НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

Факультет інформаційних технологій

УДК 004.9:070.16

ПОГОДЖЕНО	**ДОПУСКАЄТЬСЯ ДО ЗАХИСТУ**
Декан факультету інформаційних технологій	**Завідувач кафедри комп'ютерних наук**
_____ **БОЛБОТ І.М. д.т.н, професор**	_____ **ГОЛУБ Б.Л. к.т.н доцент**
(підпис) (ПІБ)	(підпис) (ПІБ)
"____" _____ 2024 р.	"____" _____ 2024 р.

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему *«СИСТЕМА ВИЯВЛЕННЯ ЕЛЕМЕНТІВ ДЕЗІНФОРМАЦІЇ В ПОТОКАХ ТЕКСТОВИХ ДАНИХ»*

Спеціальність 122 «Комп'ютерні науки»

Освітня програма Інформаційні управляючі системи та технології

Орієнтація освітньої програми освітньо-професійна

Гарант освітньої програми
к.т.н., доцент _____ ГОЛУБ Б.Л.
(підпис) (ПІБ)

Керівник магістерської кваліфікаційної роботи
к.т.н., доцент _____ СВАТКО В.В.
(підпис) (ПІБ)

Виконав _____ КАЧМАР А.В.
(підпис) (ПІБ студента)

КИЇВ – 2024

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ
Факультет інформаційних технологій

ЗАТВЕРДЖУЮ
Завідувач кафедри комп'ютерних наук

к.т.н., доцент _____ ГОЛУБ Б.Л.
(підпис)
"_____" _____2023 року

З А В Д А Н Н Я
ДО ВИКОНАННЯ МАГІСТЕРСЬКОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ
СТУДЕНТУ

Качмар Андрій Володимирович
(прізвище, ім'я, по батькові)

Спеціальність 122 «Комп'ютерні мережі»
(код і назва)

Освітня програма «Інформаційні управляючі системи та технології»
(назва)

Орієнтація освітньої програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Тема магістерської кваліфікаційної роботи «Система виявлення елементів дезінформації в потоках текстових даних»
затверджена наказом ректора НУБіП України від 26 квітня 2023 р. №10
Термін подання завершеної роботи на кафедру
(рік, місяць, число)

Вихідні дані до магістерської кваліфікаційної роботи: Словникові та довідникові джерела; електронні джерела; вітчизняні та іноземні джерела

Перелік питань, що підлягають дослідженню:
1. Дослідження традиційних моделей для виявлення фейкових новин.
2. Які метрики використовувати для оцінки ефективності моделі?
3. Вплив попередньої обробки тексту на точність моделей.

Дата видачі завдання

Керівник магістерської кваліфікаційної роботи_____ СВАТКО В.В.
(підпис) (прізвище та ініціали)

Завдання прийняв до виконання _____ КАЧМАР А.В.
(підпис) (прізвище та ініціали студента)

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

NLP - Natural Language Processing (Обробка природної мови)

ANN - Artificial neural network (Штучна нейронна мережа)

ML - Machine Learning (Машинне навчання)

RNN - Recurrent Neural Network (Рекурентна нейронна мережа)

SVM - Support Vector Machines (Метод опорних векторів)

TF-IDF - Term frequency inverse document frequency (Частотний коефіцієнт термінів)

IDF - Inverse Document Frequency (Зворотна частота документа)

SA - Sentiment Analysis (Аналіз настроїв)

LR - Logistic Regression (Логістична регресія)

AI - Artificial Intelligence (Штучний інтелект)

1. ВСТУП

1.1. Актуальність теми

Важливість даної роботи на сьогоднішній день визначена кількома об'єктивними факторами. По-перше, можна відзначити постійне зростання кількості інформації, яка щодня генерується та поширюється через соціальні мережі та інші онлайн-платформи. Постійні потоки нової інформації ускладнюють визначення достовірності фактів для пересічного користувача. Таким чином створюються сприятливі умови для поширення дезінформації, що може шкодити суспільству через вплив на суспільні погляди і тенденції.

По-друге, міжнародна політична ситуація робить актуальним завдання виявлення дезінформації, мета якої - створити нові загрози для міжнародної безпеки. Багато людей отримують інформацію через соціальні медіа, і тому зростає попит на інструменти, які забезпечуватимуть якість та достовірність інформації.

Фейкові новини — це тип новин, які не мають жодного фактичного підґрунтя, але подаються як правдиві [3]. Вони можуть містити оманливий, хибний, підроблений, маніпулятивний або сфабрикований контент, а також сатиру, пародію чи помилкові зв’язки з метою введення людей в оману. Таким чином, фейкові новини можуть суттєво впливати на різні аспекти життя.

Фейкові новини привернули значну увагу суспільства з 2016 року [5], що стало помітним завдяки їхній зростаючій популярності. Вони поширюються відомими політиками, медіа-компаніями та іншими джерелами, такими як соціальні мережі та чутки. Це вплинуло на багатьох людей, адже достовірність новинних матеріалів та заяв почала активно оскаржуватися як у політичному, так і в науковому середовищі.

Довіра до інформаційних агентств зазнала серйозної критики, а термін "фейкові новини" став синонімом суперечок щодо прийнятності різних точок зору, що призвело до емоційного сприйняття замість раціонального обговорення фактів. Кількість фейкової інформації зростає та охоплює дедалі більше тем, хоча створювати неправдиві твердження складніше, коли мова йде про технічно складні чи наукові питання. Як і справжні новини, фейкові змінюються залежно від тематики. Наприклад, під час президентських виборів у США у 2016 році було опубліковано та поширено величезну кількість політичних новин, що спричинило збільшення кількості політично спрямованих фейків.

Фейкові новини стають все поширенішими, однак боротьба з ними також активізується, як і загальна обізнаність про те, як їх розпізнавати. Потрібні інструменти, що еволюціонують разом із цими викликами, для їх мінімізації та протидії. Зміна визначення та використання терміну "фейкові новини" у суспільстві може призвести до хибного розуміння цього явища. Через це дослідники, що займаються вивченням фейкових новин, прагнуть змінити термінологію, адже вона охоплює значно ширше поняття, ніж просто новини. Це потрібно для більш точного обговорення різних аспектів цієї проблеми, що детально розглядається у розділі 2.

Отже, магістерська робота про Систему виявлення елементів дезінформації в потоках текстових даних є актуальною і важливою для забезпечення інформаційної безпеки, розбірливості та стабільності суспільства.

1.2. Предмет та об’єкт дослідження

Об’єкт дослідження - методи та моделі машинного навчання.

Предмет дослідження: система прийняття рішень, яка допоможе користувачу визначати достовірність інформації в медіапросторі шляхом аналізу показників, які вказують на достовірність інформації у тексті.

1.3. Мета дослідження

Коли йдеться про виявлення та прогнозування появи неправдивої інформації, існують два основні підходи. Найбільш поширений підхід є частиною обробки природної мови (Natural Language Processing або NLP), у якому аналізується сам текст на основі використаних евристик. Він базується на розумінні писемної мови, і цей метод досліджується та вдосконалюється з часів Георгтаунського експерименту 1954 року [21]. NLP застосовує як статичні підходи, такі як N-грами та метод опорних векторів (SVM), так і сучасніші техніки, пов’язані з машинним навчанням і нейронними мережами.

Інший підхід аналізує інформацію, яка не є безпосередньо частиною тексту, наприклад, дані користувачів, джерела та мережевий трафік. Цей підхід використовує нетекстові дані й ґрунтується виключно на їхньому аналізі. Це створює універсальний метод, який можна застосовувати до мов, для яких комп’ютери поки не мають достатніх знань для використання традиційних лінгвістичних методів. Цей підхід здебільшого є прогнозувальним і не здатний у такій же мірі, як текстові методи, визначати, чи є інформація правдивою.

Ця робота приділяє основну увагу спрощенню виявлення ознак дезінформації шляхом створення методу та алгоритму для перевірки текстових потоків на наявність лінгвістичних структур і виразів, що вказують на недостовірність представленої інформації.

1.4. Наукова новизна

Запропоновано сукупність методів для порівняльного аналізу моделей машинного навчання з акцентом на виявлення фейкових новин у текстових даних, удосконалення процесу валідації моделей шляхом використання крос-валідації з комплексною оцінкою метрик, що дозволяє підвищити об’єктивність аналізу ефективності алгоритмів. Також було досліджено застосування та порівняння ефективності декількох моделей, зокрема Logistic Regression, Random Forest, Gradient Boosting, Linear SVM, та інших, для задачі текстової класифікації з точки зору їх придатності до реальних умов класифікації фейкових новин.

1.5. Структура роботи

Робота складається із 60 сторінок основного тексту. У роботі використано 36 джерел.

Робота структурована у три розділи, кожен з яких виконує свою функцію для досягнення поставленої мети.

Перелік умовних позначень – містить пояснення основних термінів, абревіатур та позначень, які використовуються у тексті роботи.

1. Вступ – розкриває актуальність теми дослідження, формулює об’єкт, предмет і мету роботи. Вказується наукова новизна, яка полягає у вперше запропонованих методах оцінювання моделей для виявлення фейкових новин. Також наведена загальна структура роботи.

2. Аналіз предметної області – складається з шести підрозділів і детально описує актуальність проблеми дезінформації, визначення фейкових новин, та різні підходи до їхнього виявлення. Розглянуто лінгвістичний підхід, контекстуальний підхід та методи видобутку інформації, які є основою для побудови моделей виявлення дезінформації.

3. Розробка і реалізація системи виявлення дезінформації – містить сім підрозділів. У розділі описано вибір та підготовку набору

даних, попередню обробку текстової інформації, векторизацію текстів методом TF-IDF, вибір та навчання моделей машинного навчання. Проведено оцінку ефективності моделей за метриками точності, recall, F1-мірою, та виконано аналіз їхніх результатів.

4. Висновки – підсумовують основні результати роботи, роблять порівняльний аналіз моделей, окреслюють науковий внесок та дають рекомендації для подальших досліджень.

5. Список використаних джерел – охоплює наукові публікації, статті та інші джерела, використані для виконання роботи.

Робота також містить ілюстрації, таблиці, графіки та додатки, які забезпечують наочність та обґрунтованість результатів.

2. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

2.1. Вплив фейкових новин

Поширення фальшивої інформації можна поділити на багато різних груп залежно від наміру чи походження інформації. Те, про що більшість людей думає, чуючи слово "фальшиві новини", до певної міри можна назвати пропагандою. Пропаганда — це свідома маніпуляція емоціями та думками людей за допомогою сильних засобів і інструментів для створення певних сприйняттів і дій. Один із прикладів цього — це висунення заяви, яка буде, ймовірно, визнана неправдою пізніше, але, роблячи такі заяви, особливо в політиці, можна виставити інших людей у негативному світлі і, таким чином, змінити спосіб, яким люди про них думають. Крім того, деяким популярним елементом фальшивих новин в останній час є приховані платні пости, фальшиві акаунти та платний контент на соціальних мережах. Наприклад, у дебатах щодо нейтральності Інтернету з'явилися докази того, що мільйони коментарів, використаних як доказ, були фальшивими[9], що зловжили іменами реальних користувачів, а контент був підроблений для підтримки певної думки в дебатах. Є також докази того, що новинні статті під час виборів у США були створені з метою просування порядку одного з партій. Було також багато статей на підтримку іншої партії, але їхня кількість і масштаби були значно меншими. Крім того, виникли області, де фальшиві новини та поширення хибної інформації стали бізнесом. Є село в Македонії[26], де виробництво фальшивих новин приносить людям великі гроші. Така реакція схожа на виробництво підробок в інших галузях. Якщо є попит на продукт, знайдеться й той, хто його постачатиме. Ще одним аспектом фальшивих новин є просто хибна інформація. Хибна інформація може бути як неправдивими фактами, так і заявами, які є помилковими, або простими помилками, зробленими під час створення інформації чи новинних статей. Ці фальшивки важче виявити, оскільки зазвичай весь сюжет не створюється як фальшивий, а

є змішаним із правдивою та неправильною інформацією. Це може бути через використання застарілих джерел, упереджених джерел або ж через припущення без перевірки фактів. Такий тип "фальшивості", наразі, найкраще обробляється людьми, оскільки автоматизація перевірки фактів і валідації правильних тверджень є складною для комп'ютерів. Ще один спосіб поділу фальшивих новин — це намір інформації. Це, передусім, можна поділити на три частини: шахрайства, сатиру та шкідливий контент. Шахрайство — це вигадка, створена для того, щоб виглядати правдою. Це можуть бути події, такі як чутки, урбаністичні легенди, псевдонаука. Це також можуть бути практичні жарти, жартівливі новини на День дурня тощо. Шахрайства можуть варіюватися від добрих за наміром, таких як жарти, до злих і небезпечних історій, таких як псевдонаука та чутки. Окрім шахрайств, є сатира, коли щось висміюється. Наприклад, публічна особа висміюється добрими намірами, коли деякі з її більш виразних рис виводяться з контексту і стають ще більш помітними. Сатира може, як шахрайства, бути як жартівливою, так і використовуватися зі злим наміром для того, щоб знизити статус когось чи чогось. І нарешті, є контент, створений з наміром бути руйнівним, тобто шкідливий контент. Цей контент створюється для того, щоб дестабілізувати ситуацію, змінювати громадську думку і використовувати хибну інформацію для поширення меседжу з метою завдати шкоди інституціям, особам, політичним поглядам або чомусь подібному. Важливо помітити, що існує явне накладення між різними типами фальшивок. Це відбувається через намір того, хто створює інформацію. Усі згадані типи можуть бути шкідливими, якщо в інформаційний матеріал потрапляють помилкові дані. Намір буде абсолютно різним. Один із прикладів — це публікації або статті, в яких міститься неправильна інформація, що була спростована на пізнішому етапі, але досі використовується як джерело певними групами для підтвердження своєї точки зору. Це може призвести до розколу щодо того, що є правильним в науці, де

можна вибирати лише ті частини науки, в які хочеш вірити. Така руйнівна поведінка підриває сутність емпіричних досліджень і потребує втручання.

2.2. Визначення фейкових новин

Коли йдеться про фейкові новини, пропаганду чи дезінформацію, завжди важливо визначити, що саме потрібно виявляти. Легше мати спеціалізовану систему, яка націлена лише на певні аспекти фальшивих новин. У такому випадку система має глибше розуміння в межах своєї сфери, але не здатна виявляти інші види дезінформації. Загальні системи, які намагаються виявити кілька аспектів, не є настільки точними, оскільки такі системи повинні адаптуватися до змін, а правила не можуть бути такими строгими, як у спеціалізованих системах. Підхід, який працює з контекстними даними, має деякі переваги над мовними підходами. Контекстна інформація однакова кожного разу, незалежно від мови. Тому контекстну систему можна застосовувати до різних наборів даних, що охоплюють різні мови, з мінімальними змінами. Однак контекстні підходи не є такими визначеними, як підходи на основі обробки природної мови (NLP). Оскільки вони аналізують ймовірнісні та статистичні дані, їм буде складніше виявляти аномалії, наприклад, користувачів, які слідують за фальшивою інформацією, але не поширюють її і не беруть участь у поширенні дезінформації. Для контекстної системи такий користувач виглядатиме як той, хто віддає перевагу фейковій інформації, хоча насправді це не так. Аналізуючи текстову інформацію, ці деталі можна виявити. Це показує, як різні підходи на всіх рівнях залежать від технік, які вони використовують. Окрім того, як здійснюється виявлення або передбачення, результати майже завжди кращі, якщо система розроблена для виявлення певних частин спектра. Орієнтація на клікбейт, сатиру, фальшиві веб-сторінки та фальшиві реклами — це приклади такої сегментації. Різні підходи краще справляються з певними проблемами. На найвищому рівні є

лінгвістичні та мережеві методи, які працюють з різними типами даних. Лінгвістичні методи аналізують лише мову, тоді як мережеві методи враховують інформацію, що оточує мову, таку як мережевий трафік, взаємовідносини користувачів і посилання. Ці методи будуть детально розглянуті в наступних розділах разом з загальними методами, які можна використовувати разом з лінгвістичними та мережевими методами для покращення результатів.

2.3. Загальний підхід до виявлення дезінформації

Методи та техніки, згадані в цьому розділі, можуть використовуватися як у лінгвістичних, так і в контекстуальних підходах. Це методи, які приймають рішення на основі навчання, а не правил, і тому здатні покращуватися з часом залежно від вхідних даних, на відміну від статичних методів. Одним із успішно застосованих методів є машинне навчання. Воно використовується для тренування класифікаторів, що покращують процес прийняття рішень, здебільшого в лінгвістичних підходах, але може також застосовуватися в контекстуальних і мережевих методах.

Глибоке навчання також використовується разом із різними техніками, здебільшого у вигляді нейронних мереж. Для підвищення ефективності лінгвістичних систем до них інтегрують підходи машинного навчання та штучного інтелекту. Це зробило системи більш стійкими та здатними обробляти дедалі різноманітніший контент завдяки компоненту навчання, притаманному машинному навчанню та штучному інтелекту (AI), на відміну від більш статичних ранніх систем.

2.3.1. Штучні нейронні мережі

Нейронні мережі — це парадигма програмування, яка дозволяє комп'ютерам навчатися на основі спостережних даних, тим самим підвищуючи ефективність і точність з часом [29].

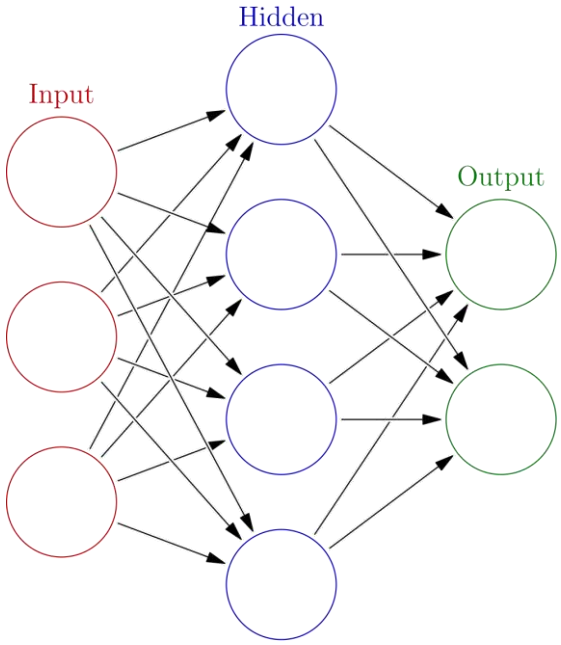


Рис. 1. Штучна нейронна мережа, приклад - Wikipedia, the free encyclopedia. Artificial neural network with layer coloring, 2013.

Штучна нейронна мережа (Artificial Neural Network, ANN) — це система, яка імітує спосіб обробки інформації біологічними нервовими системами, такими як мозок. Сам мозок — це серія взаємопов'язаних нейронів, кожен з яких працює над вирішенням тієї самої проблеми. Завдяки здатності до навчання мозок може знаходити кращі рішення, отримуючи більше вхідних даних і з часом вдосконалюючись.

ANN здебільшого налаштовуються для виконання однієї задачі, що дозволяє їм спеціалізувати свої навчальні можливості в межах конкретної теми. Чим більше мережа спеціалізується, тим краще вона визначає дані, які не відповідають створеній моделі. Завдяки цьому відхилення й інші аномалії

виявляються набагато легше. Наприклад, це можна застосовувати в аналізі податкових даних: система може обробляти звичайні податкові форми, але щойно щось виходить за межі встановлених порогів, вона може повідомити людину для подальшої перевірки.

Головна відмінність між нейронними мережами та традиційними обчислювальними системами полягає в тому, що ANN не слідують фіксованому набору інструкцій для знаходження рішення. Натомість вони органічно знаходять рішення, тому їхня поведінка може бути непередбачуваною, якщо вони не отримали правильного навчання та вхідних даних.

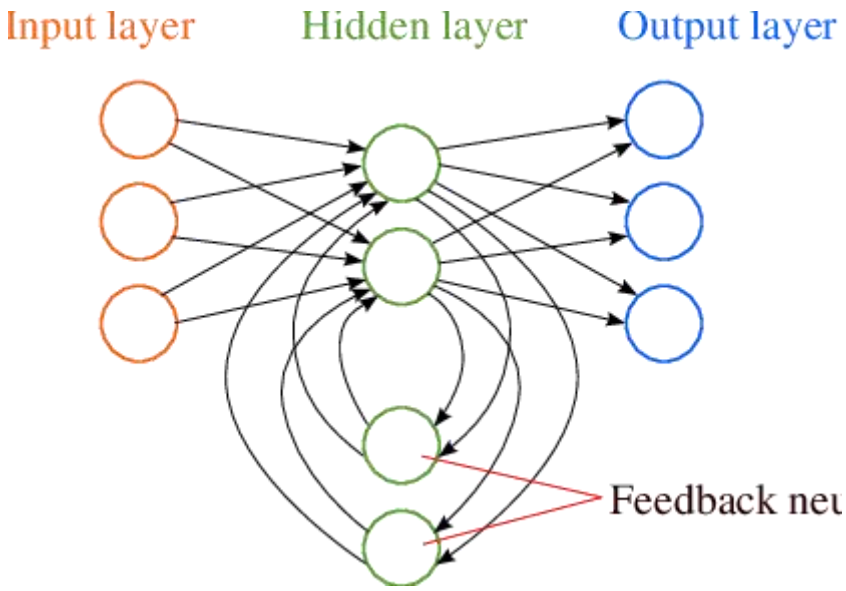


Рис. 2. Рекурентна нейронна мережа. Michael Galetzka, Lutz Strüngmann, and Christian Weber. Intelligent predictions: an empirical study of the cortical learning algorithm. University of Applied Sciences Mannheim, 2014.

Нейрони в ANN створюються так, що вони активують вихід на основі певних входів. Вони також розроблені для роботи в двох режимах: режимі навчання та режимі використання. Режим навчання використовується для того, щоб навчити нейрон активуватися для заданого входу. Оскільки кожен

нейрон аналізує невелику частину задачі, можна поєднати багато нейронів для аналізу, наприклад, зображення, і якщо достатня кількість нейронів активується для заданого входу, можна визначити, що на зображенні є обличчя, завдяки певній активації нейронів і виявленим ними шаблонам.

Нейрони можуть бути набагато складнішими, ніж описані вище. Вони можуть мати вагові входи, де певні входи мають перевагу над іншими і активуються, якщо загальний вхід перевищує поріг. Мережі існують у багатьох різних формах. Прямі мережі (Feed-forward networks), див. рис. 1, завжди працюють в одному напрямку — від входу до виходу — і зазвичай використовуються для розпізнавання шаблонів. Зворотні мережі (Feedback networks), див. рис. 2, можуть передавати сигнали назад і вперед у мережі та містять цикли. Через це стан всієї мережі постійно змінюється і видає результат лише тоді, коли система перебуває у стабільному стані.

Більш складні ANN використовують нейрони, які називаються перцептронами — це нейрони з ваговими входами та деякими додатковими, фіксованими, попередніми обчисленнями.

2.3.2 Машинне навчання

Деякі завдання надзвичайно складно програмувати вручну, і їх краще вирішувати за допомогою машин. Розпізнавання облич, добування даних, рухи роботів та інші складні завдання часто включають забагато змінних, щоб людина могла їх відслідковувати. Тому краще використовувати комп'ютер для вирішення таких завдань, використовуючи їх здатність до навчання. Машинне навчання може бути використано для вирішення складних завдань і базується на реальних даних, а не на інтуїції. Це робить його трохи відмінним від ANN[14].

Для навчання ML існують три основні підходи. Контрольоване навчання — це навчання, де дані для мають як входи, так і виходи, щоб ми завжди знали

правильну відповідь на вхід і могли навчити і адаптувати алгоритм ML для отримання того самого виходу, що й правильний. Другий підхід — це неконтрольоване навчання, коли деякі дані містять лише входи. Через це система робить певні припущення і таким чином неконтрольовано класифікує вхід без відомої правильної відповіді. І, нарешті, підкріплене навчання — це навчання, коли немає прямого доступу до правильного виходу, але якість виходу можна виміряти за допомогою входу. Підкріплене навчання використовує винагороди для кількісної оцінки виходу, і з часом модель змінюється залежно від того, як змінюється загальна винагорода. Такий евристичний підхід, коли модель змінюється з часом, подібний до того, як працюють ANN, і є накладання між методами, а також областями їх застосування.

2.4 Лінгвістичний підхід

Лінгвістичний або текстовий підхід до виявлення неправдивої інформації включає використання технік, які аналізують частоту, використання та патерни в тексті. Використання цього підходу дає можливість знаходити схожості, які відповідають використанню, відомому для певних типів текстів, таких як фейкові новини, які мають мову, схожу на сатиру, і будуть містити більше емоційної та простішої мови порівняно з статтями на ту ж тему. Нижче детальніше представлено вибір різних видатних лінгвістичних підходів. Це вибір старих, перевірених методів та деяких нових передових методів.

2.4.1 Метод опорних векторів (SVM)

Метод опорних векторів (Support Vector Machines) — це класифікатор, який працює шляхом розділення гіперплощини (n-вимірного простору), що

містить вхідні дані. Він базується на теорії статистичного навчання. За наявності мічених навчальних даних алгоритм виводить оптимальну гіперплощину, яка класифікує нові приклади. Оптимальна гіперплощина обчислюється шляхом знаходження роздільника, який мінімізує чутливість до шуму та максимізує узагальнення і маржу моделі[21]. Унікальною рисою SVM є те, що підхід за допомогою гіперплощини базується виключно на точках даних, які називаються опорними векторами. Одним з основних недоліків SVM є те, що він може працювати лише з міченими даними, а отже, працює тільки в режимі навчання з наглядом. SVM не обмежується лінійним розділенням, оскільки вони можуть перетворювати вхідні дані в простір ознак високої вимірності, де можна знайти роздільну гіперплощину, яка буде оптимальним класифікатором. Однією з переваг SVM є те, що їх можна використовувати для дуже об'ємних задач, за умови, що їхні ознаки можна лінійно відобразити в просторі ознак. Нелінійне використання SVM застосовує так званий ядровий трюк (англ. kernel trick). Ядровий трюк працює, замінюючи частини оригінальних алгоритмів на ядрову функцію замість функції добутку. Ядрові методи можуть працювати у високорозмірних просторах, оскільки вони обчислюють внутрішні добутки між даними в просторі замість використання координат даних. Варто також зазначити, що простори ознак вищих вимірів збільшують помилку узагальнення, але при достатній кількості зразків вони все одно показують хороші результати.

2.4.2 Naive Bayes

Naive Bayes — це родина лінійних класифікаторів, які працюють шляхом використання взаємно незалежних ознак у наборі даних для класифікації[10]. Він відомий своєю простотою реалізації, надійністю, швидкістю та точністю. Широко використовується для завдань класифікації, таких як діагностика хвороб та фільтрація спаму в електронній пошті. Якщо

його використовувати в системах, де ознаки сильно взаємозалежні, продуктивність зазвичай знижується.

$$posteriorprobability = \frac{conditionalprobability * priorprobability}{evidence}$$

Naive Bayes ґрунтується на ймовірнісному правилі Байєса, яке показано вище і може бути інтерпретоване як ймовірність того, що об'єкт належить до певного класу, зважаючи на ознаки, які він має. Окрім розпізнавання шаблонів, Naive Bayes також можна використовувати для класифікації тексту, представляючи текст у вигляді ряду ознак. Класифікатори Naive Bayes використовуються в багатьох різних сферах, зокрема для діагностики хвороб і прийняття рішень щодо лікування, класифікації послідовностей РНК в таксономічних дослідженнях та фільтрації спаму в електронних поштових клієнтах[10]. Наївна частина Naive Bayes полягає в припущенні, що змінні незалежні і однаково розподілені. Це означає, що змінні, що використовуються для класифікації, всі отримані з подібних ймовірнісних розподілів. Незалежність означає, що ймовірність одного результату не впливає на інші результати. Кидання монети є гарним прикладом незалежного і однаково розподіленого набору. Один результат не впливає на інший, і обидві змінні мають рівний ймовірнісний розподіл.

2.4.3 Частотний коефіцієнт термінів (TF-IDF)

Частотний коефіцієнт термінів — зворотна частота документа (TF-IDF) — це значення вагомості, яке часто використовується в пошуку інформації та дає статистичний показник для оцінки важливості слова в колекції документів або корпусі. В принципі, важливість слова збільшується пропорційно кількості разів, коли воно з'являється в документі, але компенсується частотою цього слова в колекції або корпусі. Отже, слово, яке зустрічається дуже часто,

матиме низький коефіцієнт впливу, тоді як інші менш використовувані слова будуть мати більшу значущість.

$$\mathrm{TF}(t,d) = \frac{\text{Кількість появ терміна } t \text{ в документі } d}{\text{Загальна кількість термінів у документі } d}$$

Рис. 3. Формула обчислення TF

Частота терміна — це частота появи терміна в документі. Документ тут — це окремий шматок інформації, наприклад, пост у Facebook, повідомлення в Twitter або навіть новинна стаття. Частота часто зростає в довших документах і зазвичай ділиться на довжину документа, якщо колекція складається з документів різного розміру, як спосіб нормалізації значень. Нарешті, зворотна частота документа (IDF) вимірює важливість терміна.

$$\mathrm{IDF}(t,D) = \log \frac{\text{Загальна кількість документів } D}{\text{Кількість документів, що містять термін } t}$$

Рис. 4. Формула обчислення IDF

$$\text{TF-IDF}(t,d,D) = \mathrm{TF}(t,d) \times \mathrm{IDF}(t,D)$$

Рис. 5. Формула обчислення TF-IDF

Поки частота терміна не робить розрізнення між термінами, частина IDF знає, що слова, які часто зустрічаються, зазвичай не додають великої якості колекції документів, і знижує їх вагу, в той час як рідкісні терміни збільшуються.

TF-IDF векторизація є потужним інструментом для роботи з текстовими даними, оскільки дозволяє враховувати як частоту термінів у документі, так і їх інформативність у контексті всього корпусу. Використання TF-IDF є

особливо ефективним для задач класифікації текстів та визначення важливих термінів у документах.

До таких широко використовуваних слів можуть належати стоп-слова, слова, отримані за допомогою стеммінгу та інших інструментів попередньої обробки.

2.4.4 N-грами

N-грами — це підрядки довжиною n символів у довшому рядку, але також можуть стосуватися n слів. N-грама з величиною один називається уніграмою, два — біграмою, три — тріграма і так далі. N-грами можуть використовуватися для поділу текстів на різні частини, і, використовуючи пробіли, можна також визначити, що є словами, а що — ні. N-грами частково базуються на Законі Ципфа, який можна сформулювати так: "N-те за поширеністю слово в тексті людської мови з’являється з частотою, обернено пропорційною до n." Це означає, що деякі слова будуть домінувати над іншими, і що n-грами матимуть таке ж розподілення, а документи також будуть мати однакові розподіли, і ми можемо порівнювати їх[3]. N-грами використовувались у багатьох різних застосунках з моменту їх першого запропонування, включаючи виявлення коду[1], оцінку підсумків[13] та автоматичну оцінку якості машинного перекладу[6]. Потужність N-грам як інструменту обробки природної мови є майже безпрецедентною, і її можна використовувати для майже будь-якого завдання, пов’язаного з текстом, у тих випадках, коли текст належить до домену, в якому є певний розподіл частоти.

2.4.5. Сентимент-аналіз

Сентимент-аналіз (англ. Sentiment analysis) — це інструмент обробки мови, що має на меті виявити приховані точки зору в тексті. Він намагається класифікувати полярність сентименту, що є виміром того, чи є текст позитивним чи негативним відносно чогось[23]. Це може бути зроблено на основі певного корпусу з додаванням ваг до слів або N-грам слів або навіть перевірки, чи є певні тексти "за" чи "проти" чогось. Останнє вимагає значної спеціалізованої роботи, щоб бути корисним, але це абсолютно можливо. Сентимент-аналіз може використовувати ресурси з глибокими лінгвістичними знаннями щодо індикаторів сентименту, тим самим будуючи на існуючих знаннях про мову. Сентимент-аналіз намагається витягти думки з тексту і, таким чином, бути використаним для подальшої оцінки чогось. Це щось може бути, наприклад, системами рекомендацій або редакційними сайтами, що намагаються створювати підсумки або давати рекомендації користувачам. Сентимент-аналіз є складним завданням для комп’ютерів, оскільки вони в якийсь спосіб повинні зрозуміти думки автора тексту. Це вимагає розуміння того, як користувач використовує певні слова. Оскільки люди використовують слова по-різному, а мови значно різняться за граматикою та синтаксисом, сентимент-аналіз вимагає величезних зусиль для того, щоб бути корисним, таких як корпус, що містить ваги для більшості слів, що використовуються в тій чи іншій мові[15]. Сентименти зазвичай можна витягти з думок, оскільки вони містять суб'єктивну інформацію. Думки, як правило, складаються з двох основних компонентів: об'єкта та сентименту щодо об'єкта. Об'єкт може бути будь-яким видом сутності, від осіб до подій до продуктів. Таким чином, можна сказати, що мета сентимент-аналізу — витягти всі думки з даного тексту[15]. Сентимент-аналіз можна застосовувати до більшості письмових джерел, включаючи, але не обмежуючись, Facebook, Twitter, блоги, відгуки та обговорення. Його також можна використовувати як інструмент для організацій для збору громадських думок замість опитувань і голосувань. Сирі

дані, які вони мають на різних платформах, можна використовувати для створення базового рівня через використання неупереджених даних. Галузі, де сентимент-аналіз використовувався останнім часом, включають охорону здоров'я, фінансові послуги та політичні вибори. Дослідження в галузі сентимент-аналізу включають прогнозування ефективності продажів, відгуки для ранжування продуктів, сентименти в Twitter у порівнянні з громадською думкою та багато інших[15].

2.5 Контекстуальний підхід

Контекстуальні підходи включають більшість інформації, яка не є текстом. Це включає дані про користувачів, такі як коментарі, лайки, ретвіти, шари та інше. Це також можуть бути відомості щодо походження, як хто створив інформацію і де вона була вперше опублікована. Така інформація має більш прогностичний підхід, ніж лінгвістичний, де ви можете бути більш детермінованим. Контекстуальні підказки дають хороше уявлення про те, як інформація використовується, і на основі цього можна робити припущення. Цей підхід спирається на структуровані дані для того, щоб робити ці припущення, і через це сфера застосування наразі обмежена соціальними мережами через обсяг публічної інформації, яка там доступна. Ви маєте доступ до публікаторів, реакцій, походження, шерингу та навіть віку постів. Крім того, контекстуальні системи найчастіше використовуються для покращення якості існуючої інформації та доповнення лінгвістичних систем, надаючи більше інформації для роботи цих систем, наприклад, репутації, метрик довіри або інших способів надання індикаторів того, чи є інформація статистично схильною до фальшивості чи ні. Нижче представлено низку контекстуальних методів. Вони є комбінацією сучасних методів та старих, перевірених методів.

2.5.1 Логістична регресія

Логістична регресія (LR) — це регресійний аналіз, який працює, коли залежна змінна є бінарною. Це прогностичний аналіз, що використовується для пояснення взаємозв'язку між однією залежною бінарною змінною та іншими незалежними змінними. Логістичну регресію можна використовувати в ситуаціях, де постає питання так/ні, наприклад, чи вважається пост у Facebook фальшивим чи ні. Її можна розглядати як особливий випадок лінійної моделі, і вона належить до тієї ж родини, що й лінійна регресія. Основні відмінності полягають у тому, що LR використовує розподіл Бернуллі замість розподілу Гауса та що результат є ймовірністю. LR моделює ймовірність результату на основі індивідуальних результатів, і результат, який буде отримано, потрапить у межі прийняття рішення. Використовуючи межі прийняття рішення, які визначають, яку класифікацію отримає результат, ми отримуємо результат від алгоритму, що є або True, або False. Логістична регресія базується на центральній математичній концепції логіту, натурального логарифму відношення шансів. Логістична регресія добре підходить для опису зв'язків між категоріальними результатами, такими як класифікація інформації як неправдивої чи ні[25]. Найпростіший випадок лінійної регресії має лише один предиктор і одну бінарну змінну результату. Це може бути сформульовано таким чином:

$$logit(Y) = \log_{\epsilon}(\frac{\pi}{1-\pi}) = \alpha + \beta X$$

Рис. 6. Логістична регресія

Де Y — це результат бінарної змінної, α — це перетин з віссю Y, β — це коефіцієнт регресії, а X — це предиктор. Логістична регресія широко

використовується в соціальних науках для вивчення результатів, таких як підвищення по службі, розлучення, медичні діагнози, безробіття та політичне голосування.

2.5.2 Мережевий аналіз

Мережевий аналіз є потужним інструментом для роботи з графами. Розуміння зв'язків і взаємодій між різними вершинами є важливим для того, щоб отримувати непрямі результати з таких мережевих додатків, як Facebook, Twitter або інших мережевих платформ. Аналіз можна використовувати для прогнозування поведінкових моделей у популяції, що ділиться різноманітною інформацією, такою як спільноти, лайки, друзі тощо. Соціальна поведінка серед людей свідчить, що спільні інтереси, як правило, призводять до схожої поведінки, такої як віра в фейкові новини. Мережевий підхід наголошує на структурі взаємозв'язків, а не на атрибутах окремих учасників. Необхідність статистично значущої популяції залишається першочерговою, і це те, чому ця дисертація присвячена дослідженню[11]. Мережевий аналіз успішно застосовувався для покращення задоволення клієнтів[7], для розуміння того, як учні співпрацюють[4], для аналізу атрибутів користувачів і поведінки, аналізу взаємодій, прогнозування зв'язків і розробки рекомендательних систем, щоб згадати лише кілька напрямів. Крім того, мережевий аналіз застосовується на людському рівні в контексті розвідки, контррозвідки та правоохоронної діяльності. Розуміння того, як популяція або певні її групи реагують на певний вплив, є важливим для коректного реагування. Так само мережевий аналіз можна застосовувати до будь-якої колекції даних, де елементи мають взаємозв'язки, наприклад, до текстових корпусів, досліджуючи використання слів, відносини між словами тощо.

2.5.3 Система репутації, орієнтована на контент

Системи репутації працюють, надаючи користувачам можливість оцінювати один одного на основі їхніх дій. Системи репутації варіюються від загальносайтових репутацій, як на Amazon, до сайтів відгуків, де кожен відгук має свою власну репутацію незалежно від користувача. Системи репутації допомагають іншим користувачам знаходити реальні відгуки про продавців і товари. Однією з основних проблем систем репутації є те, що вони базуються на суб'єктивних відгуках. Людям надзвичайно важко бути повністю об'єктивними, і тому репутація, яку надають продукту, продавцю, людині або іншій сутності, буде забарвлена на основі думок інших, а не на об'єктивних показниках. Для спільноти, орієнтованої на користувачів, як Wikipedia, де весь контент створюється людьми для людей, важливо, щоб контент був як правильним, так і відповідав певним стандартам. Якщо якість контенту знижується з часом, користувачі помітять це, і сайт може втратити репутацію та користувачів. Для боротьби з цим можна впровадити автоматизовану систему репутації, орієнтовану на контент, яка об'єктивно вимірює довіру та репутацію авторів. Автори оцінюються на основі їхнього внеску на сторінці. Замість того, щоб користувачі ставили іншим користувачам оцінки "плюс" чи "мінус", залишаючи редагування, вони опосередковано надають їм голос довіри, що поступово будує репутацію авторів, редагування за редагуванням. Оцінка змінюється відповідно до репутації авторів, які перевіряють ці редагування. Таким чином, автори або користувачі не можуть завдати шкоди репутації інших, просто вставляючи негативні коментарі або оцінки. Натомість вони повинні видалити зроблене редагування, потім замінити або змінити його на щось інше, ризикуючи, що інші автори відновлять оригінальне редагування, і вся робота буде марною[2]. Поняття репутації має хорошу прогнозову цінність, оскільки зміни, виконані авторами з низькою репутацією, мають вищу ймовірність бути низької якості, ніж зміни, виконані авторами з високою репутацією. Однак, як і в багатьох завданнях NLP, система репутації

обмежена через недостатнє розуміння мови розмітки, що використовувалася[2], і тому текстовий аналіз є гіршої якості, ніж мав би бути. Також слід зазначити, що системи репутації можна поділити на дві категорії: хронологічні, де репутація обчислюється на основі хронологічної послідовності оцінок, та фіксовані, де репутація обчислюється по всій кількості відгуків без урахування часової інформації.

2.6 Видобуток інформації

Видобуток інформації — це автоматизований процес видобутку структурованої інформації, такої як сутності, взаємозв'язки між сутностями та атрибути, що описують сутності, з неструктурованих джерел[12]. Це схоже на те, як знання графи використовуються в пошукових системах. Видобуток інформації включає витягування структур за допомогою машинного навчання, інформаційного пошуку, баз даних, веб- та документального аналізу. Видобуток інформації може бути використаний у багатьох різних застосунках, таких як відстеження новин. Відстеження новин полягає в автоматичному відстежуванні конкретних сутностей із новинних джерел, таких як події, особи або навіть журналісти. Іншим застосунком є очищення даних, де сховища даних повинні зберігати дані у певних форматах і забезпечувати їх сумісність. Оскільки інформація надходить з різних джерел, дані повинні бути зрозумілими, можливо, трансформованими і зрештою вставленими в сховище для подальшого зберігання та обробки, а також аналізу.

2.6.1 Видобуток джерела

Видобуток джерела включає витягування походження даних або визначення того, з якого типу джерела була зібрана інформація. Отримання походження джерела не є темою, що викликала значний інтерес у науковій

спільноті, оскільки більшість робіт були зосереджені на частині видобутку, що стосується обробки природної мови (NLP), де інформація є значно більш практичною, доступною і визначеною. У контекстуальному сенсі джерело може бути використане як точка походження, з якої можна отримати додаткову інформацію, наприклад, якщо є новинне агентство, зазначене як джерело, можна створити знання граф із них на основі трафіку, який вони отримують. Більшість робіт з видобутку джерела зосереджено на тілі та заголовках текстів і базується більше на сутностях, ключових словах і подіях. Видобуток джерела за типами джерел можна поділити на структуровані та неструктуровані джерела. Структуровані джерела вже мають логічний порядок і можуть бути легко витягнуті. Неструктуровані джерела, однак, можуть варіюватися за ступенем складності[12]. Вони часто доповнюються існуючими структурованими джерелами, такими як бази даних, вже промарковані неструктуровані тексти або використання бібліотек знань. Найпоширенішою формою неструктурованих даних є маленькі текстові фрагменти або записи, оскільки їх простота і мінімальна довжина знижують складність. Інші неструктуровані джерела — це абзаци та документи, які часто потрібні для того, щоб побачити загальну картину в тексті та зрозуміти контекст елементів тексту. Для більш неясних джерел точність видобувача сильно залежить від однорідності стилю та формату документа. Машинно згенеровані сторінки часто мають чітко визначену структуру, як HTML і XML документи. З іншого боку, є відкриті джерела. Відкриті джерела є одними з найбільш складних для обробки, оскільки вони не належать до жодної конкретної області, наприклад, медицини чи астрономії, і тому важко створити систему, яка легко зрозуміє зміст. Джерела, такі як Інтернет, є відкритими, і при видобутку інформації з них існує висока ймовірність того, що буде видобуто надлишкову інформацію, але витягнуту по-різному[12].

2.6.2 Видобуток сутностей

Видобуток сутностей — це процес, коли система видобутку здатна витягувати інформацію, що пов’язана з сутністю[12]. Сутність — це будь-який тип об'єкта, який може мати зв'язки, риси та іншу інформацію, що зберігається з ним. Це можуть бути люди, місця, компанії або продукти, а також майже будь-що інше. Видобуток сутностей також можна розглядати як сегментацію текстового запису на структуровані сутності, як місток між неструктурованими і структурованими даними. Видобуток іменованих сутностей — це завдання, де в тексті розпізнаються імена, такі як люди, організації та місця. Ці сутності явно вказані в тексті, і їх легше видобути, ніж більш абстрактні сутності, які можуть бути сховані в тексті, і це було темою багатьох досліджень. Це включає алгоритми на основі правил і системи машинного навчання, які використовують безліч різних методів функцій і оцінки. Однією з проблем у видобутку іменованих сутностей є розпізнавання, що одна сутність є тим самим, що й інша. Наприклад, автомобільна компанія, заснована Генрі Фордом у 1903 році, може називатися як Ford, так і Ford Motor Company[20]. Включення онтологій або інших систем знань може мінімізувати цю проблему, але вона все ще залишається викликом, особливо в відкритих доменах, де можуть виникати проблеми з надмірністю та подібними назвами сутностей.

2.6.3 Видобуток ключових слів

Видобуток ключових слів є важливою частиною видобутку інформації, особливо в домені пошуку, де розуміння основних аспектів тексту є вирішальним. Це автоматичне визначення набору термінів, які найкраще описують тему документа або колекції. Видобуток правильних ключових слів дасть значно кращу точність і релевантність для пошукової системи, а також допоможе визначити, які документи є взаємопов’язаними[16]. TF-IDF — це

один з алгоритмів, який широко використовується в видобутку ключових слів, оскільки він підкреслює частоту слів, нормовану по корпусу, в рейтинговому порядку. Він не вимагає великої кількості вхідних даних і є відносно незалежним від домену. TF-IDF є прикладом в основному незамкнутого методу для видобутку ключових слів, але також доступні контрольовані та напівконтрольовані методи. Іншим відомим методом видобутку ключових слів є TextRank[18]. TextRank — це графічна модель ранжування для обробки текстів. Видобуток ключових слів за допомогою TextRank ґрунтується на незамкнутому навчанні і на зв'язках співпадінь. Спочатку текст токенізується і анотовану частинами мови. Тільки окремі слова вважаються кандидатами для додавання до графа TextRank. Потім усі лексичні одиниці, які проходять синтаксичний фільтр, додаються до графа, і ті, що співпадають в межах вікна з N словами, отримують оцінку, яка визначається за допомогою ранжування протягом 20-30 ітерацій. На етапі постобробки всі кандидати на ключові слова згортаються в багатослівні ключові слова.

3. РОЗРОБКА І РЕАЛІЗАЦІЯ СИСТЕМИ ВИЯВЛЕННЯ ДЕЗІНФОРМАЦІЇ

Для розробки системи виявлення дезінформації, а також дослідження підходів його реалізації я обрав традиційний спосіб. Він передбачає використання класичних алгоритмів машинного навчання, таких як Logistic Regression, Random Forest, Decision Tree та інших. Цей підхід полягає в перетворенні текстових даних у структурований числовий формат, наприклад, за допомогою TF-IDF, і подальшому навчанні моделей на цих ознаках.

Переваги традиційного підходу включають простоту реалізації, відносно невисокі вимоги до обчислювальних ресурсів і здатність забезпечувати зрозумілі результати. Він дозволяє швидко експериментувати з різними моделями та налаштуваннями, а також отримувати результати, які легко інтерпретувати, що є важливим для аналізу та пояснення результатів класифікації.

Рис. 7. Підхід до виявлення фейкових новин

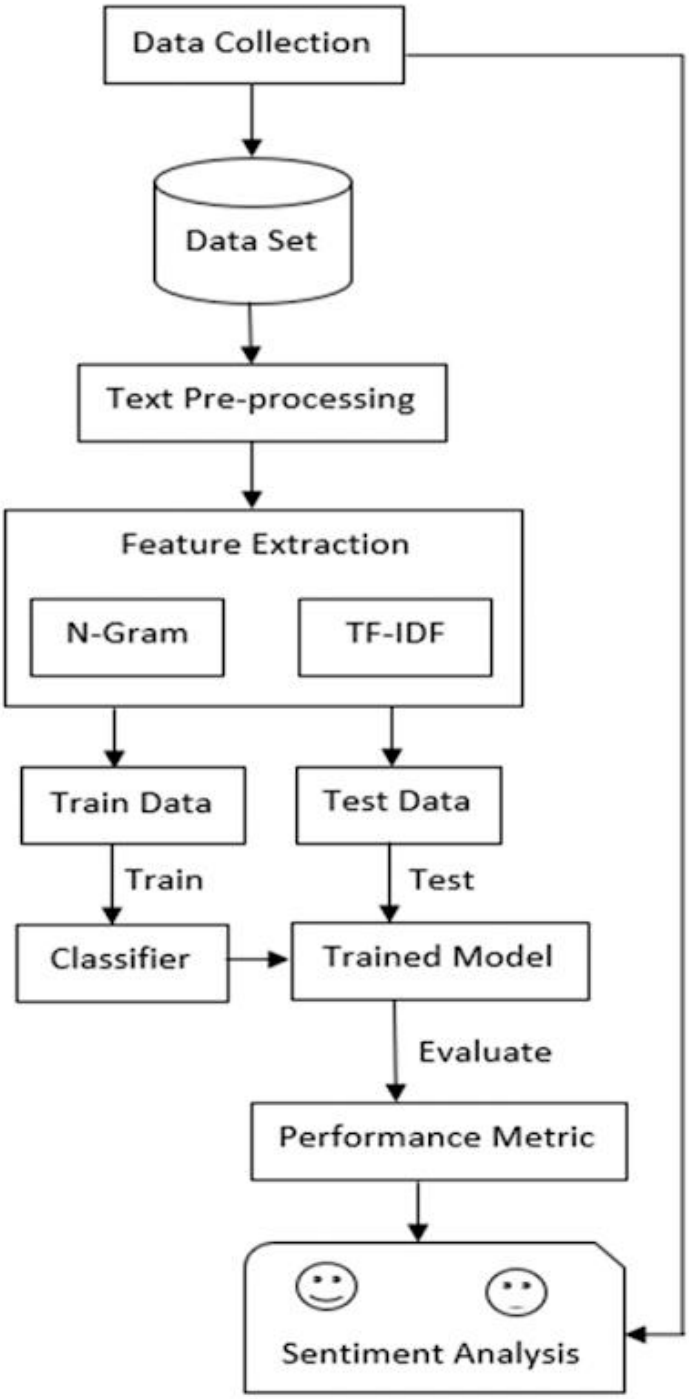


Рис. 8. Підхід до виявлення фейкових новин

3.1. Опис набору даних для навчання та тестування моделі

Збір даних є першим і ключовим етапом в процесі розробки системи виявлення дезінформації в потоках текстових даних. Цей етап включає в себе дослідження та накопичення великого обсягу текстової інформації з різних джерел, таких як новинні агентства, соціальні мережі, блоги, форуми тощо.

Важливо обрати джерела, які надають достовірну та репрезентативну інформацію для дослідження. Це можуть бути національні та міжнародні новинні сайти, акаунти в соціальних мережах з високою активністю користувачів, форуми з актуальними обговореннями тощо.

Для збору даних можна використовувати веб-скрапінг або API, що надають доступ до публічної інформації. Важливо враховувати можливість автоматизації цього процесу для ефективного збору великого обсягу даних.

Важливо збирати не лише текстові дані, але й метадані, які допомагають розуміти контекст і джерело інформації. Це може включати дату публікації, автора, тему, категорію новин тощо. Контекстуалізація даних дозволяє забезпечити правильне їх інтерпретування та аналіз у подальших етапах.

Після збору і обробки дані зберігаються у структурованому форматі, що відповідає вимогам подальшого аналізу та обробки. Важливо забезпечити доступність даних для використання в моделях машинного навчання та аналітичних інструментах.

Загальний успіх системи виявлення дезінформації значною мірою залежить від якості та репрезентативності зібраних даних, тому цей етап вимагає уваги до деталей та систематичного підходу.

Тому для точності дослідження я обрав готовий збалансований датасет.

Рис. 9. Ворд клауд опрацьованого датасету

Датасет містить текстові дані новинних статей та їхні характеристики, що використовуються для навчання моделей виявлення дезінформації. Основні атрибути датасету:

- id: Унікальний ідентифікатор кожної статті, що дозволяє відрізняти її від інших у наборі даних.
- title: Заголовок новинної статті, який є важливим елементом для аналізу, оскільки заголовки часто використовуються для привернення уваги та можуть містити ознаки маніпуляцій.
- author: Автор статті. Ця характеристика може бути корисною для аналізу джерел інформації, якщо автор має історію недостовірних публікацій.
- text: Текст статті, який може бути частково або повністю представленим. Це основне джерело інформації для аналізу, яке обробляється за допомогою лінгвістичних та машинних методів.
- label: Мітка, що позначає рівень достовірності статті:
    - 1: Ненадійна стаття.
    - 0: Надійна стаття.

Цей набір даних використовується для тренування моделей класифікації, які аналізують заголовки, тексти та інші характеристики, щоб визначати, чи є стаття потенційно недостовірною.

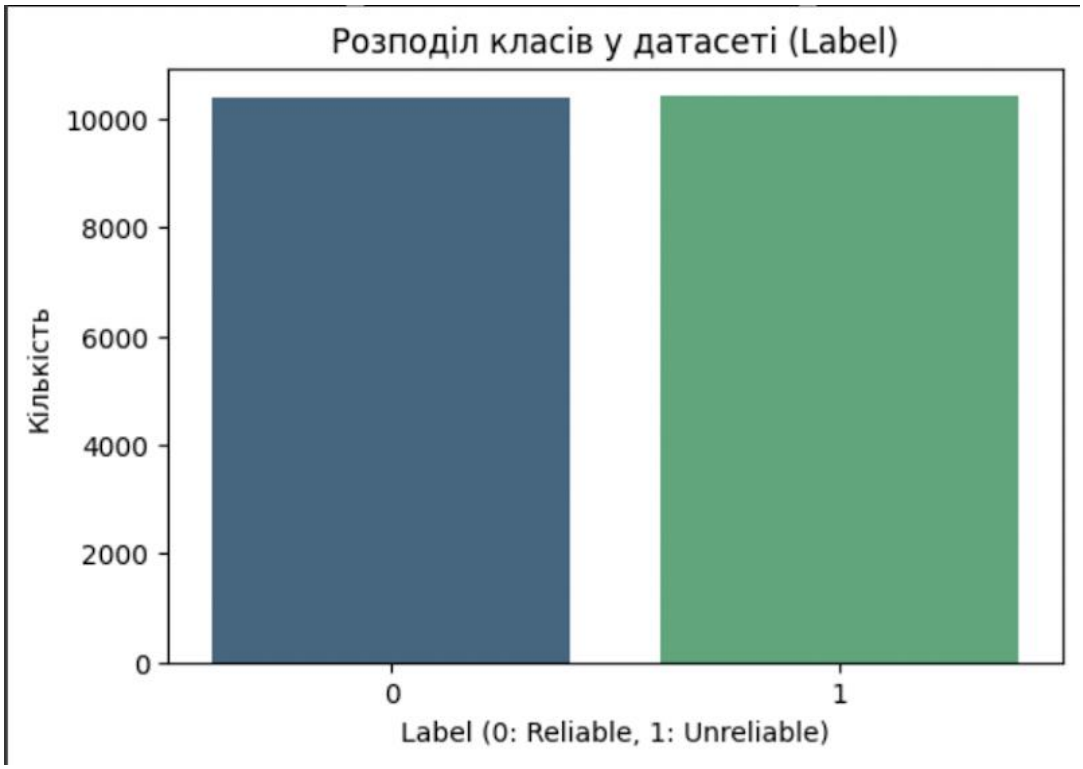


Рис. 10. Збалансований датасет правдивих і неправдивих текстів

3.2. Попередня обробка даних

Попередня обробка тексту є критично важливим етапом в аналізі текстових даних для системи виявлення дезінформації. Цей процес включає в себе низку кроків, спрямованих на очищення текстів від зайвих елементів та підготовку їх для подальшої обробки та аналізу. Ось основні аспекти попередньої обробки тексту:

Видалення непотрібних символів і знаків пунктуації. Тексти можуть містити надлишкові символи, які не несуть семантичного навантаження (наприклад, спеціальні символи, числа, знаки пунктуації). Вони можуть бути

видалені або замінені на пробіли для збереження лише значущих компонентів тексту.

Токенізація - процес розбиття тексту на окремі слова або токени. Це може включати розділення тексту на окремі слова за пробілами або за допомогою спеціалізованих інструментів, які враховують особливості мови.

Видалення стоп-слів. Стоп-слова є частотними словами (такими як "і", "в", "на", "для"), які зазвичай не несуть значущого семантичного навантаження для аналізу тексту. Вони можуть бути видалені з тексту для зменшення впливу шуму на моделі.

Лематизація або стемінг. Ці процеси використовуються для зведення словоформ до їх базових форм (лем), що допомагає зменшити кількість унікальних слів у тексті та покращує узагальнення моделей. Лематизація зазвичай враховує граматичні правила мови, в той час як стемінг зменшує слова до їх кореневих частин.

Нормалізація тексту. Включає процеси нормалізації, такі як перетворення тексту до нижнього регістру або обробка сленгу та абревіатур для забезпечення однорідного формату текстових даних.

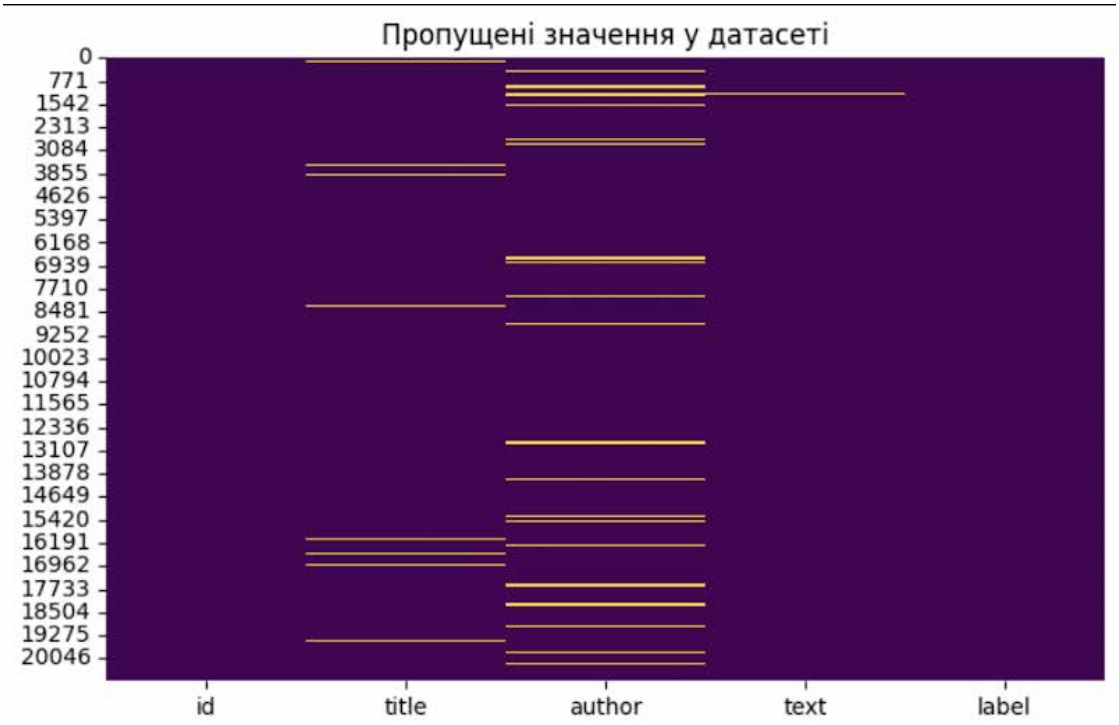
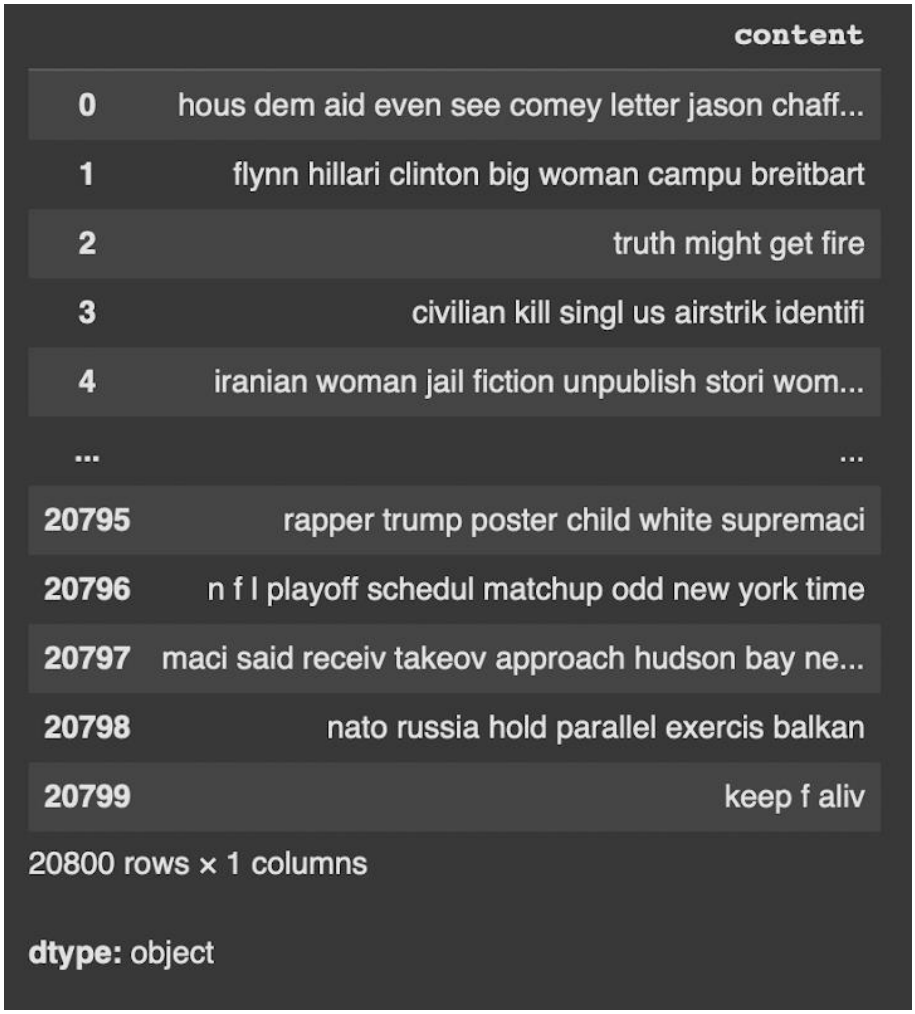


Рис. 11. Пропущені значення у датасеті

```
i
me
my
myself
we
our
ours
ourselves
you
you're
you've
you'll
you'd
your
yours
yourself
yourselves
he
him
his
himself
she
she's
```

Рис. 12. Приклад стоп-слів англійською



	content
0	hous dem aid even see comey letter jason chaff...
1	flynn hillari clinton big woman campu breitbart
2	truth might get fire
3	civilian kill singl us airstrik identifi
4	iranian woman jail fiction unpublish stori wom...
...	...
20795	rapper trump poster child white supremaci
20796	n f l playoff schedul matchup odd new york time
20797	maci said receiv takeov approach hudson bay ne...
20798	nato russia hold parallel exercis balkan
20799	keep f aliv

20800 rows × 1 columns

dtype: object

Рис. 13. Дані після стемінгу

3.3. Векторизація тексту TF-IDF

Про векторизацію ми говорили вище у роботі. Зараз саме час його застосувати. У цьому нам допоможе інструмент TfidfVectorizer.

```
tfidf = TfidfVectorizer(stop_words='english', max_features=5000)
X_train_tfidf = tfidf.fit_transform(X_train)
X_test_tfidf = tfidf.transform(X_test)
```

3.4. Вибір моделі для виявлення дезінформації

У роботі було використано широкий спектр моделей машинного навчання, які дозволяють реалізувати різні підходи до задачі класифікації

текстів. Кожна з цих моделей має свої особливості, переваги та недоліки, що робить їх цікавими для аналізу у контексті нашого завдання.

Логістична регресія — це одна з найбільш поширених моделей для бінарної класифікації. Вона базується на лінійному підході, використовуючи сигмоїдну функцію для перетворення вхідних даних у ймовірності належності до певного класу. Логістична регресія є простою у реалізації та швидкою, що робить її ефективною для великих наборів даних. Завдяки регуляризації вона добре справляється із завданнями, де важлива інтерпретація результатів.

Random Forest є ансамблевою моделлю, яка складається з множини рішучих дерев. Кожне дерево створюється з випадкової вибірки даних і вибірки ознак, а остаточне рішення приймається голосуванням. Цей підхід дозволяє уникати перенавчання, роблячи модель стійкою до шумів у даних. Random Forest добре працює з нелінійними залежностями і є одним із найпопулярніших алгоритмів для задач класифікації та регресії.

Градієнтний бустинг є ще одним ансамблевим методом, який поступово створює нові моделі для виправлення помилок попередніх. Кожна наступна модель зосереджується на тих даних, які були класифіковані з похибкою. Gradient Boosting забезпечує високу точність, але є більш ресурсозатратним у порівнянні з іншими моделями, що може бути недоліком при роботі з великими обсягами даних.

Linear SVM (Лінійний метод опорних векторів) використовує гіперплощину для поділу класів у просторі ознак. Цей метод ефективний для даних із великою кількістю вимірів, наприклад, для текстової класифікації після векторизації. Linear SVM забезпечує високу точність і є стійким до перенавчання, однак його продуктивність залежить від параметрів і налаштувань, зокрема від вибору ядра.

Naive Bayes базується на ймовірнісному підході і припускає незалежність ознак. Незважаючи на наївність припущення, цей метод є надзвичайно швидким і обчислювально ефективним, що робить його

ідеальним для класифікації текстів. Naive Bayes часто використовується як базова модель через його простоту і здатність справлятися з малими обсягами даних.

KNN (k-Nearest Neighbors). Алгоритм класифікує нові дані на основі їх схожості з найближчими зразками з навчальної вибірки. Це неконструктивний метод, який не створює моделі під час навчання, а зберігає всі дані. Хоча KNN є простим у реалізації, його недоліком є висока обчислювальна складність для великих наборів даних.

Decision Tree (Дерево рішень) створює ієрархію рішень на основі умов. Цей підхід забезпечує простоту інтерпретації результатів і гнучкість у виборі глибини дерева. Однак окремі дерева схильні до перенавчання, тому вони часто використовуються в ансамблевих моделях, таких як Random Forest або Gradient Boosting.

Ridge Classifier (Рідж-класифікатор) — це модель лінійної регресії, яка використовує регуляризацію для боротьби з перенавчанням. Регуляризація L2 дозволяє моделі добре працювати навіть у випадках, коли дані мають багато корельованих ознак. Ridge Classifier є швидким і ефективним для задач із лінійно роздільними класами.

3.5. Навчання моделі

Після того, як ми обрали моделі для нашого проекту, наступним кроком є підготовка даних для навчання. Це важливий етап, адже якість даних безпосередньо впливає на ефективність моделі. Ось основні етапи, які ми виконали після вибору моделей: Першим кроком було поділ даних на тренувальні та тестові. Ми скористалися функцією train_test_split, щоб поділити наш набір даних на дві частини: тренувальну, на якій модель буде навчатися, і тестову, на якій ми оцінюватимемо її точність. Це дозволяє нам уникнути перенавчання та перевірити, як модель працює з новими, невідомими даними.

3.6. Оцінка ефективності моделі: метрики точності, recall, F1-міра

Оцінка моделей класифікації включає кілька ключових метрик, які допомагають оцінити їхню ефективність.

Точність (Precision) визначає, яка частина об'єктів, визначених моделлю як позитивні, є дійсно позитивними.

$$\text{Precision} = \frac{TP}{TP+FP}$$

Рис. 14. Формула Точність (Precision)

Де TP - кількість правильно передбачених позитивних випадків, FP - кількість неправильно передбачених позитивних випадків.

Відновлення (Recall або Sensitivity) показує, яка частина дійсних позитивних об'єктів була виявлена моделлю.

$$\text{Recall} = \frac{TP}{TP+FN}$$

Рис. 15. Формула Відновлення (Recall або Sensitivity)

Де FN - кількість неправильно передбачених негативних випадків.

F1-оцінка (F1 score) є гармонійним середнім між точністю і відновленням і надає збалансовану оцінку моделі

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}.$$

Рис. 16. Формула F1-оцінка (F1 score)

Точність передбачення (Accuracy) визначає загальну правильність класифікації моделі, ділячи кількість правильно класифікованих випадків на загальну кількість випадків.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$$

Рис. 17. Формула Точність передбачення (Accuracy)

Де TN - кількість правильно передбачених негативних випадків.

Кожен з цих кроків важливий для успішної розробки системи інтелектуального аналізу, зокрема для системи виявлення дезінформації в текстових даних.

У нашому випадку моделі показали такі результати:

Навчання моделі: Logistic Regression

Середній результат крос-валідації для Logistic Regression: 0.9421

Logistic Regression: Accuracy=0.9421, Precision=0.9348, Recall=0.9506, F1 Score=0.9426

```
--- Logistic Regression ---
Overall Accuracy: 0.9421
Precision Score: 0.9348
Recall Score: 0.9506
AUC: 0.9879

Classification Report:
              precision    recall  f1-score   support

    Reliable       0.95      0.93      0.94      2077
  Unreliable       0.93      0.95      0.94      2083

    accuracy                           0.94      4160
   macro avg       0.94      0.94      0.94      4160
weighted avg       0.94      0.94      0.94      4160
```

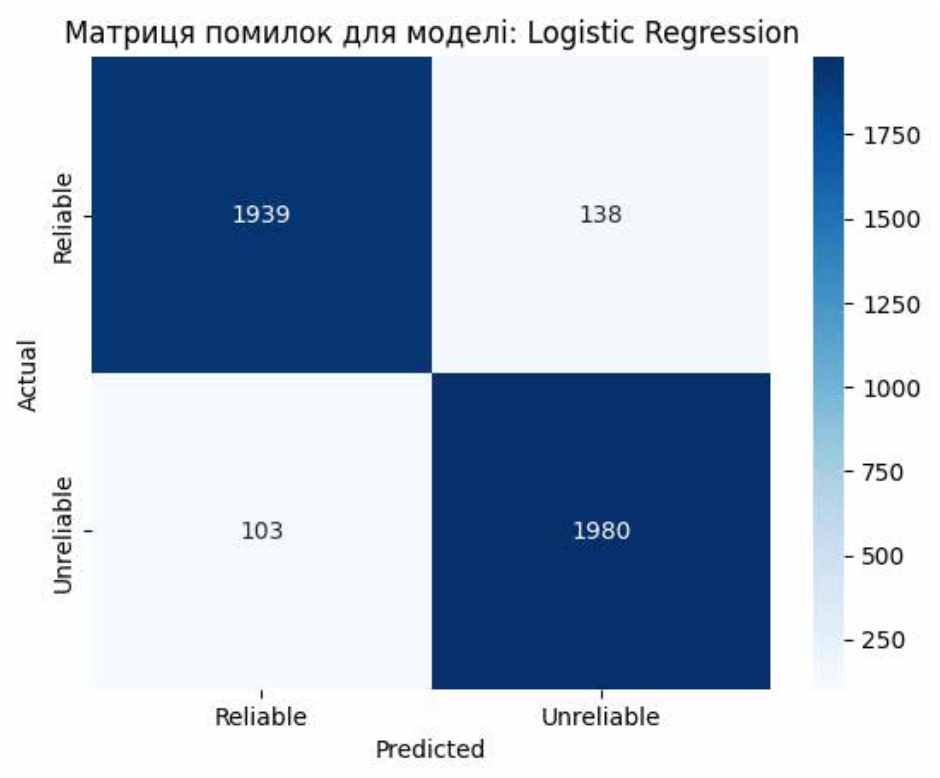


Рис. 18-19. Метрики і матриця моделі Logistic Regression

Навчання моделі: Random Forest

Середній результат крос-валідації для Random Forest: 0.9442

Random Forest: Accuracy=0.9442, Precision=0.9452, Recall=0.9434, F1 Score=0.9443

```
--- Random Forest ---
Overall Accuracy: 0.9442
Precision Score: 0.9452
Recall Score: 0.9434
AUC: 0.9875

Classification Report:
              precision    recall  f1-score   support

    Reliable       0.94      0.95      0.94      2077
  Unreliable       0.95      0.94      0.94      2083

    accuracy                           0.94      4160
   macro avg       0.94      0.94      0.94      4160
weighted avg       0.94      0.94      0.94      4160
```

Рис. 20-21. Метрики і матриця моделі Random Forest

Навчання моделі: Linear SVM

Середній результат крос-валідації для Linear SVM: 0.9513

Linear SVM: Accuracy=0.9510, Precision=0.9502, Recall=0.9520, F1 Score=0.9511

```
--- Linear SVM ---
Overall Accuracy: 0.9510
Precision Score: 0.9502
Recall Score: 0.9520
AUC: 0.9904

Classification Report:
              precision    recall  f1-score   support

    Reliable       0.95      0.95      0.95      2077
  Unreliable       0.95      0.95      0.95      2083

    accuracy                           0.95      4160
   macro avg       0.95      0.95      0.95      4160
weighted avg       0.95      0.95      0.95      4160
```

Рис. 22-23. Метрики і матриця моделі Linear SVM

Навчання моделі: Naive Bayes

Середній результат крос-валідації для Naive Bayes: 0.8914

Naive Bayes: Accuracy=0.8901, Precision=0.9191, Recall=0.8560, F1 Score=0.8864

```
--- Naive Bayes ---
Overall Accuracy: 0.8901
Precision Score: 0.9191
Recall Score: 0.8560
AUC: 0.9645

Classification Report:
              precision    recall  f1-score   support

    Reliable       0.86      0.92      0.89      2077
  Unreliable       0.92      0.86      0.89      2083

    accuracy                           0.89      4160
   macro avg       0.89      0.89      0.89      4160
weighted avg       0.89      0.89      0.89      4160
```

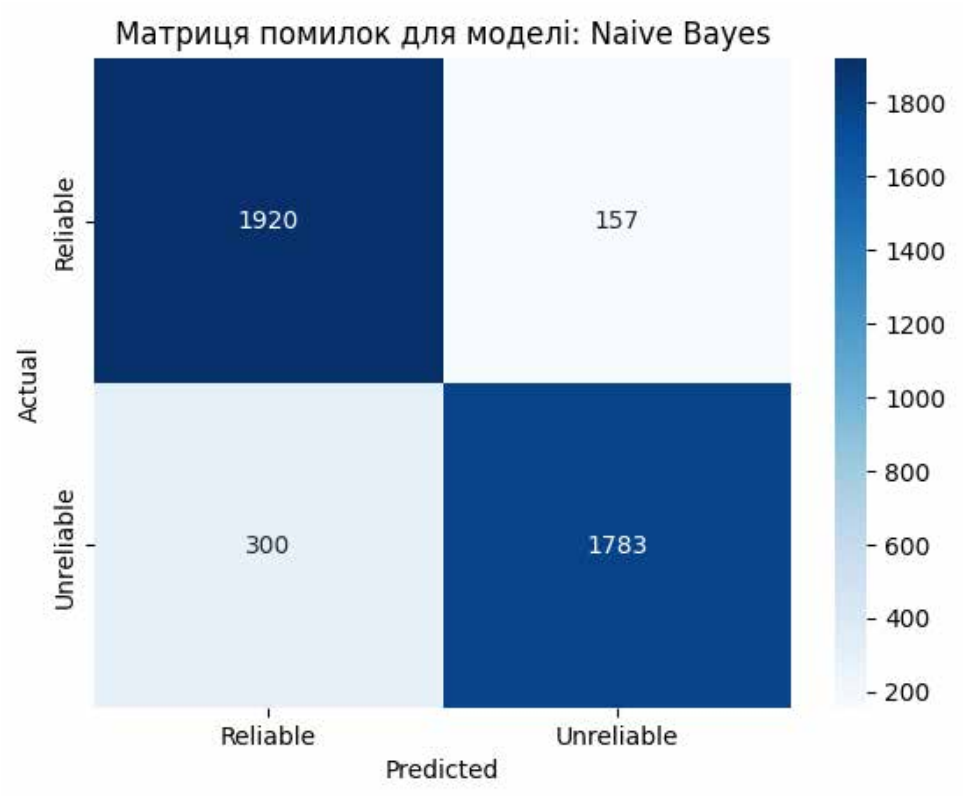


Рис. 24-25. Метрики і матриця моделі Naive Bayes

Навчання моделі: KNN

Середній результат крос-валідації для KNN: 0.6032

KNN: Accuracy=0.6293, Precision=0.5762, Recall=0.9822, F1 Score=0.7263

```
--- KNN ---
Overall Accuracy: 0.6293
Precision Score: 0.5762
Recall Score: 0.9822
AUC: 0.6980

Classification Report:
              precision    recall  f1-score   support

    Reliable       0.94      0.28      0.43      2077
  Unreliable       0.58      0.98      0.73      2083

    accuracy                           0.63      4160
   macro avg       0.76      0.63      0.58      4160
weighted avg       0.76      0.63      0.58      4160
```

Рис. 26-27. Метрики і матриця моделі Naive Bayes

Навчання моделі: Decision Tree

Середній результат крос-валідації для Decision Tree: 0.8800

Decision Tree: Accuracy=0.8755, Precision=0.8717, Recall=0.8809, F1 Score=0.8763

```
--- Decision Tree ---
Overall Accuracy: 0.8755
Precision Score: 0.8717
Recall Score: 0.8809
AUC: 0.8755

Classification Report:
              precision    recall  f1-score   support

    Reliable       0.88      0.87      0.87      2077
  Unreliable       0.87      0.88      0.88      2083

    accuracy                           0.88      4160
   macro avg       0.88      0.88      0.88      4160
weighted avg       0.88      0.88      0.88      4160
```

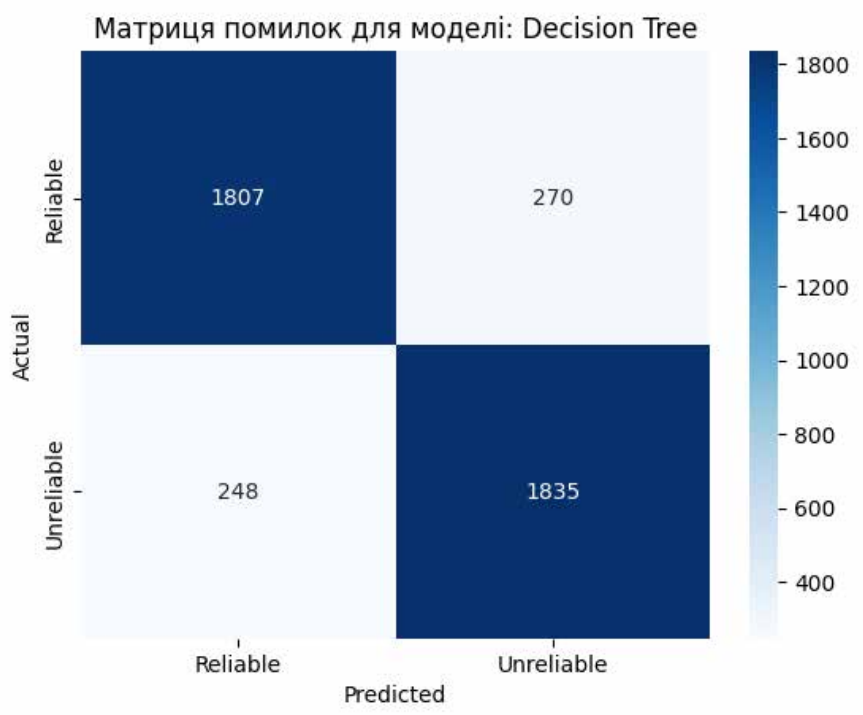


Рис. 28-29. Метрики і матриця моделі Decision Tree

Навчання моделі: Ridge Classifier

Середній результат крос-валідації для Ridge Classifier: 0.9437

Ridge Classifier: Accuracy=0.9440, Precision=0.9384, Recall=0.9506, F1 Score=0.9444

```
--- Ridge Classifier ---
Overall Accuracy: 0.9440
Precision Score: 0.9384
Recall Score: 0.9506
AUC: 0.9880

Classification Report:
              precision    recall  f1-score   support

    Reliable       0.95      0.94      0.94      2077
  Unreliable       0.94      0.95      0.94      2083

    accuracy                           0.94      4160
   macro avg       0.94      0.94      0.94      4160
weighted avg       0.94      0.94      0.94      4160
```

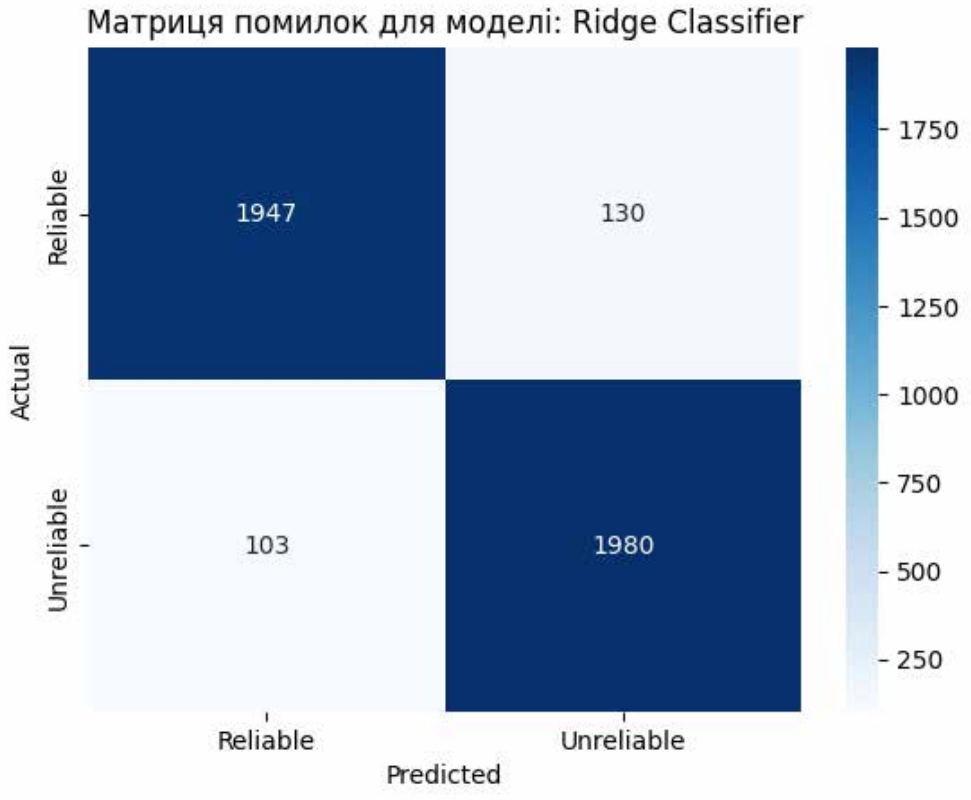


Рис. 30-31. Метрики і матриця моделі Ridge Classifier

3.7. Аналіз

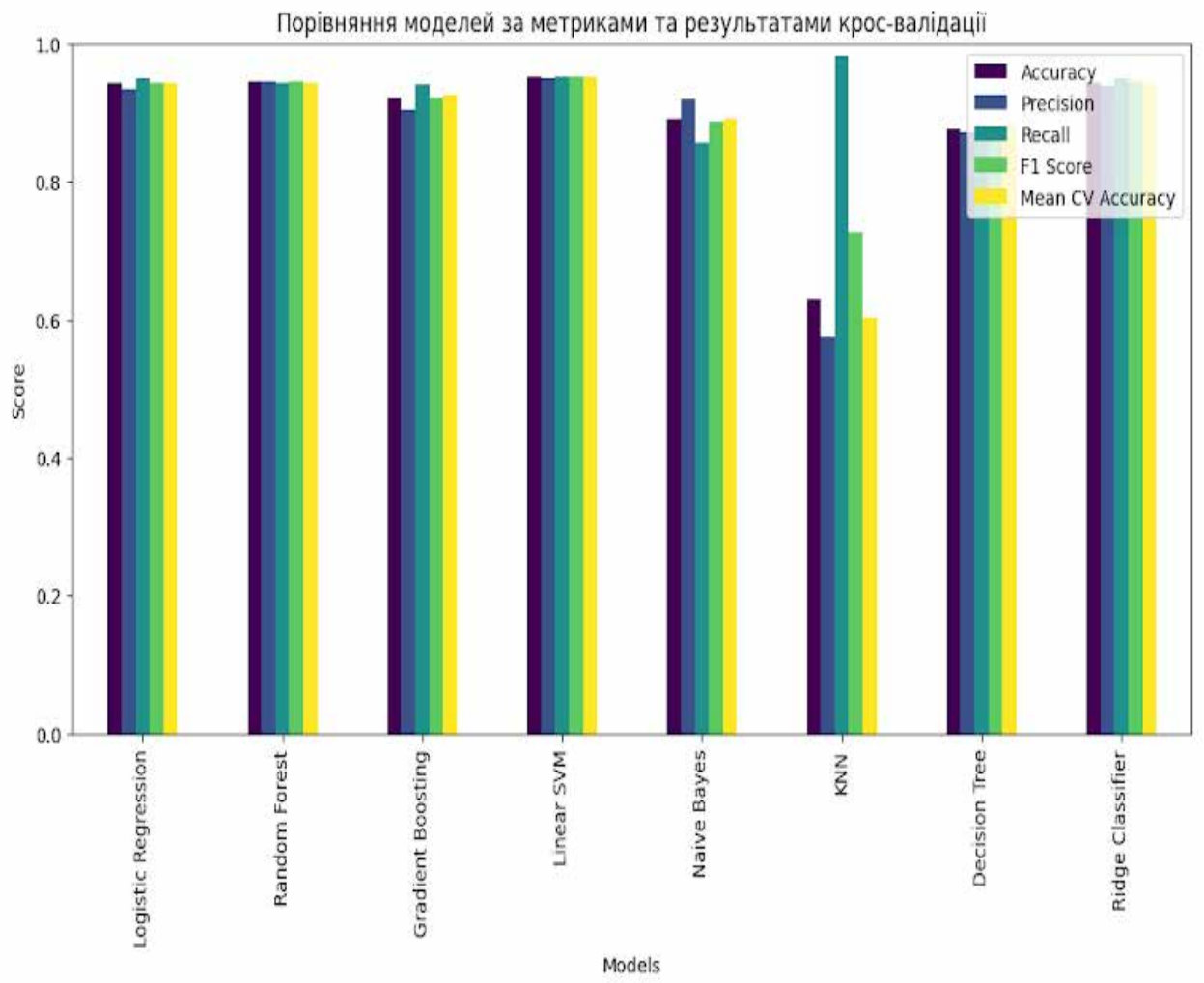


Рис. 32. Порівняння моделей

Аналіз продуктивності моделей для задачі розпізнавання фейкових новин демонструє значні відмінності, які можна пояснити характеристиками самих алгоритмів, природою даних і специфікою метрик оцінювання.

Linear SVM, яка показала найкращі результати (Accuracy = 0.9510, Precision = 0.9502, Recall = 0.9520, F1 Score = 0.9511), є ідеальним прикладом моделі, що добре працює на лінійно роздільних даних. Ця модель використовує гіперплощину для класифікації даних і забезпечує максимальний розрив між класами. Її успіх може свідчити про те, що дані мають добре виражені закономірності, які модель змогла виявити. Завдяки

простоті математичних обчислень вона також менш схильна до перенавчання на невеликих наборах ознак.

У той же час, Random Forest (Accuracy = 0.9442, Precision = 0.9452, Recall = 0.9434, F1 Score = 0.9443) показує дуже схожу продуктивність, оскільки цей алгоритм використовує ансамблевий підхід, комбінуючи результати багатьох рішень окремих дерев. Це дозволяє йому бути стійким до шуму в даних і уникати перенавчання, що часто характерно для окремих дерев ухвалення рішень. Різниця між Random Forest і Linear SVM може полягати в тому, що Linear SVM краще працює з чітко структурованими даними, тоді як Random Forest більш універсальний і ефективний на даних із складними або нерівномірними розподілами.

Щодо Ridge Classifier (Accuracy = 0.9440, Precision = 0.9384, Recall = 0.9506, F1 Score = 0.9444), його результати свідчать, що регуляризація дозволила моделі уникнути перенавчання. Ridge добре справляється з проблемами багатоколінеарності між ознаками, що може бути важливим у задачах класифікації тексту, де певні слова чи фрази мають високу кореляцію.

Моделі Naive Bayes (Accuracy = 0.8901, Precision = 0.9191, Recall = 0.8560, F1 Score = 0.8864) показали помітно нижчу продуктивність, незважаючи на їх простоту і популярність у задачах класифікації тексту. Ймовірно, розподіл даних або їх особливості, такі як залежність між ознаками, призвели до неточних апостеріорних ймовірностей, оскільки Naive Bayes припускає незалежність ознак. Хоч Precision = 0.9191 є доволі високим, низький Recall (0.8560) свідчить про те, що модель не змогла коректно виявити всі випадки фейкових новин.

Незважаючи на загальну ефективність ансамблевих методів, Gradient Boosting (Accuracy = 0.9204, Precision = 0.9048, Recall = 0.9400, F1 Score = 0.9221) показує дещо нижчі результати порівняно з Random Forest. Ця модель побудована на базі послідовного покращення похибок попередніх моделей, що може зробити її чутливішою до шуму в даних. Ймовірно, Gradient Boosting

недостатньо добре справляється з невеликим відсотком некоректно класифікованих прикладів у тренувальних даних.

Продуктивність Decision Tree (Accuracy = 0.8755, Precision = 0.8717, Recall = 0.8809, F1 Score = 0.8763) нижча, ніж у його ансамблевого аналога Random Forest. Це очікувано, оскільки окремі дерева ухвалення рішень схильні до перенавчання і погано працюють на великих наборах даних, якщо вони мають шум або складні закономірності.

Найгіршу продуктивність демонструє KNN (Accuracy = 0.6293, Precision = 0.5762, Recall = 0.9822, F1 Score = 0.7263). Дуже високий Recall = 0.9822 свідчить, що модель виявляє майже всі випадки одного з класів (ймовірно, фейкових новин), але низький Precision = 0.5762 вказує, що значна частка таких виявлень є помилковими. Це може бути наслідком того, що KNN залежить від відстаней у просторі ознак, і якщо дані високовимірні або погано нормалізовані, модель працює неефективно.

Отже, продуктивність моделей значною мірою визначається їх архітектурою та здатністю адаптуватися до специфіки даних. Linear SVM виділяється завдяки своїй здатності чітко розділяти класи, тоді як Random Forest і Ridge Classifier надають збалансовану альтернативу для більш складних даних. Низька ефективність KNN і Naive Bayes підкреслює важливість вибору моделі залежно від властивостей вхідного набору даних.

ВИСНОВКИ

У ході виконання роботи була розроблена система розпізнавання елементів дезінформації в текстових даних, яка допомагає визначати достовірність інформації в медіапросторі шляхом аналізу текстових даних. Основною метою дослідження було спрощення процесу виявлення дезінформації за допомогою створення методу та алгоритму, що аналізують текстові потоки на наявність лінгвістичних структур та виразів, які вказують на можливу недостовірність представленої інформації.

На основі аналізу предметної області розроблено підхід, що включає попередню обробку текстових даних, їх векторизацію за допомогою TF-IDF, та використання різних моделей машинного навчання. Ефективність алгоритмів оцінено за допомогою крос-валідації та основних метрик точності.

Для реалізації системи було використано набір текстових даних із маркуванням достовірності, який попередньо оброблено, векторизовано за допомогою TF-IDF і подано у форматі, придатному для аналізу моделями машинного навчання. Проведено навчання та оцінювання кількох алгоритмів, серед яких логістична регресія, метод опорних векторів (SVM), Random Forest, Gradient Boosting, Naive Bayes, Ridge Classifier, KNN і Decision Tree.

Результати показали, що Linear SVM є найбільш стабільною моделлю, яка досягла результату крос-валідації 0.9513 та продемонструвала найвищі значення метрик точності, повноти та F1-міри. Random Forest і Ridge Classifier також продемонстрували високу продуктивність і можуть бути рекомендовані для практичного використання в задачах виявлення дезінформації.

Інші моделі, зокрема Random Forest і Logistic Regression, також показали значні результати, але мали трохи нижчі значення у порівнянні з лідерами. Натомість методи KNN та Naive Bayes виявилися менш ефективними через

складність роботи з текстовими даними та високий рівень помилок класифікації.

Таким чином, запропонована система прийняття рішень довела свою ефективність у виявленні дезінформації, продемонструвавши високі показники точності. На основі аналізу отриманих результатів можна рекомендувати використання моделей Linear SVM та Ridge Classifier для побудови практичних систем виявлення недостовірної інформації. Подальші дослідження можуть бути спрямовані на удосконалення алгоритмів, розширення набору даних і використання більш складних моделей, таких як глибокі нейронні мережі.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Tony Abou-Assaleh, Nick Cercone, Vlado Keselj, and Ray Sweidan. N-gram-based detection of new malicious code. In Computer Software and Applications Conference, 2004. COMPSAC 2004. Proceedings of the 28th Annual International, volume 2, pages 41–42. IEEE, 2004.

2. B Thomas Adler and Luca De Alfaro. A content-driven reputation system for the wikipedia. In Proceedings of the 16th international conference on World Wide Web, pages 261–270. ACM, 2007.

3. William B Cavnar, John M Trenkle, et al. N-gram-based text categorization. Ann Arbor MI, 48113(2):161–175, 1994.

4. Maarten De Laat, Vic Lally, Lasse Lipponen, and Robert-Jan Simons. Investigating patterns of interaction in networked learning and computer-supported collaborative learning: A role for social network analysis. International Journal of Computer-Supported Collaborative Learning, 2(1):87–103, 2007.

5. Hunt Allcott and Matthew Gentzkow. Social media and fake news in the 2016 election. Technical report, National Bureau of Economic Research, 2017.

6. Edelman. Where do you get your news? https:// www.edelman.com/p/6-a-m/where-do-you-get-your-news/.

7. Weiguo Fan and Michael D Gordon. The power of social media analytics. Communications of the ACM, 57(6):74–81, 2014.

8. Michael Galetzka, Lutz Strüngmann, and Christian Weber. Intelligent predictions: an empirical study of the cortical learning algorithm. University of Applied Sciences Mannheim, 2014.

9. FORTUNE JOHN PATRICK PULLEN. Fcc and net neutrality: Check to see if your name was used in fake comments.

http://fortune.com/2017/11/29/fcc-and-net-neutralitycheck-to-see-if-your-name-was-used-for-fake-comments/.

10. Sebastian Raschka. Naive bayes and text classification introduction and theory. arXiv preprint arXiv:1410.5329, 2014.

11. Derek Ruths and Jürgen Pfeffer. Social media for large studies of behavior. Science, 346(6213):1063–1064, 2014.

12. Sunita Sarawagi et al. Information extraction. Foundations and Trends® in Databases, 1(3):261–377, 2008.

13. Chin-Yew Lin and Eduard Hovy. Automatic evaluation of summaries using n-gram co-occurrence statistics. In Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1, pages 71–78. Association for Computational Linguistics, 2003.

14. Pierre Lison. “an introduction to machine learning, 2015.

15. Bing Liu. Sentiment analysis and opinion mining. Synthesis lectures on human language technologies, 5(1):1–167, 2012.

16. Yutaka Matsuo and Mitsuru Ishizuka. Keyword extraction from a single document using word co-occurrence statistical information. International Journal on Artificial Intelligence Tools, 13(01):157–169, 2004.

17. Hannah Rashkin, Eunsol Choi, Jin Yea Jang, Svitlana Volkova, and Yejin Choi. Truth of varying shades: Analyzing language in fake news and political fact-checking. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, 2017.

18. Rada Mihalcea and Paul Tarau. Textrank: Bringing order into text. In Proceedings of the 2004 conference on empirical methods in natural language processing, 2004.

19. Ayush Pant, “Introduction to Logistic Regression”, towardsdatascience, https://towardsdatascience.com/introduction-to-logistic-regression-66248243c148

20. David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification. Lingvisticae Investigationes, 30(1):3–26, 2007.
21. OpenCV. Introduction to support vector machines opencv. https://docs.opencv.org/2.4/doc/tutorials/ml/introduction_to_svm/introduction_to_svm.html.
22. Carlos Castillo, Marcelo Mendoza, and Barbara Poblete. Information credibility on twitter. In Proceedings of the 20th international conference on World wide web, p. 675–684. ACM, 2011.
23. Bo Pang and Lillian Lee. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In Proceedings of the 42nd annual meeting on Association for Computational Linguistics, p. 271. Association for Computational Linguistics, 2004.
24. Svitlana Volkova, Kyle Shaffer, Jin Yea Jang, and Nathan Hodas. Separating facts from fiction: Linguistic models to classify suspicious and trusted news posts on twitter. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), 2017.
25. Chao-Ying Joanne Peng, Kuk Lida Lee, and Gary M Ingersoll. An introduction to logistic regression analysis and reporting. The journal of educational research, 96(1):3–14, 2002.
26. ALEXANDER SMITH and NBC NEWS VLADIMIR BANIC. Fake news: How a partying macedonian teen earns thousands publishing lies. https://www.nbcnews.com/news/world/fake-news-howpartying-macedonian-teen-earns-thousands-publishinglies-n692451.
27. Statista. Leading social networks used weekly for news in the united states as of February 2017. https://www.statista.com/ statistics/444708/social-networks-used-for-news-usa/.

28. Statita. Which of the following types of news is most important to you? https://www.statista.com/statistics/254511/levelof-interest-in-various-news-types-in-the-us/.

29. Christos Stergiou and Dimitrios Siganos. Neural networks. https://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report.html.svms.org.

30. Christos Stergiou and Dimitrios Siganos. Introduction to support vector machines. http:// www.svms.org/introduction.html.

31. Wikipedia, the free encyclopedia. Artificial neural network with layer coloring, 2013.

32. Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. End-to-end memory networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, Advances in Neural Information Processing Systems 28, pages 2440–2448. Curran Associates, Inc., 2015.

33. Jason Weston, Sumit Chopra, and Antoine Bordes. Memory networks. arXiv preprint arXiv:1410.3916, 2014.

34. Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu. Fake news detection on social media: A data mining perspective. ACM SIGKDD Explorations Newsletter, 2017

35. Victoria L. Rubin, Chen, Y. & Conroy N. J. Deception detection for news: three types of fakes. Proceedings of the 78th ASIS&T Annual Meeting: Information Science with Impact: Research in and for the Community, 2015. American Society for Information Science, 83.

36. Conroy, N.J., Rubin, V.L., Chen, Y.: Automatic deception detection: methods for finding fake news. In: Proceedings of the Association for Information Science and Technology, 2015