження за замовчуванням (зазвичай 1 січня). Також можлива ситуація, коли вік користувача (*Age*) підлягає сумніву або не співпадає з іншими датами на сторінці (*DatesOnProfile*). Наприклад, користувачу 15 років, проте інша інформація на сторінці свідчить, що він закінчив ВУЗ 10 років тому.

Ім’я користувача дослідити важко, оскільки існує чимало людей з таким самим ім’ям та прізвищем. Проте варто перевірити ім’я на предмет його співпадіння з ім’ям видатної людини (*IsCelebrityName*). Також слід звернути увагу на те, чи належить ім’я користувача (*UserNameCountry*) до типових імен країни цього користувача (*UserCounrty*).

Відсутність персональної інформації у профілі (*NoInfoAboutUser*) або неповністю заповнені поля для персональної інформації (*PartlyFilledInfo*) свідчить про те, що користувач не хоче, щоб його могли ідентифікувати інші користувачі, а отже це теж є ознакою фейковості. Проте, існують також облікові записи, де персональна інформація заповнена повністю (*FullyFilledInfo*), хоча і недостовірними даними.

Суперечливість інформації на сторінці є одним з найбільш достовірних показників фейковості, проте і потребує складного аналізу. Наприклад, інформація в постах користувача (*PostsInfo*) не відповідає інформації, зазначеній у профілі (*InfoAboutProfile*), або користувач знаходиться у групах (*InfoAboutGroups*), які не відповідають його зазначеним інтересам (*InfoAboutIntersts*).

До персональної інформації також відносять номер телефону (*PhoneNumber*) та посилання на облікові записи у інших соціальних мережах

(*SocialNetworksLinks*). Користувачі рідко виставляють таку інформацію у відкритий доступ, проте вона може свідчити про те, що користувач є активним у соціальних мережах і не має намірів приховувати інформацію про себе. Також цим засобом можуть користуватися спеціально створені рекламні профілі.

Структурна модель метрик у категорії «Персональна інформація про користувача» показана на рисунку 2.4.

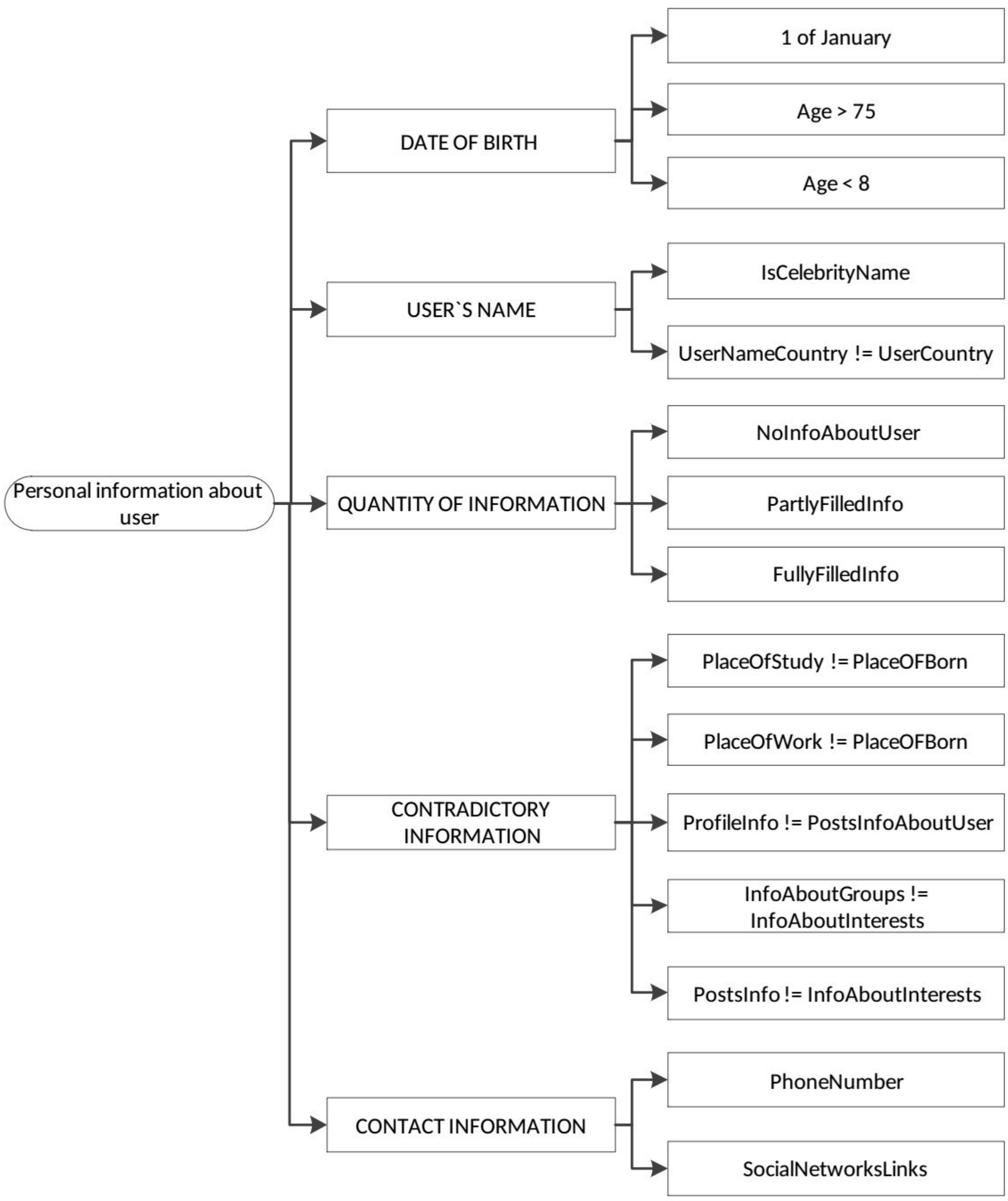


Рисунок 2.4 – Структурна модель ознак фейковості у категорії «Персональна інформація про користувача»

Параметри моделі ознак фейковості у категорії «Персональна інформація про користувача» можна записати у вигляді кортежів:

PERSONAL INFORMATION ABOUT USER = {DATE OF BIRTH; USER NAME; QUANTITY OF INFORMATION; CONTRADICTORY INFORMATION; CONTACT INFORMATION}

DATE OF BIRTH = {1 of January; Age}

USER NAME = {IsCelebrityName; UserNameCountry/UserCountry}

NUMBER OF INFORMATION = {InfoAboutUser; PartlyFilledInfo; FullyFilledInfo}

CONTRADICTORY INFORMATION = {PlaceOfStudy/PlaceOfBorn; PlaceOfWork/PlaceOfBorn; ProfileInfo/PostsInfoAboutUser; InfoAboutGroups/InfoAboutInterests; PostsInfo/InfoAboutInterests}

CONTACT INFORMATION = {isPhoneExists; SocialNetworksLinks}

Статуси та пости (*STATUSES AND POSTS ON PAGE*) на сторінці користувача аналізуються як одне ціле, оскільки вони відрізняються лише розміщенням у профілі. Їх можна аналізувати за такими ознаками: за частотою редагування/додавання (*UPDATES*) та за коментарями (*COMMENTS*). Статуси та пости іноді використовуються у якості реклами (*Advertising*) [31].

Частота редагування/додавання постів та статусів (*UpdateFrequency*) вказує на активність користувача. Якщо пости/статуси додаються рідко або дуже часто – це є однією з ознак фейковості. Якщо користувач давно додав пост/статус і протягом тривалого часу не оновлює, існує імовірність того, що цей обліковий запис є фейковим.

Кількість коментарів (*QUANTITY*) також вказує на активність самого профілю. Їх відсутність або надмірна кількість найчастіше буває саме у фейків. Коментарі можуть залишити (*FROM*) як друзі користувача (*Friends*), так і незнайомці (*Strangers*).

Структурна модель метрик у категорії «Статуси та пости» показана на рисунку 2.5.

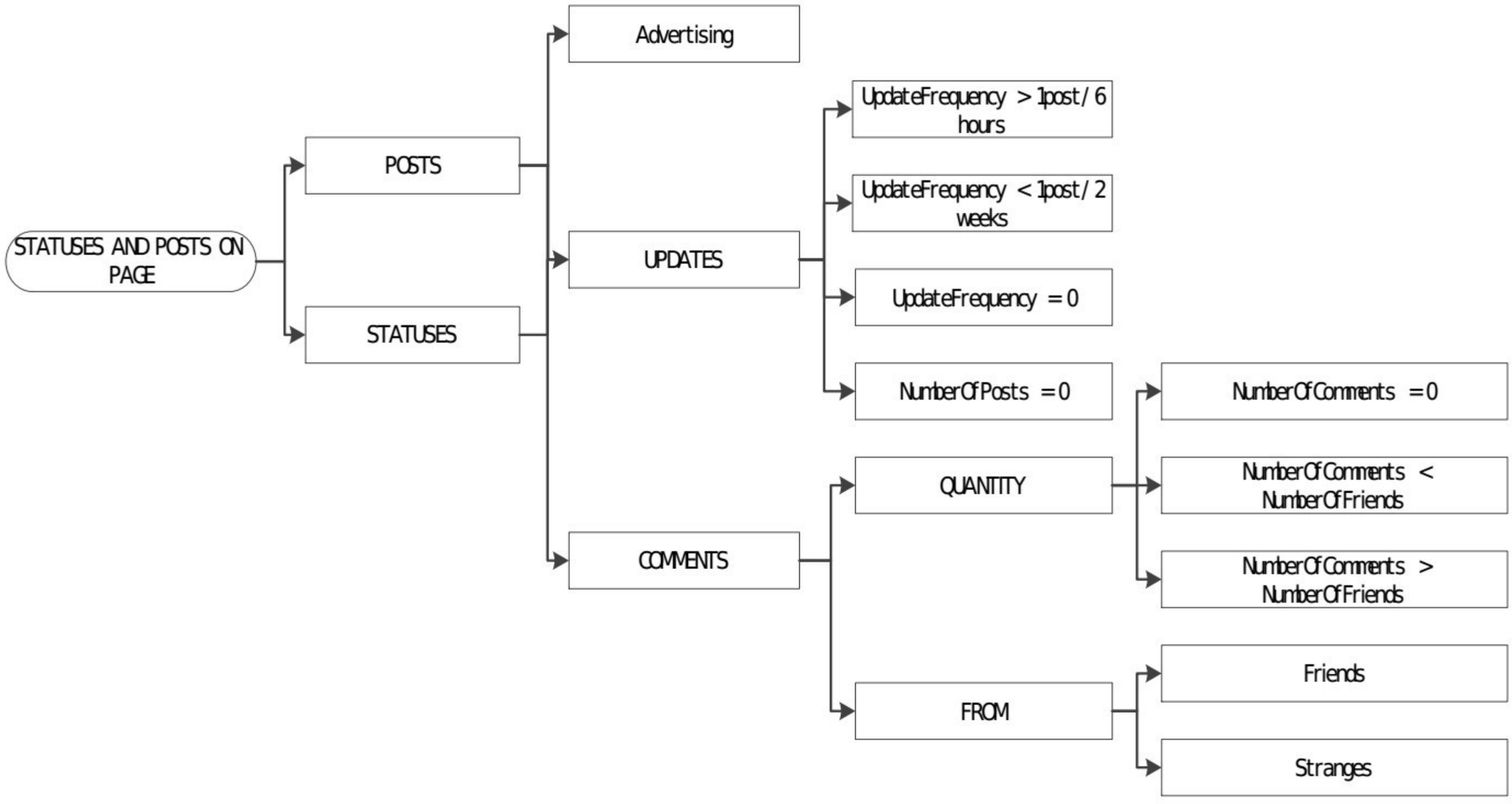


Рисунок 2.5 – Структурна модель ознак фейковості у категорії «Статуси та пости»

Параметри моделі ознак фейковості у категорії «Статуси та пости» можна записати у вигляді кортежів:

STATUS AND POSTS ON PAGE = {POSTS; STATUSES}

POSTS = {Advertising; UPDATES; COMMENTS}

STATUSES = {Advertising; UPDATES; COMMENTS}

UPDATES = {UpdateFrequency; NumberOfPosts}

COMMENTS = {QUANTITY; FROM}

UANTITY = {NumberOfComments; NumberOfComments / NumberOfFriends}

FROM = {Friends; Strangers}

Друзі користувача (*FRIENDS*) грають доволі значну роль у визначенні фейка, оскільки вони вказують як на активність профілю в соціальній мережі, так і на коло інтересів користувача [28].

Залежність фейковості від кількості друзів (*QUANTITY*) користувача проаналізувати важко, тому що для того, щоб зробити висновок про фейковість,

необхідно аналізувати самих друзів. Наприклад, якщо у списку друзів користувача є фейки (*IsFriendFake*), є імовірність, що і сам користувач – фейк. Також важливо досліджувати зв'язки між друзями та друзями друзів (LinksBetweenFriends). Якщо користувач не має друзів (*NumberOfFriends = 0*), існує велика імовірність, що його профіль використовується не для спілкування, а для інших цілей. Велика кількість друзів (*Max(NumberOfFriends)*) за короткий проміжок часу (*Min(timeline)*) після створення профілю також викликає підозри, тому, скоріше за все, такий профіль є фейковим. Структурна модель метрик у категорії «Друзі» показана на рисунку 2.6.

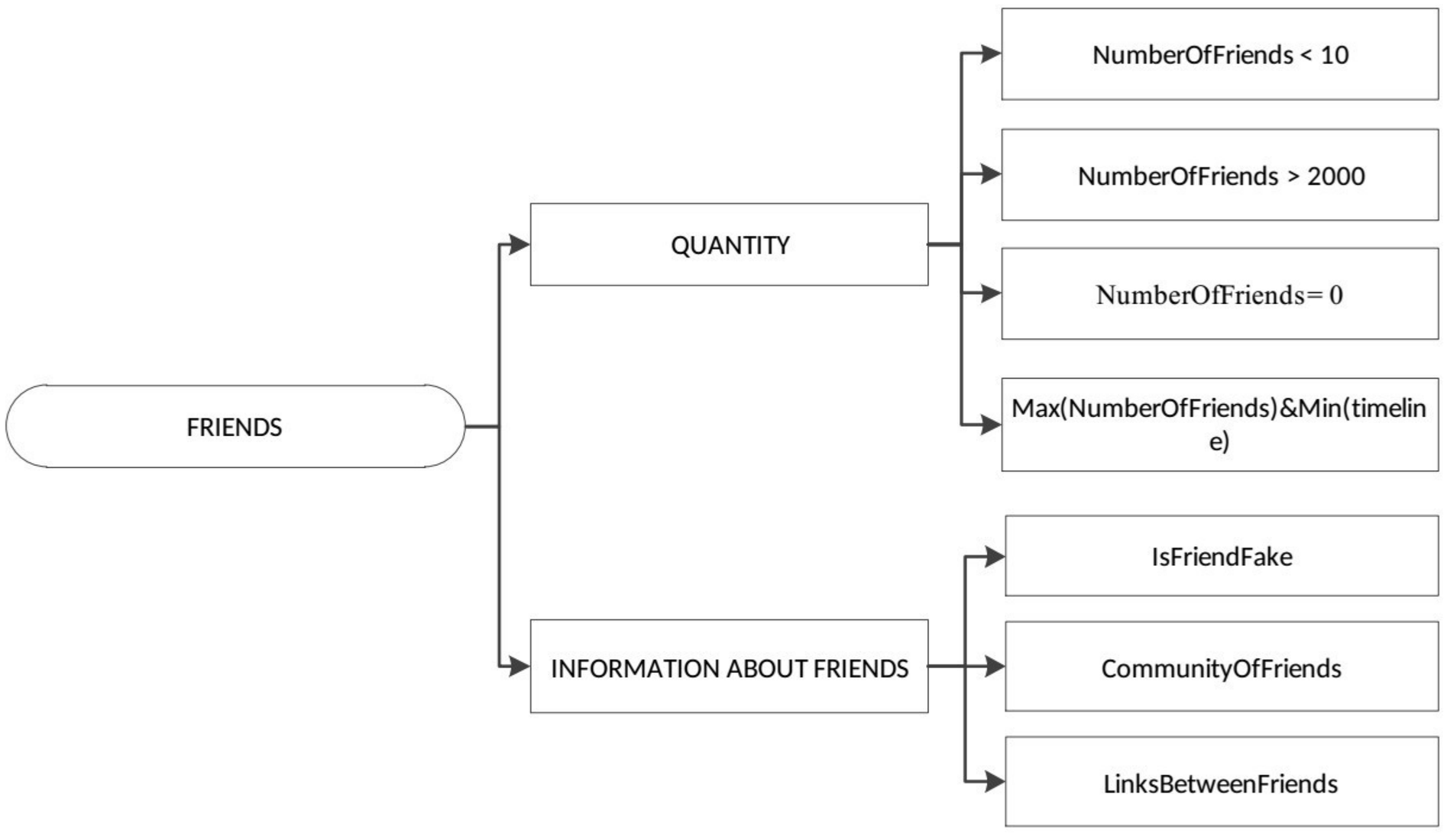


Рисунок 2.6 – Структурна модель ознак фейковості у категорії «Друзі»

Параметри моделі ознак фейковості у категорії «Друзі» можна записати у вигляді кортежів:

FRIENDS = {QUANTITY; INFORMATION ABOUT FRIENDS}

QUANTITY = {NumberOfLikes; NumberOfFriends; NumberOfLikes/ NumberOfFriends; Max(NumberOfFriends); Min(timeline)}

INFO ABOUT FRIENDS = {IsFriendFake; CommunityOfFriends; LinksBetweenFriends}

Аналіз *фотографій користувача* (*PHOTO*) відіграє найважливішу і, водночас, найважчу частину дослідження фейковості облікового запису. По-перше, необхідно проаналізувати присутність фотографій на аватарі (EXISTANCE). Відсутність фотографій як на аватарі (*AVATAR / COVER*), так і в альбомах (*PROFILE*) вже свідчать про те, що даний обліковий запис є фейковим. По-друге, за наявності фотографій на сторінці їх потрібно аналізувати на предмет співпадіння з іншими фотографіями в Інтернеті або з фотографіями інших профілів, оскільки користувач міг завантажити замість своїх фотографії знаменитостей (*Celebrities*) або інших об'єктів (*OtherPictures*). Кількість фотографій (*NumberOfPhotos*) також є важливим показником, оскільки надмірна або замала кількість фотографій вказує на неправдивість фотографій або неактивність користувача відповідно.

Структурна модель метрик у категорії «Фото» показана на рисунку 2.7.

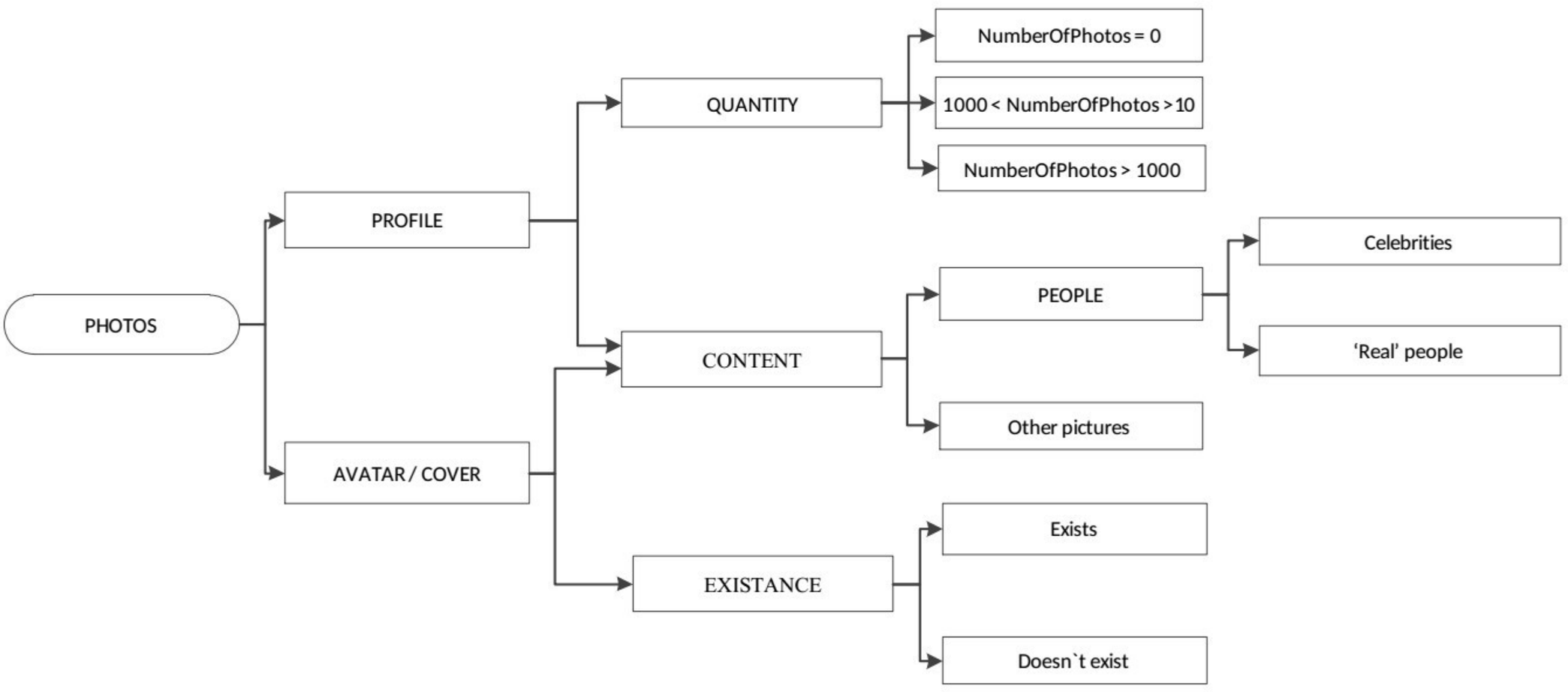


Рисунок 2.7 – Структурна модель ознак фейковості у категорії «Фото»

Параметри моделі ознак фейковості у категорії «Фото» можна записати у вигляді кортежів:

PHOTO = {PROFILE; AVATAR / COVER}

PROFILE = {QUANTITY; CONTENT}

AVATAR / COVER = {CONTENT; EXISTANCE}

QUANTITY = {NumberOfPhotos}

CONTENT = {PEOPLE; Other pictures}

PEOPLE = {Celebrities; ‘Real’ people}

EXISTANCE = {Exists; Doesn`t exist}

Звичайно, окремо ці критерії не можуть однозначно вказувати на «фейковість» облікового запису, оскільки лише аналіз їх об’єднання може поставити під сумнів справжність облікового запису. Для більш точного визначення статусу облікового запису необхідно використовувати аналіз з використанням якомога більшої кількості критеріїв.

У магістерській кваліфікаційній роботі не враховані інші важливі параметри облікових записів, такі як час створення сторінки, швидкість формування кола друзів, зв’язки між друзями користувача, перевірка друзів на фейковість, аналіз фотографій на предмет збігу з іншими зображеннями в інтернеті тощо [31]. Ці та інші параметри будуть враховані в подальших дослідженнях.

2.3 Реалізація системи прийняття рішення

У подальшому дослідженні для прийняття рішення щодо фейковості облікового запису обрано метод опорних векторів, описаний у підрозділі 1.4.

Така класифікація даних методом опорних векторів (Support Vector Machine, SVM) має досить широке застосування [32]. Завдання класифікації полягає у визначенні до якого класу з, як мінімум, двох спочатку відомих належить цей об'єкт. Для визначення того, чи є обліковий запис користувача фейковим, визначено два класи, а саме:

– «Real»;

– «Fake».

– Оскільки класів всього два («Fake» / «Real»), то завдання називається бінарною класифікацією. Також в даному випадку існують зразки кожного

класу - об'єкти, про які заздалегідь відомо до якого класу вони належать. Вони знаходяться у відповідному файлі *.csv у вигляді бази даних та в подальшому використовуватимуться як навчальна вибірка.

Отже, математична формулювання задачі класифікації така: нехай X - простір об'єктів (наприклад, P1, P2, P3, …, P10), Y - класи (наприклад, Y = {0, 1}, де 0 – справжній обліковий запис, 1 – фейковий обліковий запис). Потрібно побудувати функцію F: X→Y (класифікатор), що зіставляє клас y_i довільному об'єкту x_i.

Класифікатор працює таким чином: дані точки на площині, що представляють множини параметрів облікових записів, розбиті на два класи (рис. 2.8). Проведено лінію, що розділяє ці два класи. Далі, всі нові точки, які будуть формуватися під час дослідження облікових записів (не з навчальної вибірки) автоматично класифікуються наступним чином: точка вище прямої потрапляє до класу «Fake», точка нижче прямої - до класу «Real». Така пряма називається розділяючою прямою.

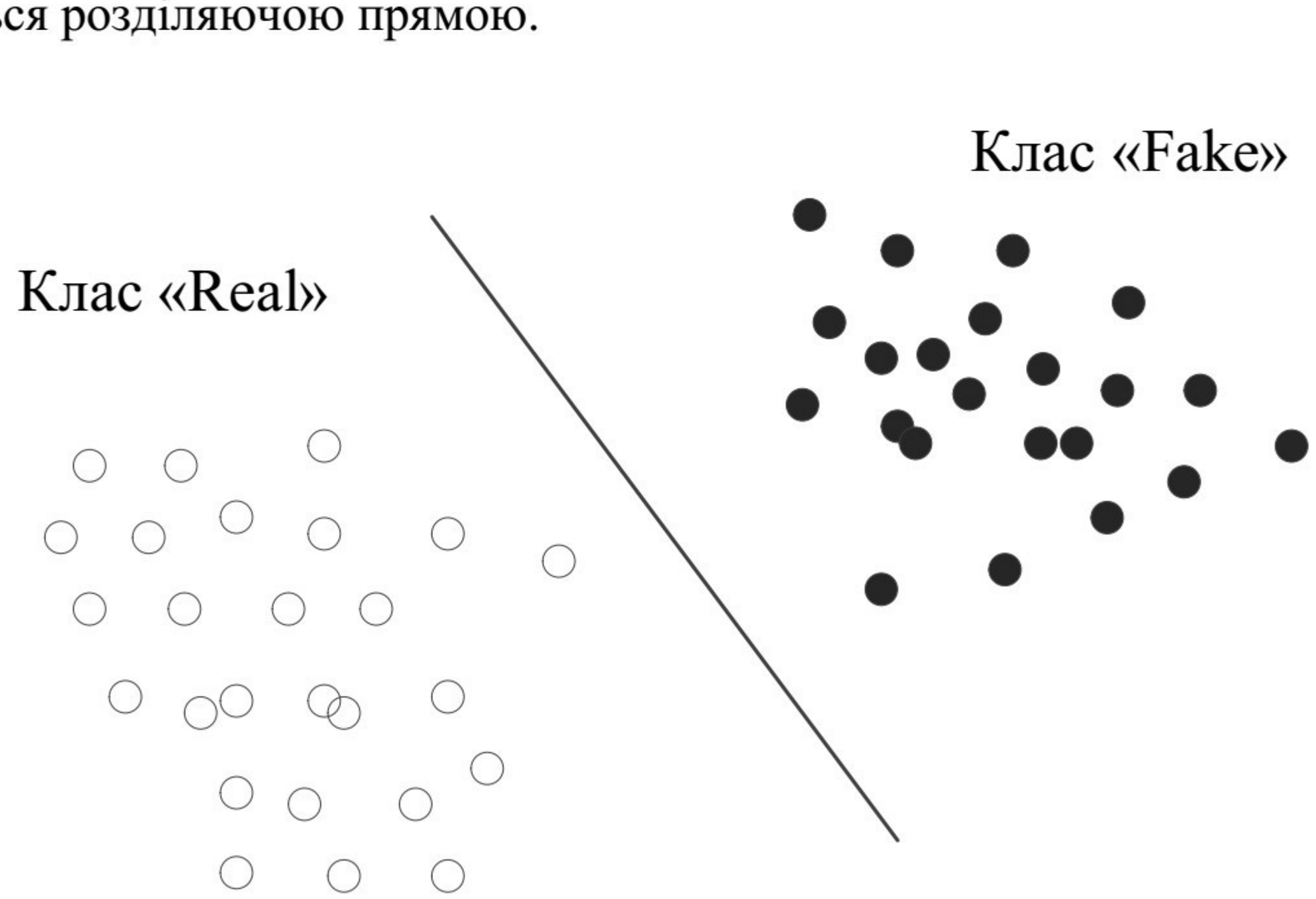


Рисунок 2.8 - Пряма, що розділяє класи «Real» та «Fake»

Однак, замість прямих доцільно розглядати гіперплощини - площини, розмірність яких на одиницю менше, ніж розмірність початкового простору.

Нехай є навчальна вибірка: $(x_i, y_i), x_i \in [0;1;2], y_i \in [0;1]$. Метод опорних векторів будує класифікуючу функцію F у вигляді $F(X) = sign((w, x) + b)$, де X - скалярний добуток, w - нормальний вектор до розділяючої гіперплощини, b - допоміжний параметр. Ті об'єкти, для яких $F(X) = 1$ потрапляють в один клас, а об'єкти з $F(X) = 0$ - в інший. Вибір саме такої функції не випадковий: будь-яка гіперплощина може бути задана у вигляді $(w, x) + b = 0$ для деяких w і b.

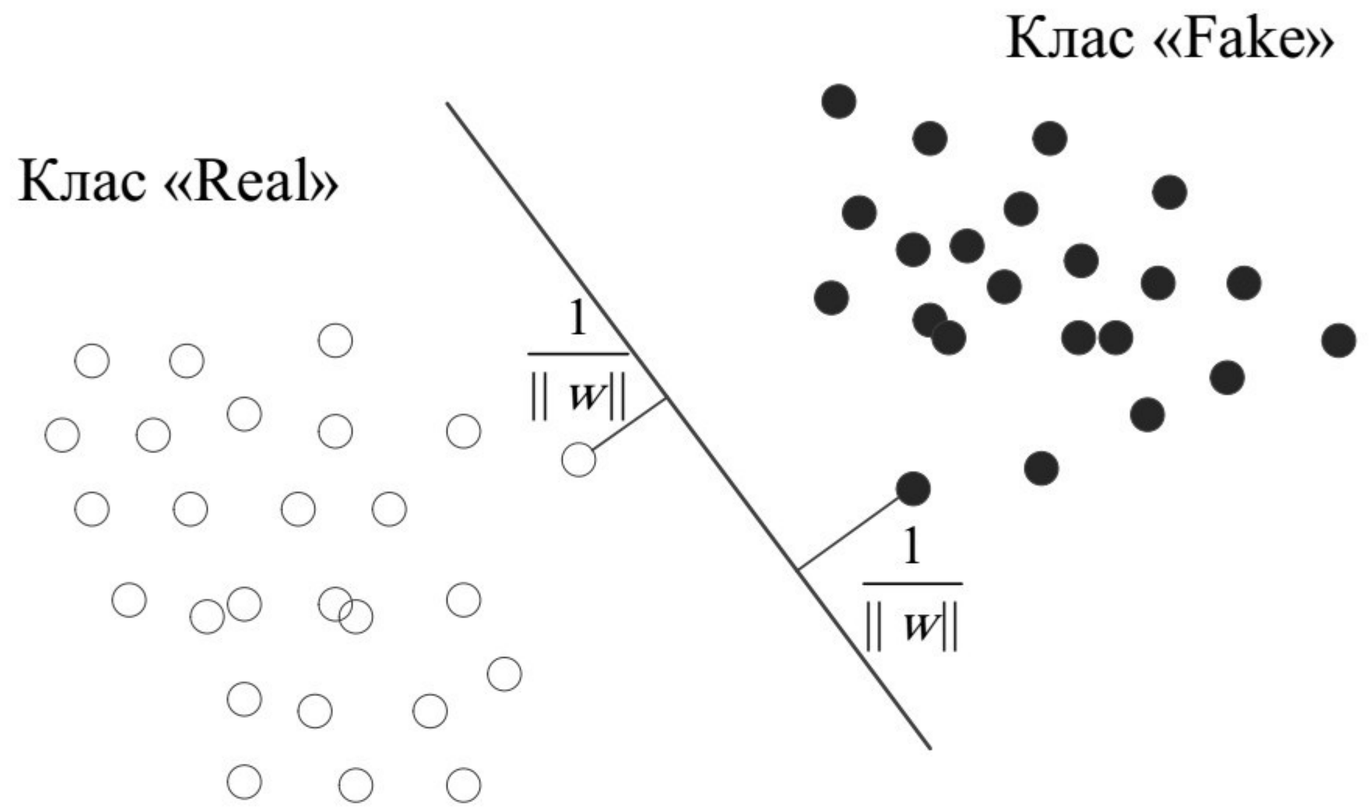


Рисунок 2.9 - Гіперплощина, що розділяє класи «Real» та «Fake»

Після цього, необхідно вибрати такі w і b, які максимізують відстань до кожного класу. Дана відстань становить $\frac{1}{\| w\|}$ (див. рис. 2.9). Проблема знаходження максимуму $\frac{1}{\| w\|}$ еквівалентна проблемі знаходження мінімуму $\| w\|^2$. Завдання оптимізації має такий вигляд (2.1):

$$\begin{cases} \arg\min\limits_{w,b} \| w\|^2, \\ y_i(\langle w, x_i \rangle + b) \geq 1, i = 1,...,10. \end{cases} \quad (2.1)$$

Задача оптимізації є стандартною задачею квадратичного програмування і вирішується за допомогою множників Лагранжа. Випадки, коли дані можна розділити гіперплощиною, або, як ще кажуть, лінійно, досить рідкісні. Тому, в цьому випадку необхідно всі елементи навчальної вибірки вкласти у простір X більш високої розмірності за допомогою спеціального відображення. При

цьому відображення φ вибирається так, щоб в новому просторі X вибірка була лінійно роздільна.

Таким чином, класифікуюча функція F приймає вигляд $F(X) = sign((w, \varphi(x)) + b)$. Вираз $k(x, x') = (\varphi(x), \varphi(x'))$ становить ядро класифікатора. З математичної точки зору ядром може служить будь-яка позитивно визначена симетрична функція двох змінних.

Найчастіше на практиці зустрічаються такі типи ядер: поліноміальні, радіальна базисна функція, гауссова радіальна базисна функція та сигмоїд. Для подальшої розробки нейронної мережі обрано тип ядра сигмоїд.

Нейронна мережа складається з великої кількості однотипних елементів - нейронів, пов'язаних між собою. На рисунку 2.10 зображено схему нейрона.

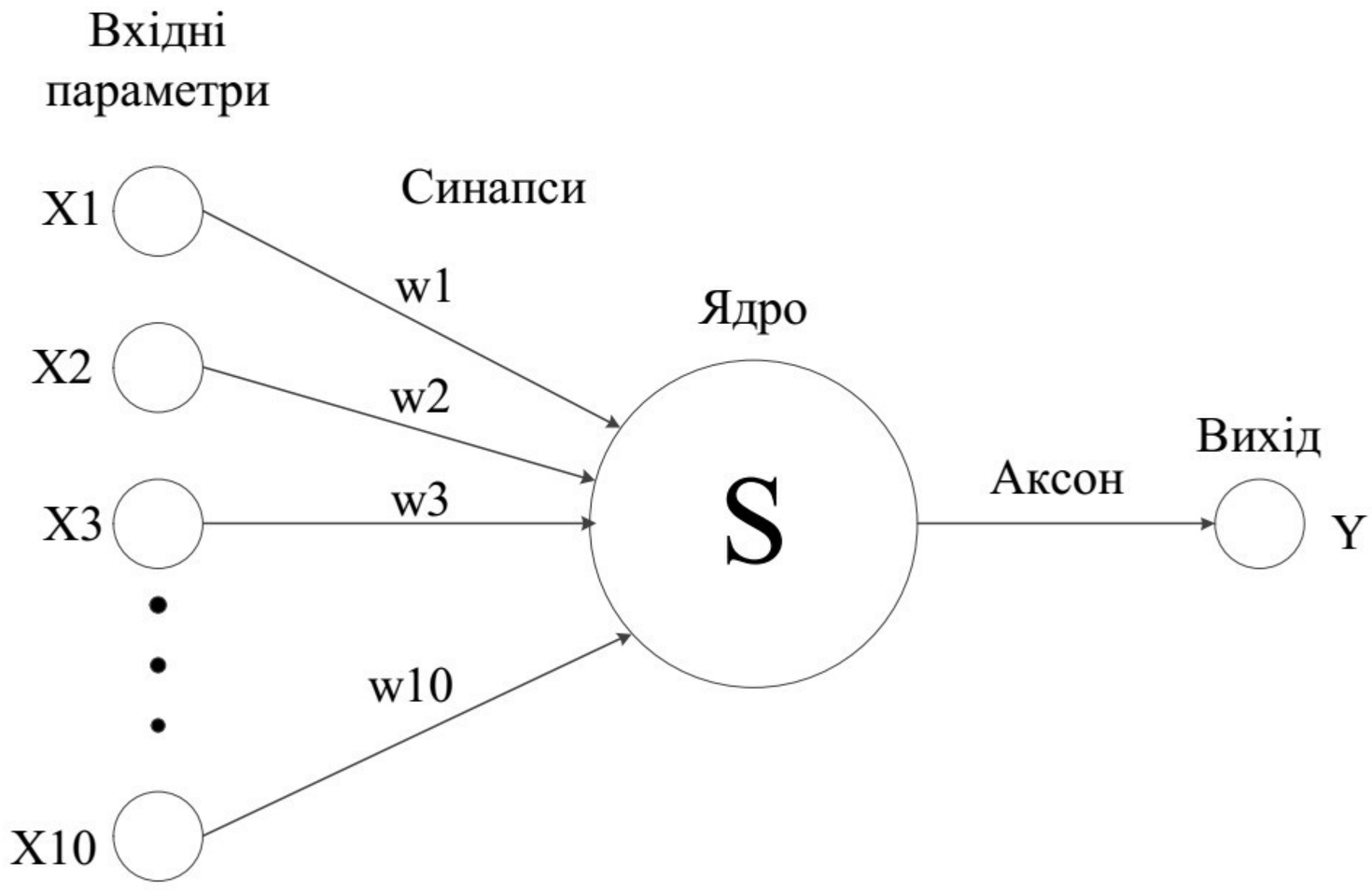


Рисунок 2.10 - Схема нейрона

Зі схеми зрозуміло, що штучний нейрон складається з таких елементів: синапсів, що пов'язують входи нейрона з ядром; безпосередньо ядра нейрона, що здійснює обробку вхідних даних; аксона, що пов'язує нейрон з наступним шаром нейронів. Кожен синапс має певну вагу, яка визначає вплив відповідного входу нейрона на його стан. Стан нейрона визначається за формулою 2.2:

$$S = \sum_{i=1}^{n} X_i W_i \qquad (2.2)$$

Де n - кількість входів нейрона, x_i - значення і-го входу нейрона, w_i - вага і-го синапсу.

Після цього визначається за формулою значення аксона нейрона $Y = F(S)$, де F - функція, яка називається функцією активації. Найбільш часто в якості функції активації використовується так званий сигмоїд, який має наступний вигляд (2.3):

$$F(x) = \frac{1}{1 + e^{-\alpha x}} \qquad (2.3)$$

У даному випадку задача навчання нейронної зводиться до знаходження функціональної залежності $Y = F(X)$ де X – вхідний вектор, а Y - вихідний вектор. Вхідний вектор X складається з нормалізованого представлення даних про обліковий запис користувача, такі як кількість фотографій, персональна інформація, дата народження тощо.

Основна відмінність цієї нейронної мережі в тому, що в неї всі вхідні і вихідні параметри представлені у вигляді цілих чисел в діапазоні [0, 1] (для вихідних параметрів) та [0, 1, 2] (для вхідних параметрів). У той же час дані предметної області часто кодуються іншим способом. Це можуть бути числові значення у довільному діапазоні, дати, символьні рядки тощо. Таким чином інформація може бути як кількісною так і якісною. Тому необхідно спочатку перетворити якісні дані в числові, а потім перетворення вхідні дані в необхідний діапазон. Таким чином для кожного якісного значення вхідних параметрів (від P1 до P10) необхідно поставити у відповідність числове значення. Оптимальними числовими значеннями вхідних параметрів є діапазон від 0 до 2, де 0 – ознака справжності облікового запису, 2 – ознака фейковості, 1 – підозріле значення параметра. Однак, це може спричинити небажану

впорядкованість, яка може спотворити дані, і сильно ускладнити процес навчання. Проте, у даному випадку для вирішення задачі виявлення фейків такий спосіб перетворення якісних показників у кількісні не модифікує подальші результати та не спотворить результат, що буде обчислено нейронною мережею.

2.4 Розробка програмного засобу

Для роботи з даними соціальної мережі Facebook обрано мову програмування Python, оскільки вона дозволяє обробляти дані великої розмірності, є простою у використанні, а також дозволяє вирішувати будь-які задачі. Для отримання даних з соціальної мережі «Facebook» обрано модуль Selenium, який дозволяє за допомогою пошуку за елементами веб-сторінок зчитувати з них дані. Для розробки графічного інтерфейсу обрано модуль Tkinter, оскільки він є відносно простим та ефективним під час розробки. Для того, щоб отримати доступ до інформації про користувача у соціальній мережі Facebook, необхідно зчитувати усі дані про користувача разом, що є складним для подальшої обробки інформації. Така складність отримання інформації з соціальної мережі «Facebook» зумовлена нещодавніми подіями масштабного витоку персональної інформації про користувачів, а також часті хакерські атаки на веб-сервери «Facebook». Тому отримання прав розробника для комфортного отримання усіх необхідних даних наразі є досить складною та тривалою процедурою. Для прийняття рішення щодо статусу облікового запису обрано нейронну мережу, яка вирішує задачу класифікації – вибору між фейковим та справжнім обліковим записом [10, 29]. Для навчання нейронної мережі створено вибірку даних, яка складається з інформації про існуючі облікові записи, а також їх класифікація.

Архітектура системи програмного засобу для виявлення фейкових облікових записів у соціальній мережі «Facebook» складається з таких компонентів (рис. 2.11):

– власне, програмного засобу, який, у свою чергу, складається з модуля зчитування та обробки інформації, а також нейронної мережі, яка має 9 входів та 1 вихід;

– веб-сторінки соціальної мережі «Facebook», де знаходиться інформація про облікові записи;

– зовнішніх файлів, у яких міститься допоміжна для роботи програми інформація.

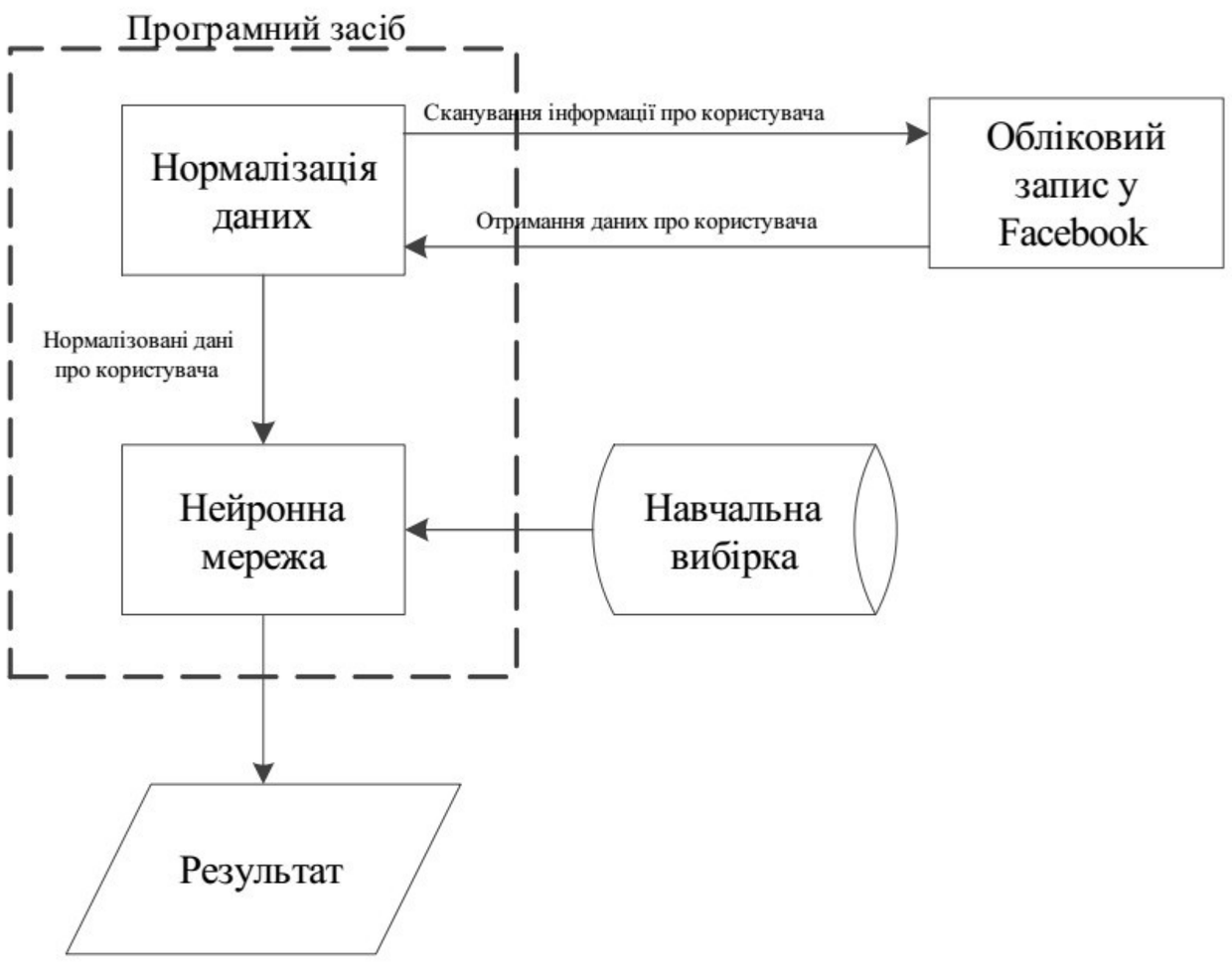


Рисунок 2.11 - Архітектура програмного засобу

Система працює таким чином: після запуску програмного засобу та вибору облікового запису для перевірки відбувається запит до веб-сторінки користувача у «Facebook». У відповідь надходить уся інформація про користувача у текстовому вигляді, що є незручним для обробки. Аналіз даних відбувається за допомогою функцій, що розділяють отриману інформацію та дістають лише необхідну для подальшого аналізу. Програмний засіб перевіряє отриману інформацію та представляє її у вигляді, зрозумілому для системи підтримки прийняття рішень та записує її у файл account.csv. Система підтримки прийняття рішень представляє собою нейронну мережу, якій, для достовірності подальших результатів, необхідно навчитися розрізняти справжні та фейкові облікові записи. Для цього використовується навчальний набір

даних, що міститься у файлі training.csv. Коли система підтримки прийняття рішень готова для обробки даних, на вхід подається файл account.csv з інформацією про обліковий запис, що потребує перевірки. Нейронна мережа обробляє інформацію та передає результат обробки головному модулю програми. Після цього програмний засіб відображає на екран результат роботи програмного засобу у вигляді статусу облікового запису та діаграми впливу параметрів на результат.

Розроблені модулі у вигляді класів дозволяють зчитувати та обробляти інформацію про фото на аватарі, фото на фоні обкладинці сторінки, кількість фотографій, кількість друзів, кількість постів, персональна інформація про користувача (місце роботи, освіта, місто проживання/народження, сімейний стан), контактна інформація, дата народження, кількість лайків на аватарі та кількість коментарів на аватарі. Модулі збору інформації працюють паралельно для пришвидшення процесу збору інформації. Таким чином, алгоритм роботи програми можна відобразити у вигляді блок-схеми (рис. 2.12).

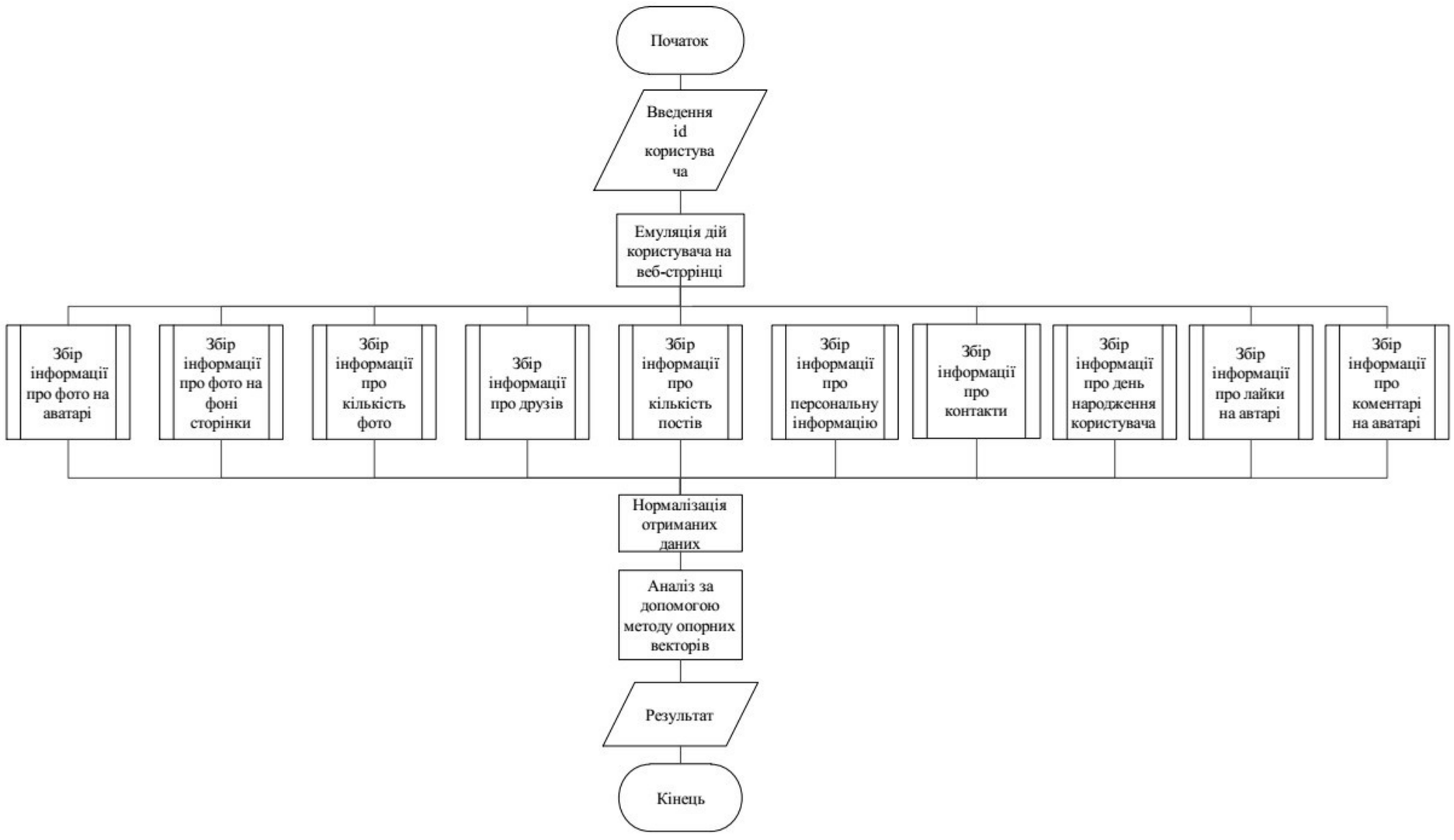


Рисунок 2.12 – Схема роботи програмного засобу

Як показник обрано систему балів, що свідчить про фейковість облікового запису користувача. Кожен з параметрів під час аналізу залежно від умови отримує певну кількість балів від 0 до 2 (таблиця 2.1).

Таблиця 2.1 – Критерії встановлення балів відповідно до значень метрик

Бали / Параметр	0	1	2
Наявність фото на аватарі	На фото зображена людина	Фото існує	Фото не існує
Наявність фото на фоні	Фото існує	-	Фото не існує
Кількість фотографій	Фото не існує	Кількість фото < 10 & Кількість фото > 1000	10 < Кількість фото < 1000
Кількість друзів	10 < Кількість друзів < 2000	0 < Кількість друзів < 10 & Кількість друзів > 2000	Друзів немає
Кількість постів	10 < Кількість постів < 500	Кількість постів > 500 & Кількість постів < 10	Пости не існують
Наявність персональної інформації	Всі поля заповнено	Частина полів заповнено	Інформація відсутня
Контактна інформація	Контакти присутні	-	Контакти відсутні
Дата народження	1932 < Рік народження < 2009	2009 < Рік народження || Рік народження < 1932	Рік народження відсутній
Лайки на аватарі	Кількість лайків	Лайки від друзів < Лайки від чужинців	Лайки відсутні
Кількість коментарів на аватарі	5 < Кількість коментарів < 100	Кількість коментарів < 5 & Кількість коментарів > 100	Коментарі відсутні

Отримані значення для кожного параметру формуються у масив даних, який подається на вхід нейронної мережі, яка, у свою чергу, аналізує їх та видає результат. Залежно від ситуації, існує необхідність проведення додаткових досліджень за участю експертів та з урахуванням інформації, що міститься у гістограмі, для найточнішого визначення статусу облікового запису. Для цього необхідно перевірити кожен зі стовпців та визначити їх рівень впливу. Всього є 3 рівня впливу (від 0 до 2), які вказують наскільки той чи інший показник

вплинув на висновок щодо фейковості облікового запису. Рівні впливу мають такі значення:

– 0 – не має впливу;

– 1 – має незначний вплив;

– 2 – має значний вплив;

Наприклад, якщо стовпець діаграми сягнув 2 рівня, це означає, що параметр, який характеризує даний стовпець, значно вплинув на фейковість облікового запису. У іншому випадку, якщо стовпець знаходиться на 0 рівні, це означає, що параметр відповідає такому, який притаманний справжньому обліковому запису. Таким чином, опираючись на рівні кожного з параметрів, можна зробити висновок про фейковість чи справжність конкретного облікового запису.

Отже, проаналізовано можливу інформацію, що міститься в облікових записах у соціальні мережі «Facebook», а також виокремлено основні метрики облікових записів та віднесено їх до відповідних категорій. Розроблено модель системи підтримки прийняття рішень з використанням нейромережі, оскільки вона підходить для роботи з нечіткими множинами та великою кількістю даних. Розглянуто структуру нейрона та принцип роботи нейронної мережі. Розроблено алгоритм, архітектуру та схему роботи програмного засобу. Розподілено ступені фейковості облікових записів у вигляді балів відповідно до значень метрик.

3 ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ

3.1 Розробка програмного засобу для досліджень

Проект програмного засобу складається з таких файлів (рис. 3.1):

– з файлу chromedriver.exe, який слугує інтерфейсом між браузером та драйвером та дозволяє зчитувати дані з веб-сторінок;

– з файлу decision_support_system.py, у якому реалізовано систему підтримки прийняття рішень на основі нейронної мережі;

– fake_detector.py, який є головним виконуваним файлом програми;

– fake_detector.uml з UML-діаграмою проекту, у якій міститься інформація про класи програми, методи та їх вхідні параметри, а також інформація про взаємодію між ними;

– з файлу Legend.JPG з легендою для гістограми;

– з файлу training.csv, де зберігається навчальна вибірка для нейромережі у вигляді масиву числових даних, що є нормалізованим результатом зчитування та обробки даних про обліковий запис та готовий для подальшого аналізу;

– з файлу database.csv, у якому міститься інформація про усі облікові записи, що аналізувалися;

– з допоміжного файлу account.csv, де зберігається інформація про обліковий запис, що перевіряється, у нормалізованому вигляді.

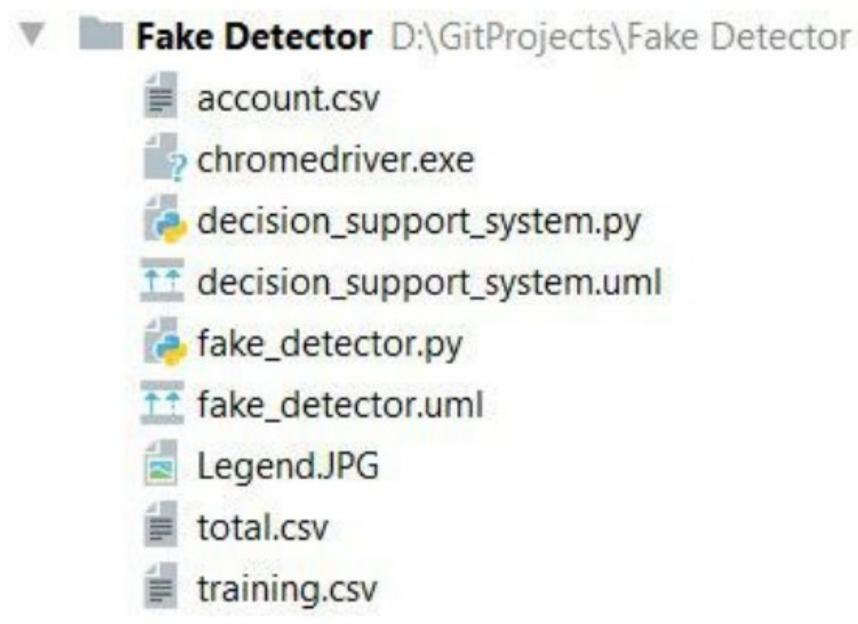


Рисунок 3.1 – Структура проекту програмного засобу у середовищі програмування PyCharm

Програмний засіб складається з двох файлів (fake_detector.py та decision_support_system.py), кожен з яких, у свою чергу, складається з класів та методів.

Файл fake_detector.py відповідає за графічний інтерфейс програми, зчитування інформації про обліковий запис, її нормалізацію, а також виведення результатів роботи системи підтримки прийняття рішень на екран. Структура файлу fake_detector.py (усі модулі та методи) зображено на UML-діаграмі проекту (рис. 3.2). На UML-діаграмі видно, що фалй складається з трьох класів, кожен з яких за замовчуванням є похідним від класу object, що є особливістю мови Python. З діаграми видно, що класи Scrapper та Analyzer містять ряд методів, для кожного з яких вхідним параметром є person_id.

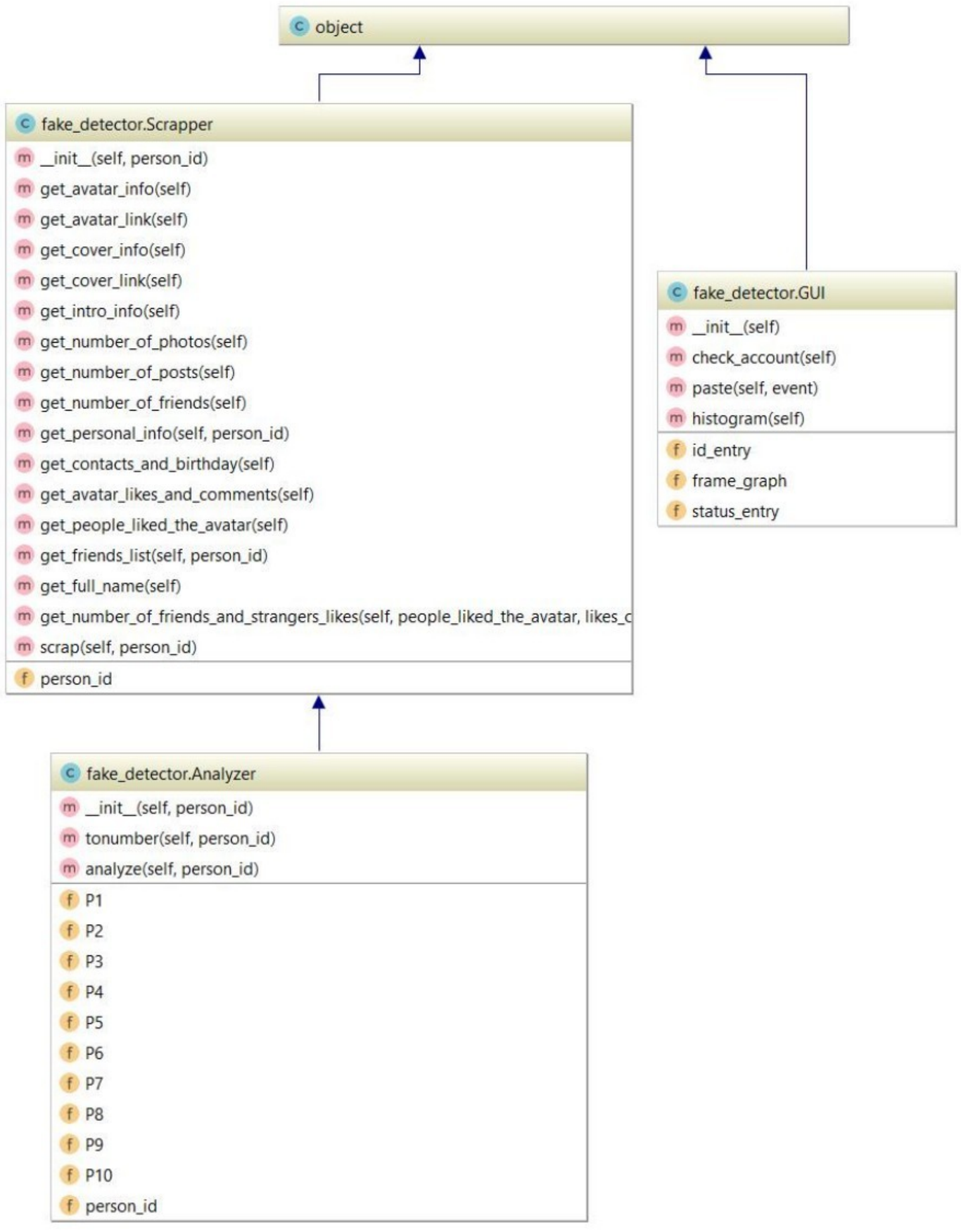


Рисунок 3.2 – UML-діаграма файлу fake_detector.py

Як видно з UML-діаграми (див. рис. 3.2), клас Scrapper, що відповідає за збір інформації про обліковий запис користувача та складається з ряду методів, кожен з яких відповідає за збір тієї чи іншої частини інформації. Так, клас Scrapper містить в собі 16 методів, серед яких 10 основних і 6 допоміжних: get_avatar_info(), get_avatar_link(), get_cover_info(), get_cover_link(), get_intro_info(), get_number_of_photos(), get_number_of_posts(), get_number_of_friends(), get_personal_info(), get_contacts_and_birthday(), get_avatar_likes_and_comments(), get_people_liked_the_avatar(), get_friends_list(), get_full_name(), get_number_of_strangers_likes() та scrap(). Для пошуку необхідної інформації на сторінці користувача використовується бот, який емулює дії користувача у Facebook. Це відбувається за використанням виконуючого файлу chromedriver.exe, який працює з браузером Google Chrome.

Клас GUI формує собою графічний інтерфейс програмного засобу за допомогою вбудованого модуля tkinter, а також представляє легенду для гістограми залежності. Метод check_account() описує функціонал кнопки «Check page» та відповідає за процес створення об'єкту класу Analyzer; метод paste() є допоміжним методом, що реалузіє можливість вставки id користувача за допомогою комбінації клавіш «Ctrl+V» для зручності роботи з програмою; метод histogram(), який відповідає за створення об'єкту класу NeuralNetwork та обчислення результату за допомогою нейронної мережі, а також представлення її користувачеві програми у вигляді статусу та гістограми залежності параметрів від результату.

У класі Analyzer, який наслідується від класу Scrapper, міститься метод tonumber(), який відповідає за перетворення інформації про обліковий запис у числовий формат, зрозумілий для нейронної мережі; метод analyze(), який формує нормалізовані дані та записує їх у файл training.csv для подальшого аналізу.

Гістограма формується зі значень кожного з методів з урахуванням відповідних коефіцієнтів. Стовпці гістограми для зручності подання

зафарбовуються у різні кольори відповідно до категорій метрик. Так, параметри Avatar та Cover зафарбовуються помаранчевим кольором; параметри Number of photos, Number of friends і Number of posts зафарбовуються зеленим кольором; параметри Personal Info, Contact Info та Birthday зафарбовуються синім кольором; параметри Likes on avatar і Comments on avatar зафарбовуються червоним кольором.

Для побудови гістограми було обрано параметри, такі як розмір та щільність гістограми, а також розміщення її у вікні програмного засобу. Для цього виконуються такі дії:

```
figure = Figure(figsize=(6, 3), dpi=100)
ax = figure.add_subplot(111)
barchart = ax.bar(y_pos, mas, align='center', alpha=0.5)
ax.yaxis.grid()
canvas = FigureCanvasTkAgg(figure, master=frame_graph)
canvas.draw()
```

Тут *mas* – список значень, повернених з методів, що відповідають за збір та обробку інформації про обліковий запис, *barchart* – ряд стовпців для кожного з параметрів, canvas – змінна для створення поля, на якому відображається гістограма.

Файл decision_support_system.py містить один клас NeuralNetwork, який складається з двох методів training() та prediction() (рис. 3.3).

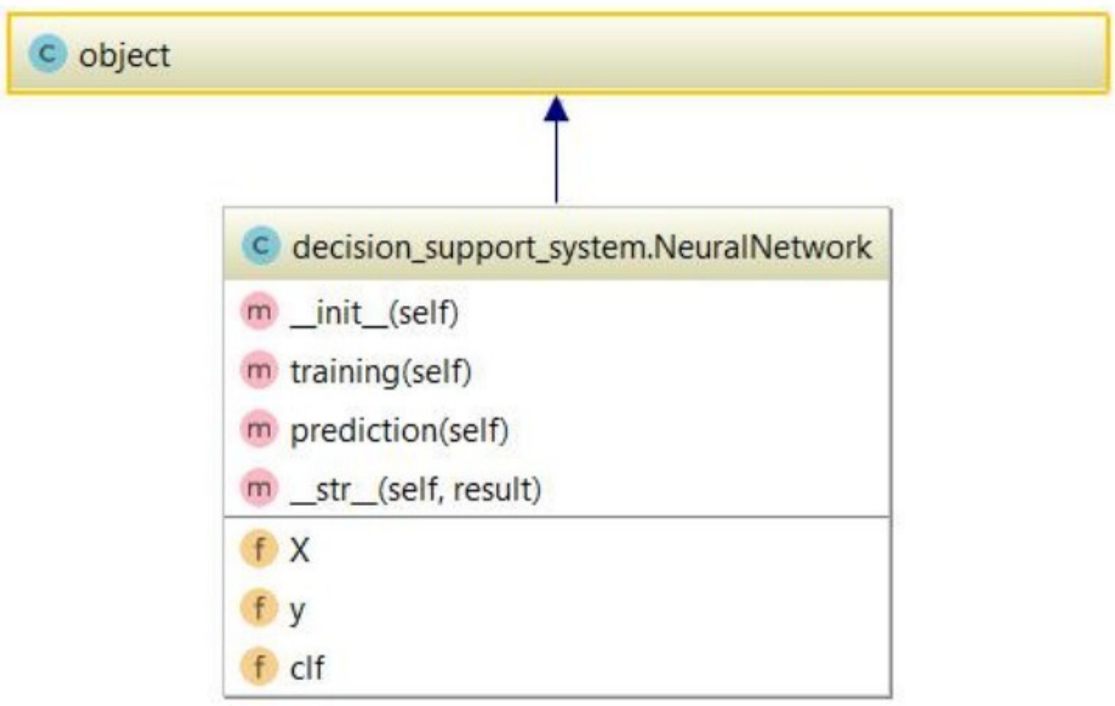


Рисунок 3.3 – UML-діаграма файлу decision_support_system.py

Метод training() відповідає за навчання нейронної мережі за допомогою набору даних, який міститься у файлі training.csv (рис. 3.4). Для цього використовується функція fit():

clf.fit(X_train, y_train.ravel())

Також у методі training() відбувається процес розподілу даних на навчальну і тестову вибірки:

X_train, X_test, y_train, y_test = train(X, y, test_size=0.6)

Безпосередньо запуск процесу аналізу облікового запису на предмет його фейковості відбувається за допомогою функції predict(), що знаходиться у методі prediction():

predicted = clf.predict(row_test)

Набір даних складється з навчальної та тестової вибірки, розподіленої за допомогою методу train() у співвідношенні 60%:40%.

Link	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	Status
https://www.facebook.com/100011341317183	0	0	1	0	1	1	2	2	0	0	Real
https://www.facebook.com/zuck	0	0	0	2	1	0	2	0	0	1	Fake
https://www.facebook.com/100027339220625	0	0	0	0	0	1	2	2	0	2	Real
https://www.facebook.com/james.clemons.5688	1	2	1	2	1	1	2	2	1	1	Fake
https://www.facebook.com/ford.krieger	0	0	1	2	1	1	2	2	1	1	Fake
https://www.facebook.com/OneToBeCherished	0	0	0	2	0	1	2	2	1	1	Fake
https://www.facebook.com/100011592244012	0	0	0	0	0	1	2	2	0	1	Real
https://www.facebook.com/Draugdin	0	0	0	0	0	1	2	2	0	0	Real
https://www.facebook.com/Amber.Seward.33	1	0	0	0	1	1	2	2	0	2	Real
https://www.facebook.com/richard.grove.796	0	0	0	0	0	0	2	2	0	0	Real

Рисунок 3.4 – Вигляд файлу training.csv

Набір даних складається з посилань на досліджені облікові записи, числової інформації про реальні облікові записи у соціальній мережі «Facebook» (колонки від P1 до P10), а також статусів, що відповідають цим обліковим записам (колонка Status), де:

- P1 - наявність фото на аватарі;
- P2 - наявність фото-обкладинки на сторінці;
- P3 - кількість фотографій;
- P4 - кількість друзів;

- P5 - кількість постів;
- P6 - персональна інформація про користувача:
    - місце роботи;
    - освіта;
    - місто проживання/народження;
    - сімейний стан;
    - P7 - контактна інформація;
    - P8 - дата народження;
    - P9 - кількість лайків на аватарі;
    - P10 - кількість коментарів на аватарі.

Набір даних містить інформацію про 350 існуючих облікових записів. Навчання нейронної мережі відбувається за допомогою метода fit():

clf.fit(X_train, y_train.ravel())

Після навчання нейронної мережі відображається параметр accuracy, що вказує на точність навчання та достовірність подальших результатів.

Метод prediction() відповідає за обробку інформації, що надходить з соціальної мережі «Facebook» з використанням розставлених вагових коефіцієнтів вже «навченою» нейронною мережею. Обчислення результату відбувається за допомогою методу predicted():

predicted = clf.predict(row_test)

Таким чином отриманий результат передається до файлу fake_detector.py та виводиться програмою на екран.

Для того, щоб отримати дані з соціальної мережі «Facebook» за допомогою бібліотеки Selenium необхідно увійти у будь-який (тестовий) обліковий запис Facebook, здійснити запит до веб-сторінки користувача, що досліджується, після чого бот переходить по розділам сторінки та збирає усі необхідні дан про обліковий запис, після чого вони формуються у єдиний об’єкт та записуються до бази даних. Для обробки деяких даних та зручності

подальшого аналізу вони попередньо обробляються за допомогою регулярних виразів.

У результаті обробки нормалізованих даних нейронною мережею програмного засобу на екран виводиться інформація про фейковість чи справжність облікового запису користувача. Вікно програмного засобу містить такі елементи:

– поле для введення ідентифікатора користувача у соціальній мережі «Facebook» [33];

– кнопка, при натисканні на яку відбувається процес перевірки облікового запису;

– висновок програми у вигляді статусу;

– гістограма, у якій містяться показники соціальної мережі та їх вплив на результат перевірки облікового запису.

Програмний засіб аналізує такі параметри: наявність фото на аватарі, наявність фото на фоні обкладинці сторінки, кількість фотографій, кількість друзів, кількість постів, персональна інформація про користувача (місце роботи, освіта, місто проживання/народження, сімейний стан), контактна інформація, дата народження, кількість лайків на аватарі, кількість коментарів на аватарі. Таким чином, аналіз цих параметрів дозволить зробити висновок щодо фейковості облікових записів.

3.2 Тестування програмного засобу

Для тестування роботи програмного засобу розроблено тестові випадки для перевірки поведінки програми під час виключних ситуацій. До таких ситуацій належать введення коректного та некоректного id користувача, реакція програми на присутність та відсутність файлу training.csv, а також перевіркароботи програми з коректними та некоректними даними у файлі training.csv (табл. 3.1).

Таблиця 3.1 – Тестові випадки до програми

№	Назва тестового випадку	Очікувана дія програми	Чи пройшла програма тест?
1	Введення коректного id користувача	Збір інформації про обліковий запис	Так
2	Введення некоректного id користувача	Повідомлення про помилку	Так
3	Навчання нейронної мережі, коли файл training.csv існує	Навчання нейронної мережі	Так
4	Аналіз облікового запису, коли файл training.csv не існує	Повідомлення про помилку	Так
5	Навчання нейронної мережі за тестовою вибіркою, що містить коректні дані	Навчання нейронної мережі	Так
6	Навчання нейронної мережі за тестовою вибіркою, що містить некоректні дані	Повідомлення про помилку	Так

Вікно програмного засобу містить такі графічні елементи:

– кнопка «Check page», при натисканні на яку відбувається процес перевірки облікового запису;

– текстове поле «User id:» для введення ідентифікатор користувача;

– текстове поле «Status:» з висновком програми у вигляді статусу;

– гістограма, у якій містяться показники соціальної мережі та їх вплив на результат перевірки облікового запису.

Так, на рисунку 3.5 зображено вікно програми після перевірки облікового запису, що має id користувача emilyipearce, на фейковість. Обліковий запис є справжнім. У результаті перевірки програмний засіб розпізнав обліковий запис як справжній. Про це свідчить статус The page is real, а також діаграма, за якою видно, що на сторінці користувача міститься основна інформація та фото, а також присутня персональна інформація, проте у нього прихований список друзів, а також на аватарі користувача скоріш за все розміщене не його фотографія.

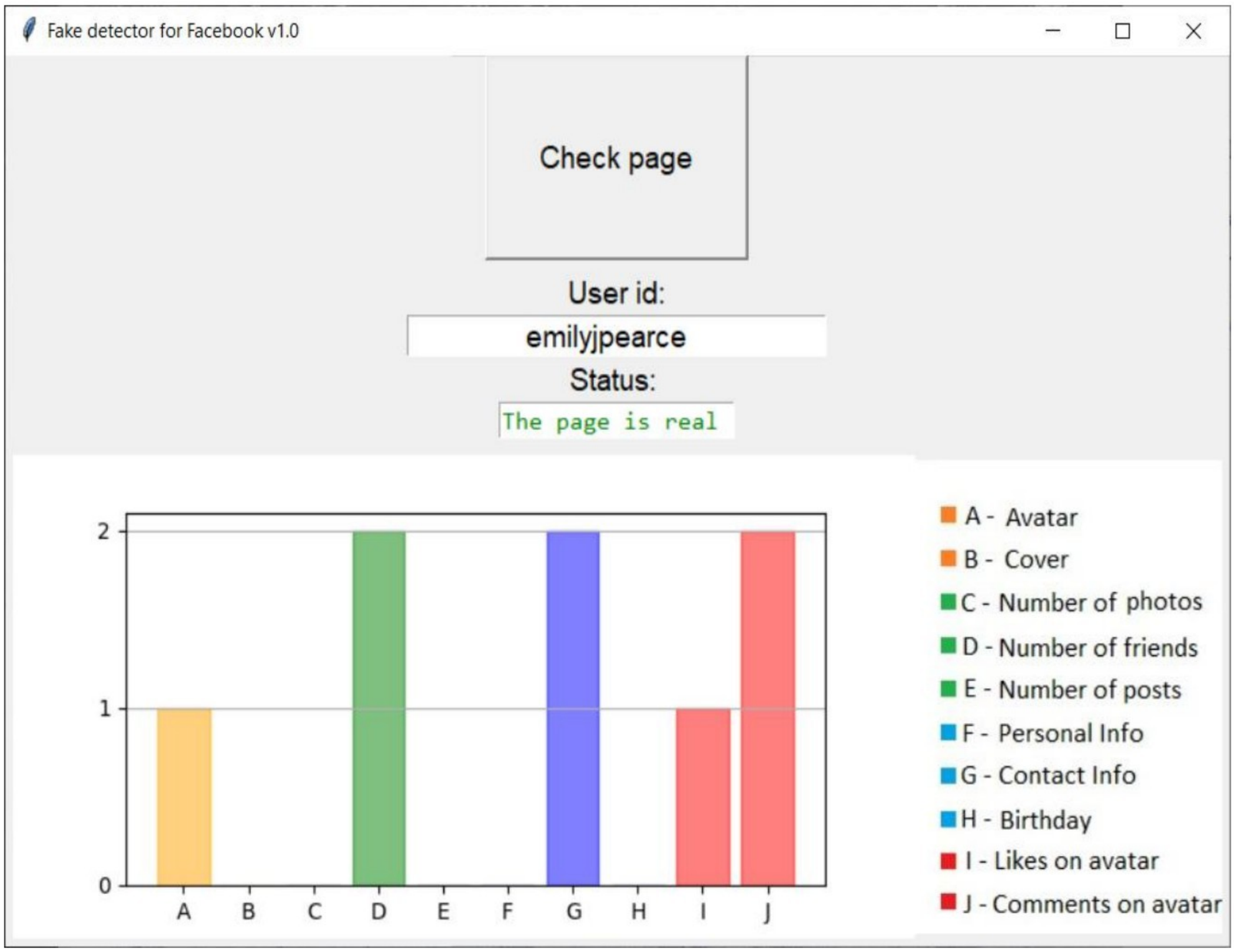


Рисунок 3.5 – Вигляд вікна програмного засобу під час перевірки справжнього облікового запису

Також для перевірки роботи програмного засобу обрано фейковий обліковий запис, що має id користувача alinka.mykolaichuk, на якому відсутня майже вся інформація про користувача, є незначна кількість постів, інформація про дату народження, а також користувач є майже неактивним. Хоча на сторінці присутній аватар та обкладинка профіля, проте програмний засіб визначив даниий обліковий запис як фейковий, про що свідчить статус The page is fake (рис. 3.6).

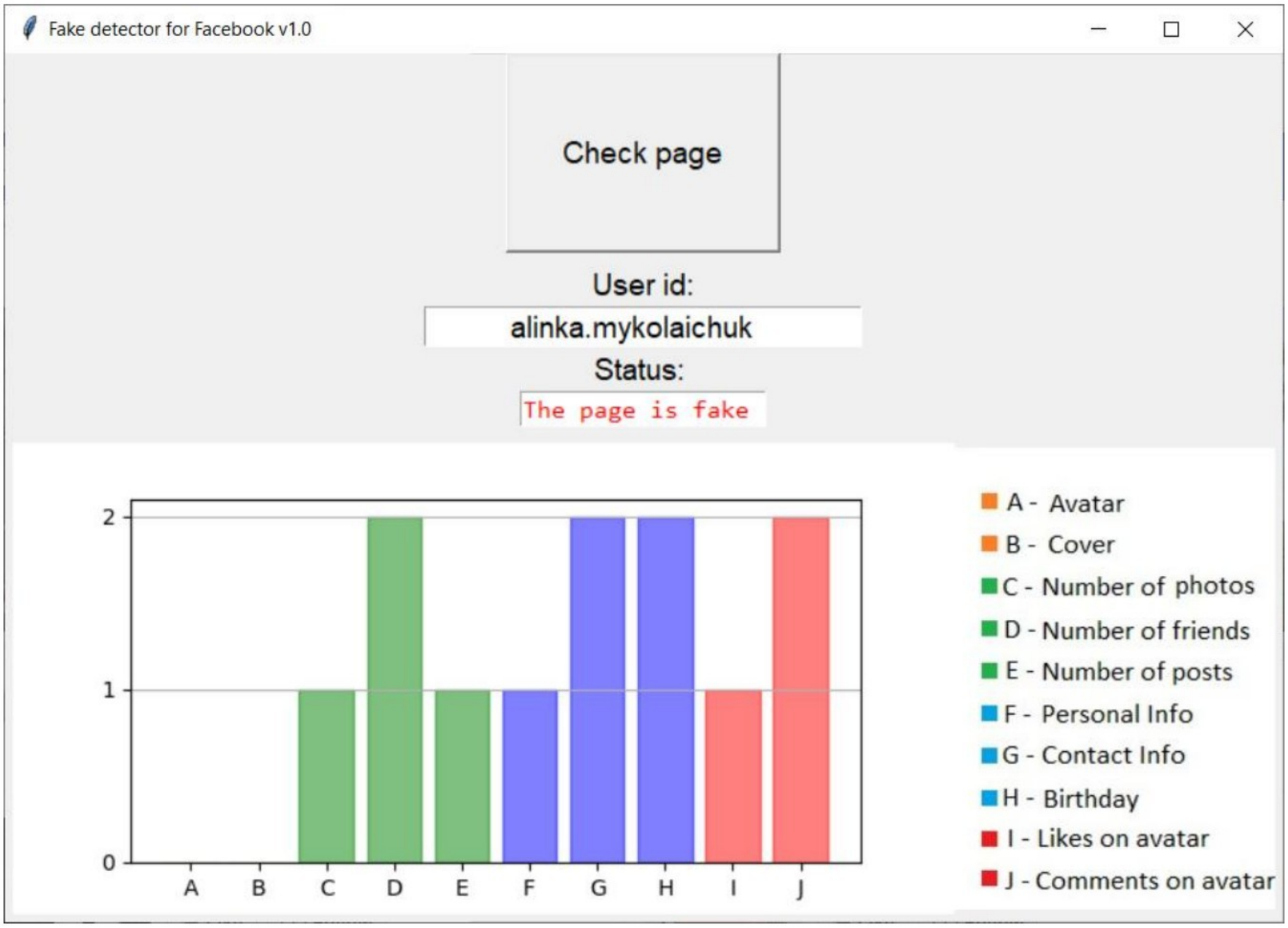


Рисунок 3.6 – Вигляд вікна програмного засобу під час перевірки фейкового облікового запису

У разі введення некоректного id користувача або пустого поля «User id:» програмний засіб видає повідомлення про помилку, що необхідно ввести коректний id користувача (рис. 3.7).

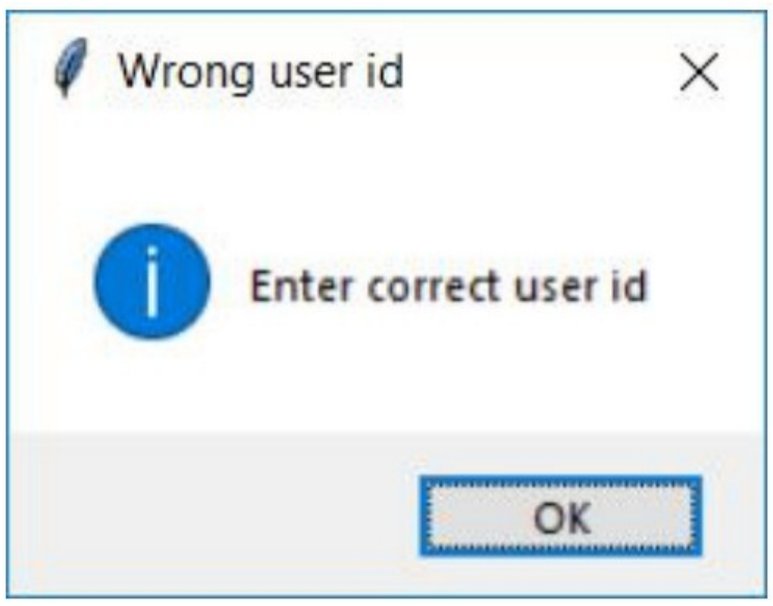


Рисунок 3.7 – Повідомлення про помилку введення некоректного id користувача

Також можлива ситуація, коли файл з тренувальною вибіркою нейронної мережі training.csv пошкоджено або видалено. У такому разі програма видає повідомлення про помилку, що файл training.csv не існує (рис. 3.8).

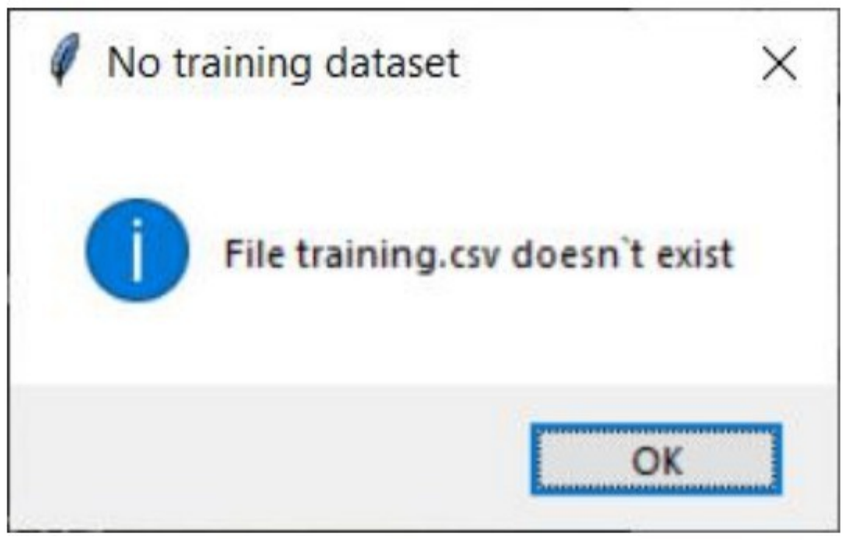


Рисунок 3.8 – Повідомлення про помилку введення некоректного id користувача

Програмний засіб аналізує такі параметри: кількість друзів, наявність фото на аватарі, відмітки на фото з користувачем, фото на фоні сторінки, кількість постів, дата народження, персональна інформація про користувача, оновлення сторінки, ім’я користувача, кількість лайків, поставлених користувачем та кількість лайків на постах користувача. Таким чином, аналіз цих параметрів дозволить зробити висновок щодо фейковості облікових записів.

Отже, перевірено роботу основного функціоналу програмного засобу та протестовано його роботу під час виникнення виключних ситуацій та помилок, таких як відсутність навчального файлу для нейромережі, а також під час введення некоректного id користувача.

3.2 Оцінка ефективності програмного засобу

Ефективність розробленого програмного засобу оцінюється за правильністю виявлення нейронною мережею фейкових та справжніх облікових записів та за мінімальною кількістю помилок першого та другого роду. Для цього перевірено 100 облікових записів. Результати аналізу 20 облікових записів зі 100 наведено у таблиці 3.2.

Таблиця 3.2 – Результати аналізу облікових записів користувачів

Користувач	Статус облікового запису	Висновок програми
Vitalii Holovenko	Справжній	Справжній
Татьяна Головенько	Фейк	Фейк
Oleksandr Topchii	Фейк	Фейк
IvanVorobyov	Справжній	Справжній
Alex Rudyk	Фейк	Фейк
Ольга Гнатюк	Справжній	Справжній
Petro Petrovich	Фейк	Фейк
Andrii Beatle	Справжній	Справжній
Den Ivanov	Фейк	Фейк
Олеся Войтович	Справжній	Справжній
Talii Santie	Фейк	Фейк
Jenny Rahl	Справжній	Справжній
Sergey Hubchakevych	Справжній	Справжній
Zheka Oleynik	Справжній	Фейк
Alice Black	Фейк	Фейк
Ira Vinn	Фейк	Справжній
Dimon Anipchenko	Фейк	Фейк
Сергей Тарахта	Справжній	Справжній
Владислав Караван	Справжній	Справжній
Jessica Dowling	Справжній	Справжній

Після аналізу необхідно перевірити достовірність роботи програми, визначивши помилки першого та другого роду [34]. Для цього застосовано перевірку гіпотез та введено два поняття: нульова гіпотеза H_0 та альтернативна гіпотеза H_1. Нульова гіпотеза H_0 - це гіпотеза про те, що обліковий запис є справжнім, а альтернативна гіпотеза H_1 - обліковий запис є фейковим. Таким чином, якщо приймається нульова гіпотеза H_0, то робиться висновок, що обліковий запис справній. Якщо приймається альтернативна гіпотеза H_1, то робиться висновок про те, що обліковий запис є фейковим (табл. 3.3).

Таблиця 3.3 – Помилки першого і другого роду під час прийняття рішень

	H_0 приймається	H_0 не приймається, H_1 приймається
H_0 вірна	Правильно визначений справжній обліковий запис	Фейковий обліковий запис визначено як справжній
H_0 не вірна, H_1 вірна	Справжній обліковий запис визначено як фейковий	Правильно визначений фейковий обліковий запис

При перевірці гіпотези експериментальні дані можуть суперечити гіпотезі H_0, тоді ця гіпотеза відхиляється. В іншому випадку, якщо експериментальні дані не суперечать гіпотезі H_0, то ця гіпотеза не відхиляється.

Зрозуміло, що при перевірці гіпотез система прийняття рішень може з деякою імовірністю прийняти помилкове рішення. Існують такі помилки: помилка першого роду та помилка другого роду.

Помилка першого роду означає, що приймається рішення відхилити гіпотезу H_0, хоча в дійсності вона є вірною. Помилка другого роду означає, що приймається рішення не відхиляти гіпотезу H_0, хоча в дійсності вона буде невірна.

Помилки першого та другого роду для системи виявлення фейкових облікових записів становлять:

- помилка першого роду $\beta = 0{,}04$;
- помилка другого роду $\alpha = 0{,}02$.

Тому, для оцінки роботи системи виявлення фейкових облікових записів у соціальних мережах доцільно визначити імовірності правильного визначення фейкових та справжніх облікових записів, а також імовірності появи помилок першого та другого роду.

Імовірність правильно визначеного справжнього облікового запису становить $P_{H_0}(H_0) = 1 - \alpha$ і дорівнює 98%.

Імовірність правильно визначеного фейкового облікового запису становить $P_{H_1}(H_1) = 1 - \beta$ і дорівнює 96%.

Імовірність появи помилки 1-го роду становить $P_{H_1}(H_0) = \beta$ і дорівнює 4%.

Імовірність появи помилки 2-го роду становить $P_{H_0}(H_1) = \alpha$ і дорівнює 2%.

Перевірка точності роботи системи виявлення фейкових облікових записів у соціальній мережі «Facebook» показала, що помилки першого та другого роду дорівнюють 0,04 і 0,02 відповідно, а отже – система доволі точно виявляє фейкові та справжні облікові записи. Загальна достовірність системи

виявлення фейкових облікових записів відповідно до отриманих результатів становить $P(H) = 1 - \alpha - \beta = 0,94 = 94\%$.

Очевидно, що система підтримки прийняття рішень може помилитися при ухваленні рішення. Виключити помилки повністю при прийнятті гіпотез неможливо, тому необхідно мінімізувати можливі наслідки прийняття неправильного рішення (невірної статистичної гіпотези). Тому, для мінімізації можливих помилок у подальших дослідженнях необхідно збільшити обсяг вибірки.

Отже, розроблено програмний засіб, який реалізує процес виявлення фейкових облікових записів у соціальній мережі «Facebook». Проект складається з двох файлів, які, у свою чергу, складаються з класів і методів, а також з допоміжних файлів, у яких містититься навчальна вибірка для нейронної мережі та файл з базою даних про користувачів, що аналізуються. У результаті тестування виявлено, що програмний засіб майже завжди правильно виявляє фейки та справжні сторінки, про що свідчить достовірність прийняття рішення, яка становить 94%. Із 100 досліджуваних облікових записів програма правильно визначила статус для 94 облікових записів. При цьому, помилки першого та другого роду становлять 4% і 2% відповідно. Для поліпшення результату роботи програмного засобу необхідно збільшити розмір навчальної вибірки, розширити кількість параметрів, що перевіряються, а також розробити алгоритм перевірки зв’язків друзів користувача, частоту оновлення облікового запису користувачем та спільнот і інтересів користувача. Також необхідно розробити функції, що дозволять аналізувати текстову інформацію на сторінці користувача з використанням вдосконаленого семантичного аналізу.

4 ЕКОНОМІЧНА ЧАСТИНА

4.1 Аналіз комерційного потенціалу розробки (технологічний аудит розробки) системи виявлення фейкових облікових записів у соціальних мережах

4.1.1 Визначення рівня комерційного потенціалу розробки системи виявлення фейкових облікових записів у соціальних мережах

Метою проведення технологічного аудиту є оцінювання комерційного потенціалу розробки системи виявлення фейкових облікових записів у соціальних мережах. В результаті оцінювання можна буде зробити висновок щодо напрямів (особливостей) організації подальшого її впровадження з врахуванням встановленого рейтингу.

Для проведення технологічного аудиту залучимо 3-х незалежних експертів. У нашому випадку такими експертами будуть керівник магістерської роботи та провідні викладачі випускової та споріднених кафедр.

Оцінювання комерційного потенціалу розробки системи виявлення фейкових облікових записів у соціальних мережах будемо здійснювати за 12-ю критеріями згідно рекомендацій.

Результати оцінювання комерційного потенціалу розробки системи виявлення фейкових облікових записів у соціальних мережах заносимо до таблиці 4.1.

За даними таблиці 4.1 робимо висновок щодо рівня комерційного потенціалу розробки системи виявлення фейкових облікових записів у соціальних мережах. При цьому користуємося рекомендаціями, наведеними в таблиці 4.2.

Таблиця 4.1. - Результати оцінювання комерційного успіху розробки системи виявлення фейкових облікових записів у соціальних мережах

Критерії	Експерти		
	к.т.н., доцент Войтович О.П.	к.т.н., ст. викл Лукічов В. В.	д. т. н., проф., Лужецький В.А.
	Бали, виставлені експертами		
1	2	2	3
2	3	3	2
3	4	4	3
4	4	2	3
5	4	3	3
6	4	4	3
7	3	3	2
8	3	3	3
9	3	4	4
10	2	3	3
11	3	3	3
12	2	3	3
Сума балів	37	37	35
Середньоарифметична сума балів, СБ	36		

Таблиця 4.2 – Рівні комерційного потенціалу розробки

Середньоарифметична сума балів, розрахована на основі висновків експертів	Рівень комерційного потенціалу розробки
0 – 10	Низький
11 – 20	Нижче середнього
21 – 30	Середній
31 – 40	Вище середнього
41 – 50	Високий

Таким чином, робимо висновок, щодо рівня комерційного потенціалу нашої розробки системи виявлення фейкових облікових записів у соціальних мережах – вище середнього.

4.1.2 Визначення рівня якості розробки системи виявлення фейкових облікових записів у соціальних мережах

Оцінювання якості розробки системи виявлення фейкових облікових записів у соціальних мережах проводиться з метою порівняльного аналізу і

визначення найбільш ефективного, з технічної точки зору, варіанта інженерного рішення.

Рівень якості – це кількісна характеристика міри придатності певного виду продукції для задоволення конкретного попиту на неї при порівнянні з відповідними базовими показниками за фіксованих умов споживання.

Абсолютний рівень якості розробки системи виявлення фейкових облікових записів у соціальних мережах знаходимо обчисленням вибраних для її вимірювання показників, не порівнюючи їх із відповідними показниками аналогічних виробів. Для цього необхідно визначити зміст основних функцій, які повинні реалізовувати розробка, вимоги замовника до неї, а також умови, які характеризують експлуатацію, визначають основні параметри, які будуть використані для розрахунку коефіцієнта технічного рівня виробу. Система параметрів, прийнята до розрахунків, повинна достатньо повно характеризувати споживчі властивості інноваційного товару (його призначення, надійність, економічне використання ресурсів, стандартизація тощо).

Далі визначаємо величину параметрів якості в балах та встановлюємо граничні його значення (кращі, гірші, середні). Всі ці дані для кожного параметра заносимо в табл. 4.3.

Таблиця 4.3 – Основні параметри системи виявлення фейкових облікових записів у соціальних мережах

Параметри	Абсолютне значення параметра			Коефіцієнт вагомості параметра
	Краще +5…+4	Середнє +3	Гірше +1…+2	
Кількість соціальних мереж			1	0,1
Кількість параметрів, що аналізуються	4			0,5
Достовірність прийняття рішення		3		0,3
Швидкість роботи			2	0,1

Із врахуванням коефіцієнтів вагомості відповідних параметрів можна визначити абсолютний рівень якості інноваційного рішення за формулою 4.1:

$$K_{я.а.} = \sum_{i=1}^{n} P_{H_i} \cdot a_i \qquad (4.1)$$

де Рні – числове значення і-го параметра інноваційного рішення, n – кількість параметрів інноваційного рішення, що прийняті для оцінювання, аі – коефіцієнт вагомості відповідного параметра (сума коефіцієнтів вагомості всіх параметрів повинна дорівнювати 1).

Отже, абсолютний рівень якості системи виявлення фейкових облікових записів у соціальних мережах становитиме – 3,2 бали.

Одночасно визначаємо відносний рівень якості системи виявлення фейкових облікових записів у соціальних мережах, що виробляється (проектується), порівнюючи її показники з абсолютними показниками якості найліпших аналогів (товарів-конкурентів) (табл. 4.4).

Таблиця 4.4 – Основні параметри системи виявлення фейкових облікових записів у соціальних мережах та товару-конкурента

Параметри	Варіанти		Відносний показник якості	Коефіцієнт вагомості параметра
	Базовий (конкурент)	Новий		
Кількість соціальних мереж	2	1	2	0,1
Кількість параметрів, що аналізуються	4	10	2,5	0,5
Достовірність прийняття рішення	75%	90%	1,2	0,3
Швидкість роботи	2	2	1	0,1

Відносний рівень якості методу та засобу завадостійкого розподілу секрету визначаємо за формулою 4.2:

$$K_{я.в.} = \sum_{i=1}^{n} q_i \cdot a_i \qquad (4.2)$$

За розрахунками відносний рівень якості системи виявлення фейкових облікових записів у соціальних мережах становитиме – 1,91. Це означає, що наша розробка краща за якістю на 91% від товару-аналога.

4.1.3 Визначення конкурентоспроможності розробки системи виявлення фейкових облікових записів у соціальних мережах

У найширшому розумінні конкурентоспроможність товару – це можливість його успішного продажу на певному ринку і в певний проміжок часу. Водночас конкурентоспроможною можна вважати лише однорідну продукцію з технічними параметрами і техніко-економічними показниками, що ідентичні аналогічним показникам уже проданого товару. Для того, щоб високоякісний товар був одночасно і конкурентоспроможним, він має відповідати критеріям оцінювання споживачів конкретного ринку в конкретний період часу.

Дані для розрахунку загального показника конкурентоспроможності розробки необхідно занести до таблиці 4.5.

Таблиця 4.5 – Нормативні, технічні та економічні параметри системи виявлення фейкових облікових записів у соціальних мережах і товару-конкурента

Параметри	Варіанти		Відносний показник якості	Коефіцієнт вагомості параметра
	Базовий (конкурент)	Новий		
Кількість соціальних мереж	2	1	2	0,1
Кількість параметрів, що аналізуються	4	10	2,5	0,5
Достовірність прийняття рішення	75%	90%	1,2	0,3
Швидкість роботи	2	2	1	0,1
Ціна за продукт, тис. грн.	25000	20000	0,8	-

Загальний показник конкурентоспроможності розробки (К) з урахуванням вищезазначених груп показників визначаємо за формулою 4.3:

$$K = \frac{I_{Т.П.}}{I_{Е.П.}} = \frac{1,91}{0,8} = 2,4, \quad (4.3)$$

де Іт.п. – індекс технічних параметрів (відносний рівень якості інноваційного рішення); Іе.п. – індекс економічних параметрів (4.4).

$$I_{Е.П.} = \frac{РН_{ЕІ}}{РБ_{ЕІ}} = \frac{20000}{25000} = 0,8, \quad (4.4)$$

де $РН_{ЕІ}$, $РБ_{ЕІ}$ – економічні параметри (ціна придбання та споживання товару) відповідно нового та базового товарів.

Згідно розрахунків загальний показник конкурентоспроможності – 2,4 . Це означає, що наша розробка системи виявлення фейкових облікових записів у соціальних мережах більш конкурентна на 140% від товару-аналога.

4.2 Прогнозування витрат на виконання науково-дослідної, дослідно-конструкторської та конструкторсько-технологічної роботи

4.2.1 Розрахунок витрат, що стосуються виконавців розробки системи виявлення фейкових облікових записів у соціальних мережах

Основна заробітна плата кожного із розробників (дослідників) Зо, якщо вони працюють в наукових установах бюджетної сфери (4.5):

$$З_О = \frac{М}{Т_Р} \cdot t, \quad (4.5)$$

де М – місячний посадовий оклад конкретного розробника (інженера, дослідника, науковця тощо), грн.

У 2019 році величини окладів (разом з встановленими доплатами і надбавками) рекомендується брати в межах (5000…10000) грн. за місяць; Тр – число робочих днів в місяці; приблизно Тр = (21…23) дні; t – число робочих днів роботи розробника (дослідника).

Зроблені розрахунки зводимо до таблиці 4.6.

Таблиця 4.6 – Заробітна плата розробників

Посада	Місячний посадовий оклад, грн.	Оплата за робочий день, грн.	Число днів роботи	Витрати на заробітну плату, грн.
Керівник	25000	1190	15	17850
Інженер-програміст	15000	714	14	9996
Тестувальник	12000	571	1	571
Всього:				28417

Основна заробітна плата робітників Зр, якщо вони беруть участь у виконанні даного етапу роботи і виконують роботи за робочими професіями у випадку, коли вони працюють в наукових установах бюджетної сфери, розраховується за формулою 4.6:

$$З_P = \sum_{i=1}^{n} t_i \cdot C_i \qquad (4.6)$$

де ti – норма часу (трудомісткість) на виконання конкретної роботи, годин; n – число робіт по видах та розрядах; Сі – погодинна тарифна ставка робітника відповідного розряду, який виконує дану роботу. Сі визначається за формулою 4.7:

$$C_i = \frac{M_M \cdot K_i}{T_P \cdot T_{ЗМ}}, \qquad (4.7)$$

де Мм – розмір мінімальної заробітної плати за місяць, грн.; в 2019 році мінімальна заробітна плата становить – 4173 грн., Кі – тарифний коефіцієнт робітника відповідного розряду, Тр – число робочих днів в місяці; приблизно Тр = 21…23 дні; Тзм – тривалість зміни, зазвичай Тзм = 8 годин.

Таблиця 4.7 – Заробітна плата робітників

Найменування робіт	Трудомісткість, н-год.	Розряд роботи	Погодинна тарифна ставка	Тариф. коеф.	Величина, грн.
Розробка	8	5	38,25	1,54	306
Тестування	8	4	36	1,45	288
Впровадження	2	2	27,1	1,09	55
Всього					649

Додаткова заробітна плата Зд всіх розробників та робітників, які брали участь у виконанні даного етапу роботи, розраховується як (10…12)% від суми основної заробітної плати всіх розробників та робітників, тобто:

$$З_Д = 0,1 \cdot (З_Р + З_О) = 0,1 \cdot (28417 + 649) = 2906,6 \text{ грн.} \quad (4.8)$$

Нарахування на заробітну плату Нзп розробників та робітників, які брали участь у виконанні даного етапу роботи, розраховуються за формулою 4.9:

$$Н_{ЗП} = 0,22 \cdot (З_Р + З_О + З_Д) = 0,22 \cdot (28417 + 649 + 2906,6) = 7034 \text{ грн.} \quad (4.9)$$

де Зо – основна заробітна плата розробників, грн.; Зр – основна заробітна плата робітників, грн.; Зд – додаткова заробітна плата всіх розробників та робітників, грн.; β – ставка єдиного внеску на загальнообов'язкове державне соціальне страхування, % (приймаємо для 1-го класу професійності ризику 22%).

Амортизація обладнання, комп'ютерів та приміщень А, які використовувались під час (чи для) виконання даного етапу роботи.

Дані відрахування розраховують по кожному виду обладнання, приміщенням тощо.

У спрощеному вигляді амортизаційні відрахування А в цілому бути розраховані за формулою 4.10:

$$A = \frac{Ц \cdot Н_а}{100} \cdot \frac{Т}{12}, \qquad (4.10)$$

де Ц – загальна балансова вартість всього обладнання, комп'ютерів, приміщень тощо, що використовувались для виконання даного етапу роботи, грн.; На – річна норма амортизаційних відрахувань. Для нашого випадку можна прийняти, що На = (10...25)%; Т – термін, використання обладнання, приміщень тощо, місяці.

Таблиця 4.8 - Амортизаційні відрахування

Найменування	Ціна, грн.	Норма амортизації, %	Термін використання, м.	Сума амортизації
ПК	18000	20	24	7200
Всього	7200			

Витрати на силову електроенергію Ве, якщо ця стаття має суттєве значення для виконання даного етапу роботи, розраховуються за формулою 4.11:

$$B_E = B \cdot П \cdot Ф \cdot K_П, \text{ грн} \qquad (4.11)$$

В – вартість 1 кВт-год. електроенергії, в 2019 р. В ≈ 2,44 грн./кВт; П – установлена потужність обладнання, кВт; Ф – фактична кількість годин роботи обладнання, годин, Кп – коефіцієнт використання потужності; Кп < 1.

Потужність обладнання складає – 0,5 кВт.

Кількість годин роботи складає – 700 годин.

Коефіцієнт викор. потужності -0,9.

Ве=769 грн.

Інші витрати Він охоплюють: витрати на управління організацією, оплата службових відряджень, витрати на утримання, ремонт та експлуатацію основних засобів, витрати на опалення, освітлення, водопостачання, охорону праці тощо.

Інші витрати Ів можна прийняти як (100…300)% від суми основної заробітної плати розробників та робітників, які були виконували дану роботу, тобто (4.12):

$$I_B = 1 \cdot (З_O + З_P) = 1 \cdot (28417 + 649) = 29066 \text{ грн.} \quad (4.12)$$

Сума всіх попередніх статей витрат дає витрати на виконання даної частини (розділу, етапу) роботи – В (4.13).

$$B = 77934 \text{ грн.} \quad (4.13)$$

Отже, витрати на виконання даної частини (розділу, етапу) роботи становить 77934 грн.

4.2.2 Розрахунок собівартості розробки системи виявлення фейкових облікових записів у соціальних мережах

Витрати на силову електроенергію Ве, якщо ця стаття має суттєве значення для виконання даного етапу роботи, розраховуються за формулою:

$$B_E = B \cdot П \cdot Ф \cdot K_П, \text{ грн} \quad (4.14)$$

В – вартість 1 кВт-год. електроенергії, в 2019 р. В ≈ 2,44 грн./кВт; П – установлена потужність обладнання, кВт; Ф – фактична кількість годин роботи обладнання, годин, Кп – коефіцієнт використання потужності; Кп < 1.

Потужність обладнання складає – 0,5 кВт.

Кількість годин роботи складає – 700 годин.

Коефіцієнт викор. потужності -0,9.

Ве=769 грн.

Основна заробітна плата робітників Зр, якщо вони беруть участь у виконанні даного етапу роботи і виконують роботи за робочими професіями у випадку, коли вони працюють в наукових установах бюджетної сфери, розраховується за формулою 4.15:

$$З_P = \sum_{i=1}^{n} t_i \cdot C_i \qquad (4.15)$$

де ti – норма часу (трудомісткість) на виконання конкретної роботи, годин; n – число робіт по видах та розрядах; Сі – погодинна тарифна ставка робітника відповідного розряду, який виконує дану роботу. Сі визначається за формулою 4.16:

$$C_i = \frac{M_M \cdot K_i}{T_P \cdot T_{ЗМ}}, \qquad (4.16)$$

де Мм – розмір мінімальної заробітної плати за місяць, грн.; в 2019 році мінімальна заробітна плата становить – 4173 грн., Кі – тарифний коефіцієнт робітника відповідного розряду, Тр – число робочих днів в місяці; приблизно Тр = 21…23 дні; Тзм – тривалість зміни, зазвичай Тзм = 8 годин.

Таблиця 4.9 – Заробітна плата робітників

Найменування робіт	Трудомісткість, н-год.	Розряд роботи	Погодинна тарифна ставка	Тариф. коеф.	Величина, грн.
Розробка	8	5	38,25	1,54	306
Тестування	8	4	36	1,45	288
Впровадження	2	2	27,1	1,09	55
Всього					649

Додаткова заробітна плата Зд всіх робітників, які брали участь у виконанні даного етапу роботи, розраховується як (10…12)% від суми основної заробітної плати всіх розробників та робітників, тобто (4.17):

$$З_Д = 0,1 \cdot З_О = 0,1 \cdot 649 = 64,9 \text{ грн.} \quad (4.17)$$

Нарахування на заробітну плату Нзп розробників та робітників, які брали участь у виконанні даного етапу роботи, розраховуються за формулою:

де Зо – основна заробітна плата розробників, грн.; Зр – основна заробітна плата робітників, грн.; Зд – додаткова заробітна плата всіх розробників та робітників, грн.; β – ставка єдиного внеску на загальнообов'язкове державне соціальне страхування, % (приймаємо для 1-го класу професійності ризику 22%).

$$НЗП = 0,22 \cdot (ЗО + ЗД) = 0,22 \cdot (649 + 64,9) = 157 \text{ грн.} \quad (4.18)$$

«Загальновиробничі витрати» належать витрати: пов'язані з управлінням виробництвом (утримання працівників апарату управління виробництвом, оплата службових відряджень персоналу цехів, витрати на інформаційне забезпечення управління тощо); на повне відновлення та капітальний ремонт основних фондів загальновиробничого призначення; витрати некапітального характеру, пов'язані з удосконаленням технологій та організацією виробництва, поліпшенням якості продукції; на утримання, обслуговування, поточний ремонт виробничих приміщень; на контроль за виробничими процесами та кістю продукції.

Крім того, загальновиробничі витрати з розрахунку на одиницю продукції можна розрахувати за нормативами відносно до основної заробітної плати основних робітників, які виготовляють продукцію (4.19):

$$ЗВВ = Н_В \cdot З_О, \quad (4.19)$$

Норматив загальновиробничих витрат для програмних продуктів становить 230-270% (4.20).

$$ЗВВ = 2,7 \cdot 649 = 1752 \text{ грн.} \quad (4.20)$$

Сума попередніх витрат утворює виробничу собівартість розробки (4.21):

$$S_B = 5285 \text{ грн.} \quad (4.21)$$

4.3 Розрахунок мінімальної ціни та чистого прибутку від реалізації розробки системи виявлення фейкових облікових записів у соціальних мережах

Ціна – це грошовий вираз вартості товару (продукції, послуги). Вона завжди коливається навколо ціни виробництва (перетвореної форми вартості одиниці товару, що дорівнює сумі витрат виробництва й середнього прибутку) та відображає рівень суспільне необхідних витрат праці.

Виходячи з того, що розробки, як правило, приймаються та впроваджуються за завданням замовника, або коли результатом розробки є продукція, що підлягає державному регулюванню, то нижню межу ціни реалізації розробки можна розрахувати за формулою 4.22:

$$Ц = S_B \cdot (1 + \frac{P}{100}) \cdot (1 + \frac{\omega}{100}), \quad (4.22)$$

де Sв – виробнича собівартість інноваційного рішення, грн.; Р – норматив рентабельності узгоджений із замовником або встановлений державою, (Р=30…60%); w – ставка податку на додану вартість, % (в 2019 році w=20%) (4.23).

$$Ц = 5285 \cdot (1 + \frac{60}{100}) \cdot (1 + \frac{20}{100}) = 10150 \text{ грн.}$$

(4.23)

Чистий прибуток від реалізації розробки можна розрахувати за формулою 4.24:

$$П = \left(Ц - \frac{(Ц - МР) \cdot f}{100} - S_B - \frac{q \cdot S_B}{100} \right) \cdot \left(1 - \frac{h}{100} \right) \cdot РП, \quad (4.24)$$

де Ц – ціна розробки, грн.; МР – вартість матеріальних та інших ресурсів, що були придбані виробником для виготовлення розробки (МР=(0,1…0,2) Цр), грн.; f – зустрічна ставка податку на додану вартість, %; Sв – виробнича собівартість розробки, грн.; q – норматив, який визначає величину адміністративних витрат, витрат на збут та інші операційні витрати, % (рекомендовано q=5…10%); h – ставка податку на прибуток, %, РП – прогнозований попит продажів (4.25):

$$П = 38900 \text{ грн.} \quad (4.25)$$

4.4 Розрахунок терміну окупності коштів, вкладених в наукову розробку системи виявлення фейкових облікових записів у соціальних мережах

Термін окупності вкладених у реалізацію наукового проекту інвестицій Ток можна розрахувати за формулою 4.26:

$$T_{ОК} = \frac{B}{П} = \frac{77934}{38900} = 2 \text{ роки.} \quad (4.26)$$

Оскільки Ток < 3 років, то фінансування даної наукової розробки є доцільним.

Отже, здійснено економічне обґрунтування та доведено доцільність розробки системи виявлення фейкових облікових облікових записів у соціальних мережах. У ході дослідження виявленно, що чистий прибуток сягає 38900 грн, а період окупності становить 2 роки.

ВИСНОВКИ

Розглянуто основні поняття фейкових облікових записів, соціальних мереж та використання фейків як для особистої анонімності користувача, так і для ведення інформаційних протиборств. Проаналізовано основні методи аналізу інформації, отриманої з соціальних мереж, а також розглянуто теоретичні відомості про параметри облікових записів - метрики, що використовують для подальшого аналізу отриманих даних про користувачів, групи та інші елементи соціальних мереж. Поставлено завдання на розробку системи виявлення фейкових облікових записів у соціальних мережах, базуючися на проаналізованих літературних джерелах.

Проаналізувавши структуру облікового запису у соціальних мережах, виділено інформацію про користувача, яка міститься у них. Розглянуто та структуровано метрики за категоріями за значеннями та їх впливом на фейковість облікового запису. Виділено такі категорії: фото, друзі, персональна інформація про користувача та пости та статуси. Для зручності подальшої обробки всі метрики були віднесені до певних категорій. Розглянуто та проаналізовано існуючі системи підтримки прийняття рішень та обрано для подальшого дослідження бінарний класифікатор, що працює за методом опорних векторів. На основі обраної системи підтримки прийняття рішень реалізовано нейромережу для виявлення фейкових облікових записів у соціальній мережі «Facebook».

Розроблено архітектуру та схему роботи програми, а також основні основні компоненти. Створено графічний інтерфейс програмного засобу, що є зручним для користування. Програмний засіб розроблено за допомогою об’єктно-орієнтованої мови програмування Python версії 3.7, яка дозволяє працювати з даними великої розмірності. Проведено ряд експериментальних досліджень для тестування роботи програми шляхом перевірки на фейковість 100 облікових записів у соціальній мережі «Facebook». Експериментальні

дослідження показали достовірність роботи системи виявлення фейкових облікових записів у соціальній мережі «Facebook» на рівні 94%, при цьому помилки першого та другого роду становлять 4% і 2% відповідно.

Здійснено економічне обґрунтування та доведено доцільність розробки системи виявлення фейкових облікових облікових записів у соціальних мережах. У ході дослідження виявленно, що чистий прибуток сягає 38900 грн, а період окупності становить 2 роки.

У подальших дослідженнях планується розширити кількість метрик для аналізу, серед яких зв’язки між друзями користувача, частота оновлення інформації у обліковому записі, аналіз спільнот та інтересів користувача. Аналіз цих метрик дозволить задіяти більше даних про обліковий запис. Також планується аналізувати облікові записи у інших популярних соціальних мережах, таких як «Twitter».

ПЕРЕЛІК ІНФОРМАЦІЙНИХ ДЖЕРЕЛ

1. Theohary C. Information warfare: The role of social media in conflicts [Електронний ресурс]. – Режим доступу : URL : https://digital.library.unt.edu/ark:/6753 1/metadc503647/m1/1/high_res_d/IN10240_2015Mar04.pdf – Назва з екрану.

2. Voitovych O., Holovenko V. Research of social networks as a source of information in warfare // Inżynier XXI wieku projectujemy przyszlosc: monografia / pod red: Jacek Rysiński. – Bielsko-Biała, 2016. – С. 111-119.

3. Дудатьєв А. В., Войтович О. П. Інформаційна безпека соціотехнічних систем: Модель інформаційного впливу // Інформаційні технології та комп'ютерна інженерія. – 2017. – № 38. – С. 16 - 21.

4. Войтович О. П., Дудатьєв А. В., Головенько В. О. Модель та засіб для виявлення фейкових облікових записів у соціальних мережах // Вчені записки таврійського національного університету ім. В.І. Вернадського. Серія: Технічні науки. Частина 1 – 2018. – № 1 Том 29 (68). – С. 112 – 119.

5. Войтович О. П., Буда А. Г., Головенько В. О. Дослідженння методів аналізу соціальних мереж як середовища інформаційних війн //Тези доповідей Шостої Міжнародної науково-практичної конференції «Методи та засоби кодування, захисту й ущільнення інформації» м. Вінниця, 24-25 жовтня 2017 року. – Вінниця: ВНТУ, 2017. – С. 67-70.

6. Войтович О. П., Дудатьєв А. В., Головенько В. О., Виявлення фейкових облікових записів у соціальній мережі «Facebook» // Тези доповідей міжнародної науково-практичної конференції «Інформаційні технології та комп'ютерне моделювання» м. Івано-Франківськ, 14-19 травня 2018 року. Івано-Франківськ: 2018. - С. 190-193.

7. Дудатьєв А. В., Войтович О. П., Головенько В. О., Рудик О. А. Генератор мемів для тестування соціальної складової соціотехнічної системи //

Тези доповідей III Міжнародної науково-практичної конференції «Інформаційна безпека та комп’ютерні технології» м. Кропивницький, 19-20 квітня 2018 року. Кропивницький: ЦНТУ, 2018. – С. 63-65.

8. Комп'ютерна програма «Засіб для виявлення фейкових облікових записів у соціальній мережі «Facebook»» // Свідотцтво про реєстрацію авторського права на твір № 77297 від 03.03.2018.

9. Коршунов А., Белобородов И., Бузун Н. Анализ социальных сетей: методы и приложения // Труды Института системного программирования РАН. – 2014. – Т. 26. – № 1. – С. 439-456.

10. Катерина Прогнімак. Фабрика фоловерів — як продається вплив у соціальних мережах [Електронний ресурс]. – Режим доступу : URL : https://www.imena.ua/blog/follower-factory-network - Назва з екрану.

11. Дудатьєв А. В. Комплексна інформаційна безпека СТС: моделі впливу та захисту : монографія – Вінниця: ВНТУ, 2017. – 128 с.

12. Ольга Карпенко. От лайков к шантажу: как работает индустрия фейковых профилей на Facebook [Електронний ресурс]. – Режим доступу : URL : https://ain.ua/2017/11/21/industriya-fejkovyx-profilej-na-facebook - Назва з екрану.

13. Ryan Mac, Charlie Warzel. Scammers Are Impersonating Elon Musk And Donald Trump To Take Your Bitcoin [Електронний ресурс]. – Режим доступу : URL : https://www.buzzfeed.com/ryanmac/scammers-are-impersonating-elon-musk-and-donald-trump-steal?bftwnews&utm_term=.kjV1k9Z4d#.xvpNGbqEg - Назва з екрану.

14. Smith Jamie. 65+ Social Networking Sites You Need to Know About [Електронний ресурс]. – Режим доступу : URL : https://makeawebsitehub.com/social-media-sites/ - Назва з екрану.

15. Батура Т. В., Копылова Н. С., Мурзин Ф. А., Проскуряков А. В. Методы анализа данных из социальных сетей // Вестник НГУ. Серия: Информационные технологии. – 2013. – Т. 11. – Вып. 3. – С. 5-21.

16. Прохоров А. Социальные сети и Интернет [Електронний ресурс]. – Режим доступу : URL : https://compress.ru/article.aspx?id=16723 - Назва з екрану.

17. Берни Хоган. Анализ социальных сетей в интернете [Електронний ресурс]. – Режим доступу: URL:https://postnauka.ru/longreads/20259-Назва з екрану.

18. Будыльский Д.В. Автоматизация мониторинга общественного мнения на основе интеллектуального анализа сообщений в социальных сетях[Електронний ресурс]. – Режим доступу : URL : http://docplayer.ru/28555072-Budylskiy-dmitriy-viktorovich-avtomatizaciya-monitoringa-obshchestvennogo-mneniya-na-osnove-intellektualnogo-analiza-soobshcheniy-v-socialnyh-setyah.html - Назва з екрану.

19. DSS - система поддержки принятия решений. [Електронний ресурс]. – Режим доступу : URL : https://pro-spo.ru/erp/1816-dss - Назва з екрану.

20. Чалый С.Ф., Чередниченко А.А. Исследование методов анализа социальных сетей для определения групп пользователей программного продукта // АСУ и приборы автоматики. – 2013. – №165.

21. Базенков Н. И., Губанов Д. А. Обзор информационных систем анализа социальных сетей // Управление большими системами. – Вып.41. – М.: ИПУ РАН, 2013. – С. 357-394.

22. Горчинская О., Ривкин А. Анализ данных социальных сетей // Открытые системы. СУБД. – 2015. – №3. – С. 22-23.

23. Cohen David. FB Checker Analyzes Facebook Photos To Identify Fake Profiles [Електронний ресурс]. – Режим доступу : URL : https://www.adweek.com/digital/fb-checker/ - Назва з екрану.

24. FAKE FB - Fake profile examiner [Електронний ресурс]. – Режим доступу : URL : https://chrome.google.com/webstore/detail/fake-fb-fake-profile-exam/ikibemjjmdaoeheahednadigldppjdom - Назва з екрану.

25. Borison Rebecca. This App Could Solve The No.1 Problem On Facebook: Fake Accounts [Електронний ресурс]. – Режим доступу : URL : https://www.businessinsider.com/fakeoff-app-weeds-out-fake-facebook-profiles-2014-6 - Назва з екрану.

26. Michal Kosinski, Sandra C. Matz, Samuel D. Gosling, Vesselin Popov, David Stillwe. Facebook as a Research Tool for the Social Sciences. Opportunities,

Challenges, Ethical Considerations, and Practical Guidelines / Michal Kosinski, Sandra C. Matz, Samuel D. Gosling, Vesselin Popov, David Stillwe // American Psychologist. – 2015. – Vol. 70. – No. 6. – 543-556 pp.

27. Нежданов И. Ю. Технологии информационных войн в Интернете [Електронний ресурс]. – Режим доступу : URL : http://bash.rosmu.ru/activity/attach/events/1283/01.pdf - Назва з екрану.

28. Ольга Карпенко. Как кремлевская фабрика троллей влияла на выборы в США: реклама, хештеги, фейки [Електронний ресурс]. – Режим доступу : URL : https://ain.ua/2018/02/19/fabrika-trollej-i-vybory-v-ssha - Назва з екрану.

29. Tony Romm. Facebook, Google and Twitter have been asked to testify before Congress on Russia and the 2016 election [Електронний ресурс]. – Режим доступу : URL : https://www.recode.net/2017/9/27/16376228/facebook-google-twitter-testify-congress-senate-russia-presidential-election - Назва з екрану.

30. Aaron Aguis. 10 Metrics to Track for Social Media Success [Електронний ресурс]. – Режим доступу : URL : https://www.socialmediaexaminer.com/10-metrics-to-track-for-social-media-success/ - Назва з екрану.

31. Губанов Д.А., Новиков Д.А., Чхартишвили А.Г. Социальные сети: модели информационного влияния, управления и противоборства. – М.: Физматлит, 2010.

32. Кветный Р. Н., Коцюбинский В. Ю., Кислица Л. Н.; Казимирова Н. В.; Кириленко А. А. Адаптивная система поддержки принятия решений на основе нечеткого логического вывода // Наукові праці ВНТУ. – 2011. – №3. [Електронний ресурс]. – Режим доступу : URL : https://trudy.vntu.edu.ua/index.php/trudy/article/view/301/301 - Назва з екрану.

33. Find your Facebook ID [Електронний ресурс]. – Режим доступу : URL : https://findmyfbid.com/ - Назва з екрану.

34. Войтович О. П. Достовірність прийняття рішення в системах захисту інформації // Інформаційні технології та комп'ютерна інженерія, - № 27, - Том 2, - 2013. - С. 15-20.

ДОДАТКИ

Додаток А

Міністерство освіти і науки України

Вінницький національний технічний університет

Факультет інформаційних технологій та комп’ютерної інженерії

Кафедра захисту інформації

ЗАТВЕРДЖУЮ

Завідувач кафедри ЗІ, д.т.н, проф.

_____ В. А.Лужецький

_____ 2019 р.

ТЕХНІЧНЕ ЗАВДАННЯ

на магістерську кваліфікаційну роботу

«Система виявлення фейкових облікових записів у соціальних мережах»

08-20.МКР 003.00.000 ТЗ

Керівник роботи

к. т. н. доц. кафедри ЗІ

_____Войтович О. П.

«___»_____2019 р.

Вінниця 2019

1 Назва та область використання

Система виявлення фейкових облікових записів у соціальних мережах. Програмний засіб дозволяє виявляти фейкові облікові записи у соціальній мережі «Facebook» та може бути використаний як для домашнього користування, так і для виявлення ботів під час ведення інформаційних війн.

2 Основа для розробки

Робота проводиться на підставі наказу ректора ВНТУ № 254 від 02.10.2019 р.

3 Мета та призначення розробки

Виявлення фейкових облікових записів у соціальних мережах з метою протидії під час ведення інформаційних війн та контролю за інформацією, що розповсюджується соціальними мережами.

4 Джерела розробки

4.1 Voitovych O., Holovenko V. Research of social networks as a source of information in warfare / O. Voitovych, V. Holovenko // Inżynier XXI wieku projectujemy przyszlosc: monografia / pod red: Jacek Rysiński. – Bielsko-Biała, 2016. – С. 111-119.

4.2 Дудатьєв А. В., Войтович О. П. Інформаційна безпека соціотехнічних систем: Модель інформаційного впливу / А. В. Дудатьєв, О. П. Войтович // Інформаційні технології та комп'ютерна інженерія. – 2017. – № 38. – С. 16 - 21.

4.3 Войтович О. П., Буда А. Г., Головенько В. О. Дослідженння методів аналізу соціальних мереж як середовища інформаційних війн / О. П. Войтович // Тези доповідей Шостої Міжнародної науково-практичної конференції «Методи та засоби кодування, захисту й ущільнення інформації» м. Вінниця, 24-25 жовтня 2017 року. – Вінниця: ВНТУ, 2017. – С. 67-70.

4.4 Michal Kosinski, Sandra C. Matz, Samuel D. Gosling, Vesselin Popov, David Stillwe. Facebook as a Research Tool for the Social Sciences. Opportunities, Challenges, Ethical Considerations, and Practical Guidelines / Michal Kosinski, Sandra C. Matz, Samuel D. Gosling, Vesselin Popov, David Stillwe // American Psychologist. – 2015. – Vol. 70. – No. 6. – 543-556 pp.

4.5 Войтович О. П., Дудатьєв А. В., Головенько В. О. Модель та засіб для виявлення фейкових облікових записів у соціальних мережах / О. П. Войтович // Вчені записки таврійського національного університету ім. В.І. Вернадського. Серія: Технічні науки. Частина 1 – 2018. – № 1 Том 29 (68). – С. 112 – 119.

4.6 Войтович О. П., Дудатьєв А. В., Головенько В. О., Виявлення фейкових облікових записів у соціальній мережі «Facebook» / О. П. Войтович // Тези доповідей міжнародної науково-практичної конференції «Інформаційні технології та комп'ютерне моделювання» м. Івано-Франківськ, 14-19 травня 2018 року. Івано-Франківськ: 2018. - С. 190-193.

5 Вимоги до програмного засобу

5.1 Параметри розроблюваного програмного засобу:

- операційна система - Windows;
- мова програмування – Python з використанням модуля Facebook-SDK;
- середовище розробки – JetBrains PyCharm 2017.1;
- підключення до мережі Інтернет.

5.2 Програмний засіб повинен виконувати такі дії:

- відображати імовірність того, що обліковий запис є фейковим;
- відображати гістрограму параметрів впливу;
- відображати статус перевірки облікового запису.

6 Вимоги до супровідної документації

Графічна і текстова документація повинна відповідати діючим стандартам України.

7 Стадії та етапи розробки

№ з/п	Назва етапів магістерської кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз завдання. Вступ	01.09.2019 – 04.09.2019	Чорновик вступу
2	Розробка технічного завдання	23.09.2019 – 29.09.2019	Проект технічного завдання
3	Аналіз літературних джерел за напрямком магістерської кваліфікаційної роботи	05.09.2019 – 15.09.2019	Чорновик першого розділу. Схеми та алгоритми
4	Практична реалізація, моделювання, експериментування, результати	14.10.2019 – 10.11.2019	Програмний засіб. Розділ пояснювальної записки
5	Аналіз виконання ТЗ, висновки	18.11.2019 – 24.11.2019	Висновки, інструкції
6	Оформлення пояснювальної записки	25.11.2019 – 30.11.2019	Пояснювальна записка, графічний матеріал

8 Порядок контролю та прийому

До прийому магістерської кваліфікаційної роботи подається:

- заключний звіт (пояснювальна записка);
- ілюстративний матеріал;
- програмний засіб;
- інструкція по роботі з програмним засобом.

Початок розробки	01.09.2019
Крайній термін виконання магістерської кваліфікаційної роботи	20.12.2019

Розробив студент групи 1БС-18м_____Головенько В. О.

Додаток Б
Лістинг програмного засобу

Файл fake_detector.py

```
import csv, time, ast
import decision_support_system
import tkinter
import numpy as np
from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg
from matplotlib.figure import Figure
from selenium.common.exceptions import *
from selenium import webdriver
from pandas import read_csv as read
from tkinter import *
import tkinter as tk
import tkinter as Tk
from PIL import ImageTk, Image

class Scrapper:
    def __init__(self, person_id):
        self.person_id = person_id

    def get_avatar_info(self):
        try:
            main_photos = driver.find_element_by_css_selector(
                '#fbTimelineHeadline > div.name > div > div > a > img').get_attribute('alt')
            # print('Info about avatar: ' + str(main_photos))
        except NoSuchElementException:
            main_photos = 'No info about avatar'
            # print('No info about avatar')
            pass
        return main_photos

    def get_avatar_link(self):
        try:
            avatar_link = driver.find_element_by_css_selector(
                '#fbTimelineHeadline > div.name > div > div > a > img').get_attribute('src')
            # print('Avatar: ' + str(avatar_link))
        except NoSuchElementException:
            avatar_link = 'Avatar doesn`t exist'
            # print('Avatar doesn`t exist')
            pass
        return avatar_link

    def get_cover_info(self):
        try:
            info_about_photo = driver.find_element_by_css_selector('img.coverPhotoImg.photo.img').get_attribute('alt')
            # print('Info about cover: ' + str(info_about_photo))
        except NoSuchElementException:
            info_about_photo = 'No info about cover photo'
            # print('No info about cover photo')
            pass
        return info_about_photo

    def get_cover_link(self):
        try:
            cover_link = driver.find_element_by_css_selector('img.coverPhotoImg.photo.img').get_attribute('src')
            # print('Cover: ' + str(cover_link))
        except NoSuchElementException:
            cover_link = 'Cover photo doesn`t exist'
            # print('Cover photo doesn`t exist')
            pass
        return cover_link

    def get_intro_info(self):
```

```
    try:
        intro = driver.find_element_by_id('intro_container_id').text
        # print('Info about user: ' + intro)
    except NoSuchElementException:
        intro = 'No info about user'
        # print('No info about user')
        pass
    return intro

def get_number_of_photos(self):
    try:
        photos = driver.find_element_by_xpath('//span[text()="Photos"]').click()
        time.sleep(2)
        # Get scroll height
        SCROLL_PAUSE_TIME = 2
        last_height = driver.execute_script("return document.body.scrollHeight")
        while True:
            # Scroll down to bottom
            driver.execute_script("window.scrollTo(0, document.body.scrollHeight);")
            # Wait to load page
            time.sleep(SCROLL_PAUSE_TIME)
            # Calculate new scroll height and compare with last scroll height
            new_height = driver.execute_script("return document.body.scrollHeight")
            if new_height == last_height:
                break
            last_height = new_height
        photos_count = driver.find_elements_by_class_name('fbPhotoStarGridElement')
        # print('Number of photos: ' + str(len(photos_count)))
        photos_count = str(len(photos_count))
    except NoSuchElementException:
        photos_count = 'No info about photos'
        # print('No info about photos')
        pass
    return photos_count

def get_number_of_posts(self):
    try:
        posts = driver.find_elements_by_class_name('profileLink')
        # print('Number of posts: ' + str(len(posts)))
        posts = str(len(posts))
    except NoSuchElementException:
        posts = 'No info about posts'
        # print('No info about posts')
        pass
    return posts

def get_number_of_friends(self):
    try:
        friends = driver.find_element_by_css_selector(
            '#profile_timeline_tiles_unit_pagelets_friends > li > div > div > div > div > div > div > div > span:nth-child(3) > a').text
        # print('Number of friends: ' + str(friends))
    except NoSuchElementException:
        friends = 'No info about friends'
        # print('No info about friends')
        pass
    return friends

def get_personal_info(self, person_id):
    driver.get(person_id + '/about')
    driver.find_element_by_tag_name('body').click()
    time.sleep(5)
    personal_info = driver.find_element_by_css_selector(
        '.uiList > li > div > div:nth-child(2) > div > div > div:nth-child(1)').text
    return personal_info

def get_contacts_and_birthday(self):
    try:
        contacts_and_birthday = driver.find_element_by_css_selector(
```

```
        '.uiList > li > div > div:nth-child(2) > div > div > div:nth-child(2)').text

    birthday = re.findall('\d{2} [a-zA-Z]{3,9} \d{4}', contacts_and_birthday)
    # print(birthday)
    if birthday != []:
        birthday = birthday[0]
    else:
        birthday = 'No info about birthday'

    try:
        contacts_and_birthday = contacts_and_birthday.replace('\n', '.')
        contacts = re.search('%s(.*)%s' % ('Social Links.', '.Birthday'), contacts_and_birthday).group(1)
        contacts = contacts.replace('.', '\n')
    except AttributeError:
            contacts = 'No contacts'
    except NoSuchElementException:
        contacts = 'The "Contacts" container is empty'
        birthday = 'The "Birthday" container is empty'
        # print('The "Contacts" container is empty')
        pass
    return birthday, contacts

def get_avatar_likes_and_comments(self):
    try:
        driver.find_element_by_css_selector('.name .photoContainer > div > a').click()
        time.sleep(3)
        try:
            likes_on_avatar = driver.find_element_by_css_selector(
                '.fbPhotosSnowliftFeedbackForm > div:nth-child(4) > div > div > div:nth-child(1) > div:nth-child(1)').text
            likes_on_avatar = re.findall(r'^\w*', likes_on_avatar)
            likes_on_avatar = int(likes_on_avatar[0])

            people = driver.find_element_by_css_selector(
                '.fbPhotosSnowliftFeedbackForm > div:nth-child(4) > div > div > div:nth-child(1) > div:nth-child(1)').text
        except:
            likes_on_avatar = 'No likes'
            pass
        try:
            comments_on_avatar = driver.find_element_by_css_selector(
                '.fbPhotosSnowliftFeedbackForm > div:nth-child(4) > div > div > div:nth-child(1) > div:nth-child(3)').text
            comments_on_avatar = re.findall(r'^\w*', comments_on_avatar)
            if comments_on_avatar == ['']:
                comments_on_avatar = 'No comments'
            else:
                comments_on_avatar = comments_on_avatar[0]
        except:
            comments_on_avatar = 'No comments'
            pass
    except:
        likes_on_avatar = 'No likes'
        comments_on_avatar = 'No comments'
    return likes_on_avatar, comments_on_avatar

def get_people_liked_the_avatar(self):
    try:
        driver.find_element_by_css_selector(
            '.fbPhotosSnowliftFeedbackForm > div:nth-child(4) > div > div > div:nth-child(1) > div:nth-child(1) > a:nth-child(2)').click()
        time.sleep(2)
        people_liked_the_avatar = driver.find_element_by_css_selector('.uiScrollableAreaContent > div > ul').text
        people_liked_the_avatar = people_liked_the_avatar.replace('Add Friend\n', '')
        people_liked_the_avatar = people_liked_the_avatar.replace('Add Friend', '')
        people_liked_the_avatar = people_liked_the_avatar.split('\n')
    except NoSuchElementException:
        people_liked_the_avatar = 'No likes on avatar'
    return people_liked_the_avatar

def get_friends_list(self, person_id):
    driver.get(person_id + '/friends')
```

```
time.sleep(2)
driver.find_element_by_tag_name('body').click()
SCROLL_PAUSE_TIME = 2
last_height = driver.execute_script("return document.body.scrollHeight")
while True:
    # Scroll down to bottom
    driver.execute_script("window.scrollTo(0, document.body.scrollHeight);")
    # Wait to load page
    time.sleep(SCROLL_PAUSE_TIME)
    # Calculate new scroll height and compare with last scroll height
    new_height = driver.execute_script("return document.body.scrollHeight")
    if new_height == last_height:
        break
    last_height = new_height

    friends_list = driver.find_element_by_id('pagelet_timeline_medley_friends').text
    friends_list = friends_list.replace('Add Friend\n', '')
    friends_list = friends_list.split('\n')
    return friends_list

def get_full_name(self):
    data_fullname = driver.find_element_by_css_selector('#fb-timeline-cover-name > a').text
    return data_fullname

def get_number_of_friends_and_strangers_likes(self, people_liked_the_avatar, likes_on_avatar, friend_list):
    people_liked_the_avatar = people_liked_the_avatar
    likes_on_avatar = likes_on_avatar
    friend_list = friend_list

    if people_liked_the_avatar != 'No likes on avatar':
        friends_liked_the_avatar = list(set(friend_list) & set(people_liked_the_avatar))
        number_of_fiends_liked_the_avatar = int(len(friends_liked_the_avatar))
        number_of_strangers_liked_the_avatar = likes_on_avatar - number_of_fiends_liked_the_avatar
        number_of_likes_on_avatar = {}
        number_of_likes_on_avatar['Number of friends` likes: '] = number_of_fiends_liked_the_avatar
        number_of_likes_on_avatar['Number of strangers` likes: '] = number_of_strangers_liked_the_avatar
    else:
        number_of_likes_on_avatar = 'No info about likes on avatar'
    return number_of_likes_on_avatar

def scrap(self, person_id):
    self.person_id = person_id

    driver.get(person_id)
    time.sleep(2)
    driver.find_element_by_tag_name('body').click()
    likes_on_avatar, comments_on_avatar = self.get_avatar_likes_and_comments()
    people_liked_the_avatar = self.get_people_liked_the_avatar()

    driver.get(person_id)
    time.sleep(2)
    driver.find_element_by_tag_name('body').click()

    time.sleep(3)
    SCROLL_PAUSE_TIME = 2

    # Getting user`s full name

    full_name = self.get_full_name()

    # Getting info about avatar
    avatar_info = self.get_avatar_info()

    # Getting link on avatar
    avatar_link = self.get_avatar_link()

    # Getting info about cover
    cover_info = self.get_cover_info()
```

```
# Getting link on cover
cover_link = self.get_cover_link()

# Getting info from intro container
intro_info = self.get_intro_info()

time.sleep(2)

# Getting number of user`s photo
number_of_photos = self.get_number_of_photos()

driver.get(person_id)
time.sleep(2)
driver.find_element_by_tag_name('body').click()

# Get scroll height
last_height = driver.execute_script("return document.body.scrollHeight")
while True:
    # Scroll down to bottom
    driver.execute_script("window.scrollTo(0, document.body.scrollHeight);")
    # Wait to load page
    time.sleep(SCROLL_PAUSE_TIME)
    # Calculate new scroll height and compare with last scroll height
    new_height = driver.execute_script("return document.body.scrollHeight")
    if new_height == last_height:
        break
    last_height = new_height

# Getting number of posts
number_of_posts = self.get_number_of_posts()

time.sleep(2)

# Getting number of user`s friends
number_of_friends = self.get_number_of_friends()

# Getting personal info about user
personal_info = self.get_personal_info(person_id)

# Getting user contact information and birthday
birthday, contacts = self.get_contacts_and_birthday()

friend_list = self.get_friends_list(person_id)

get_number_of_friends_and_strangers_likes = self.get_number_of_friends_and_strangers_likes(
    people_liked_the_avatar, likes_on_avatar, friend_list)

print('\n\n')

arr = []
arr.append(str(full_name))
arr.append(str(avatar_info))
arr.append(str(avatar_link))
arr.append(str(likes_on_avatar))
arr.append(str(get_number_of_friends_and_strangers_likes))
arr.append(str(people_liked_the_avatar))
arr.append(str(comments_on_avatar))
arr.append(str(cover_info))
arr.append(str(cover_link))
arr.append(str(intro_info))
arr.append(str(number_of_photos))
arr.append(str(number_of_posts))
arr.append(str(number_of_friends))
arr.append(str(friend_list))
arr.append(str(personal_info))
arr.append(str(birthday))
arr.append(str(contacts))
```

```
        print(arr)

        with open('total.csv', "a", newline='', encoding="utf-8") as file:
            writer = csv.writer(file, delimiter=",", quoting=csv.QUOTE_MINIMAL)
            writer.writerow(arr)

        return arr

class Analyzer(Scrapper):
    def __init__(self, person_id):
        self.person_id = person_id

    def tonumber(self, person_id):
        data = Scrapper.scrap(self, person_id)
# AVATAR P1
if 'oh=cfbb962aa7a58f425d07881def1ebc01&oe=5E7215E0' in str(
        data[2]) or 'oh=794bea5187e612775f8b753766129863&oe=5E7FD7F3' in str(data[2]):
    self.P1 = 2
elif 'person' in str(data[1]):
    self.P1 = 0
else:
    self.P1 = 1

# COVER P2
if str(data[7]) == 'No info about cover photo':
    self.P2 = 2
else:
    self.P2 = 0

# PHOTOS P3
if str(data[10]) == 'No info about photos':
    self.P3 = 2
elif int(str(data[10]).replace(',', '')) < 10 or int(str(data[10]).replace(',', '')) > 1000:
    self.P3 = 1
else:
    self.P3 = 0

# FRIENDS P4
if str(data[12]) == 'No info about friends':
    self.P4 = 2
elif int(str(data[12]).replace(',', '')) < 10 or int(str(data[12]).replace(',', '')) > 2000:
    self.P4 = 1
else:
    self.P4 = 0

# POSTS P5
if str(data[11]) == 'No info about posts':
    self.P5 = 2
elif int(data[11]) < 10 or int(data[11]) > 500:
    self.P5 = 1
else:
    self.P5 = 0

# PERSONAL INFO P6
if str(data[
           14]) == 'No workplaces to show\nNo schools / universities to show\nNo places to show\nNo relationship info to show':
    self.P6 = 2
elif 'No ' in str(data[14]):
    self.P6 = 1
else:
    self.P6 = 0

# CONTACTS AND BIRTHDAY P7
if str(data[16]) == 'The "Contacts" container is empty' or str(data[16]) == 'No contacts':
    self.P7 = 2
else:
    self.P7 = 0
```

```
# P8
if str(data[15]) == 'The "Birthday" container is empty' or str(data[15]) == 'No info about birthday':
    self.P8 = 2
elif int(str(str(data[15])[-4:])) > 2009 or int(str(str(data[15])[-4:])) < 1932:
    self.P8 = 1
else:
    self.P8 = 0

# AVATAR LIKES AND COMMENTS P10
if str(data[6]) == 'No comments':
    self.P10 = 2
elif int(data[6]) < 5 or int(data[6]) > 100:
    self.P10 = 1
else:
    self.P10 = 0

# LIKES ON AVATAR P9
if str(data[5]) == 'No likes on avatar' or str(data[5]) == 'No info about likes on avatar':
    self.P9 = 2
elif int(ast.literal_eval(str(data[4]))['Number of friends` likes: ']) < int(
        ast.literal_eval(str(data[4]))['Number of strangers` likes: ']):
    self.P9 = 1
else:
    self.P9 = 0

return self.P1, self.P2, self.P3, self.P4, self.P5, self.P6, self.P7, self.P8, self.P9, self.P10

def analyze(self, person_id):
    P1, P2, P3, P4, P5, P6, P7, P8, P9, P10 = self.tonumber(person_id)
    with open('account.csv', "w", newline='') as file:
        writer = csv.writer(file)
        writer.writerow(['ID', 'P1', 'P2', 'P3', 'P4', 'P5', 'P6', 'P7', 'P8', 'P9', 'P10'])
        writer.writerow(
            [str(person_id), str(P1), str(P2), str(P3), str(P4), str(P5), str(P6), str(P7), str(P8), str(P9),
             str(P10)])

class GUI:
    def __init__(self):
        root = tk.Tk()

        root.resizable(width=True, height=True)
        root.title("Fake detector for Facebook v1.0")

        root.bind_all("<Key>", self.paste, "+")

        frame_info = Frame(root)
        frame_progressbar = Frame(root, bd=5)
        frame_status = Frame(root, bd=5)
        self.frame_graph = Frame(root, bd=5)

        label_id = Label(frame_status, font='TimesNewRoman 14')
        label_id['text'] = "User id:"
        label_id.pack()

        self.id_entry = Entry(frame_status, font='TimesNewRoman 14', width=25, borderwidth=1, justify=CENTER)
        self.id_entry.pack()

        check_button = Button(frame_info, width=15, height=5, fg="black", font='TimesNewRoman 14',
                              command=self.check_account)
        check_button["text"] = "Check page"
        check_button.bind("Check page")
        check_button.pack()

        label_status = Label(frame_status, font='TimesNewRoman 14')
        label_status['text'] = "Status: "
        label_status.pack()

        self.status_entry = Text(frame_status, height=1, width=17, font='Consolas 12')
```

```
self.status_entry.pack()

# LEGEND
im = Image.open('Legend.jpg')
tkimage = ImageTk.PhotoImage(im)
label_image = tkinter.Label(self.frame_graph, image=tkimage)
label_image.config(height=294, width=200)
label_image.pack(side=RIGHT)

frame_info.pack()
frame_progressbar.pack()
frame_status.pack()
self.frame_graph.pack()
root.mainloop()

def check_account(self):
    person_id = str('https://www.facebook.com/' + self.id_entry.get())
    print(person_id)
    login = 'n00basya@mail.ru'
    password = '1234nubasik1234'
    driver.get("https://www.facebook.com")
    driver.find_element_by_name('email').send_keys(login)
    driver.find_element_by_name('pass').send_keys(password)

    try:
        driver.find_element_by_id('loginbutton').click()
    except:
        driver.find_element_by_name('login').click()
    time.sleep(2)

    analyzer = Analyzer(person_id)
    analyzer.analyze(person_id)
    result = self.histogram()
    print(result)

def paste(self, event):
    ctrl = (event.state & 0x4) != 0
    if event.keycode == 86 and ctrl and event.keysym.lower() != "v":
        event.widget.event_generate("<<Paste>>")

def histogram(self):
    res = decision_support_system.NeuralNetwork()
    result = res.prediction()

    # Histogram
    figure = Figure(figsize=(6, 3), dpi=100)
    ax = figure.add_subplot(111)

    objects = ('A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J')

    data_file = read("account.csv", delimiter=",").head()
    mas = str(data_file.values[::, 1:11])
    mas = mas.replace('[', '')
    mas = mas.replace(']', '')
    mas = mas.split(' ')
    print(mas)
    mas = list(mas)
    i = 0
    while i < len(mas):
        mas[i] = int(mas[i])
        i+=1

    barchart = ax.bar(objects, mas, align='center', alpha=0.5, bottom=0)
    barchart[0].set_color('orange')
    barchart[1].set_color('orange')
    barchart[2].set_color('green')
    barchart[3].set_color('green')
    barchart[4].set_color('green')
```

```
barchart[5].set_color('blue')
barchart[6].set_color('blue')
barchart[7].set_color('blue')
barchart[8].set_color('red')
barchart[9].set_color('red')

ax.set_xticks(np.arange(len(objects)))
ax.set_xticklabels(objects, fontdict=None, minor=False)

y_labels = (0, 1, 2)
ax.set_yticks(y_labels)
ax.set_yticklabels(y_labels, fontdict=None, minor=False)

ax.yaxis.grid()
```

```
        canvas = FigureCanvasTkAgg(figure, master=self.frame_graph)

        canvas.draw()
        canvas.get_tk_widget().pack(side=Tk.TOP, fill=Tk.BOTH, expand=1)

        if result == 'Real':
            self.status_entry.insert(1.0, "The page is real")
            self.status_entry.tag_add("here", "1.0", "1.64")
            self.status_entry.tag_config("here", foreground="green")
        elif result == 'Fake':
            self.status_entry.insert(1.0, "The page is fake")
            self.status_entry.tag_add("here", "1.0", "1.64")
            self.status_entry.tag_config("here", foreground="red")
        return result

driver = webdriver.Chrome()
gui = GUI()
driver.close()
```

Файл decision_support_system.py

```
from pandas import read_csv as read
from sklearn.model_selection import train_test_split as train
from sklearn.ensemble import RandomForestClassifier
from tkinter import messagebox as mb

class NeuralNetwork:
    def __init__(self):
        try:
            path = "training.csv"
            data = read(path, delimiter=",")
            data.head()
            self.X = data.values[::, 1:11]
            self.y = data.values[::, 11:12]
            self.clf = RandomForestClassifier(n_estimators=100, n_jobs=-1)
        except OSError as error:
            mb.showinfo("No training dataset", "File training.csv doesn`t exist")

    def training(self):
        X = self.X
        y = self.y
        clf = self.clf
        X_train, X_test, y_train, y_test = train(X, y, test_size=0.6)
        clf.fit(X_train, y_train.ravel())
        accuracy = clf.score(X_test, y_test)
        print('Accuracy: ' + str(accuracy))
        return accuracy

    def prediction(self):
        self.training()
        clf = self.clf
        path_test = "account.csv"
```

```
    data_test = read(path_test, delimiter=",")
    data_test.head()
    row_test = data_test.values[::, 1:11]
    predicted = clf.predict(row_test)
    result = predicted[0]
    return result

def __str__(self, result):
    result = self.result
    return result
```

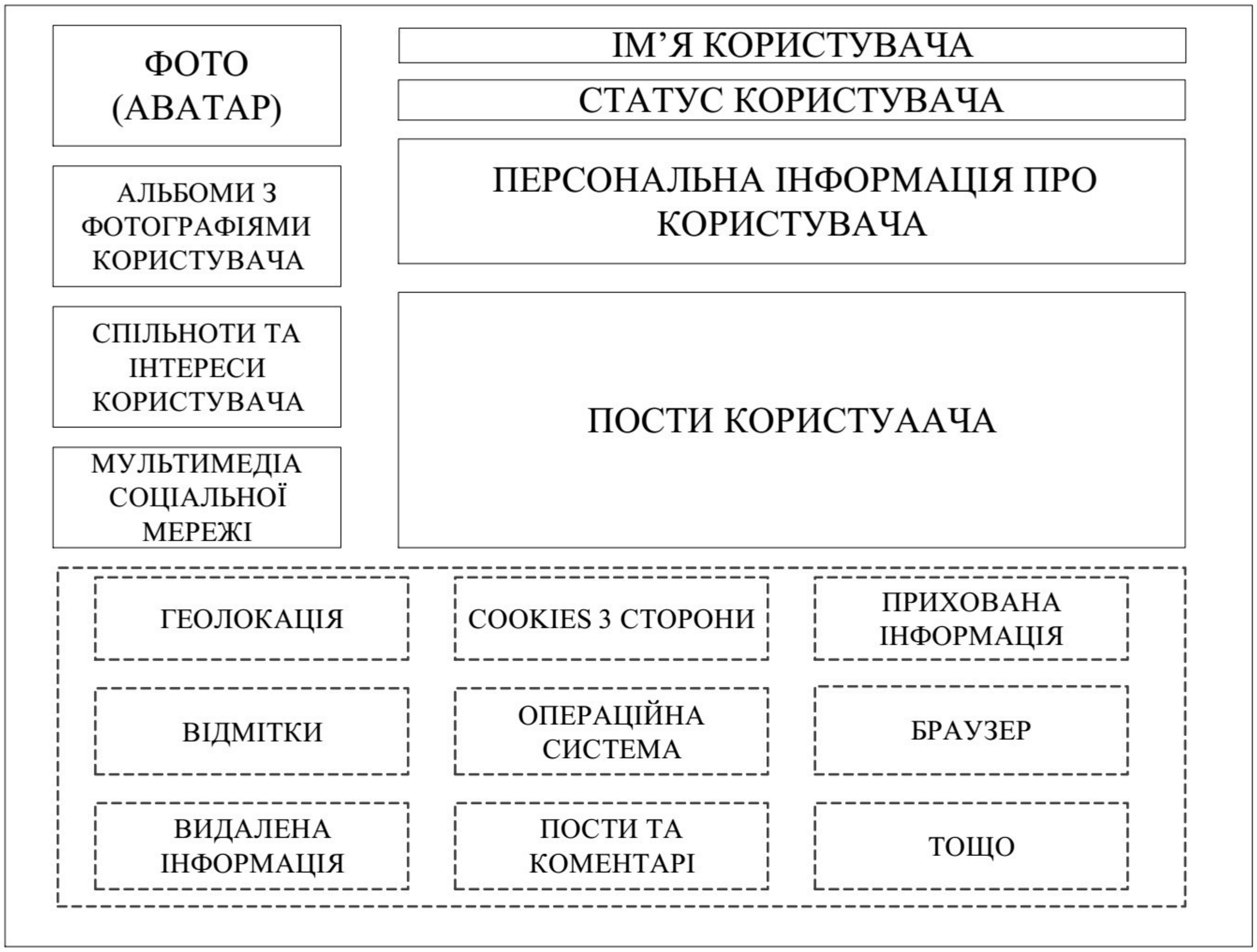
ІЛЮСТРАТИВНА ЧАСТИНА

Критерії встановлення балів відповідно до значень метрик

Бали / Параметр	0	1	2
Наявність фото на аватарі	На фото зображена людина	Фото існує	Фото не існує
Наявність фото на фоні	Фото існує	-	Фото не існує
Кількість фотографій	Фото не існує	Кількість фото < 10 & Кількість фото > 1000	10 < Кількість фото < 1000
Кількість друзів	10 < Кількість друзів < 2000	0 < Кількість друзів < 10 & Кількість друзів > 2000	Друзів немає
Кількість постів	10 < Кількість постів < 500	Кількість постів > 500 & Кількість постів < 10	Пости не існують
Наявність персональної інформації	Всі поля заповнено	Частина полів заповнено	Інформація відсутня
Контактна інформація	Контакти присутні	-	Контакти відсутні
Дата народження	1932 < Рік народження < 2009	2009 < Рік народження || Рік народження < 1932	Рік народження відсутній
Лайки на аватарі	Кількість лайків	Лайки від друзів < Лайки від чужинців	Лайки відсутні
Кількість коментарів на аватарі	5 < Кількість коментарів < 100	Кількість коментарів < 5 & Кількість коментарів > 100	Коментарі відсутні

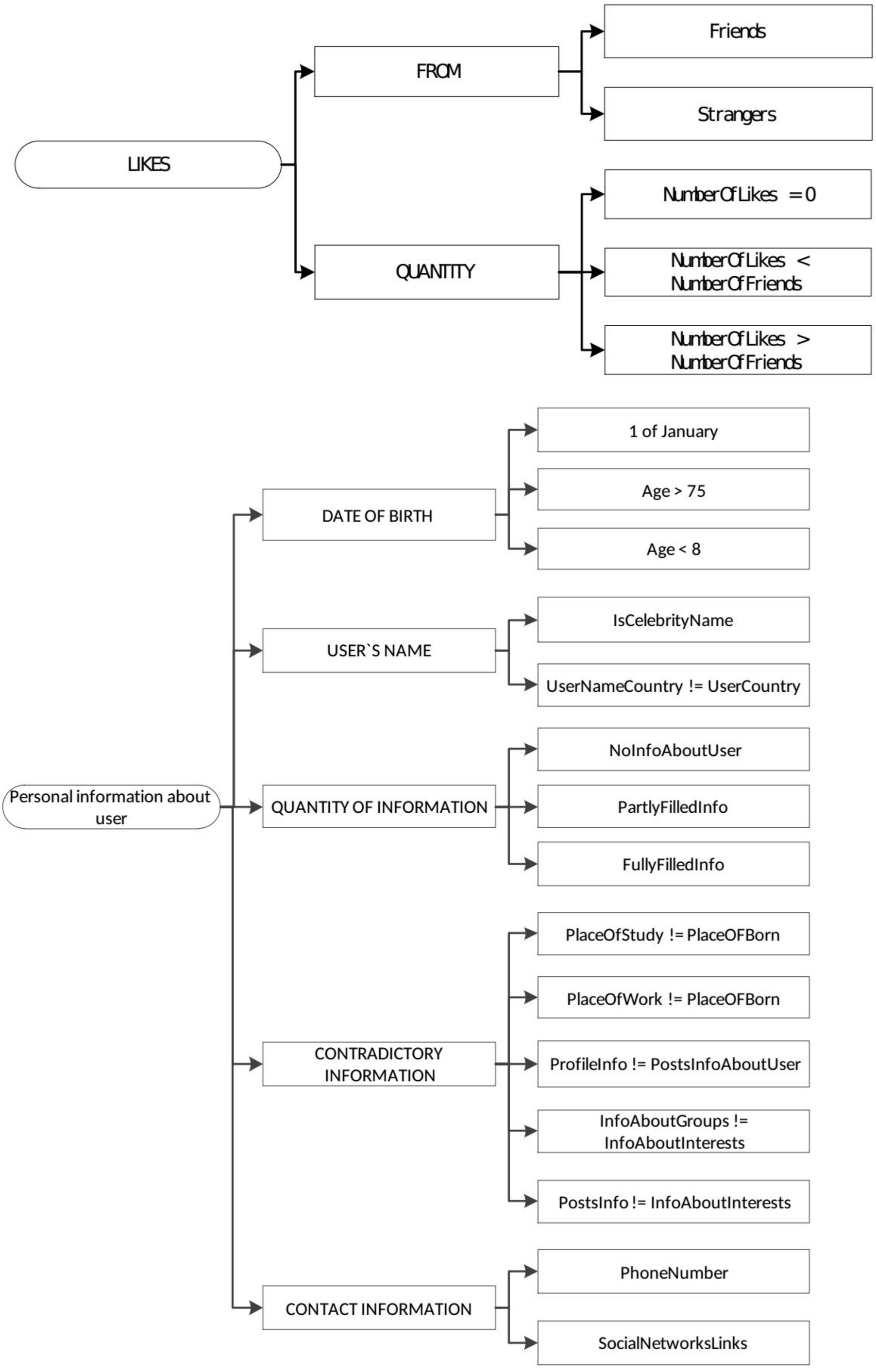
					08-20.МКР.003.00.000 ІЧ1		
Змн	Арк.	№ докум.	Підпис	Дата			
Розроб.		ГоловенькоВ.О.			Система виявлення фейкових облікових записів у соціальних мережах. Критерії встановлення балів відповідно до значень метрик	Літ.	Маса / Масштаб
Перевір.		Войтович О.П.					
Реценз.		Азарова А. О.				ВНТУ, гр. 1 БС-18 м	
Н. Контр.		Войтович О.П.					
Затверд.		ЛужецькийВ.А.					

Структура облікового запису



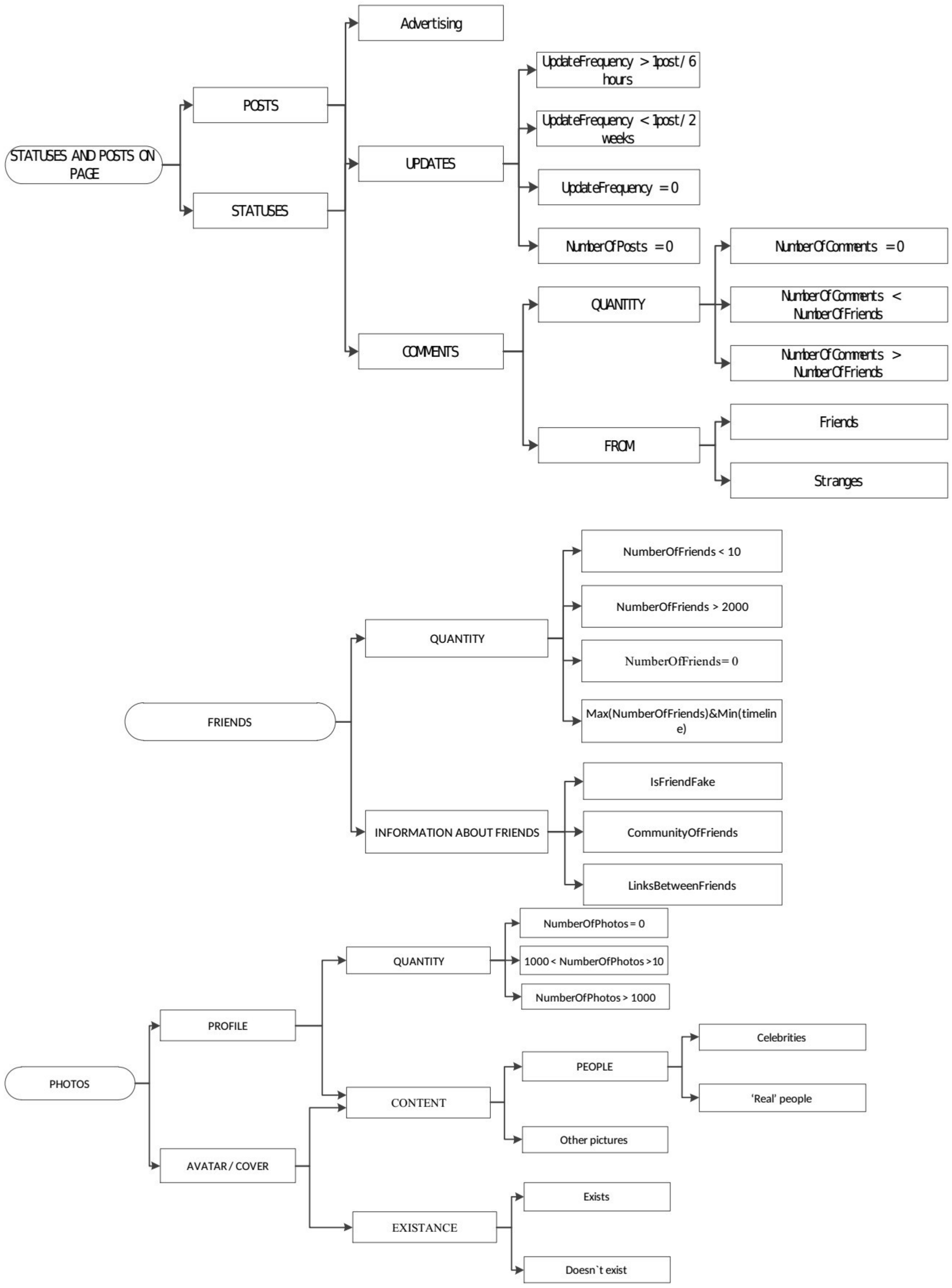
					08-20.МКР.003.00.000 ІЧ2			
Змн	Арк.	№ докум.	Підпис	Дата				
Розроб.		ГоловенькоВ.О.			Система виявлення фейкових облікових записів у соціальних мережах. Структура облікового запису	Літ.	Маса	Масштаб
Перевір.		Войтович О.П.						
Реценз.		Азарова А. О.				ВНТУ, гр. 1 БС-18 м		
Н. Контр.		Войтович О.П.						
Затверд.		ЛужецькийВ.А.						

Структурна модель ознак фейковості у категоріях «Лайки» та «Персональна інформація про користувача»

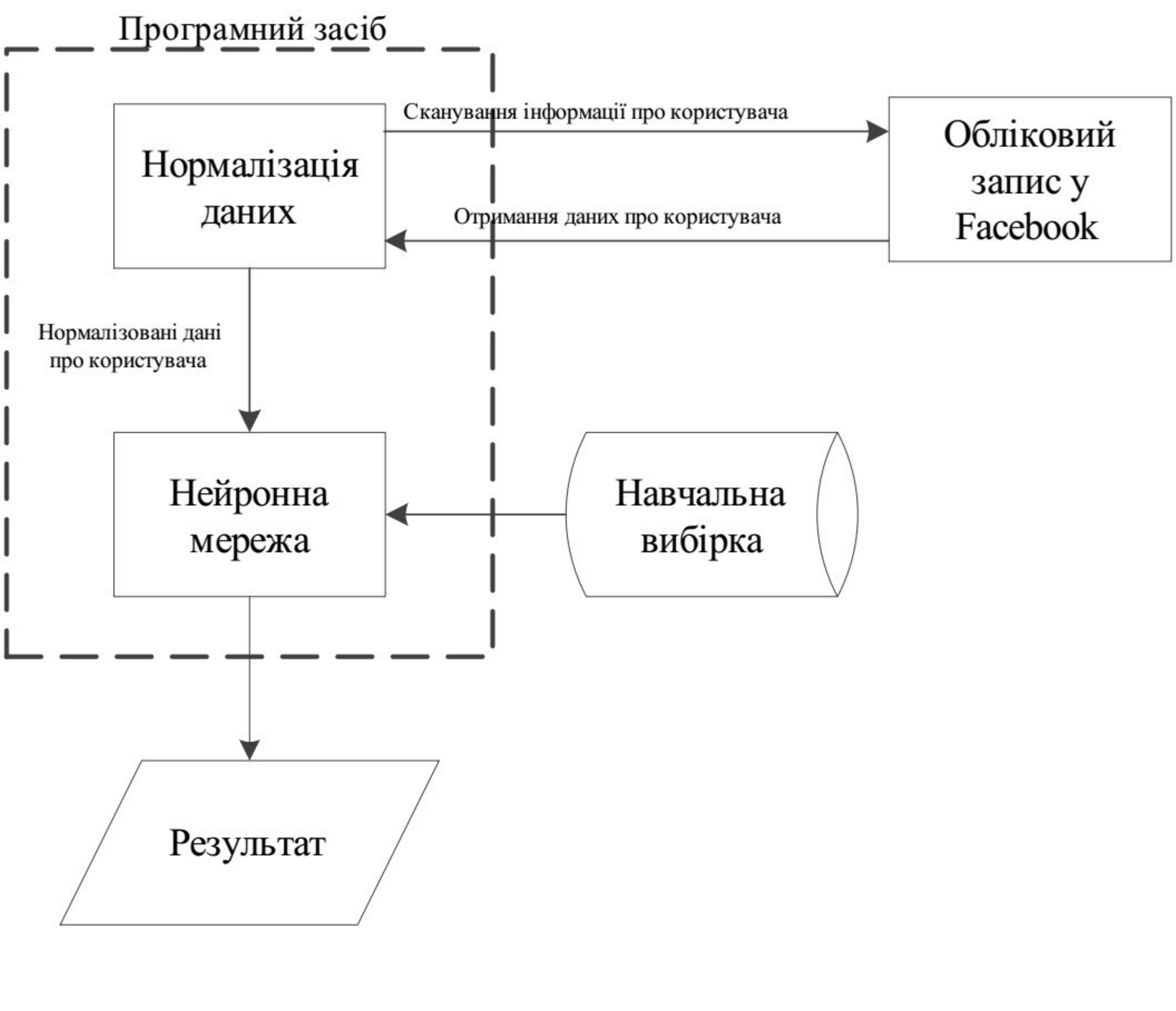


					08-20.МКР.003.00.000 ІЧЗ			
Змн	Арк.	№ докум.	Підпис	Дата				
Розроб.		ГоловенькоВ.О.			Система виявлення фейкових облікових записів у соціальних мережах. Структурна модель ознак у категоріях «Лайки» та «Песональна інформація про користувача»	Літ.	Маса	Масштаб
Перевір.		Войтович О.П.						
Реценз.		Азарова А. О.				ВНТУ, гр. 1 БС-18 м		
Н. Контр.		Войтович О.П.						
Затверд.		ЛужецькийВ.А.						

Структурна модель ознак фейковості у категоріях «Статуси і пости», «Друзі» та «Фото»

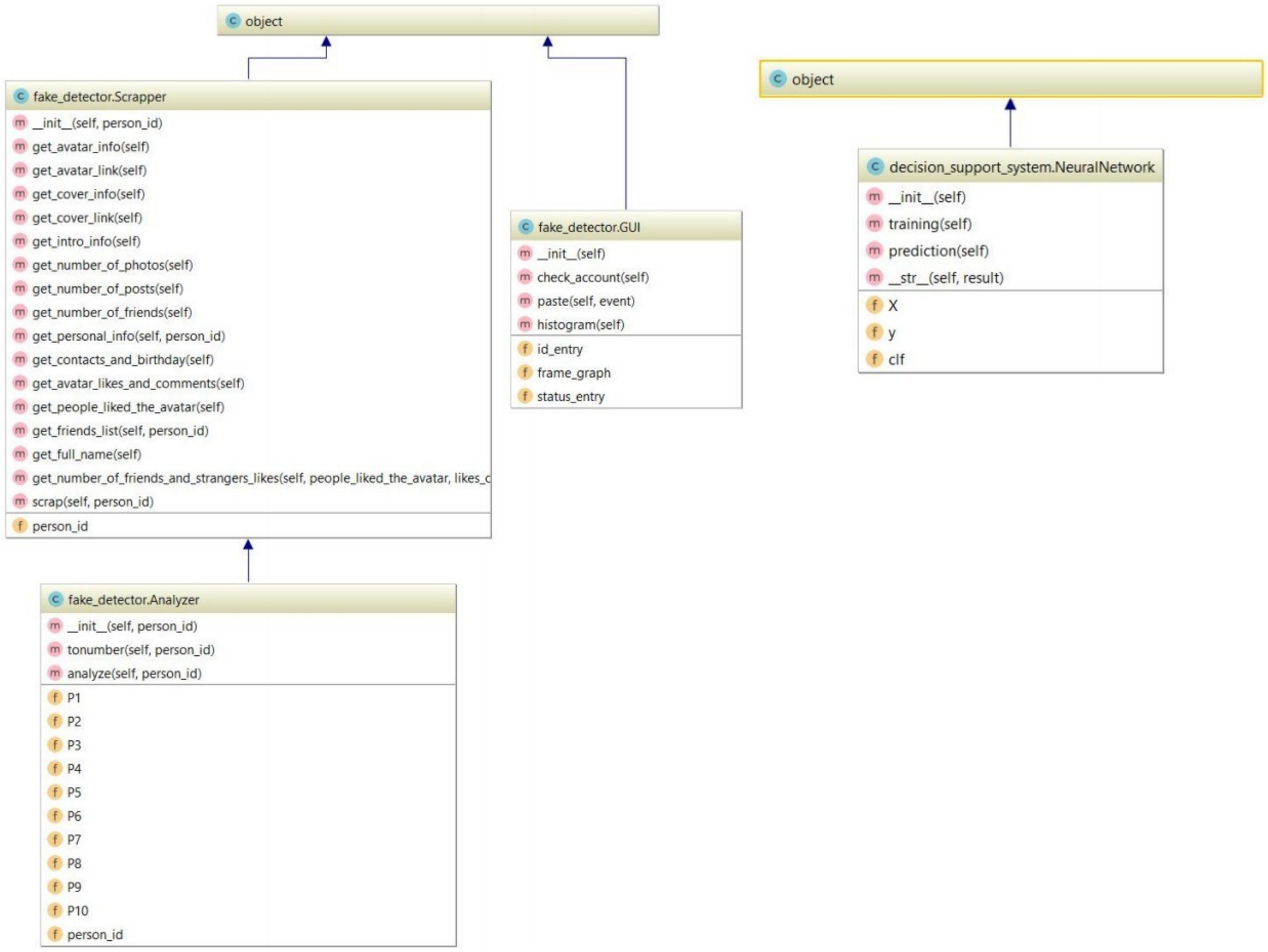


Архітектура програмного засобу



					08-20.МКР.003.00.000 ІЧ5		
Змн	Арк.	№ докум.	Підпис	Дата			
Розроб.		ГоловенькоВ.О.			Система виявлення фейкових облікових записів у соціальних мережах. Архітектура програмного засобу	Літ.	Маса / Масштаб
Перевір.		Войтович О.П.					
Реценз.		Азарова А. О.				ВНТУ, гр. 1 БС-18 м	
Н. Контр.		Войтович О.П.					
Затверд.		ЛужецькийВ.А.					

UML-діаграма класів програмного засобу

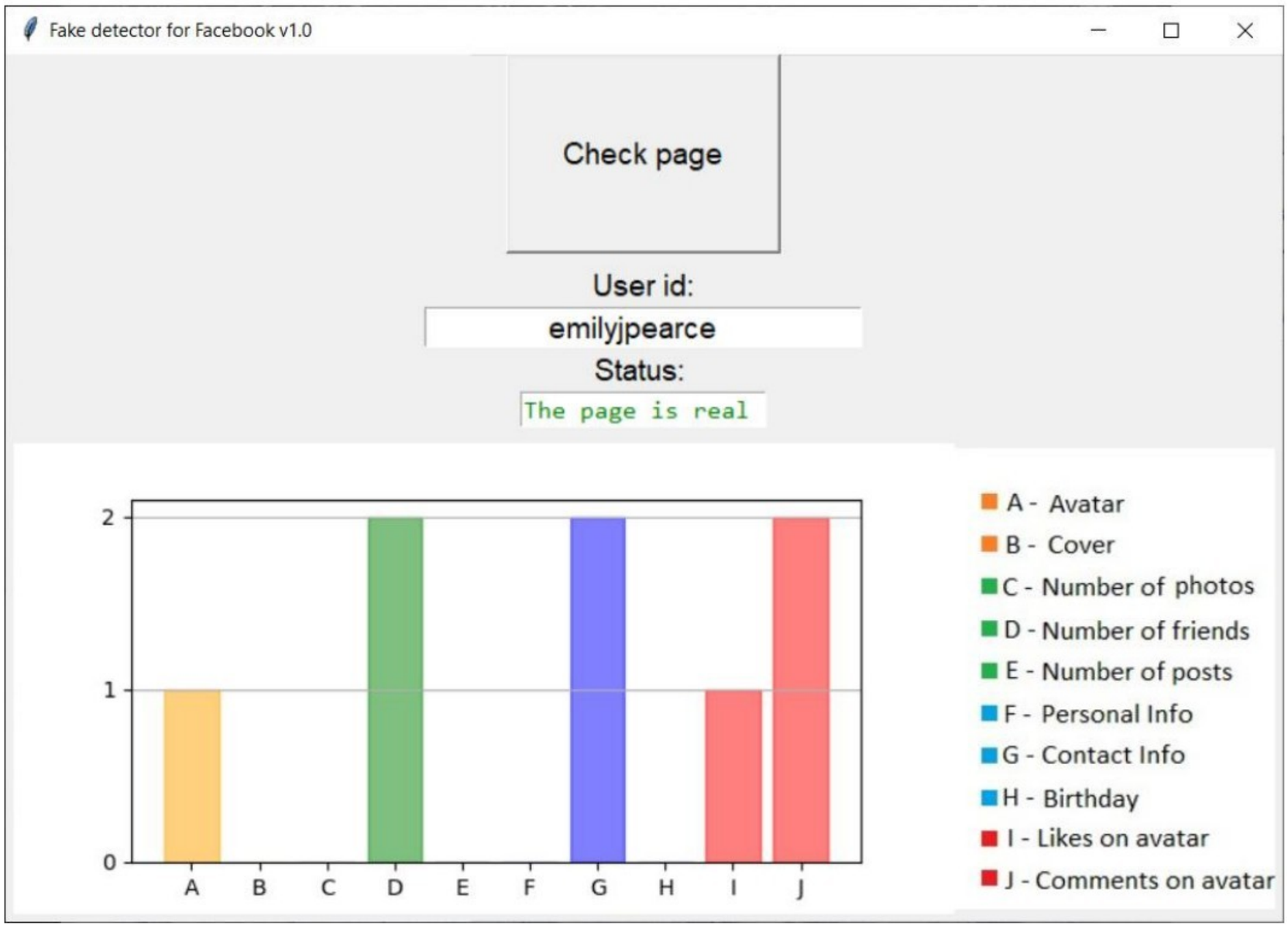


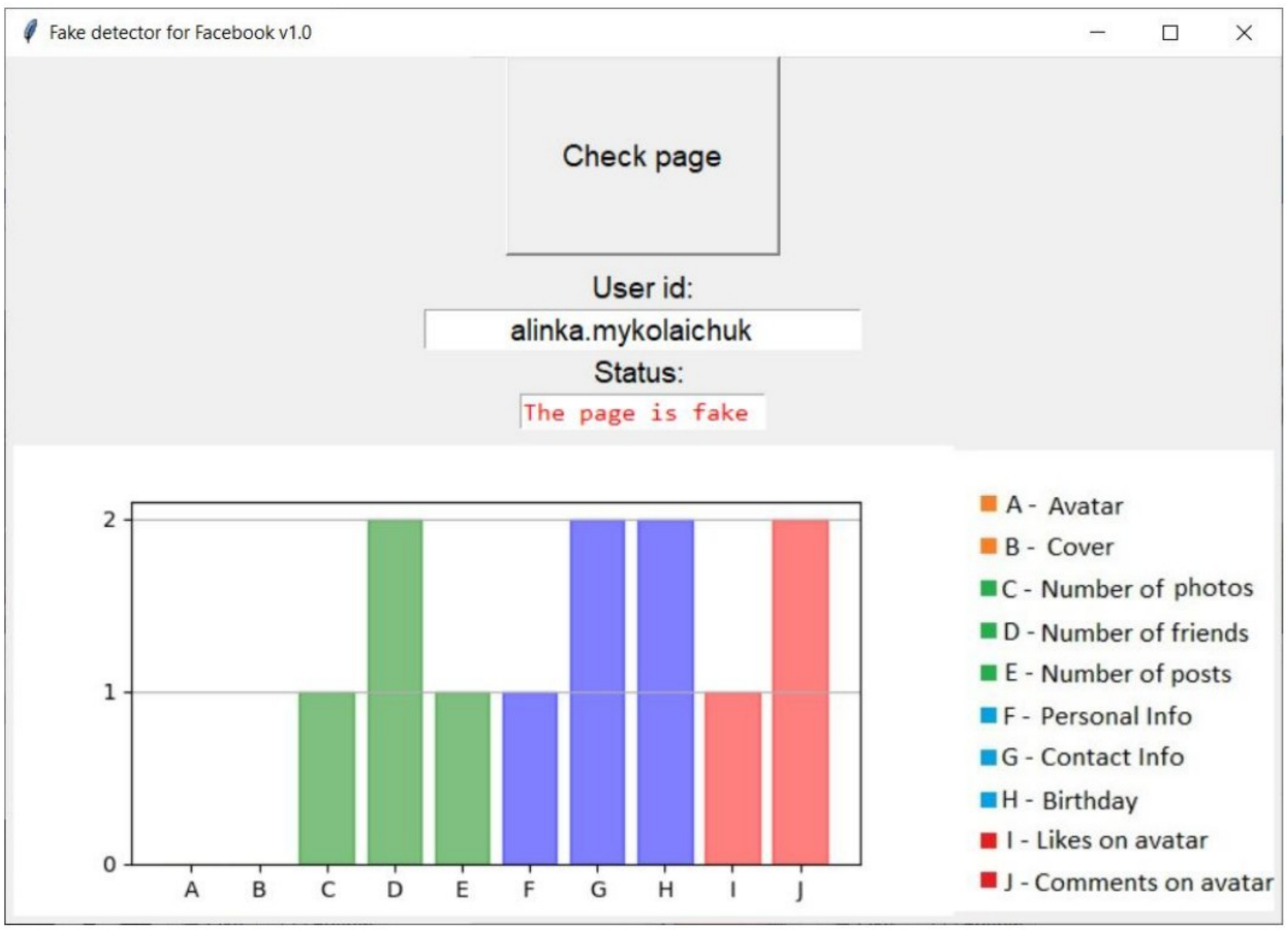
					08-20.МКР.003.00.000 ІЧ6
Змн	Арк.	№ докум.	Підпис	Дата	
Розроб.		ГоловенькоВ.О.			Система виявлення фейкових облікових записів у соціальних мережах. UML-діаграма класів програмного засобу
Перевір.		Войтович О.П.			
Реценз.		Азарова А. О.			
Н. Контр.		Войтович О.П.			
Затверд.		ЛужецькийВ.А.			

Літ.	Маса	Масштаб

ВНТУ, гр. 1 БС-18 м

Інтерфейс програмного засобу





					08-20.МКР.003.00.000 ІЧ7		
Змн	Арк.	№ докум.	Підпис	Дата			
Розроб.		ГоловенькоВ.О.			Система виявлення фейкових облікових записів у соціальних мережах. Інтерфейс програмного засобу	Літ.	Маса / Масштаб
Перевір.		Войтович О.П.					
Реценз.		Азарова А. О.				ВНТУ, гр. 1 БС-18 м	
Н. Контр.		Войтович О.П.					
Затверд.		ЛужецькийВ.А.					