Міністерство освіти і науки України
Київський національний університет імені Тараса Шевченка

Факультет інформаційних технологій
Кафедра кібербезпеки та захисту інформації

ПОЯСНЮВАЛЬНА ЗАПИСКА
Дипломної роботи
магістра
(назва освітньо-кваліфікаційного рівня)

галузь знань 12 Інформацімайні технології
(шифр і назва галузі знань)
спеціальність 125 Кібербезпека
(код і назва спеціальності)
освітній ступень магістр
(назва освітньої програми)
освітньо-наукова програма кібербезпека

на тему: «Система детектування фішингу на основі використання нейронних мереж»

Виконавець: студента II курсу, групи КБм-21
(підпис) **Хроленко Ярослав Олексійович** (прізвище ім'я по-батькові)

	Прізвище, ініціали	**Оцінка**	**Підпис**
Науковий керівник	Бучик С.С.		
Рецензент	Корніенко Б.Я.		
Нормоконтроль	Даков С.Ю.		

Київ 2022

Міністерство освіти і науки України
Київський Національний університет імені Тараса Шевченка

Факультет інформаційних технологій
Кафедра кібербезпеки та захисту інформації

ЗАТВЕРДЖЕНО:
завідувач кафедри кібербезпеки
та захисту інформації
_____Лукова-Чуйко Н.В.
«__» _____ 2021року

ЗАВДАННЯ
на виконання дипломної роботи

спеціальності 125 Кібербезпека
(код і назва спеціальності)

студенту КБм-21 **Хроленку Ярославу Олексійовичу**
(група) (прізвище ім'я по-батькові)

Тема дипломного роботи Система детектування фішингу на основі використання нейронних мереж

1. ПІДСТАВИ ДЛЯ ПРОВЕДЕННЯ РОБОТИ

Рішення засідання кафедри кібербезпеки та захисту інформації факультету інформаційних технологій протокол № 5 від 29.10.2021

2. МЕТА ТА ВИХІДНІ ДАНІ ДЛЯ ПРОВЕДЕННЯ РОБІТ

Об'єкт досліджень	Процес захисту від фішингових атак .
Предмет досліджень	Методи захисту від фішингових атак з використанням нейромереж
Мета	Підвищення ефективності засобів захисту від фішингових атак на основі використання нейромереж
Вихідні дані для проведення роботи	Сучасні дослідження виявлення фішингових атак засобами нейронних мереж.

3. ОЧІКУВАНІ НАУКОВІ РЕЗУЛЬТАТИ

Наукова новизна удосконалення засобів захисту від фішингових атак на основі застосування гібридних архітектур нейронних мереж

Практична цінність Розроблений нейромережгний модуль вирішує задачу детектування фішингу і може бути використаний як складова частина технологій кіберзахисту.

4. ВИМОГИ ДО РЕЗУЛЬТАТІВ ВИКОНАННЯ РОБОТИ

Робота виконана у повному обсязі відповідно до теми.

5. ЕТАПИ ВИКОНАННЯ РОБОТИ

Найменування етапів робіт	Строки виконання робіт (початок-кінець)
Розробка плану для досягнення мети роботи	29.10.2021 – 23.01.2022
Аналіз літературних джерел	24.01.2022 – 14.02.2022
Розробка методу захисту від витоку даних платіжних карток через інтернет-браузер	15.02.2022 – 24.04.2022
Оформлення і друк пояснювальної записки	25.04.2022 – 19.05.2022

6. РЕАЛІЗАЦІЯ РЕЗУЛЬТАТІВ ТА ЕФЕКТИВНІСТЬ

Економічний ефект Зниження збитків через викрадення даних

Соціальний ефект Покращення технологій забезпечення захисту інформації як особисто так і на підприємствах.

7. ДОДАТКОВІ ВИМОГИ

Завдання видав _____ _____
(підпис) (прізвище, ініціали)

Завдання прийняв до виконання _____ _____
(підпис) (прізвище, ініціали)

Дата видачі завдання: _____
Термін подання дипломної роботи до ЕК _____

РЕФЕРАТ

Пояснювальна записка до дипломної роботи «Система детектування фішингу на основі використання нейронних мереж»: 65 сторінка, 12 рисунків та 3 таблиці. 51 літературних джерел.

Об’єкт дослідження – процес захисту від фішингових атак .

Мета роботи – розробка методу захисту даних платіжних карток від витоку через інтернет-браузер.

Методи дослідження – Підвищення ефективності засобів захисту від фішингових атак на основі використання нейромереж.

У роботі досліджено сучасні загрози та методи протидії фішинговим атакам задля отримання персональних даних. Проведено аналіз існуючих методів та засобів захисту. Запропоновано метод захисту фішингових атак засобами нейронних мереж.

Наукова новизна: запропоновано удосконалення засобів захисту від фішингових атак на основі застосування гібридних архітектур нейронних мереж.

Актуальність теми:. В даний момент фішинг є одним із найпоширеніших видів інтернет злочину. Фішинг атаки призводять до великих втрат включаючи особисті дані, конфіденційну інформацію, комерційному або державну таємницю. Таким чином, забезпечення інформаційної безпеки, використовуючи системи детектування фішингу на основі використання нейронних мереж є актуальними сьогодні.

Ключові слова: фішинг, штучна нейронна мережа, глибинне навчання, згорткова нейрона мережа, рекурентна нейрона мережа, мережа довгої короткостроковій пам'яті, , гібридна нейрона мережа.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

AOL	–	America Online
CNN	–	Convolutional neural network
CSS	–	Cascading Style Sheets
DDoS	–	Distributed denial-of-service attack
DL	–	Deep learning
DNN	–	Deep neural network
DoS	–	Denial-of-service attack
FC	–	Fully connected layer
GRU	–	Gated recurrent units
HTML	–	HyperText Markup Language
IP	–	Internet Protocol
ISP	–	Internet service provider
LSTM	–	Long short-term memory
ML	–	Machine learning
ReLU	–	Rectified Linear Unit
SEO	–	Chief Executive Officer
SMS	–	Short Message Service
URL	–	Uniform Resource Locator

ЗМІСТ

ВСТУП

Актуальністю даної роботи є опис нового методу захисту конфіденційної інформації. Даний метод можливо буде впровадити та використовувати для будь-якого браузеру.

Основною задачею цієї кваліфікаційної роботи є знайти, розглянути, проаналізувати найефективніші варіанти захисту конфіденційних .

Науковою новизною цієї кваліфікаційної роботи є удосконалення засобів захисту від фішингових атак на основі застосування гібридних архітектур нейронних мереж .

Об’єктом дослідження э процес захисту від фішингових атак, адже в наш час проблеми пов’язані з втратою даних або їх крадіжкою дуже актуальні та небезпечні, так як несанкціоновані дії користувачів призводять до виникнення загроз.

Предметом дослідження є методи захисту від фішингових атак з використанням нейромереж.

Метою даної дипломної роботи є підвищення ефективності засобів захисту від фішингових атак на основі використання нейромереж

Питання безпеки в мережі інтернет має беззаперечний рівень, адже майже кожен пристрій котрий ми маємо підтримує доступ до всесвітньої павутини. Телефони, комп’ютери, планшети, приставки, телевізори та багато іншого, все це потребує інтернету.

Таке дослідження є дуже актуальним на даний момент,адже в наш час. Адже маючи натреновані та добре навчені нейронні мережі ми можемо зростити задачу аналізу іфшингових загроз.

Сформовані в результаті, теоретичні та практичні рекомендації можуть бути використані, як організаційна складова в процесі роботи над створенням та впровадження даного методу в інтернет браузери.

РОЗДІЛ 1
МЕТОДИ БОРОТЬБИ З ФІШИНГОВИМИ АТАКАМИ

1.1 Кіберзлочинність та її зв’язок з фішингом

Зі стрімким зростанням користувачів інтернету, люди почали розповсюджувати свої особисті дані, а тому величезна кількість інформації та фінансових записів стали доступними для кіберзлочинців. Фішинг це приклад надефективної форми кіберзлочину , що дозволяє обманути користувача та поцупити персональні дані. Перші фішингові атаки були зафіксовані в 1990 роках і почали ускладнюватися. В даний момент фішинг є одним із найпоширеніших видів інтернет злочину. Фішинг атаки призводять до великих втрат включаючи особисті дані, конфіденційну інформацію, комерційному або державну таємницю. Дослідження класифікують фішингові атаки за основними механізмами фішингу та контрзаходів(див рис 1.1).

Кіберзлочин – це злочин, коли комп’ютер є об’єктом злочину або використовується як інструмент для вчинення правопорушення. Кіберзлочинець може використовувати пристрій для доступу до особистої інформації користувача, конфіденційної ділової інформації, державної інформації . Продаж або отримання вищевказаної інформації в Інтернеті також є кіберзлочином.

Кіберзагроза – це сукупність факторів та умов, що створюють небезпеку порушення інформаційної безпеки.Дії зловмисників можуть бути спрямовані на IT-інфраструктуру, робочі комп'ютери, мобільні пристрої, інші технічні засоби і, нарешті, людину як елемент кіберпростору.

Кібератака – несанкціонований вплив на інформаційні системи та користувачів інформаційних систем з боку кіберзлочинців з використанням технічних засобів та програмного забезпечення з метою отримання доступу до інформаційних ресурсів, порушення нормальної роботи або доступності систем, крадіжки, викривлення чи видалення інформації.

Масова атака – кібератака, спрямована на широке коло організацій та приватних осіб. Під час проведення масової атаки зловмисники можуть обмежуватися однією галуззю економіки або не враховувати галузеву приналежність компаній, їх завданням є охоплення максимальної кількості жертв.

Цільова атака – кібератака, спрямована на конкретну компанію, галузь економіки або обмежене коло приватних осіб. У рамках цільової атаки зловмисники зазвичай проводять попередню розвідку з метою зібрати інформацію про обрану жертву.

Кіберзлочини поділяються на три великі категорії: індивідуальні, майнові та державні. Спираючись на категорію кіберзлочинів, кіберзлочинці використовують різні рівні та типи загроз.

Індивідуальні: ця категорія кіберзлочинності включає поширення шкідливої або незаконної інформації через Інтернет і цифрові програми однією особою. Кіберзмови, розповсюдження порнографії та торгівля людьми – це кілька прикладів цієї категорії кіберзлочинності.

Майнові: цей кіберзлочин схожий на інцидент у реальному житті, коли злочинець незаконно отримує інформацію про банк або кредитну картку. Хакер краде банківські реквізити особи, щоб отримати гроші, або здійснює фішингові шахрайства в Інтернеті, щоб отримати інформацію від людей.

Державні: Це найменш частий кіберзлочин, але це найсерйозніший проступок. Кіберзлочин проти уряду також вважається кібертероризмом. Урядові кіберзлочини включають злом веб-сайтів, військових веб-сайтів або розповсюдження урядової пропаганди.

Види кіберзлочинів:

DDoS атаки – це зловмисна спроба порушити нормальний трафік цільового сервера, служби або мережі шляхом перевантаження цілі або навколишньої інфраструктури потоком Інтернет-трафіку.

Ботнети – це мережі зі зламаних комп’ютерів, які контролюються ззовні віддаленими хакерами. Віддалені хакери потім надсилають спам або атакують інші

комп’ютери через ці ботнети. Ботнети також можуть використовуватися як шкідливі програми та виконувати інші завдання.

Cyberstalking – Цей вид кіберзлочинності передбачає переслідування в Інтернеті , коли користувач отримує безліч онлайн-повідомлень та електронних листів. Зазвичай кіберсталкери використовують соціальні мережі, веб-сайти та пошукові системи, щоб залякати користувача та вселяти страх. Зазвичай кібер-переслідувач знає свою жертву і змушує людину відчувати страх або стурбованість за свою безпеку.

Віруси – потенційно небажані програми менш небезпечні, ніж інші кіберзлочини, але є різновидом зловмисного програмного забезпечення. Вони видаляють необхідне програмне забезпечення у вашій системі, включаючи пошукові системи та попередньо завантажені програми. Вони можуть включати шпигунське або рекламне програмне забезпечення.

Exploit Kits – це готові інструменти, які злочинці можуть придбати в Інтернеті та використати проти будь-кого, хто має комп’ютер. Набори експлойтів регулярно оновлюються, як і звичайне програмне забезпечення, і доступні на форумах про хакерство темної мережі.

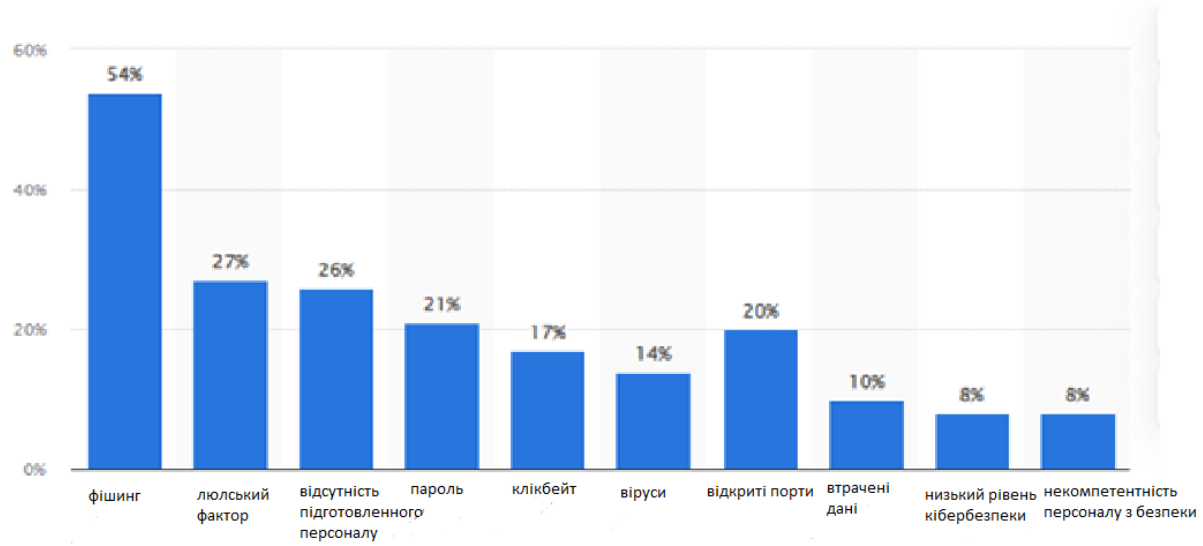

Рисунок 1.1 – Розподіл атак за видами

Фішинг – тип атаки який передбачає, що хакери надсилають користувачам шкідливі вкладення електронною поштою або URL-адреси, щоб отримати доступ до

їхніх облікових записів або комп’ютера.[1] Кіберзлочинці стають все більш підготовленими, і багато з цих листів не позначаються як спам . Користувачі отримують електронні листи, які стверджують, що їм потрібно змінити пароль або оновити платіжну інформацію, надавши доступ злочинцям.

1.2 Фішинг як різновид кіберзагрози

Фішинг – це автоматизована форма соціальної інженерії, за допомогою якої зловмисники використовують Інтернет для шахрайського вилучення конфіденційної інформації компаній та окремих осіб, часто видаючи себе за легальні веб-сайти. Високий потенціал здобичі, наприклад, через доступ до банківських рахунків і номерів кредитних карток, простота надсилання підроблених повідомлень електронної пошти, що видають себе за законні органи влади, і труднощі правоохоронних органів у переслідуванні злочинців призвели до сплеску фішингових атак в останні роки . У звіті «State of the Phish»[2] за 2019 рік було виявлено, що майже 90% організацій зазнали цілеспрямованих фішингових атак у 2019 році, 84% повідомили про фішинг через SMS/текст, 83% стикалися з голосовим фішингом, а обсяг повідомлень електронної пошти зріс на 67% за рік. Дані свідчать про те, що все більше людей ухиляються від інтернет-комерції через загрозу шахрайства з ідентифікацією, незважаючи на тенденцію компаній брати на себе ризик шахрайства.

1.3 Фішинг розповсюдження та напрямки атак

Країною з найбільшою часткою атакованих фішерами користувачів у першому кварталі 2019 стала Бразилія – тут цей показник досяг 21,66% (див. рис.1.1).

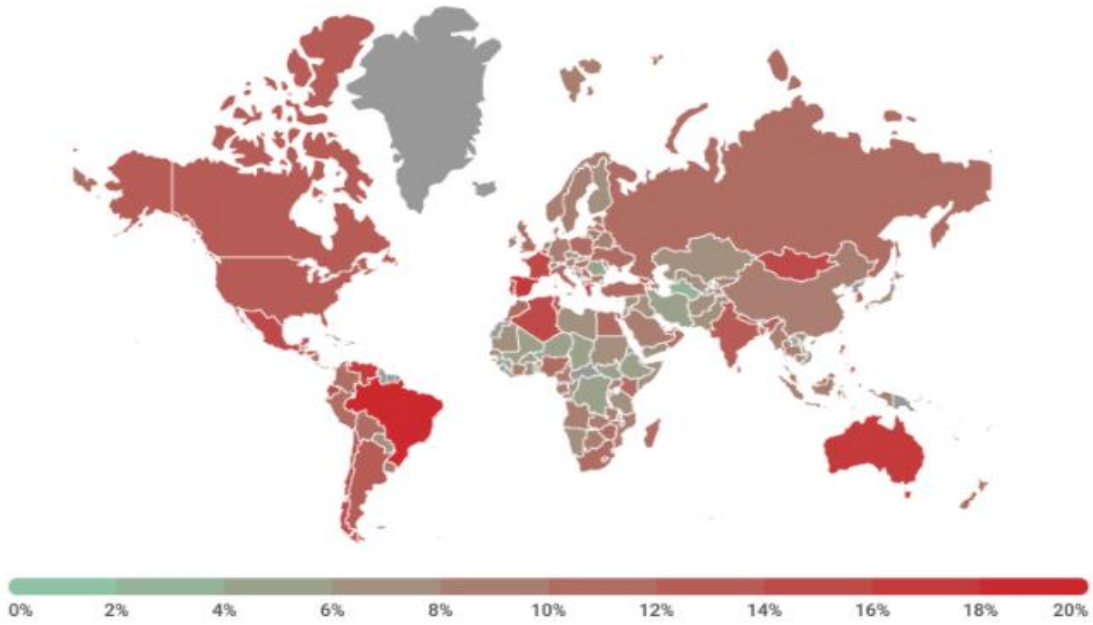

Рисунок 1.2 – Розподіл фішинг-атак у світі

Рейтинг атакованих фішерами організацій ґрунтується на спрацюваннях евристичного компонента системи «Антифішинг» на комп'ютерах користувачів.При цьому не важливо, яким чином відбувається перехід: внаслідок натискання на посилання у фішинговому листі, у повідомленні в соціальній мережі або через дії шкідливої програми. Після спрацювання компонента користувач бачить у браузері банер, який попереджає про можливу загрозу (див. рис.1.3).

У першому кварталі 2021 року на першому місці за кількістю атак залишається банківський сектор.

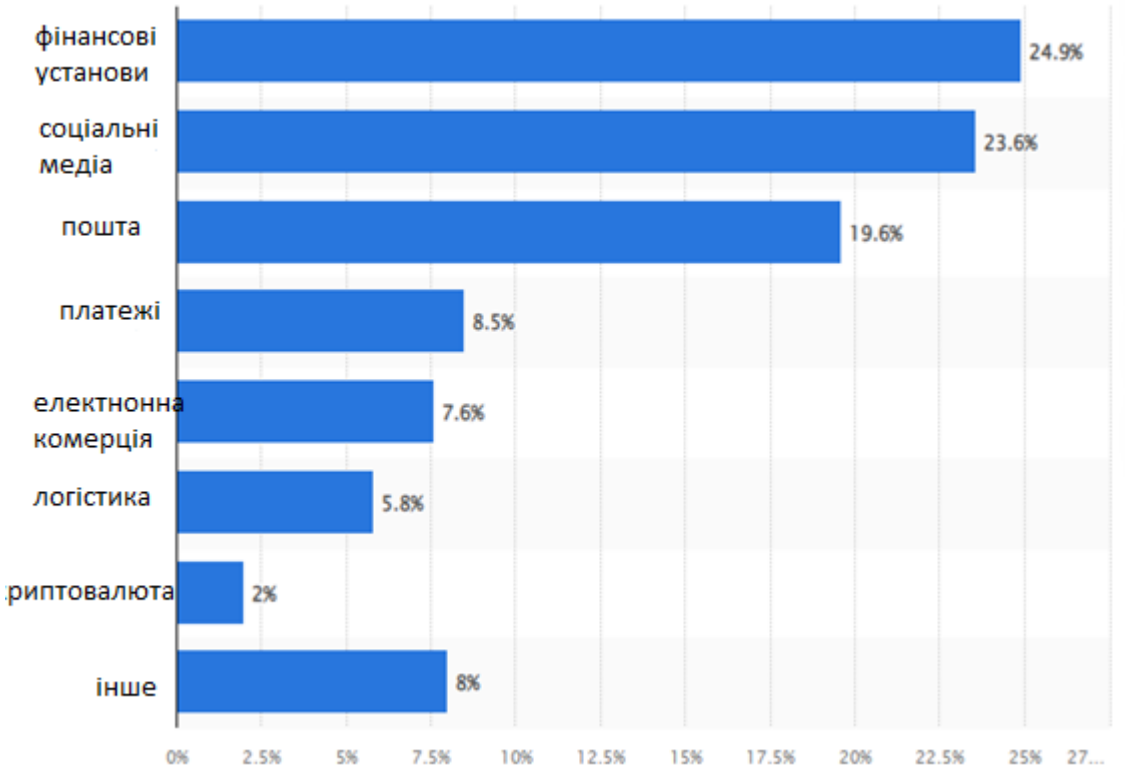

Рисунок 1.3 – Розподіл фішинг-атак за цілями

Зловмисники швидко підхопили тему загального занепокоєння з приводу коронавірусної інфекції та почали використовувати її для фішингових листів. За нашими підрахунками, у I кварталі близько 13% атак, у яких кіберзлочинці задіяли методи соціальної інженерії, пов'язані з коронавірусом(див. рис.1.4).

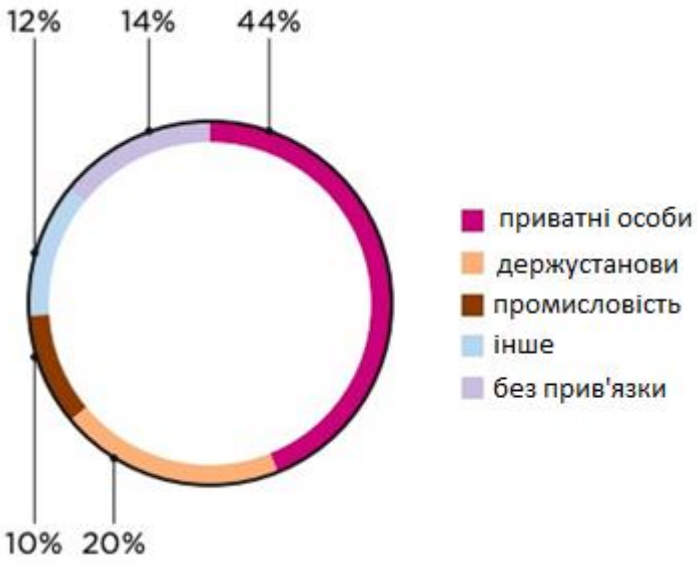

Рисунок 1.4 – Розподіл фішинг-атак за цілями підчас коронавірусу

1.4 Історія фішингу

Ще до того, як термін «фішинг» міцно узвичаївся, методи фішингу були докладно описані в доповіді та презентації, які підготувала в 1987 році компанія Interex (International HP Users Group)[3].

Використання цього терміну починається в середині 1990-х років, а його перша згадка приписується сумнозвісному спамеру і хакеру Хану Сі Сміту (Khan C Smith). Крім того, в Інтернеті зберігся перший випадок публічної згадки терміна "фішинг". Це сталося 2 січня 1996 року в Usenet – у групі новин AOHell. На той момент компанія America Online (AOL) була найбільшим інтернет-провайдером, який щодня обслуговував мільйони підключень. Очевидно, популярність компанії AOL зробила її мішенню шахраїв. Хакери та розповсюджувачі піратських програм використовували її ресурси для обміну повідомленнями, а також для фішингових атак на комп'ютери законослухняних користувачів. Коли AOL вжила заходів та закрила групу AOHell, зловмисники взяли на озброєння інші методи. Вони надсилали користувачам мереж AOL повідомлення, в яких представлялися співробітниками AOL та просили користувачів перевірити дані своїх облікових записів або передати їм свої платіжні реквізити. В результаті проблема стала настільки гострою, що компанія AOL почала додавати попередження до кожного електронного листа, особливо вказуючи, що жоден співробітник AOL не буде просити повідомити йому пароль або платіжні реквізити користувачів[3].

З настанням 2000-х років фішингові шахраї почали звертати увагу на вразливості систем електронних платежів. Клієнти банків і платіжних систем стали дедалі частіше ставати жертвами фішингу, а деяких випадках як показало подальше розслідування – зловмисникам навіть вдавалося точно ідентифікувати своїх жертв та дізнаватися, яким банком вони користувалися. Соціальні мережі також стали однією з головних мішеней фішингу через свою привабливість для шахраїв: особиста інформація, що публікується в соціальних мережах, є чудовою підмогою для крадіжки ідентифікаційних даних.

Кіберзлочинці реєстрували десятки доменів, які настільки витончено імітували такі ресурси, як eBay і PayPal, що багато не надто уважних користувачів просто не помічали підміни. Клієнти системи PayPal отримували фішингові електронні листи (що містять посилання на підставний веб-сайт) із проханням оновити номер кредитної картки та інші персональні дані. У вересні 2003 року про першу фішингову атаку проти банку повідомив журнал The Banker (що належить компанії The Financial Times Ltd.).

У середині 2000-х років на чорному ринку можна було замовити «під ключ» шкідливе програмне забезпечення для фішингу. У той же час, хакери почали координувати свої дії, щоб організовувати все більш витончені фішингові атаки. Важко оцінити навіть приблизні втрати від успішних фішингових атак: як повідомляв у 2007 році звіт компанії Gartner, за період з серпня 2006 по серпень 2007 року близько 3,6 мільйона дорослих користувачів втратили 3,2 мільярда доларів.

У 2011 році фішингові шахраї навіть нібито знайшли державних спонсорів, коли китайська влада запустила передбачувану фішингову кампанію, яка була спрямована проти облікових записів Gmail, що належать високопоставленим чиновникам та військовим у США та Південній Кореї, а також китайським політичним активістам.

Можливо, найвідомішою фішинговою атакою став випадок, коли у 2013 році було викрадено 110 мільйонів записів кредитних карток та облікових даних, що належать клієнтам торгової мережі Target. Виною всьому виявився скомпрометований обліковий запис одного субпідрядника.

Ще більшу погану славу здобула фішингова атака, зроблена в першому кварталі 2016 року хакерською групою Fancy Bear (діяльність якої пов'язують із російськими спецслужбами та військовою розвідкою). Ця атака була спрямована на адреси електронної пошти Національного комітету Демократичної партії США. Зокрема, Джон Подеста, керівник агітаційної кампанії Хілларі Клінтон на президентських виборах 2016 року, заявив, що зловмисники зламали його обліковий запис Gmail і викрали листування, оскільки він попався на найстаріший

шахрайський прийом: йому на електронну пошту надійшло фішинговий лист облікового запису було скомпрометовано (тому потрібно «натиснути тут», щоб змінити його).

У 2017 році було здійснено масовану фішингову атаку на Google і Facebook, що змусила бухгалтерські служби цих компаній перерахувати загалом понад 100 мільйонів доларів на закордонні банківські рахунки хакерів.[3]

1.5 Види фішингу

Адресний фішинг – це цілеспрямована атака хакера на конкретну особу або організацію, мета якої - отримання особистих даних користувача. Спрямованість на конкретну жертву - головна риса адресного фішингу. У разі адресного фішингу зловмисник спочатку збирає інформацію про ціль і в залежності від мети атаки це може бути: адреса електронної пошти, імена колег, знайомих, партнерів, хоббі цілі, покупки в інтернет-магазинах тощо. Після цього на основі отриманих даних шахрай готує фішинговий лист або підроблений сайт. У тексті складеного повідомлення може створюватись відчуття терміновості та переконання, щоб змусити жертву виконати необхідні дії. Співробітник компанії через корпоративну пошту відкриває отриманий фішинговий лист. У листі він може прочитати рекламний текст або повідомлення від служби підтримки банку, складений таким чином, щоб співробітник перейшов на фейковий сайт, відкрив вкладення або зробив ще щось, що необхідно зловмиснику. У тексті такого листа може пропонуватися будь-що. Атака вважається успішною, якщо жертва виконала те, що вимагалося у листі. Йдучи на поводу у шахрая, співробітник може завдати шкоди компанії. Це виявляється у крадіжці конфіденційних даних компанії: документи, дані облікових записів, бази даних, крадіжка інтелектуальної власності.

Клоновий фішинг – це тип атаки, в якому шахраї беруть за основу готові листи відомих компаній і замінюють посилання або файли на шкідливі об'єкти.Клоновий фішинг немає спрямованості на певну жертву. Він спрямований широку аудиторію користувачів. Суть полягає в тому, що зловмисники беруть за

основу реальний лист відомої компанії, банку, тощо, замінюють у ньому оригінальні посилання на підроблені, які переводять на хибний сайт або містять шкідливе ПЗ.

Підміна CEO (Whaling) – це метод, який використовують кіберзлочинці, щоб маскуватися під старшого в організації та безпосередньо націлюватись на головних чи інших важливих осіб організації з метою викрадення грошей чи конфіденційної інформації чи отримання доступу до їхніх комп'ютерних систем у злочинних цілях. Цей метод також відомий як Whaling подібний фішингу , оскільки він використовує такі методи, як підробка електронної пошти та веб-сайту, щоб обманом змусити ціль виконати певні дії, наприклад, розкрити конфіденційні дані або переказати гроші [4].

У той час як клоновий фішинг націлений на неконкретних осіб, а адресний фішинг спрямований на конкретних осіб. Whaling робить ставку на останніх, не тільки націлюючись на цих ключових осіб, але й працюючий таким чином, що шахрайські повідомлення, які вони надсилають зловмисники, надходять від когось старшого чи впливового у організації («великих риб» або «китів» у компанії) , наприклад, генеральний директор чи фінансовий менеджер. Це є додатковий елемент соціальної інженерії, коли співробітники неохоче відмовляють у запиті тому, кого вони вважають важливими.

Мобільний фішинг – це телефонний дзвінок (vishing) або SMS-повідомлення (smishing), в якому шахрай намагається переконати співробітника компанії переказати гроші на рахунок шахраїв. Мобільний фішинг, у свою чергу, може бути спрямований як на широку аудиторію, так і певного співробітника компанії. Вам коли-небудь дзвонили з невідомого номера, представляючись співробітниками банку або держслужб, повідомляючи, що у вас не погашений кредит або ви порушили правила дорожнього руху, але у вас немає машини і кредит ви не брали? Якщо такий випадок був, то це спроба фішингової атаки. SMS-фішинг - це злісний брат-близнюк телефонного фішингу, який здійснює ті ж дії, що і телефонний фішинг, але за допомогою смс-повідомлень, додаючи до них шкідливі посилання.

Фармінг – маскування підробленого сайту під оригінальний і приховане перенаправлення користувача на клон оригінального сайту з метою отримання

конфіденційних даних. При використанні фармінга зловмисники створюють підроблений сайт, який не відрізняється від оригіналу, на який через редирект перенаправляють користувача[5].

Рисунок 1.4 – Види фішингових атак

1.6 Контрдії щодо невільювання фішингових втак

Дослідники обговорюють і пропонують низку рішень для подолання проблем фішингу, але все одно немає єдиного рішення, якому можна довіряти чи здатному пом’якшити ці атаки. Пропоновані в літературі заходи протидії фішингу можна розділити на три основні стратегії захисту. Перша лінія захисту – це рішення, засновані на людському факторі, які навчають кінцевих користувачів розпізнавати фішинг і уникати приманки. Друга лінія захисту – це технічні рішення, які передбачають запобігання атаці на ранніх етапах, наприклад на рівні вразливості, щоб запобігти матеріалізації загрози на пристрої користувача, що означає зменшення впливу людини та виявлення атаки.Це також включає застосування конкретних методів для відстеження джерела атаки (наприклад, вони можуть включати ідентифікацію нових зареєстрованих доменів, які тісно збігаються з відомими доменними іменами). Третя лінія захисту – використання правоохоронних органів як стримуючого контролю. Ці підходи можна комбінувати, щоб створити набагато сильніші рішення для боротьби з фішингом.

Освіта людини є ефективним контрзаходом для уникнення та запобігання фішингових атак. Обізнаність і навчання людей є першою лінією оборони у методології боротьби з фішингом, навіть якщо він не передбачає повного захисту. Навчання кінцевих користувачів зменшує сприйнятливість користувачів до

фішингових атак і доповнює інші технічні рішення. Згідно з аналізом, проведеним [6], 95% фішингових атак спричинені людським фактором.

Тим не менш, існуючого навчання виявлення фішингу недостатньо для боротьби зі складними атаками. Дані дослідження, суперечать статистиці ефективності та зручності навчання користувачів[7]. Крім того, деякі експерти з безпеки стверджують, що навчання користувачів неефективне, оскільки безпека не є головною метою для користувачів, а користувачі не мають мотивації вивчати фішинг. Тоді як інші дослідники підтверджують, що навчання користувачів може бути ефективний, якщо розроблено належним чином . Більше того, багато досліджень згадують навчання користувачів як ефективний спосіб захисту користувачів, коли вони користуються онлайн-сервісами. Щоб виявити та уникнути фішингових листів, автори в дослідженні запропонували комбінований підхід до навчання[8].

Запропоноване рішення використовує комбінацію інструментів і навчання людини, при цьому програма поінформованості про безпеку представляється користувачеві як перший крок. Другим кроком є Gвикористання інтелектуальної системи, яка виявляє атаки на рівні електронної пошти. Після цього електронні листи класифікуються експертною системою на основі нечіткої логіки. Основна критика цього методу полягає в тому, що дослідження вибирає лише обмежені характеристики електронних листів як відмінні ознаки. Більше того, більшість навчальних програм з фішингу зосереджені на тому, як розпізнавати та уникати фішингових листів і веб-сайтів, тоді як іншим загрозливим типам фішингу приділяється менше уваги, таким як голосовий фішинг і зловмисне програмне забезпечення або фішинг з рекламою. Було виявилено, що найбільш використовувані рішення для навчання людей не є корисними, якщо вони ігнорують сповіщення/попередження про підроблені веб-сайти. Навчання користувачів має включати три основні напрямки: перший – це підвищення обізнаності через проведення семінарів або онлайн-курсів для співробітників як в організації, так і для окремих осіб.

Другий – використання імітаційних фішингових атак щоб перевірити вразливість користувачів і дозволити їм оцінити власні знання про фішинг. Однак лише 38% світових організацій стверджують, що готові протистояти складній кібератаці. Звіт Wombat Security про стан Phish 2018 показав, що приблизно дві п’ятих американських компаній щомісяця використовують комп’ютерні онлайн-тренінги та імітовані фішингові атаки як інструменти навчання, тоді як лише 15% компаній у Великобританії роблять це [9].

Третій напрямок – навчання людей шляхом розробки ігор для навчання людей фішингу. Розробник гри повинен враховувати різні аспекти, перш ніж розробляти гру, такі як вік аудиторії та стать, оскільки сприйнятливість людей до фішингу різна.

Автори дослідження (Sheng et al., 2007) розробили гру для навчання користувачів, щоб вони могли визначати фішингові атаки під назвою Anti-Phishing Phil, яка розповідає фішингові веб-сторінки, а потім тестує користувачів на ефективність та ефективність гри [10]. Результати дослідження показали, що учасники гри покращили свою здатність розпізнавати фішинг на 61%, що вказує на те, що інтерактивні ігри можуть виявитися приємним способом навчання людей. Хоча навчання та тренування користувачів можуть бути дуже ефективними для пом’якшення загроз, фішинг стає все складнішим, і кіберзлочинці можуть обдурити навіть експертів з безпеки, створюючи переконливі фішингові електронні листи через соціальні мережі. Тому окремі користувачі та співробітники повинні мати принаймні базові знання щодо роботи з підозрілими електронними листами та повідомляти про це IT-персоналу та певним органам. На додачу, фішери постійно змінюють свої стратегії, що ускладнює організаціям, особливо малим/середнім підприємствам, витрати на навчання своїх співробітників. Оскільки мільйони людей щодня входять у свої акаунти в соціальних мережах, фішинг у соціальних мережах є улюбленим засобом фішерів для обману своїх жертв.

Наприклад, фішери користуються перевагами поширеності Facebook для створення креативних фішингових атак, використовуючи функцію входу у Facebook, яка дозволяє фішеру зламати всі облікові записи користувачів з однаковими обліковими даними VadeSecure[11]. Соціальні мережі вживають певних

контрзаходів, щоб зменшити підозрілу діяльність у соціальних мережах, наприклад двофакторну аутентифікацію для входу, яку вимагає Facebook, Corrata (2018). [12]. Однак контрзаходи для контролю Soshing і телефонних фішингових атак можуть включати:

• Встановіть антивірусне та антиспамове програмне забезпечення як першу дію та оновлюйте його, щоб виявити та запобігти будь-якому несанкціонованому доступу.

• Дізнайтеся про останню інформацію про фішинг, останні тенденції та заходи протидії.

• Ніколи не натискайте гіперпосилання, прикріплені до підозрілих електронних листів, дописів, твітів, прямих повідомлень.

• Ніколи не довіряйте соцмережам, не роздавайте конфіденційну інформацію по телефону або в ненадійний обліковий запис. Не приймайте запити в друзі від незнайомих вам людей.

• Використовуйте унікальний пароль для кожного облікового запису.

Навчання та навчання користувачів є ефективним заходом проти фішингу, яке вже показало багатообіцяючі початкові результати. Основним недоліком цього рішення є те, що воно вимагає великих витрат. Більше того, це рішення вимагає базових знань з комп’ютерної безпеки від підготовлених користувачів[13]

.

1.7 Технічні рішення

Запропоновані технічні рішення для виявлення та блокування фішингових атак можна розділити на два основних підходи: рішення без вмісту та рішення на основі контенту. Методи, не засновані на вмісті, включають чорні та білі списки, які класифікують підроблені електронні листи або веб-сторінки на основі інформації, яка не є частиною електронної пошти чи веб-сторінки, наприклад, URL-адреси та функції доменного імені [14]. Зупиняючи фішингові сайти за допомогою підходів до чорного та білого списків, у яких зберігається список відомих URL-адрес і сайтів, веб-сайт, що перевіряється з такими списками, щоб бути класифікованим як

фішинговий або законний сайт. Недоліком цього підходу є те, що він не ідентифікує всі фішингові веб-сайти. Оскільки після видалення фішингового сайту фішер може легко зареєструвати новий домен [15].

Методи на основі вмісту класифікують сторінку або електронну пошту на основі інформації в її вмісті, наприклад, текстів, зображень, а також кодів HTML, java-скриптів і каскадних таблиць стилів (CSS). Рішення на основі вмісту включають машинне навчання, евристики, візуальну схожість та методи обробки зображень [16].

І, нарешті, багатоаспектні методи, які використовують комбінацію попередніх підходів для виявлення та запобігання фішингових атак. Для фільтрації електронної пошти зазвичай використовуються методи ML, наприклад, у 2007 році перший фішинговий фільтр електронної пошти був розроблений авторами. Ця техніка використовує набір функцій, таких як URL-адреси, які використовують різні доменні імена. Методи фільтрації спаму і статистичні класифікатори також використовуються для ідентифікації фішингової електронної пошти. Технології аутентифікації та верифікації також використовуються у фільтрації спаму як альтернатива евристичним методам. Наприклад, Sender Policy Framework (SPF) перевіряє, чи дійсний відправник, коли приймає пошту від віддаленого поштового сервера або клієнта електронної пошти.

Технічні рішення для захисту від фішингу доступні на різних рівнях ланцюга доставки, таких як поштові сервери та клієнти, постачальники послуг Інтернету, та інструменти веб-браузера. Виходячи з запропонованої анатомії фішингових атак у Proposed Phishing Anatomy , автори класифікують технічні рішення за такими підходами:

1. Прийоми виявлення атаки після її запуску. Наприклад, скануючи Інтернет, щоб знайти підроблені веб-сайти. Підходи виявлення фішингу на основі вмісту широко розповсюджені в Інтернеті. Функції елементів веб-сайту, таких як зображення, URL-адреса та текстовий вміст, аналізуються за допомогою підходів на основі правил та машинного навчання, які перевіряють наявність спеціальних символів (@), IP-адрес замість імені домену, префіксу/суфіксу, HTTPS у частині

домену та інші функції [17]. Fuzzy Logic (FL) також використовувалася як модель захисту від фішингу, щоб допомогти класифікувати веб-сайти на легітимні або «фішівські», оскільки ця модель має справу з інтервалами, а не з конкретними числовими значеннями .

2. Прийоми запобігання проникненню атаки до системи користувача. Запобігання фішингу – це важливий крок для захисту від фішингу, блокуючий доступ користувача до атаки та попередження її. Під час фішингу електронної пошти програмні засоби захисту від спаму можуть блокувати підозрілі листи. Фішери зазвичай надсилають справжнє, схоже електронне повідомлення, яке обманює користувача, щоб відкрити вкладений файл або натиснути посилання. Деякі з цих листів проходять фільтр спаму, оскільки фішери використовують слова з помилкою. Тому все частіше використовуються методи, які виявляють підроблені електронні листи шляхом перевірки орфографії та виправлення граматики, щоб запобігти потраплянню електронної пошти до поштової скриньки користувача. Автори дослідження [18] запропонували алгоритм класифікації алгоритм класифікації на основі алгоритму Random Forest після вивчення фішингу електронної пошти з використанням алгоритму генератора дерева рішень “C4.5”. Розроблений метод називається «Ідентифікація фішингу шляхом вивчення особливостей отриманої електронної пошти» (PILFER), який може класифікувати фішингові електронні листи залежно від різних функцій, таких як IP-адреси, кількість посилань у HTML-частинах електронного листа, кількість доменів, кількість точок, невідповідні URL-адреси та наявність JavaScript. Розроблений метод показав високу точність у виявленні фішингових листів [18]

3. Коригувальні методи, які можуть знищити зламаний веб-сайт, вимагаючи від постачальника Інтернет-послуг (ISP) веб-сайту закрити підроблений веб-сайт, щоб запобігти тому, щоб більше користувачів стали жертвами фішингу. Провайдери несуть відповідальність за видалення підроблених веб-сайтів. Видалення зламаних і незаконних веб-сайтів є складним процесом; До цього процесу залучено багато суб'єктів із приватних компаній, органів саморегулювання, державних установ, волонтерських організацій, правоохоронних органів та постачальників послуг.

Зазвичай незаконні веб-сайти видаляються наказами про видалення, які видаються судами або в деяких юрисдикціях правоохоронними органами. З іншого боку, вони можуть бути добровільно видалені самими провайдерами в результаті виданих повідомлень про видалення. За даними PHISHLABS у звіті, видалення фішингових сайтів корисно, але це не зовсім ефективно, оскільки ці сайти можуть існувати кілька днів, крадучи облікові дані клієнтів, перш ніж виявити атаку [19].

4. Інструменти попередження або індикатори безпеки, вбудовані у веб-браузер для інформування користувача після виявлення атаки. Наприклад, eBay Toolbar і Account Guard захищають паролі клієнтів eBay і PayPal відповідно, попереджаючи користувачів про справжність сайтів, на яких користувачі намагаються ввести пароль. Численні рішення для боротьби з фішингом покладаються в основному на на попередження, які відображаються на панелі інструментів безпеки. Крім того, деякі панелі інструментів блокують підозрілі сайти, щоб попередити про загрозу, наприклад McAfee і Netscape. Дослідження, представлене в містить результати тестування оцінки продуктивності восьми антифішингових рішень, включаючи Microsoft Internet Explorer 7, EarthLink, eBay, McAfee, GeoTrust, Google за допомогою Firefox, Netscape і Netcraft. Ці інструменти є засобами попередження та блокування, які дозволяють використовувати законні сайти, одночасно блокуючи та попереджаючи про відомі фішингові сайти. Дослідження також показало, що Internet Explorer і Netcraft Toolbar показали найефективніші результати, ніж інші засоби захисту від фішингу. Однак панелі інструментів безпеки все ще не в змозі нівелювати людський фактор, через який люди стають жертвами фішингу, незважаючи на те, що ці панелі інструментів покращують безпеку в Інтернеті загалом [20]

5. Аутентифікація та авторизація методи, які забезпечують захист від фішингу шляхом перевірки особи законної особи. Це перешкоджає фішерам отримати доступ до захищеного ресурсу та здійснити атаку. Існує три типи аутентифікації; однофакторна автентифікація вимагає лише імені користувача та пароля. Другий тип - це двофакторна аутентифікація, яка вимагає додаткової інформації на додаток до імені користувача та пароля, наприклад, одноразового пароля, який надсилається

на електронну адресу або телефон користувача. Третій тип – це багатофакторна аутентифікація, що використовує більше ніж одну форму ідентифікації (тобто комбінацію чогось, що ви знаєте, чогось ви є, і того, що ви маєте). Деякі широко використовувані методи в процесі авторизації – це авторизація API та OAuth 2.0, які дозволяють раніше створеному API отримати доступ до системи [21].

Висновок до першого розділу

На сьогоднішній день фішингові атаки залишаються однією з основних загроз для окремих осіб та організацій. Як зазначено вище, це в основному зумовлено участю людини в циклі фішингу. Часто фішери використовують людські вразливості на додаток до технологічних умов.Прогресивне зростання фішингових атак показує, що попередні методи не забезпечують необхідного захисту від більшості існуючих фішингових атак. Можна помітити, що засоби, що використовуються для фішингових атак, змінилися від традиційних електронних листів до фішингу в соціальних мережах. Існує явне відставання між складними фішинговими атаками та існуючими контрзаходами. Контрзаходи, що з’являються, мають бути багатовимірними, щоб протидіяти як людським, так і технічним елементам атаки, та одними из найсучасніших заходів є нейромережі.

РОЗДІЛ 2
АНАЛІЗ НЕЙРОМЕРЕЖЕВИХ АРХІТЕКТУР, ВИКОРИСТОВУВАНИХ ДЛЯ РОЗПІЗНАВАННЯ ФІШИНГОВИХ ПОСИЛАНЬ

2.1 Стратегії боротьби з фішингом, засновані на механізмах DL

Протягом останніх кількох років фішинг веб-сайтів став однією з найпоширеніших загроз у кіберпросторі. Тому були розроблені різні антифішингові рішення для раннього виявлення загроз фішингу, щоб мінімізувати ризики безпеки та захистити кінцевих користувачів. Стратегії безпеки, засновані на механізмах DL (глибоких нейронних мережах), стають все більш популярними для боротьби з фішинговими атаками, що розвиваються [22-24]. Існує безліч типів методів DL, призначених для вирішення конкретної проблеми або задоволення певних вимог системи; кожен має свої переваги та недоліки [25,26].

Зловмисники постійно змінюють свою тактику атаки, щоб використати вразливості системи та неінформованість користувачів. Вибір невідповідного алгоритму протидії може призвести до непередбачуваних результатів та втрати зусиль, що в кінцевому підсумку впливає на точність та ефективність DL моделі Deep Cybersecurity [27]. Тому вибір ефективної моделі виявлення фішингу з високою точністю продуктивності та низькою обчислювальною потужністю є складним завданням. Процес тонкого налаштування архітектур DL – це ще одне питання, яке необхідно розглянути. У цьому розділі буде виконаний загальний огляд чотирьох різних алгоритмів DL, включаючи глибоку нейронну мережа (DNN), згорткову нейронну мережу (CNN), рекурентну мережу довготривалої короткочасної пам'яті (LSTM) і мережу з керованими рекурентними модулями (GRU), а також проведений аналіз існуючих досліджень цих чотирьох типів архітектур DL в області виявлення фішингу. Для кожної з архітектур буде проаналізована структура нейронної мережі, застосована оптимізація параметрів і

показники продуктивності, щоб досягти всебічного розуміння дизайну, реалізації та оцінки кожної моделі DL.

2.2 Глибокі нейронні мережі прямого поширення

Глибока нейронна мережа (DNN) є одним з найпоширеніших типів алгоритмів DL, які широко використовуються в області кібербезпеки. DNN добре відома серед архітектур DL завдяки своєму успіху в широкому діапазоні додатків [26], його здатності виражати складні функції з меншою кількістю параметрів і здатності полегшувати вилучення ознак і репрезентаційного навчання [27]. Однак DNN вимагає значної кількості розмічених даних для навчання. Крім того, ця архітектура все ще страждає від недостатньої техніки вибору параметрів [28], а процес навчання займає багато часу [26]. Незважаючи на вказані недоліки, існує кілька дослідницьких робіт для вивчення ефективності застосування DNN для виявлення фішингових веб-сторінок.

В роботах [29-31] глибокі нейронні мережі прямого поширення використовувалися як єдина(моно) архітектура для навчання системи класифікації для виявлення фішингових веб-сайтів.

Замість того, щоб використовувати DNN як окремий класифікатор, автори в [32,33] поєднали його з іншими алгоритмами DL, щоб побудувати модель для розрізнення шкідливих і доброякісних URL-адрес. Було помічено, що серед цих моделей на основі DNN налаштування параметрів відіграють істотну роль у визначенні точності продуктивності системи. Тим не менш, в деяких дослідженнях [32,33] не згадується жоден із гіперпараметрів при розробці архітектури нейронної мережі, тоді як в інших роботах [30,31], вказано лише деякі з них без оптимізації параметрів.

Крім того, показники продуктивності є ще одним важливим фактором, який необхідно враховувати при аналізі та оцінці системи виявлення фішингу. Попередні дослідження показали, що для оцінки ефективності моделей DL у виявленні фішингових веб-сайтів було використано обмежену кількість показників.

Наприклад, у [30] використано лише дві метрики, а в [32] виміряно три. У дослідженнях [29,33] було використано більше показників, але лише деякі були використані для порівняння з іншими класифікаторами машинного навчання.

Структура DNN складається з вхідного шару, вихідного шару та одного або кількох прихованих шарів [29], як показано на рисунку 2.1.

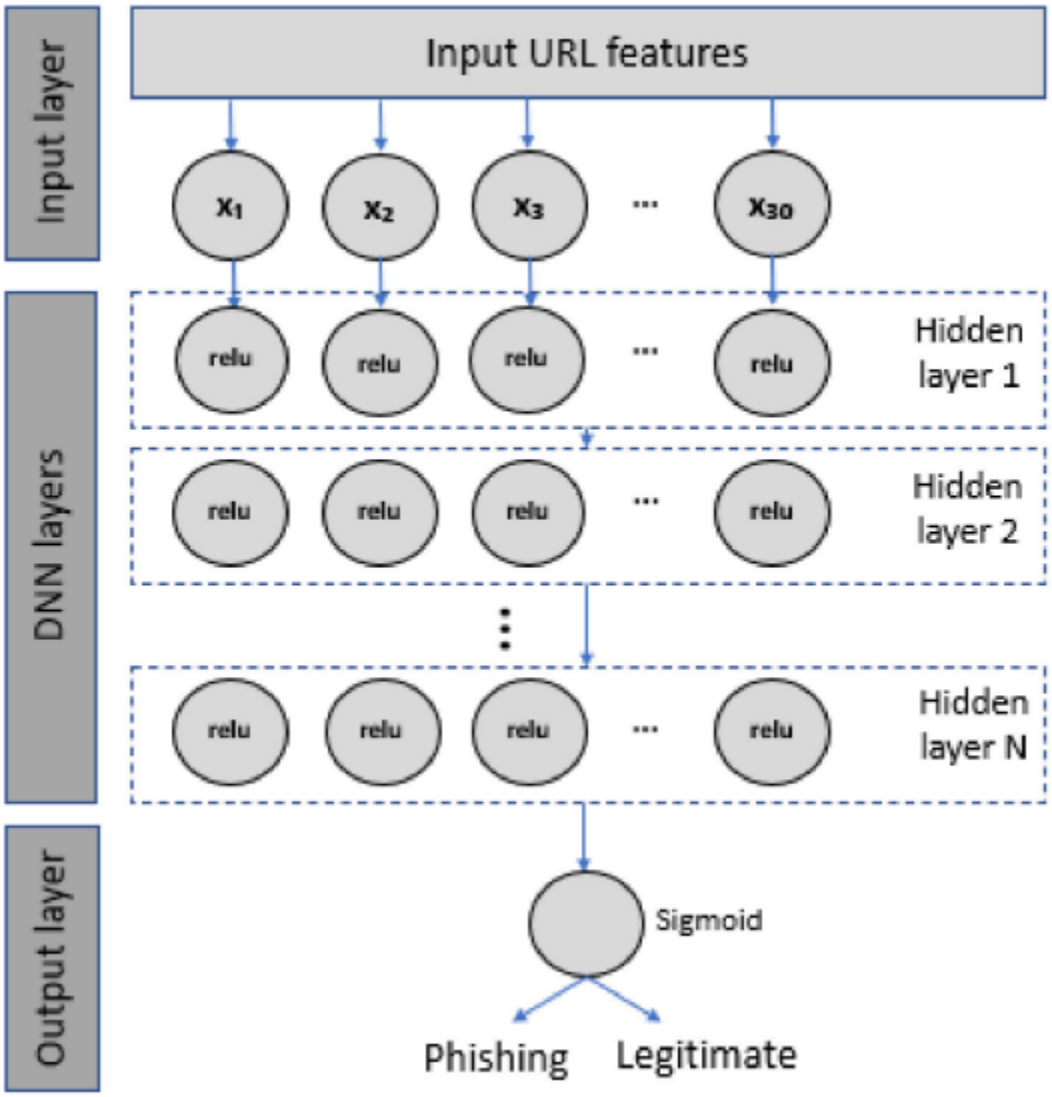

Рисунок 2.1 – Глибока нейронна мережа прямого поширення

Кожен вузол або нейрон в одному шарі з’єднаний з іншими вузлами наступного шару, щоб утворити щільний або повністю зв’язаний шар [30]. Кількість прихованих шарів і нейронів у кожному прихованому шарі може змінюватися. Функціями активації, які використовуються в прихованих шарах і вихідному шарі, є

ReLU і sigmoid відповідно. Дослідникам потрібно точно налаштувати ці параметри, щоб знайти оптимальні значення, які забезпечують найвищу точність виявлення.

2.3 Згорткові нейронні мережі

Згорткова нейронна мережа (CNN) є ще одним популярним типом техніки DL в області кібербезпеки. CNN добре підходить до багатовимірних даних і спеціалізується на обробці зображень і сигналів [34]. Крім того, CNN може ефективніше виділяти ознаки з необроблених даних і вирішувати складні завдання. Він також більш масштабований і вимагає менше часу на навчання [35]. Тим не менш, архітектура CNN потребує високої обчислювальної потужності та великого набору даних при роботі з даними зображення [26]. Незважаючи на те, що CNN досягла величезного успіху в області комп'ютерного зору, цю архітектуру також застосовують у сфері кібербезпеки.

CNN використовувався як єдиний класифікатор у численних дослідженнях, щоб розрізняти фішингові та легальні веб-сайти [36,37]. Його також можна використовувати в поєднанні з іншими методами DL для формування моделі ансамблю та покращення точності виявлення фішинг. Різниця між архітектурами CNN і DNN полягає у використанні згорткових шарів і ядер. Усвідомлюючи важливу роль цих елементів у визначенні точності роботи моделей виявлення фішингу, більшість дослідників приділяли більше уваги параметрам згорткових шарів, а не іншим, таким як швидкість навчання, параметри вилучення (dropout), епоха або розмір пакету. Хоча в [23] цієї проблеми було уникнуто, деталі оптимізації цих параметрів у роботі не наводилися. Так само автори [36,37] описали процес оптимізації, але лише за певними параметрами, наприклад, кількістю згорткових шарів, кількістю ядер і розміром ядра. Крім того, з точки зору показників продуктивності, було помічено, що параметри accuracy і precision, recall та оцінка F1 були найпоширенішими показниками [36,37]. Іншими показниками оцінки були час навчання, час виявлення, потреба в пам'яті графічного процесора тощо [36,37].

Архітектура моделі CNN, як правило, складається з трьох основних шарів: згорткового шару, шару об’єднання та повністю зв’язаного шару [37].

Згортковий шар використовується для вилучення ознак і складається з кількох згорткових ядер або фільтрів, які розділяють вхідні вектори на невеликі блоки. Потім генерується серія карт ознак шляхом виконання згорткових операцій над вхідними векторами з вибраними ядрами [23].

Шар об’єднання використовується для зменшення розмірів шляхом зменшення розмірності карт об’єктів. Рівень об’єднання виконує дві функції: прискорює роботу мережі та покращує продуктивність усієї згорткової мережі [24].

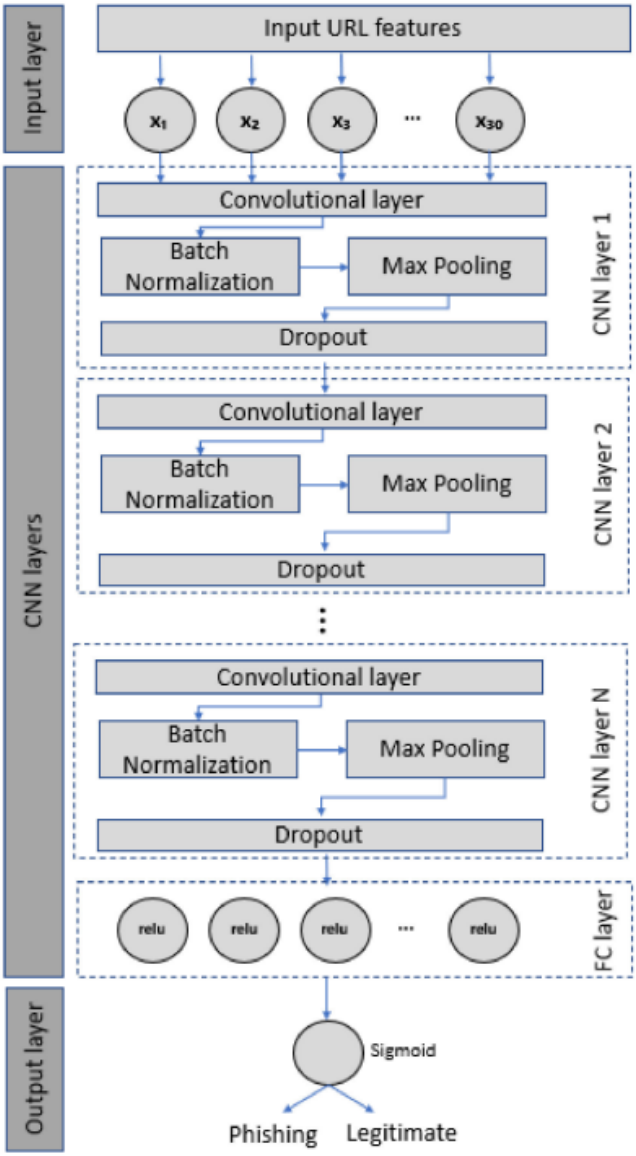

Рисунок 2.2 – Згорткова нейронна мережа

Шар повного зв’язку (FC) відповідає за цілі класифікації. Рівень FC (повнозв'язного шару) – це традиційна нейронна мережа, яка використовує виділені ознаки з попередніх шарів для виконання остаточного завдання класифікації [29].

Щоб уникнути проблем із перенавчанням, між рівнями CNN використовуються стратегії пакетної нормалізації та вилучення. ReLU використовується як функція активації на згортковому та FC рівнях, тоді як сигмоподібний реалізований у вихідному шарі (рис 2.2).

2.4 Рекурентні мережі довготривалої короткочасної пам'яті

Мережі довготривалої короткочасної пам'яті (LSTM) – це тип рекурентної нейронної мережі (RNN), яка включає рекурсивні зв’язки між нейронами в кожному шарі [25]. LSTM підходить для послідовних або часових рядів даних, оскільки він може підтримувати безперервність інформації. LSTM є більш популярним, ніж оригінальний RNN, тому що зникаючий або вибухаючий градієнт і проблеми довгострокової залежності в традиційних RNN були подолані в LSTM [35].

Незважаючи на ці переваги, порівняно з іншими алгоритмами DL, мережі довготривалої короткочасної пам'яті вимагають значно тривалішого часу для навчання.Крім того, LSTM розглядає лише пряму інформацію і не розглядає зворотну інформацію. Однак цю проблему можна вирішити в двонаправленій LSTM [35].

LSTM – це варіант RNN, який має специфічну структуру комірки пам’яті. Комірка пам’яті типового блоку LSTM складається з трьох вентилів: вхідного шлюза, шлюза забуття та вихідного шлюза [23]. На відміну від нейронної мережі з прямим зв’язком, вихід нейрона в архітектурі LSTM в певний момент може стати вхідним для того самого нейрона. У моделі виявлення фішингу на основі LSTM може бути більше одного рівня LSTM та застосовується стратегія dropout використовується в наступному шарі, щоб запобігти проблемам з перенавчанням. LSTM шар і шар повного зв’язку використовують ReLU, тоді як вихідний шар використовує Sigmoid як функції активації (рис 2.3).

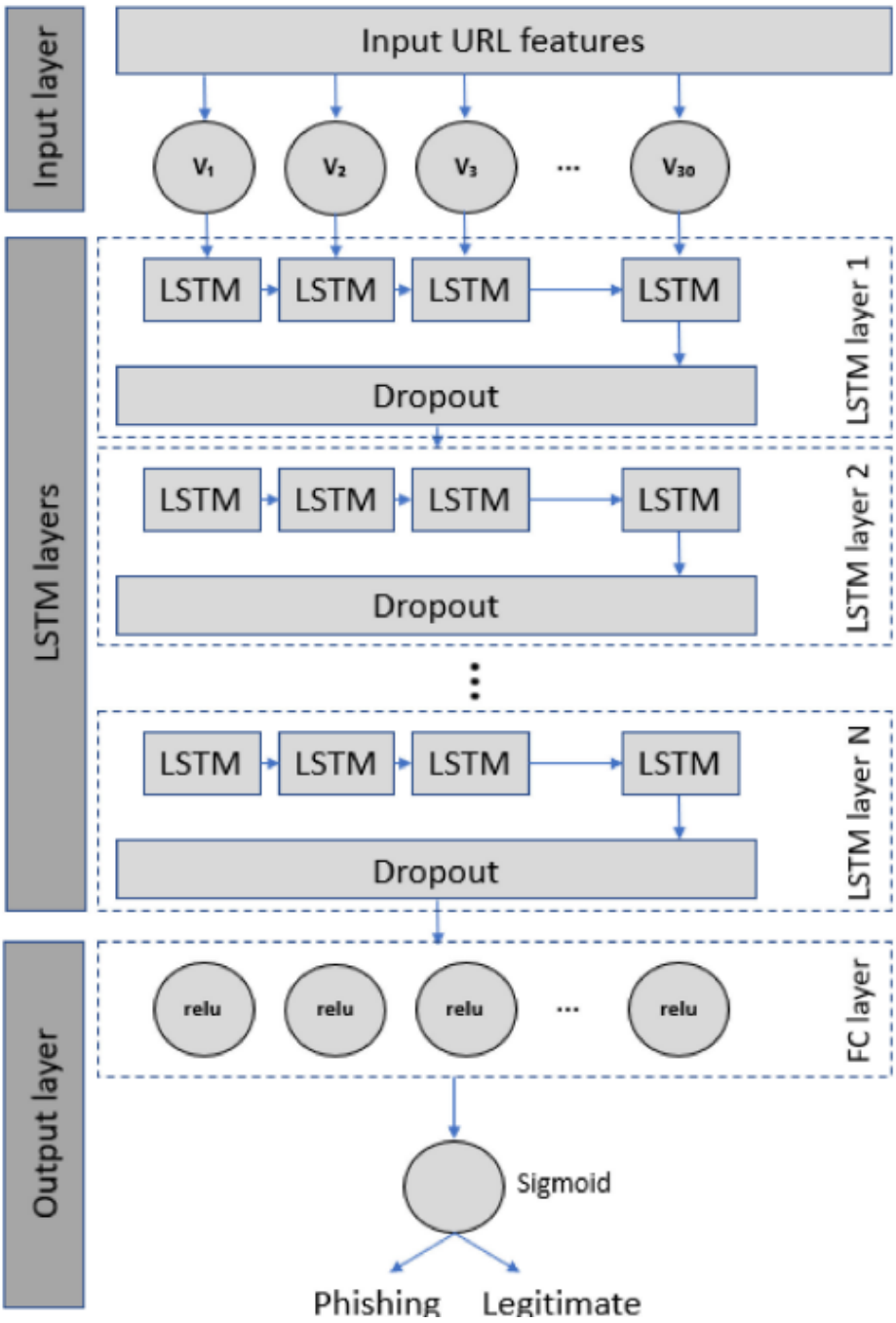

Рисунок 2.3 – Рекурентні мережі довготривалої короткочасної пам'яті

2.5 Мережа з керованими рекурентними модулями

Мережа з керованими рекурентними модулями (GRU) є ще одним варіантом рекурентних мереж і є полегшеною версією LSTM [38]. Під час роботи з невеликими наборами даних продуктивність GRU подібна до LSTM [40]. Існує обмежена кількість досліджень щодо використання GRU для виявлення фішингу.

GRU і Bidirectional GRU можуть використовуватися як єдиний класифікатор [38,39], або як заміна шару max-pooling в моделі CNN [38]. Подібно до LSTM, під час реалізації моделей виявлення фішингу на основі GRU дослідники, як правило, вказують лише архітектуру нейронної мережі, швидкість навчання та епоху, але не розмір пакету та частоту вилучення [38,39].

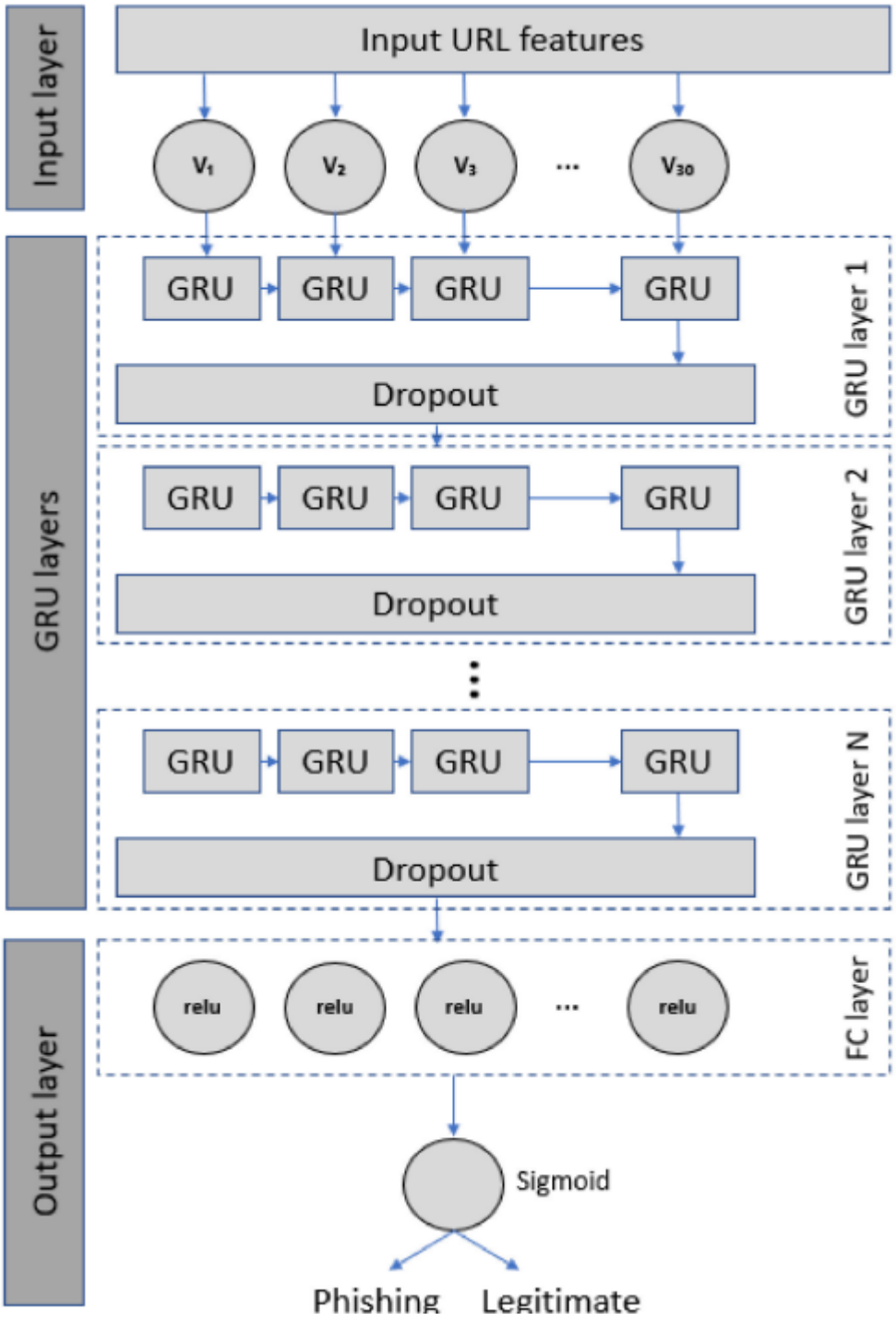

Рисунок 2.4 – Мережа з керованими рекурентними модулями

Крім того, жодна зі знайдених робіт про GRU не включала опису оптимізації параметрів у своїх експериментах. Що стосується показників продуктивності, то всі дослідження [38,39] використовували accuracy, precision, recall та F1-показник для оцінки ефективності алгоритму DL. Додаткові показники включали вимоги до пам’яті графічного процесора та розмір набору параметрів [38].

Подібно до LSTM, GRU сконструйовано з вентилями та осередками пам'яті. Проте він простіший у реалізації та обчисленні [39]. Замість структури з трьома вентилями, як-от в LSTM, у комірці пам’яті GRU є лише два вентилі: вхідний та вентиль забування (input and forget gates). Загальна архітектура моделей виявлення фішингу на основі GRU подібна до LSTM.(рис 2.4)

2.6 Змішані архітектури

На сьогоднішній день існує багато різних алгоритмів DL які численні дослідники реалізували для виявлення фішингових веб-сайтів. Однак вибір правильного підходу, який найкраще підходить для конкретної програми або набору даних, є складним завданням. Для оцінки ефективності моделі виявлення фішингу на основі DL були виміряні різні показники продуктивності. Отримані в результаті експериментів показники вказують, що серед чотирьох методів DL (DNN, CNN, LSTM, GRU) не було жодного алгоритму, який би давав найкращі значення за всіма показниками продуктивності. Необхідно вибрати той, що найкраще підходить для їх конкретних застосувань або відповідно до конкретних вимог. Моделі також можуть поєднувати різні алгоритми DL в гібридну або ансамблеву модель, щоб об’єднати свої переваги та усунути їхні недоліки.

Висновок до другого розділу

З огляду на проведені дослідження можна зробити висновок, що перспективним напрямом в задачі підвищення ефективності виявлення фішингу є

використання гібридних моделей, тобто таких, що поєднують в собі шари різної природи.

Перспективними комбінаціями для дослідження можна вважати такі:

1) CNN + DNN
2) DNN +LSTM
3) CNN +RNN

У рамках наступного розділу будуть проведені обчислювальні експерименти з вказаними гібридними моделями DL, які є відносно новими в області виявлення фішингу.

РОЗДІЛ 3
РОЗРОБКА ПРОГРАМНОГО МОДУЛЯ

3.1 Архітектура програмного застосунку

Метою даної роботи було розробити програмний застосунок для усунення загрозу фішингу. Архітектурно розроблений програмний продукт буде реалізований як браузерне розширення (див рис 3.1).

Головне призначення розширень зрозуміло вже із самої назви – розширити функціонал браузера. Ця програма буде збирати дані що є на сторінці і перевіряти їх на загрозу фішингу. Переваги розширення:

- Швидкість доступу, зручність та зрозумілість використання;
- Кросплатформеність – здатність працювати на будь-якій платформі, де є браузер;

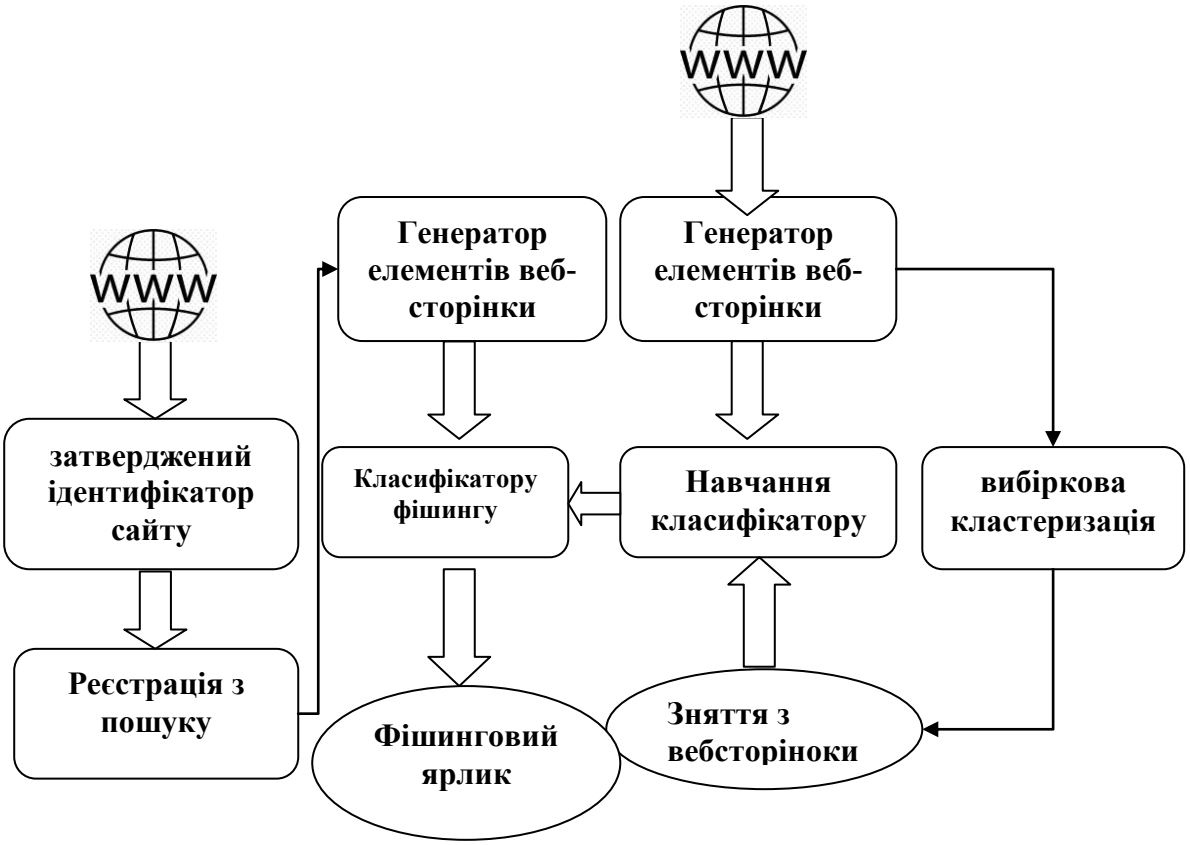

Рисунок 3.1 – Схема роботи програмного застосунку.

Застосунок працює в двох режимах, а саме: режим ідентифікації фішингових посилань, та режим донавчання мережі.

В режимі ідентифікації при появі посилань на сторінці застосунок зчитує їх всі і ті, що здаються підозрілими будуть викреслиними з коду сторінки.

В режимі донавчання застосунок оновлює старий датасет посилань, донавчає мережу, після чого до клієнтів надсилається нова версія.

3.2 Програмне середовище

Задача розпізнавання фішингу вимагає реалізації процесів підготовки даних і їх обробки методами машинного навчання. При виборі технологій для імплементації усього математичного апарату, закладеного у систему, слід врахувати уже готові допоміжні компоненти для спрощення рішення описаних у попередньому розділі методів. До таких методів відносяться методи аналізу даних, методи обробки файлів і методи машинного навчання. Наявність програмних бібліотек для вирішення цих типів задач була головним критерієм вибору мови програмування Python як головної програмної платформи, на основі якої слід будувати систему розпізнавання респіраторних захворювань, адже вона містить потужний апарат наукових інструментів. Відповідно до цього даний підрозділ містить огляд мови Python та тих її бібліотек, які містять програмні компоненти, корисні для імплементації розроблюваної системи.

Python – об'єктно-орієнтована мова програмування, яку можна порівняти з такими мовами програмування як Perl, Scheme.Python є кросплатформною мовою, яка працює на будь-якій системі, включаючи Mac OS X, Windows, Linux і Unix, а неофіційні збірки доступні також для Android і iOS.

З точки зору програмування, до особливостей мови можна також віднести автоматичне керування пам'яттю, підтримку базових типів даних (чисел, рядків, списків, словників), динамічну типізацію даних, можливість об'єктно-орієнтованого програмування, згруповування модулей у бібліотеки. Слід зазначити, що мова Python розроблена під ліцензією Open Source з відкритим вихідним кодом, що

робить її вільною як для застосування, так і для розповсюдження, навіть у комерційних цілях.

Для системи, що проектується у даній роботі, ключовими є бібліотеки саме екосистеми Sciру для наукових досліджень. Серед основних її пакетів варто згадати:

● numpy- розширення мови Python, яке додає підтримку великих багатовимірних масивів і матриць.[40]

● Keras - це API для нейронних мереж високого рівня, що працює на основі такого програмного інструментарію для створення глибинних мереж[41]

● Tqdm - розширення мови Python, яке використовує інтелектуальні алгоритми для прогнозування відновлення часу та виробництва непередбачуваних ітерацій[42]

● h5py - розширення мови Python, яке дозволяє зберігати величезну кількість числових даних та легко маніпулювати цими даними з NumPy.[43]

● matplotlib - комплексна бібліотека для створення 2D-графіки[44]

● Scikit Learn - це безкоштовна бібліотека машинного навчання. Він містить різні алгоритми класифікації, регресії та кластеризації, включаючи підтримуючі векторні машини, випадкові ліси, збільшення градієнта, k-засоби та DBSCAN, і призначений для взаємодії з числовими та науковими бібліотеками Python NumPy та SciPy.[45]

● tensorflow - відкрита програмна бібліотека для машинного навчання[46]

● pandas - програмна бібліотека мовою Python для обробки і аналізу даних[47].

3.3 Навчальний набір даних

Для проведенння експериментів був використовували контрольний набір даних із [48]. У таблиці 2.1 представлено короткий опис атрибутів. Набір даних складається з 10000 екземплярів, отриманих з 5000 фішингових і 5000 законних веб-сайтів. Атрибути конкретних показників у наборі даних наведені в таблиці нижче: всього 30 атрибутів (див. табл. 3.1).

Таблиця 3.1

Атрибути датасету

Номер атрибуту	**Атрибут**	**Можливі значення**
1	havingIPAddress	-1,1
2	URLLength	-1,0,1
3	ShorteningService	-1,1
4	havingAtSymbol	-1,1
5	doubleslashredirecting	-1,1
6	PrefixSuffix	-1,1
7	havingSubDomain	-1,0,1
8	SSLfinalState	-1,0,1
9	Domainregistrationlength	-1,1
10	Favicon	-1,1
11	Port	-1,1
12	HTTPStoken	-1,1
13	RequestURL	-1,1
14	URLofAnchor	-1,0,1
15	Linksintags	-1,0,1
16	SFH	-1,0,1
17	Submittingtoemai	-1,1

Продовження табл. 3.1

18	AbnormalURL	-1,1
19	Redirectpage	0,1
20	onMouseOver	-1,1
21	RightClick	-1,1
22	Using pop-upwidnow	-1,1
23	Iframe	-1,1
24	Ageofdomain	-1,1
25	DNSRecord	-1,1
26	Webtraffic	-1,0,1
27	PageRank	-1,1
28	GoogleIndex	-1,1
29	Linkspointingtopage	-1,0,1
30	Statisticalreport	-1,1
Результат		-1,1

Загальний опис атрибутів:

- havingIPAddress – перевірка на наявність у лінку IP адреси;
- URLLength – перевірка на те скільки кількість знаків у лінку;
- ShorteningService – перевірка на те чи відображається лінк у скороченому форматі;
- havingAtSymbol – перевірка на те чи є у лінку знак «@»;
- doubleslashredirecting - перевірка на те чи є у лінку знак «\\»;

- PrefixSuffix – перевірка на те що до лінка не прив’язано префікс або суфікс;
- havingSubDomain- перевірка на те що до лінку не прив’язаний субдомен;
- SSLfinalState – перевірка SSL сертифікату;
- Domainregistrationlength – перевірка життєвого циклу домену;
- Favicon – перевірка на те що до лінку прив’язано його особистий Favicon;
- Port – перевірка на те що до лінку не прив’язано порти і їх статус;
- HTTPStoken – перевірка HTTPS-сертифікату;
- RequestURL – перевірка на те що до лінку не прив’язано автоматично скачуваних даних;
- URLofAnchor – перевірка що не прив’язано «якорних» лінків;
- Linksintags – перевірка на те що до лінка не прив’язано SQL ін’єкцій;

- SFH(server form handler) – перевірка на те що до лінка не прив’язано SFH ін’єкцій;
- Submittingtoemai – перевірка на прив’язку до пошти;
- AbnormalURL – перевірка на підробний домен;
- Redirectpage – перевірка на реадресацію на іншу сторінку;
- onMouseOver – перевірка на прихований лінк;
- RightClick – перевірка на те чи відображається лінк як елемент «<a>»;
- Using pop-upwidnow – перевірка на спливаючі вікна;
- Iframe – перевірка на наявність елементу Iframe;
- Ageofdomain – перевірка на довжину життєвого циклу;
- DNSRecord – перевірка на реадресацію через додаткові DNS сервери;
- Webtraffic – перевірка об’єму трафіку;
- PageRank – перевірка рейтингу лінку у Чорно/Білих списках;
- GoogleIndex – перевірка рейтингу лінку у Google;
- Linkspointingtopage – перевірка на наявність «магнітних» лінків;
- Statisticalreport – отримання підтвердження безпеки із відкритих баз;

3.4 Метрики оцінювання результатів

В загальному випадку для оцінки якості класифікації використовуються такі показники, як точність (precision), відгук (recall) і F1-метрика (F1-measure) ,частка правильних відповідей (accuracy).[49]

Найбільш очевидною мірою якості в задачі класифікації є частка правильних відповідей (accuracy):

$$accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{3.1}$$

де [49]:

TP (True Positives) - кількість вірно класифікованих позитивних прикладів (так звані істинно позитивні випадки).

TN (True Negatives) - кількість вірно класифікованих негативних прикладів (істинно негативні випадки).

FN (False Negatives) - кількість позитивних прикладів, класифікованих як негативні (помилка I роду). Це так званий «помилковий пропуск», коли подія, що нас цікавить, помилково не виявляється (хибно негативні приклади).

FP (False Positives) - негативні приклади, класифіковані як позитивні (помилка II роду). Це «помилкове виявлення», тому що при відсутності події помилково виноситься рішення про її присутність(хибно позитивні випадки).

Точність і повнота

Точність показує, яка частка об'єктів, визначених класифікатором як позитивні, дійсно є позитивними. Повнота показує, яка частина позитивних об'єктів була виділена класифікатором.

$$precision = \frac{tp}{tp + fp} \tag{3.2}$$

$$recall = \frac{tp}{tp + fn} \tag{3.3}$$

Точність можна інтуїтивно зрозуміти як здатність класифікатора не прогнозувати позитивну мітку для по-справжньому негативного зразка, а відгук – як здатність класифікатора прогнозовувати позитивну мітку для всіх по-справжньому позитивних зразків.

F1-метрика є середнім гармонійним точності і відгуку.

$$F_1 = 2\frac{precision \ * \ recall}{precision + \ recall} \tag{3.4}$$

Середнє гармонійне має важливу властивість - воно близьке до нуля, якщо хоча б один з аргументів близький до нуля. Саме тому воно є кращим, ніж середнє арифметичне (якщо алгоритм буде відносити всі об'єкти до позитивного класу, то він буде мати recall = 1 і precision «1, а їх середнє арифметичне буде більше 1/2, що неприпустимо).

Після підрахунку усіх двійкових метрик результат усереднюються декількома способами:

макро–усереднення – обчислення середнього значення двійкових метрик з однаковою вагою кожного класу, що дозволяє підкреслити ефективність класифікації для класів з меншою репрезентативністю (середнє за класами);

зважене середнє – обчислення середнього значення двійкових метрик з різною вагою для кожного класу для збалансування репрезентативності зразків кожного з класів (середнє за класами). Оскільки на етапі попередньої обробки планується здійснити аугментацію даних для досягнення однакової репрезентативності зразків кожного класу, у даній роботі порівнюватимуться результати мікро-усереднення різних методів. Тоді метрики точності, відгуку і F-метрики будуть еквівалентними і дорівнюватимуть точності класифікації (accuracy).

Матриця невідповідностей – це спосіб розбити результати на чотири категорії в залежності від комбінації істинної відповіді і відповіді алгоритму

	Справжня мітка класу	
Прогнозована мітка класу	правильний результат(TP)	неправильний результат(FP)
	неправильна відсутність результату(FN)	правильна відсутність результату(TN)

Рисунок 3.3 – Матриця невідповідностей бінарної класифікації

Для багатокласової класифікації матриця невідповідностей будується за тим же принципом:

На рисунку 3.4 наведено матрицю невідповідностей для випадку багатокласової класифікації.

Прогнозована мітка класу	Клас 1 $(\mathbf{C_1})$	Клас 2 $(\mathbf{C_2})$	Клас 3 $(\mathbf{C_3})$
1 $(\mathbf{P_1})$	$\mathbf{T}_1$	$\mathbf{F}_{12}$	$\mathbf{F}_{13}$
2 $(\mathbf{P_2})$	$\mathbf{F}_{21}$	$\mathbf{T}_2$	$\mathbf{F}_{23}$
3 $(\mathbf{P_3})$	$\mathbf{F}_{31}$	$\mathbf{F}_{32}$	$\mathbf{T}_3$

Рисунок 3.4 – Матриця невідповідностей багатокласової класифікації

Розрахункові формули компонентів матриці невідповідностей для багато класової класифікації:

$$TP_i = T_i \tag{3.5}$$

$$FP_i = \sum_{c \in Classes} F_{i,c} \tag{3.6}$$

$$FN_i = \sum_{c \in Classes} F_{c,i} \tag{3.7}$$

$$TN_i = All - TP_i - FP_i - FN_i \quad (3.8)$$

ROC–аналіз дозволяє провести оцінку якості моделі класифікатора, порівняти прогностичну силу декількох моделей, визначити оптимальну точку відсікання для віднесення об'єктів до того чи іншого класу. При цьому передбачається, що у класифікатора є додаткові параметри, що дозволяють вже після проведеного навчання варіювати співвідношення помилок першого й другого роду. В основі ROC-аналізу лежить побудова графіків - ROC-кривих (Receiver Operator Characteristic).

ROC-крива показує залежність кількості вірно класифікованих позитивних прикладів від кількості невірно класифікованих негативних прикладів. У термінології ROC-аналізу перші називаються істинно позитивною, другі - хибно негативною множиною (див. рис 3.3).

У класифікатора є деякий параметр, який часто називають порогом, або точкою відсікання (cutoff value).

Модель із високою чутливістю часто дає істинний результат при наявності позитивного результату (виявляє позитивні приклади). Навпаки, модель із високою специфічністю частіше дає істинний результат при наявності негативного результату (виявляє негативні приклади).

При аналізі використовуються значення з таблиці невідповідностей , але найчастіше оперують не абсолютними показниками, а відносними - частками (rates) вираженими у відсотках:

Частка істинно позитивних прикладів (True Positives Rate):

$$TPR = \frac{TP}{TP + FN} \quad (3.9)$$

Частка хибно позитивних прикладів (False Positives Rate):

$$FPR = \frac{FP}{FP + TN} \quad (3.10)$$

Чутливість (Sensitivity) - це і є частка істинно позитивних випадків:

$$Se = FPR = \frac{FP}{FP + TN} \tag{3.11}$$

Специфічність (Specificity) - частка істинно негативних випадків, які були правильно ідентифіковані моделлю:

$$\mathbf{Sp} = \frac{\mathbf{TN}}{\mathbf{TN} + \mathbf{FP}} \tag{3.12}$$

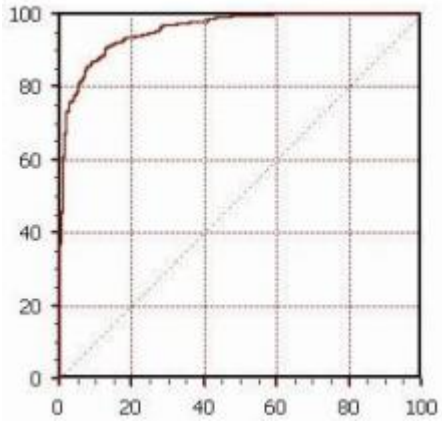

Рисунок 3.5 – ROC-крива

ROC-крива будується у такий спосіб: для кожного значення порога відсікання, що змінюється від 0 до 1 із кроком dx (наприклад, 0.01) розраховуються значення чутливості Se і специфічності Sp. Як альтернатива порогом може бути кожне наступне значення приклада у вибірці. Будується графік залежності: по осі Y відкладається чутливість Se, по осі X - 100%-Sp (сто відсотків мінус специфічність). [50]

3.5 План експерименту

У попередніх розділах були описані архітектури нейронних мереж, які можуть служити ядром системи. Для порівняння їх ефективності їх роботи була запропонована наступна схема проведення експерименту:

- 1 етап - порівняльний аналіз мереж, що побудовані на єдиній(моно) базовій моделі, а саме CNN, RNN.
- 2 етап - порівняльний аналіз роботи мережі-переможця 1-го етапу з гібридною мережею CNN-RNN.
- 3 етап - порівняльний аналіз роботи мережі-переможця 2-го етапу з гібридною мережею DNN-LSTM.

3.5.1 Згорткова мережа(CNN)

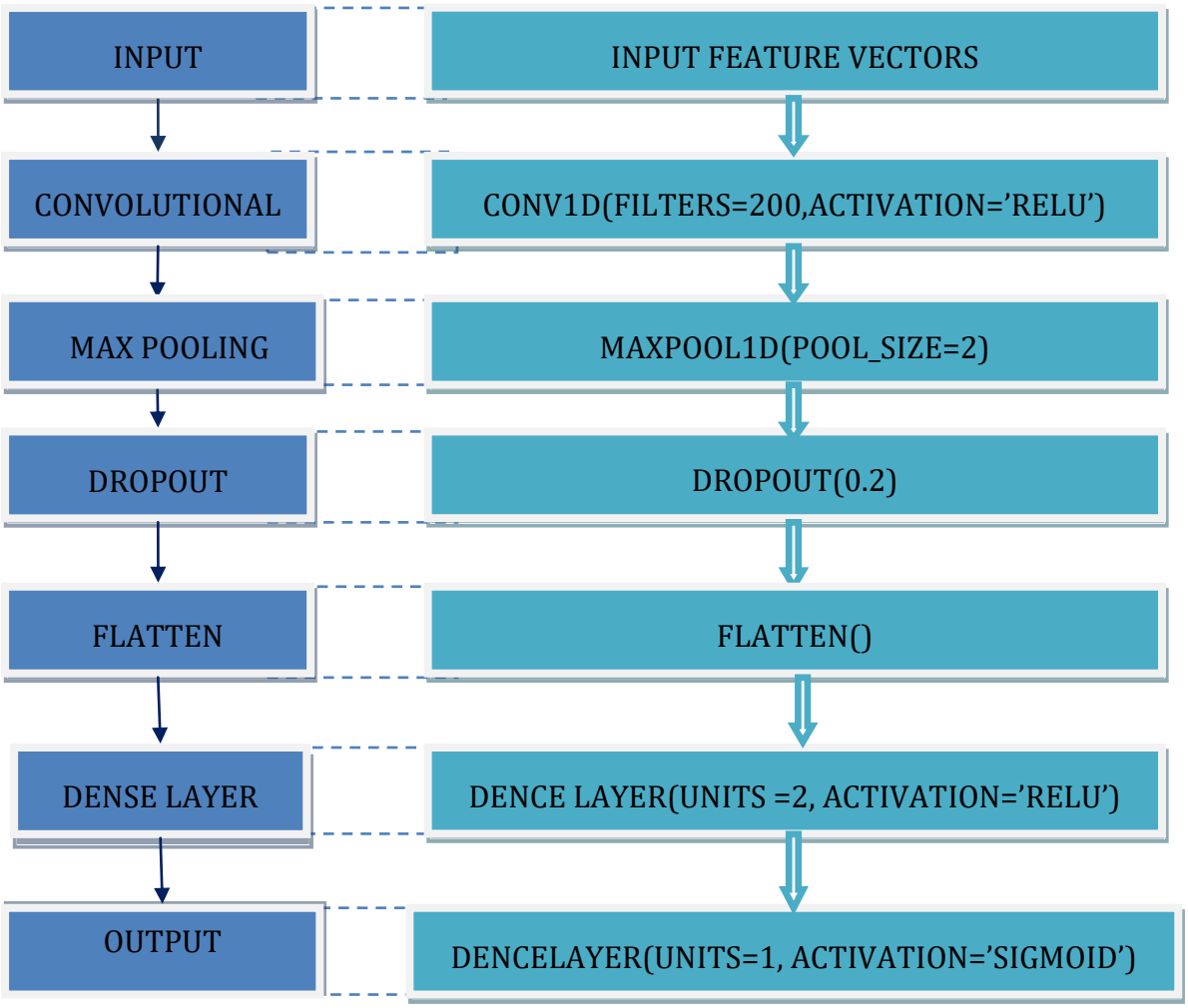

Рисунок 3.5 – CNN-архітектура

В експерименті була використана наступна архітектура згорткової мережі:

- згортковий шар(Conv1D) розміром(batch_size) 200, фільтрів(filters) 200;
- шар вибірки(MaxPooling1D) розміром 2;

- шар відбору 20% від існуючих нейронів(Dropout);
- шар перебудови вектору(Flatten);
- шар нейронів(Dense) розміром 2;

3.5.2 Рекурентна мережа(RNN)

В експерименті була використана наступна архітектура рекурентної мережі:

- рекурентний шар(SimpleRNN) розміром(batch_size) 200 ,на 128 нейронів;
- шар відбору 20% від існуючих нейронів(Dropout);
- шар нейронів(Dense) розміром 2;

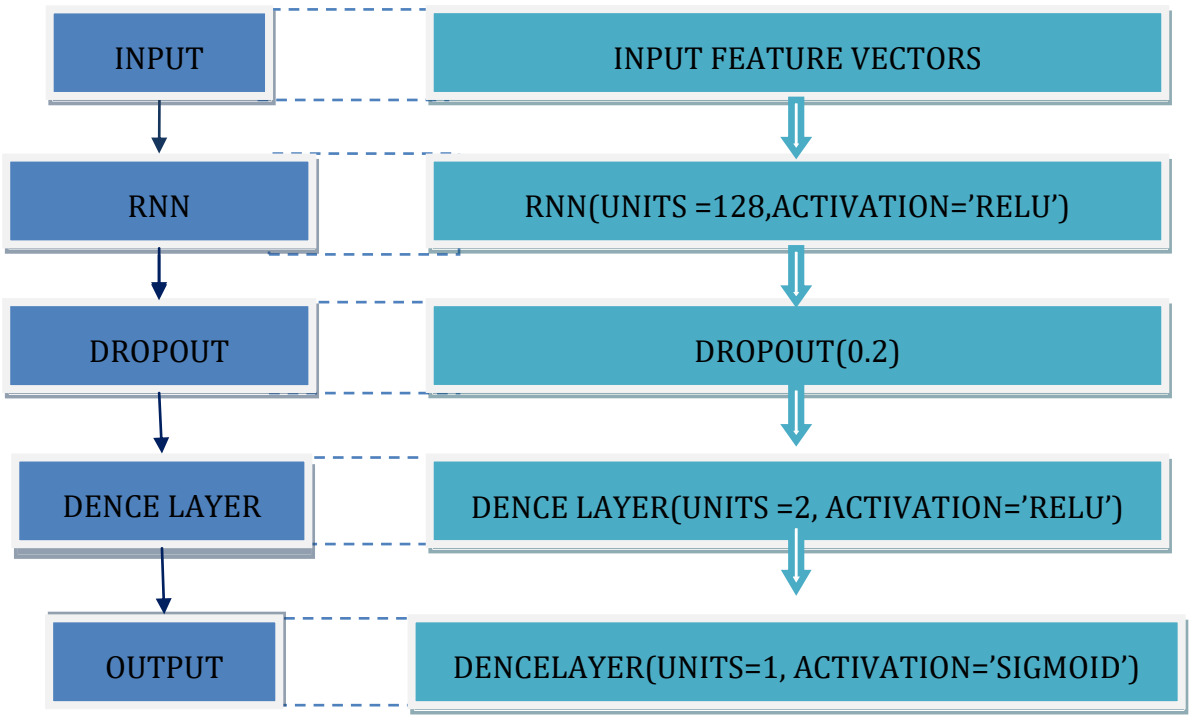

Рисунок 3.6 – RNN-архітектура

3.5.3 Гібридна мережа CNN-RNN

В експерименті була використана наступна архітектура гібридноії CNN-RNN мережі:

- згортковий шар(Conv1D) розміром(batch_size) 200 ,фільтрів(filters) 150;
- шар вибірки(MaxPooling1D) розміром 2;
- рекурентний шар(SimpleRNN) на 50 нейров;
- шар відбору 10% від існуючих нейронів(Dropout);
- шар нейронів(Dense) розміром 1 з ключем sigmoid;

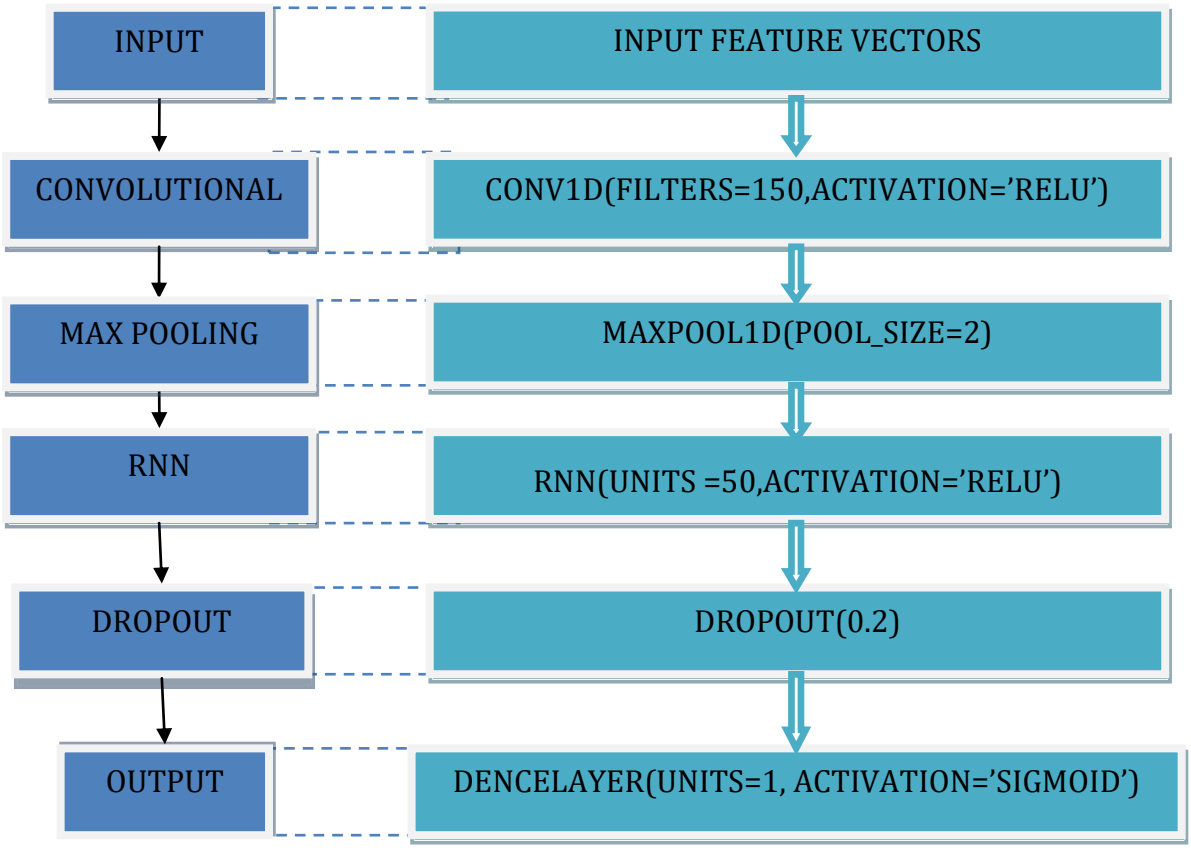

Рисунок 3.7 – CNN-RNN архітектура

3.5.4 Гібридна мережа DNN-LSTM

В експерименті була використана наступна архітектура гібридноїї мережі [51]:

Нейронна мережа складається з двох паралельних мереж , одна з них це DNN мережа, що має у собі 4 Dense layer на 48,64,32 та 16 вузлів відповідно. А інша LSTM мережа складається з двох LSTM на 32 вузла, Dropout та Dense layer на 16

вузлів. Обидві об`єднуються Dense layer на 2 вузли після чого Dense layer видає результат усієї системи(див рис 3.8).

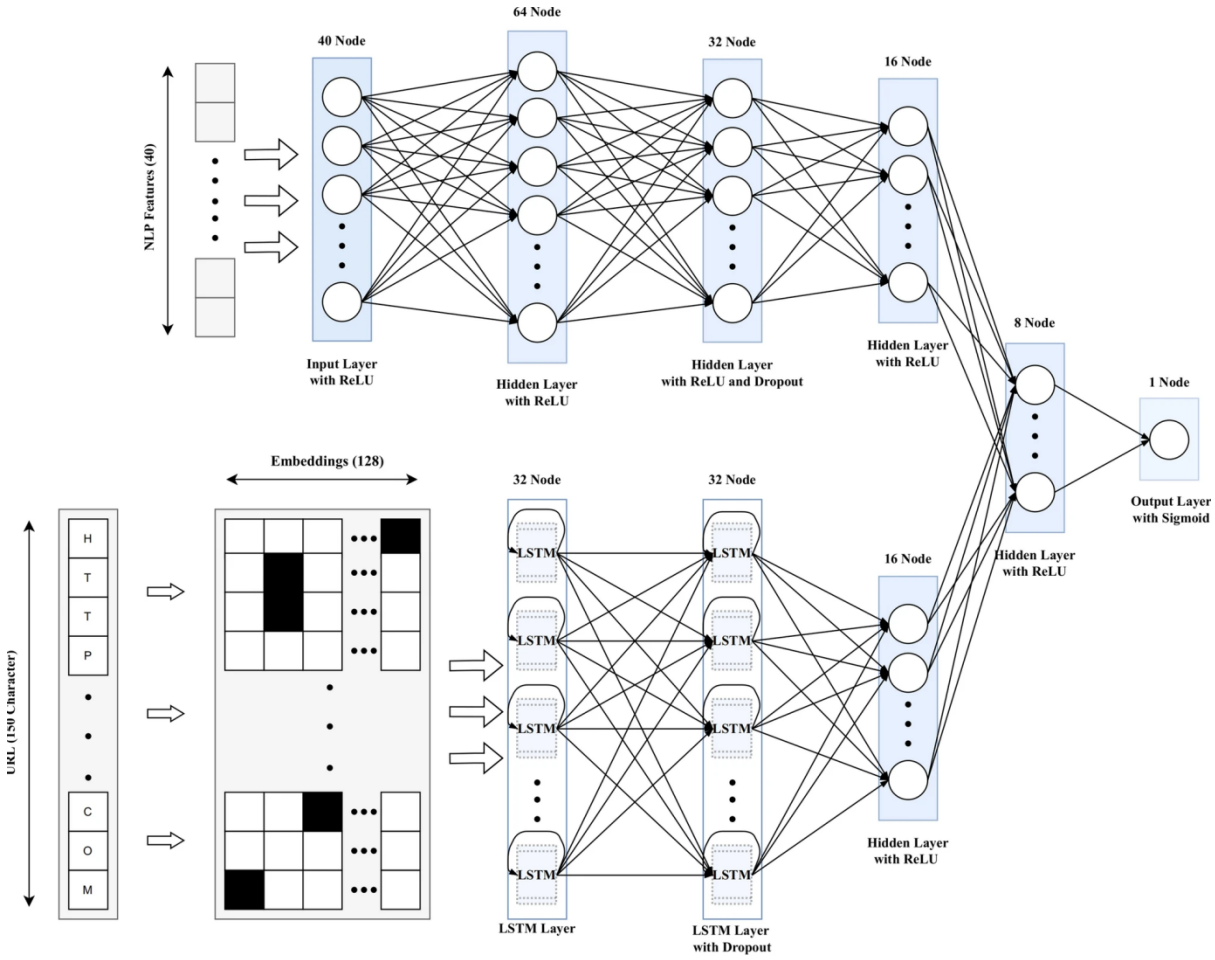

Рисунок 3.8 – DNN-LSTM архітектура**[**51**]**

3.5.5 Порівняння ефективності мереж

В ході експерименту було навчено три моделі (CNN, RNN, CNN- RNN) на однакових вхідних даних та з однаковою кількістю епох(30).

На рисунках 3.4-3.5 зображено графіки з показниками метрик для порівняння моделей.

На першому етапі (порівняння простих архітектур) найкращі результати показала модель CNN (див рис 3.9).

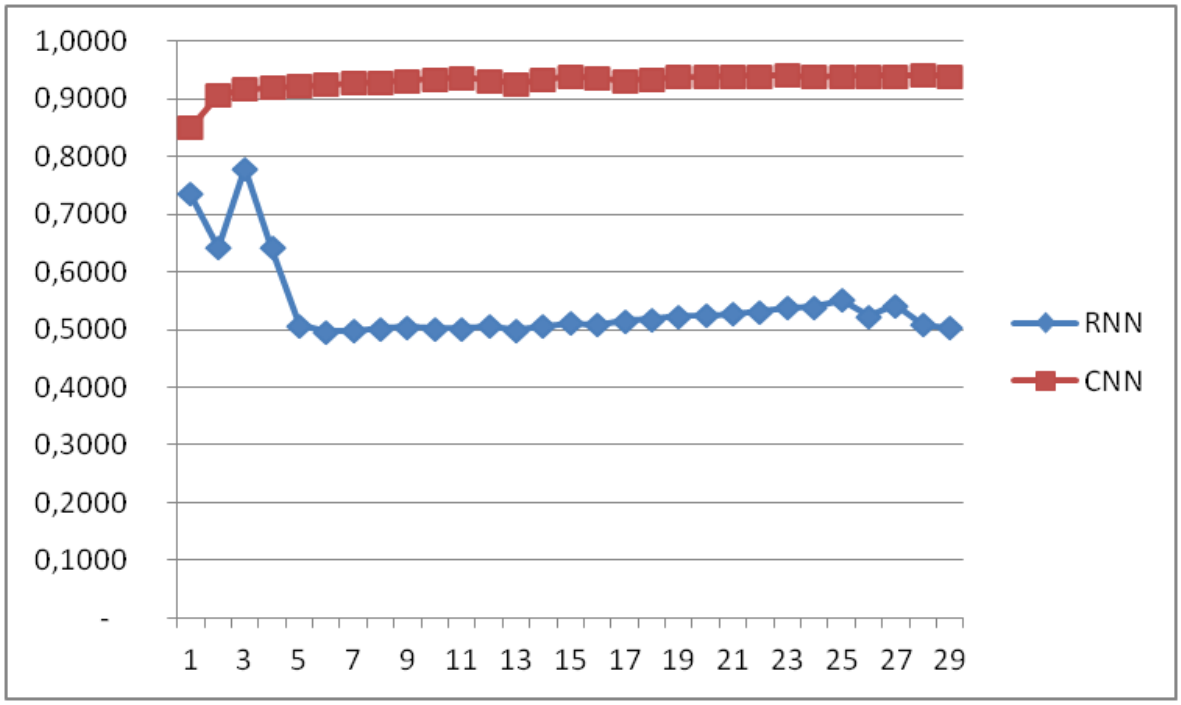

Рисунок 3.9 – Порівняння CNN та RNN архітектур за метрикою accuracy

На другому етапі порівняння кращої з простих архітектур, а саме CNN, та гібридної архітектури CNN-RNN. Найкращі результати показала модель гібрид (див рис 3.10)

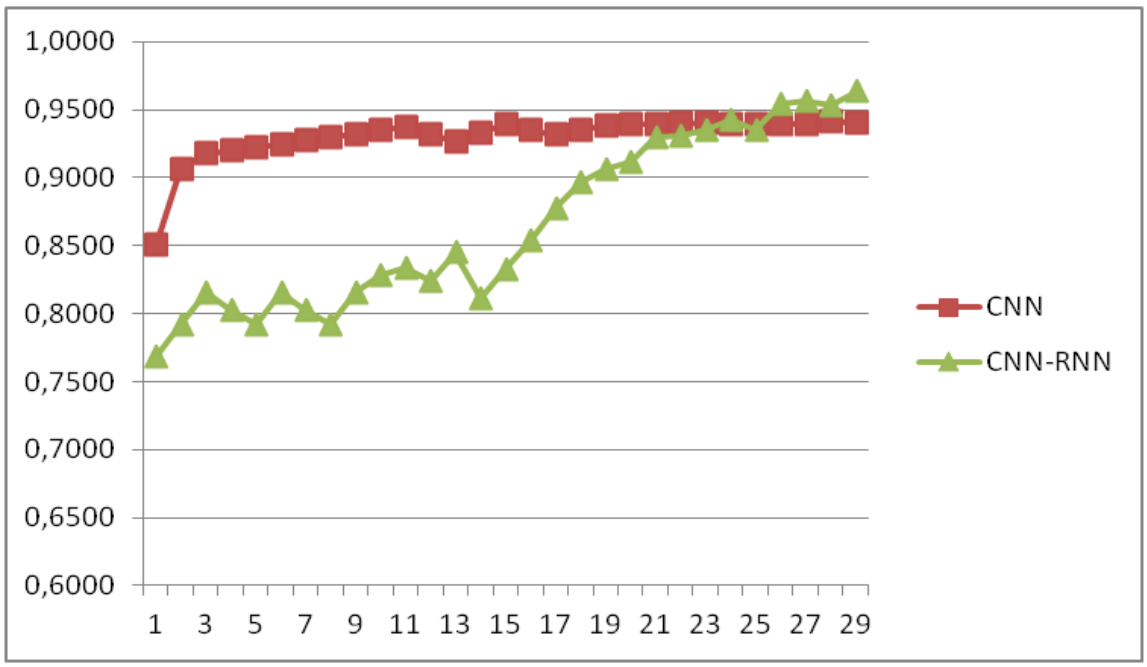

Рисунок 3.10 – Порівняння CNN та гібридної архітектур за метрикою accuracy

На третьому етапі при порівнянні гібридних архітектур (CNN-RNN, DNN-LSTM) найкращі результати за метрикою accuracy показала модель CNN-RNN (див рис 3.11), а за метрикою loss показала модель CNN-RNN (див рис 3.12).

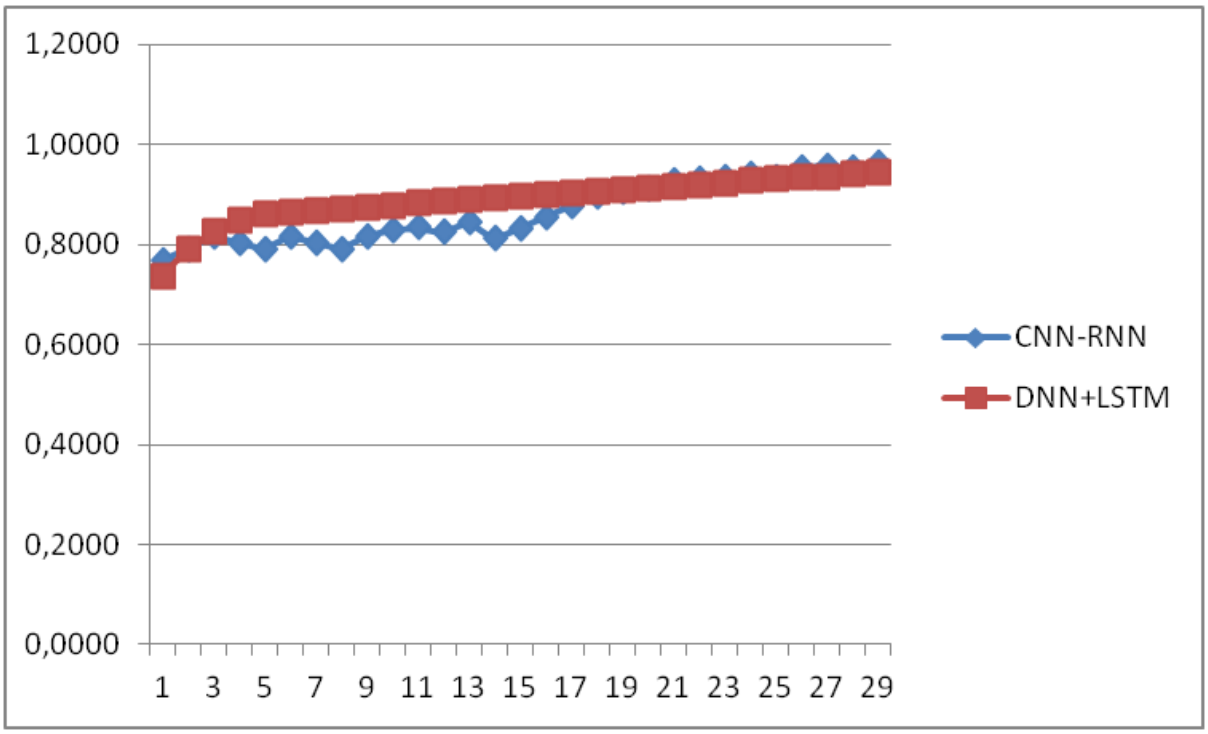

Рисунок 3.11 – Порівняння DNN-LSTM та гібридної архітектур за метрикою accuracy

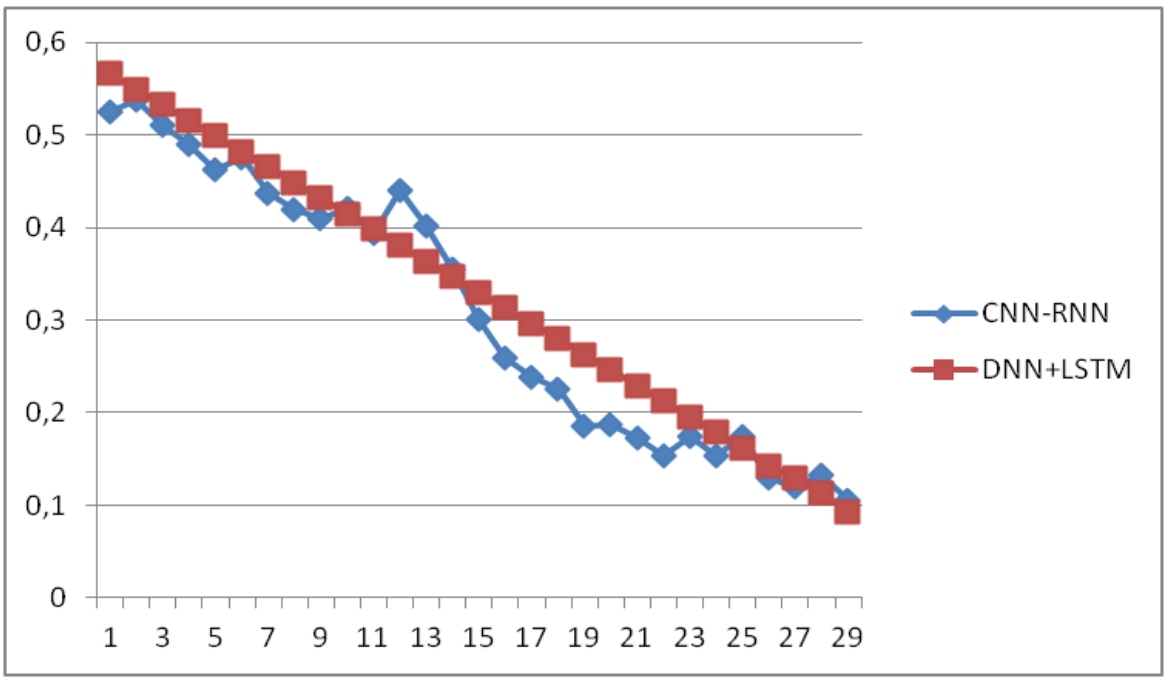

Рисунок 3.12 – порівняння DNN-LSTM та гібридної архітектур за метрикою loss

Висновок третього розділу

Як показав експеримент з простими та гібридними моделями , прості мережі програють гібридним. А серед усіх архітектур модель CNN-RNN є однією із самих складних і ефективних, що дає їй достатню гнучкість. Таким чином, CNN-RNN є найкращою моделлю серед тих, що були представлені у даному дослідженні.

ВИСНОВОК

Фішинг – це різновид кіберзлочину який полягає в тому ,що зловмисники надсилають користувачам шкідливі вкладення електронною поштою або URL-адреси, щоб отримати доступ до їхніх облікових записів або комп’ютера.

У даній кваліфікаційній роботі дослідження стосувалися розробки ефективних засобів виявлення фігингових посилань.

На даний момент вже існує багато різноманітних результатів у боротьбі з фішинговими атаками, заснованих на застосуванні новітніх концепцій, а саме інструментів штучного інтелекту. Це надало можливість проаналізувати існуючі архітектури та методи, порівняти їх, зробити висновки з приводу доцільності їх використання.

На основі даних, отриманих та проаналізованих у першому розділі, було досліджено що таке фітингові атаки їх типи та властивості.

В рамках другого розділу роботи був проведений аналіз існуючих досліджень що до виявлення фішингових посилань на основі використання різних типів архітектур неромереж. Роглянуті архітектури:CNN ,DNN, RNN, LSTM мереж їх недоліки та переваги.

У третьому розділі проводилася навчання та тестування моно та гібридних моделей нейронних мереж . За показниками ефективності порівнювалися такі архітектурні моделі: CNN , RNN, CNN –RNN, DNN-LSTM.Обчислювадьний експеремент показав, що най ефективнішою моделлю є мережа . CNN –RNN.

Перспективним напрямом майбутніх досліджень є розробка нейромережних архітектур з використанням ансамблевих методів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Types of Cybercrime - Panda Security Mediacenter. Panda Security Mediacenter. URL: https://www.pandasecurity.com/en/mediacenter/panda-security/types-of-cybercrime/ (date of access: 24.05.2022).

2. Egan, G. (2020). State of the Phish. Proofpoint. https://www.proofpoint.com/us/security-awareness /post/2020-state-phish-security-awareness-training-email-reporting-more-critical

3. Roberts J. J. Facebook and Google Were Victims of $100M Payment Scam. Fortune. URL: https://fortune.com/2017/04/27/facebook-google-rimasauskas/ (date of access: 24.05.2022).

4. What is Spam and a Phishing Scam - Definition. www.kaspersky.com. URL: https://www.kaspersky.com/resource-center/threats/spam-phishing (date of access: 24.05.2022).

5. Types of phishing - What it is an how to prevent it. Everything About Online Reputation Management. URL: https://blog.reputationx.com/guest/whats-phishing (date of access: 24.05.2022).

6. Bailey, J. L., Mitchell, R. B., and Jensen, B. k. (2008). “Analysis of student vulnerabilities to phishing,” in 14th americas conference on information systems, AMCIS 2008, 75–84. Available at: https://aisel.aisnet.org/amcis2008/271.

7. Khonji, M., Iraqi, Y., and Jones, A. (2013). Phishing detection: a literature survey. IEEE Commun. Surv. Tutorials 15, 2091–2121. doi:10.1109/SURV.2013.032213.00009

8. Scaife, N., Carter, H., Traynor, P., and Butler, K. R. B. (2016). “Crypto lock (and drop it): stopping ransomware attacks on user data,” in 2016 IEEE 36th international conference on distributed computing systems (ICDCS) (IEEE, 303–312. doi:10.1109/ICDCS.2016.46

9. Cybint Cyber Solutions (2018). 13 alarming cyber security facts and stats. Available at: https://www.cybintsolutions.com/cyber-security-facts-stats/ (Accessed July 20, 2019).

10. Sheng, S., Magnien, B., Kumaraguru, P., Acquisti, A., Cranor, L. F., Hong, J., et al. (2007). "Anti-Phishing Phil: the design and evaluation of a game that teaches people not to fall for phish," in Proceedings of the 3rd symposium on usable privacy and security - SOUPS '07 (New York, NY: ACM Press), 88–99. doi:10.1145/1280680.1280692.

11. VadeSecure (2021). Phishers favorites 2019. Available at: https://www.vadesecure.com/en/ (Accessed October 29, 2019).

12. Corrata (2018). The rising threat of social media phishing attacks. Available at: https://corrata.com/the-rising-threat-of-social-media-phishing-attacks/%0D (Accessed October 29, 2019).

13. Dodge, R. C., Carver, C., and Ferguson, A. J. (2007). Phishing for user security awareness. Comput. Security 26, 73–80. doi:10.1016/j.cose.2006.10.009

14. Bin, S., Qiaoyan, W., and Xiaoying, L. (2010). "A DNS based anti-phishing approach." in 2010 second international conference on networks security, wireless communications and trusted computing, Wuhan, China, April 24–25, 2010. (IEEE), 262–265. doi:10.1109/NSWCTC.2010.196

15. Miyamoto, D., Hazeyama, H., and Kadobayashi, Y. (2009). "An evaluation of machine learning-based methods for detection of phishing sites," ininternational conference on neural information processing ICONIP 2008: advances in neuro-information processing lecture notes in computer science. Editors M. Köppen, N. Kasabov, and G. Coghill (Berlin, Heidelberg: Springer Berlin Heidelberg), 539–546. doi:10.1007/978-3-642-02490-0_66

16. Chanti, S., and Chithralekha, T. (2020). Classification of anti-phishing solutions. SN Comput. Sci. 1, 11. doi:10.1007/s42979-019-0011-2

17. Deshmukh, M., and raddha Popat, S. (2017). Different techniques for detection of phishing attack. Int. J. Eng. Sci. Comput. 7, 10201–10204. Available at: http://ijesc.org/

18. Afroz, S., and Greenstadt, R. (2009). “Phishzoo: an automated web phishing detection approach based on profiling and fuzzy matching,” in Proceeding 5th IEEE international conference semantic computing (ICSC), 1–11.

19. Phish Labs (2019). 2019 phishing trends and intelligence report the growing social engineering threat. Available at: https://info.phishlabs.com/hubfs/2019 PTI Report/2019 Phishing Trends and Intelligence Report.pdf.

20. Abu-Nimeh, S., and Nair, S. (2008). “Bypassing security toolbars and phishing filters via dns poisoning,” in IEEE GLOBECOM 2008–2008 IEEE global telecommunications conference, New Orleans, LA, November 30–December 2, 2008 (IEEE), 1–6. doi:10.1109/GLOCOM.2008.ECP.386

21. Hutchings, A., Clayton, R., and Anderson, R. (2016). “Taking down websites to prevent crime,” in 2016 APWG symposium on electronic crime research (eCrime) (IEEE), 1–10. doi:10.1109/ECRIME.2016.7487947

22. Feng, J.; Zou, L.; Nan, T. A Phishing Webpage Detection Method Based on Stacked Autoencoder and Correlation Coefficients. J. Comput. Inf. Technol. 2019, 27.

23. Feng, J.; Zou, L.; Ye, O.; Han, J. Web2Vec: Phishing Webpage Detection Method Based on Multidimensional Features Driven by Deep Learning. IEEE Access 2020, 8, 221214–221224.

24. . Huang, Y.; Yang, Q.; Qin, J.; Wen, W. Phishing URL Detection via CNN and Attention-Based Hierarchical RNN. In Proceedings of the 2019 18th IEEE International Conference on Trust, Security and Privacy in Computing and Communications/13th IEEE International Conference on Big Data Science And Engineering (TrustCom/BigDataSE), Rotorua, New Zealand, 5–8 August 2019; pp. 112–119

25. Chen, Z. Deep Learning for Cybersecurity: A Review. In Proceedings of the 2020 International Conference on Computing and Data Science (CDS), Stanford, CA, USA, 1–2 August 2020; pp. 7–18.

26. Naway, A.; LI, Y. A Review on The Use of Deep Learning in Android Malware Detection. arXiv 2018, arXiv:181210360. Available online: http://arxiv.org/abs/1812.10360 (accessed on 3 April 2021).

27. Wu, Y.; Wei, D.; Feng, J. Network Attacks Detection Methods Based on Deep Learning Techniques: A Survey. Secur. Commun. Netw. 2020, 2020, e8872923.

28. Mahdavifar, S.; Ghorbani, A.A. Application of deep learning to cybersecurity: A survey. Neurocomputing 2019, 347, 149–176.

29. Mahdavifar, S.; Ghorbani, A.A. DeNNeS: Deep embedded neural network expert system for detecting cyber attacks. Neural Comput. Appl. 2020, 32, 14753–14780.

30. Sahingoz, O.K.; I˛sılay Baykal, S.; Bulut, D. Phishing detection from urls by using neural networks. In Computer Science & Information Technology (CS & IT); AIRCC Publishing Corporation: Chennai, India, 2018; pp. 41–54.

31. Khan, M.F.; Al, E. Detection of Phishing Websites Using Deep Learning Techniques. Turk. J. Comput. Math. Educ. TURCOMAT 2021, 12, 3880–3892

32. Sountharrajan, S.; Nivashini, M.; Shandilya, S.K.; Suganya, E.; Bazila Banu, A.; Karthiga, M. Dynamic Recognition of Phishing URLs Using Deep Learning Techniques. In Advances in Cyber Security Analytics and Decision Systems; Shandilya, S.K., Wagner, N., Nagar, A.K., Eds.; EAI/Springer Innovations in Communication and Computing; Springer International Publishing: Cham, Switzerland, 2020; pp. 27–56, ISBN 978-3-030-19353-9.

33. Selvaganapathy, S.; Nivaashini, M.; Natarajan, H. Deep belief network based detection and categorization of malicious URLs. Inf. Secur. J. Glob. Perspect. 2018, 27, 145–161.

34. Aldweesh, A.; Derhab, A.; Emam, A.Z. Deep learning approaches for anomaly-based intrusion detection systems: A survey, taxonomy, and open issues. Knowl.-Based Syst. 2020, 189, 105124.

35. Asharf, J.; Moustafa, N.; Khurshid, H.; Debie, E.; Haider, W.; Wahab, A. A Review of Intrusion Detection Systems Using Machine and Deep Learning in Internet of Things: Challenges, Solutions and Future Directions. Electronics 2020, 9, 1177.

36. Wei, W.; Ke, Q.; Nowak, J.; Korytkowski, M.; Scherer, R.; Wo ´zniak, M. Accurate and fast URL phishing detector: A convolutional neural network approach. Comput. Netw. 2020, 178, 107275.

37. Yerima, S.Y.; Alzaylaee, M.K. High Accuracy Phishing Detection Based on Convolutional Neural Networks. In Proceedings of the 2020 3rd International Conference on Computer Applications Information Security (ICCAIS), Riyadh, Saudi Arabia, 19–21 March 2020; pp. 1–6.

38. Yuan, L.; Zeng, Z.; Lu, Y.; Ou, X.; Feng, T. A Character-Level BiGRU-Attention for Phishing Classification. In Information and Communications Security; Zhou, J., Luo, X., Shen, Q., Xu, Z., Eds.; Springer International Publishing: Cham, Switzerland, 2020; pp. 746–762.

39. Feng, T.; Yue, C. Visualizing and Interpreting RNN Models in URL-based Phishing Detection. In Proceedings of the 25th ACM Symposium on Access Control Models and Technologies, Barcelona, Spain, 10–12 June 2020; pp. 13–24.

40. NumPy [Електронний ресурс] // NumPy developers – Режим доступу до ресурсу: . Дата доступу: 05.06.2020

41. Keras [Електронний ресурс] – Режим доступу до ресурсу: https://keras.io/. Дата доступу: 05.06.2020

42. Tqdm [Електронний ресурс] – Режим доступу до ресурсу: https://pypi.org/project/tqdm/. Дата доступу: 05.06.2020

43. H5py [Електронний ресурс] – Режим доступу до ресурсу: http://docs.h5py.org/en/stable/. Дата доступу: 05.06.2020

44. Matlab [Електронний ресурс] – Режим доступу до ресурсу: https://matplotlib.org/. Дата доступу: 05.06.2020

45. scikit-learn [Електронний ресурс] – Режим доступу до ресурсу: https://scikit-learn.org/ Дата доступу: 05.06.2020

46. Tensorflow [Електронний ресурс] – Режим доступу до ресурсу: https://www.tensorflow.org/. Дата доступу: 05.06.2020

47. Pandas [Електронний ресурс] – Режим доступу до ресурсу: https://pandas.pydata.org/. Дата доступу: 05.06.2020

48. UCI Machine Learning Repository, “Phishing Websites Dataset” https://archive.ics.uci.edu/ml/datasets/phishing+websites

49. Інформаційні системи та технології в управлінні. Методичні вказівки, теоретичні відомості і завдання до лабораторних робіт для студентів та магістрів денної форми навчання спеціальності 7.803060101 Менеджмент організацій і адміністрування. Частина 3. Класифікація в бізнес-аналітиці. / Укл.: Біла Н.І. – Запоріжжя: ЗНТУ, 2014. – с. 50.

50. Хроленко Я. О. Дипломна робота на здобуття ступеня бакалавра за освітньо-професійною програмою «Інтелектуальні сервіс-орієнтовані розподілені обчислювання». 2020. С. 47–52.\

51. A hybrid DNN–LSTM model for detecting phishing URLs https://link.springer.com/article/10.1007/s00521-021-06401-z#Fig1

ДОДАТОК А

Тези наукових доповідей:

1. Khrolenko Volodymr, Khrolenko Yaroslav DETERMINATION OF PHISHING LINKS USING NEURAL NETWORKS// Ninth international scientific-practical conference «Management of the development of technologies» Kyiv, 28 March 2022 // KYIV NATIONAL UNIVERSITY OF CONSTRUCTION AND ARCHITECTURE . –C. 57 -58

ДОДАТОК Б

Лістинг коду:

```
from keras.models import Sequential
from keras.layers import Dense,Flatten,LSTM,Dropout,Bidirectional
from keras.layers.recurrent import SimpleRNN
import numpy as np
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import StratifiedKFold
from keras.wrappers.scikit_learn import KerasClassifier
from keras.layers.embeddings import Embedding
from keras.layers.convolutional import Conv1D, MaxPooling1D
from preprocess_char_word_cnn_lstm import convert_urls_to_vector
from evaluating_indicator import metric_F1score,metric_precision,metric_recall
import matplotlib.pyplot as plt
seed = 7
np.random.seed(seed)
url_len = 300
out_dimension = 64
word_size=121+1
taccuracy_count=[]
vaccuracy_count = []
F1_count=[]
precision_count=[]
recall_count=[]


def create_model():
    model = Sequential()
```

```
    model.add(Embedding(word_size, out_dimension, input_length=url_len))
    model.add(Conv1D(filters=200, kernel_size=2, padding='same', activation='relu'))
    model.add(MaxPooling1D(pool_size=2))
    model.add(SimpleRNN(100))
    model.add(Dropout(0.5))
    model.add(Dense(1, activation='sigmoid'))
    model.compile(loss='binary_crossentropy', optimizer='adam',
metrics=['accuracy',metric_precision,metric_recall,metric_F1score])
    model.summary()
    return model


def main():
    file_names = ["dataset/phishing_url.txt", "dataset/cc_1_first_9617_urls"]
    is_phishing = [True, False]
    x,y = convert_urls_to_vector(file_names, is_phishing)
    model =create_model()
    history=model.fit(x, y, batch_size=100, epochs=30,validation_split=0.2).history

    taccuracy_count=history["accuracy"].copy()
    vaccuracy_count=history["val_accuracy"].copy()
    precision_count=history["val_metric_precision"].copy()
    recall_count=history["val_metric_recall"].copy()
    F1_count=history["val_metric_F1score"].copy()
    f = open(r"E:\daima-sx\test2\result\evaluating_indicator_cnn_rnn", "w+")
    f.writelines('taccuracy_count'+str(taccuracy_count)+'\n')
    f.writelines('vaccuracy_count'+str(vaccuracy_count)+'\n')
    f.writelines('precision_count' + str(precision_count)+'\n')
    f.writelines('recall_count' + str(recall_count)+'\n')
    f.writelines('F1_count' + str(F1_count) + '\n')
```

```
    f.close()
    plt.plot(history['loss'])
    plt.plot(history['val_accuracy'])
    plt.plot(history['accuracy'])
    plt.ylabel('accuracy/loss')
    plt.xlabel('epochs')
    plt.show()


if __name__ == '__main__':
    main()
```