MINISTRY OF EDUCATION AND SCIENCE OF UKRAINE
WEST UKRAINIAN NATIONAL UNIVERSITY

MINISTRY OF EDUCATION AND SCIENCE OF UKRAINE
WEST UKRAINIAN NATIONAL UNIVERSITY

Qualifying scientific work on the rights of the manuscript

PAN TIANDE

UDC: 004.932.72:004.94:316.472.4

DISSERTATION

METHODS AND SOFTWARE TOOLS FOR RECOGNIZING FAKE OR IRRELEVANT INFORMATION IN THE CONTENT OF NEWS-ORIENTED SOCIAL NETWORKS

Specialty 121 – Software Engineering

Field of knowledge 12 – Information technology

It is submitted for obtaining the degree of Doctor of Philosophy

The dissertation contains the results of own research. The use of ideas, results and texts of other authors have references to the relevant source.

潘天德 PAN TIANDE

Scientific supervisor: Mykola Dyvak, Doctor of Technical Sciences, Professor

Ternopil – 2025

АНОТАЦІЯ

Пань Тяньде. Методи та програмні засоби розпізнавання неправдивої або нерелевантної інформації у контенті новинних соціальних мереж. – Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття ступеня доктора філософії за спеціальністю 121 «Інженерія програмного забезпечення» – Західноукраїнський національний університет, Тернопіль, 2025.

Підготовка здійснювалась на кафедрі комп'ютерних наук Західноукраїнського національного університету Міністерства освіти і науки України.

Дисертаційна робота присвячена розв'язуванню актуального науково-технічного завдання підвищення ефективності виявлення та аналізу фейкового контенту в новинних соціальних мережах в умовах обмеженої вибірки даних.

Сучасне інформаційне суспільство характеризується стрімким розвитком соціальних мереж, які стали одним із основних каналів поширення новин, комунікації та формування суспільної думки. Водночас із розширенням можливостей цифрової комунікації зростає і масштаб проблеми поширення неправдивої або нерелевантної інформації, що безпосередньо впливає на інформаційну безпеку, політичну стабільність і довіру до медіа.

Особливої актуальності проблема набуває у контексті новинних соціальних мереж, де інформаційні потоки мають високу швидкість оновлення, різнорідну структуру та значну частку користувацького контенту. Відсутність централізованого контролю, домінування емоційно забарвлених повідомлень і наявність координованих інформаційних кампаній створюють сприятливі умови для появи та швидкого поширення фейкових новин. Це формує загрозу маніпулювання суспільною свідомістю, зниження рівня критичного мислення та довіри до достовірних джерел інформації.

Існуючі підходи до перевірки достовірності контенту базуються переважно на ручному фактчекінгу або використанні окремих програмних сервісів (Google

Fact Check Explorer, ClaimBuster, Logically Facts, Hoaxy тощо). Однак ці системи, як показує аналіз, вирішують лише окремі підзавдання: або пошук раніше перевірених тверджень, або виявлення потенційно сумнівних фраз, або ж візуалізацію поширення інформації. Жоден із них не забезпечує комплексного аналізу достовірності контенту з урахуванням поведінкових характеристик користувачів, динаміки реакцій та невизначеності даних, що є типовими для соціальних мереж.

У зв’язку з цим актуальною науковою задачею є підвищення ефективності виявлення та аналізу фейкового контенту в новинних соціальних мережах в умовах обмеженої вибірки даних. Її вирішення потребує поєднання інтервальних, онтологічних та поведінкових підходів із використанням програмних агентів, які здатні забезпечити автоматизований збір, оцінювання та валідацію інформації з різних веб-джерел у режимі реального часу.

У першому розділі розглядаються загальні теоретичні засади проблеми виявлення фейкової інформації у контенті новинних соціальних мереж, зокрема особливості формування інформаційних потоків, природа фейкових повідомлень, причини їхнього поширення та наслідки для користувацького середовища. Особлива увага приділяється факторам, які ускладнюють автоматичне розпізнавання неправдивого контенту — динамічності оновлення, контекстній варіативності, багатомовності та високій емоційності повідомлень.

Далі подано аналіз методів виявлення фейкового контенту, який охоплює основні підходи, що використовуються у сучасних дослідженнях і практичних реалізаціях. Зокрема, виділено групи методів на основі штучних нейронних мереж, аналізу джерел походження інформації, порівняння з апріорі достовірними даними, спільнотної перевірки, аналізу мереж поширення та поведінки користувачів. Проведено оцінку їхніх переваг, обмежень та умов ефективного застосування, що дозволяє сформулювати вимоги до створення удосконалених алгоритмів оцінювання достовірності.

У наступній частині здійснено аналіз програмних сервісів для виявлення фейкового контенту, таких як Google Fact Check Explorer, ClaimBuster, Logically

Facts (AI) та Hoaxy. Порівняльний аналіз їхньої функціональності, технологічних принципів і сфер застосування показав, що кожен із них вирішує лише окремі аспекти проблеми — від пошуку перевірених фактів до візуалізації мереж поширення. Відсутність комплексного підходу, орієнтованого на інтеграцію багатофакторного аналізу та поведінкових моделей користувачів, визначає потребу у створенні нових програмних рішень.

На основі проведеного теоретичного та прикладного аналізу у завершальній частині розділу сформульовано постановку задачі дослідження, у якій обґрунтовано необхідність розроблення гібридного методу оцінювання достовірності інформації на основі інтервальних моделей користувацьких портретів. Визначено основні цілі, підходи та етапи реалізації запропонованої системи, що стане фундаментом для подальших розділів дисертаційної роботи.

У другому розділі запропоновано використовувати математичну модель, для прийняття рішень щодо правдивості контенту, який розміщено в соціальних мережах, на підставі встановлення взаємозв’язку між результатом, на основі якого приймається рішення про достовірність чи недостовірність контенту та чинниками, які на нього впливають. При цьому основними кількісними чинниками запропоновано вважати такі: кількість постів, поширень або лайків, зроблених користувачами протягом короткого часу після появи контенту; кількість коментарів або реакцій через певні інтервали часу; час, за який інформація поширюється через соціальні мережі, наприклад, скільки людей взаємодіють із контентом протягом перших хвилин, годин або днів після публікації; коефіцієнт вірусного поширення контенту наприклад, кількість поширень від кожного користувача. Результуючим показником такої моделі є ступінь достовірності певного контенту в межах від 0 до 1. Запропоновано та обґрунтовано для представлення та аналізу цього показника на підставі дослідження контенту експертами використати методи аналізу інтервальних даних.

Далі у розділі поставлено оптимізаційну задачу для двохетапної ідентифікації моделі на основі аналізу інтервальних даних: формування поточної структури на основі моделей претендентів (синтез структури моделі); оцінювання

її параметрів та перевірка адекватності моделі. Запропоновано на обґрунтовано гібридний метод ідентифікації інтервальних моделей портрету користувачів у соціальній мережі, який ґрунтується на поєднанні метаевристичного алгоритму синтезу структури моделі на підставі поведінкової моделі бджолиної колонії та градієнтних методів ідентифікації параметрів моделей-претендентів.

В завершальній частині розділу розглянуто задачі безпосереднього застосування гібридного методу ідентифікації інтервальних моделей портрету користувачів соціальної мережі та показано можливість прийняття рішень щодо достовірності контенту на підставі побудованої моделі.

У третьому розділі розглянуто концепцію, структуру та реалізацію програмних агентів, які забезпечують реалізацію запропонованого методу у вигляді багаторівневої системи. Зокрема, метод оцінки достовірності виступає теоретичною основою для побудови обчислювальних модулів, що формують інтегральний показник достовірності контенту. Цей показник дозволяє кількісно оцінювати правдоподібність новинних повідомлень, враховуючи їх джерело, зміст, мережеве поширення та емоційні характеристики.

Особливу увагу приділено процедурі вибору порогового значення інтегрального показника, яке визначає межу між достовірним і сумнівним контентом. Це значення обґрунтовано на основі компромісу між точністю та повнотою виявлення фейкових повідомлень, що дозволяє системі адаптуватися до специфіки різних інформаційних доменів (оперативні новини, соціально-політичні повідомлення, аналітичні матеріали тощо).

У подальших підрозділах детально описано особливості реалізації програмних агентів, їхню взаємодію через API та модульну архітектуру, а також принципи етичного отримання даних із соціальних мереж (Facebook, X/Twitter, Telegram). Розкрито питання структури даних, формування інтервальних профілів користувачів, нормалізації текстового контенту та логування результатів аналізу в базі даних MongoDB.

Завершальна частина розділу присвячена експериментальним дослідженням, у межах яких проведено тестування роботи агентів на реальних і симульованих

даних соціальних мереж. На основі порівняльного аналізу продемонстровано, що впровадження запропонованих агентів дозволяє підвищити точність виявлення недостовірного контенту, скоротити час верифікації новин та забезпечити прозорість і пояснюваність прийнятих рішень.

У четвертому розділу розглядаються концептуальні та архітектурні засади створення програмного середовища, призначеного для виявлення та аналізу фейкового контенту у новинних соціальних мережах. У цій частині детально описано архітектуру програмного забезпечення, що побудована за модульним принципом і включає підсистеми збору даних, попередньої обробки, оцінювання достовірності контенту, інтервального моделювання користувачів, зберігання результатів і візуалізації аналітичної інформації.

Особливу увагу приділено механізмам інтеграції з соціальними мережами, структурі аналітичного ядра системи (модуль CIEngine) та ролі інтервального підходу в підвищенні точності й адаптивності оцінювання достовірності новин. Архітектура системи подається через опис основних UML-діаграм — варіантів використання, класів, пакетів і розгортання, що демонструють логічну й фізичну структуру програмного комплексу.

Далі у розділі зосереджено увагу на підсистемах аналізу та зберігання інформації, які забезпечують функціональну цілісність системи. Детально розкривається принцип побудови бази даних на основі MongoDB, структура основних колекцій (posts, user_profiles, facts, config, logs), їх призначення, логіка взаємодії з аналітичними та поведінковими модулями, а також методи індексації, шардінгу та оптимізації продуктивності. Описано реалізацію класів доступу до даних, механізми контролю цілісності, резервного копіювання та відтворюваності результатів. Окрему увагу приділено підсистемі інтервального моделювання користувачів, яка забезпечує накопичення і динамічне оновлення поведінкових характеристик, що використовуються для уточнення показників довіри (TrustRate) і інтегрального коефіцієнта достовірності (CI).

У наступній частині розділу розглянуто організацію графічного інтерфейсу системи, який реалізовано у сучасному веборієнтованому середовищі (HTML5,

CSS3, JavaScript, TailwindCSS, Chart.js). Інтерфейс побудований у вигляді багатокомпонентного дашборду, що охоплює головну сторінку моніторингу, сторінки аналізу постів, оцінки достовірності, інтервального портрету користувача, профілю та налаштувань. Передбачено інтерактивні елементи для фільтрації контенту за часовими інтервалами («24 год», «7 днів», «30 днів»), побудови графіків зміни CI, відображення емоційних heatmap-карт і мережевих графів поширення новин. Користувач має змогу здійснювати інтерактивний пошук, перевірку достовірності тверджень, перегляд графів підтвердження та аналіз поведінкової активності у реальному часі. Така організація інтерфейсу забезпечує поєднання аналітичної глибини з наочністю представлення результатів.

У завершальній частині розділу наведено оцінку ефективності розробленого програмного середовища, проведену за інтегральним показником ефективності (IE), який враховує аналітичні, мережеві, поведінкові та користувацькі характеристики системи. Порівняльний аналіз із відомими інструментами (Google Fact Check Explorer, ClaimBuster, Logically Facts, Hoaxy) показав, що запропонована система досягає найвищого рівня комплексності, що відповідає категорії “високоефективних систем”. Це підтверджує, що інтеграція інтервального моделювання користувачів, комплексного індикатора достовірності CI, автоматичного збору даних із соціальних мереж і сучасних засобів аналітики забезпечує суттєву перевагу розробленого рішення над існуючими аналогами.

Практичне значення отриманих результатів полягає у створенні програмного середовища для розпізнавання неправдивої або нерелевантної інформації у контенті новинних соціальних мереж.

Ключові слова: новинні соціальні мережі, достовірність контенту, фейки, портрет користувача соціальної мережі, обмежена вибірка даних, інтервальний аналіз, структурна ідентифікація, параметрична ідентифікація, поведінкова модель бджолиної колонії, градієнтні методи, програмні агенти, програмні середовища, інтелектуальні асистенти.

ПЕРЕЛІК ОПУБЛІКОВАНИХ ПРАЦЬ ЗА ТЕМОЮ ДИСЕРТАЦІЇ

Наукові праці, в яких опубліковано основні наукові результати дисертації:

1. Dyvak, M., Yushko, A., Melnyk, A., Pan, T. An Intelligent Information System for Generating a Scientist's Scientometrics Using Content Analysis Methods. CEUR-WS. 2024. Vol. 3942. P. 66-82. (0,8 д.а. / 0,2 д.а.; особистий внесок: здобувачем запропоновано підхід до вибору показників, які формують портрет сооби в спеціалізованому середовищі).

https://ceur-ws.org/Vol-3942/S_06_Dyvak.pdf

2. Dyvak, Mykola, Tyande Pan, and Oleksandr Kindzerskyi. 2025. “Mathematical Model of a Social Network User Profile Based on Interval Data Analysis”. International Journal of Computing 24 (3):452-59. (0,9 д.а. / 0,6 д.а.; особистий внесок: здобувачем запропоновано та обґрунтовано гібридний метод ідентифікації інтервальних моделей портрету користувачів у соціальній мережі та інтервальна математична модель, що встановлює взаємозв’язок між достовірністю контенту новинної соціальної мережі та портретом поведінки користувачів).

https://www.computingonline.net/computing/article/view/4182.

3. Dyvak, Mykola, Volodymyr Manzhula, Andriy Melnyk, Nataliia Petryshyn, Tiande Pan, Arkadiusz Banasik, Piotr Pikiewicz, and Wojciech M. Kempa. 2025. "Modeling the Electricity Generation Processes of a Combined Solar and Small Hydropower Plant" Energies 18, no. 9: 2351. (1,1 д.а. / 0,2 д.а.; особистий внесок: здобувачем обґрунтовано використання гібридних методів ідентифікації інтервальних моделей).

https://doi.org/10.3390/en18092351

4. Melnyk, A., Tymchyshyn, V., Pukas, A., Matiichuk, L., Shcherbiak, I., Yurchyshyn, T., Pan, T. Automatic Generation of Test Tasks Using ChatGPT API. CEUR-WS. 2025. Vol. 3974. P. 263-271. (0,9 д.а. / 0,5 д.а.; особистий внесок: здобувачем обґрунтовано використання інтелектуалізованих підходів до отримання інформації через API).

https://ceur-ws.org/Vol-3974/short09.pdf

5. Mistriakov , V.V., and Pan Tiande. 2024. “Processing Content Query Requests for CSAF Documents Using a GrapHQL-BASED API”. Optoelectronic Information-Power Technologies 48 (2):152-61. (0,8 д.а. / 0,6 д.а.; особистий внесок: здобувачем запропоновано мультиагентну архітектуру до опрацювання слабкоструктурованого контенту).

https://doi.org/10.31649/1681-7893-2024-48-2-152-161.

6. Tiande Pan. 2025. "Research on Identification Methods for False or Unrelated Information in Network Resource Content" International Journal of High Speed Electronics and SystemsVol. 34, No. 04, 2540203.

https://doi.org/10.1142/S0129156425402037

Наукові праці, які засвідчують апробацію матеріалів дисертації:

7. Пань Тяньде, Дудник Ю.Ю., Гордіюк В.Ю., Даньків А.В., Колодій А.О. Метод багатомодального профілювання особистості користувачів соціальних мереж. Комп’ютерні інформаційні технології: матеріали школи-семінару молодих вчених і студентів СІТ’2024. Тернопіль: ЗУНУ, 2024. С. 95-96. (0,1 д.а. / 0,03 д.а.; особистий внесок: здобувачем запропоновано компонентну структуру інтелектуалізованої системи для профілювання особистості користувачів соціальних мереж).

https://dspace.wunu.edu.ua/bitstream/316497/52868/1/CIT%272024_Last.pdf

8. Пань Тяньде, Забчук В.Д., Судейченко Д.В., Биц С.С., Самсонович В.В. Математичне та програмне забезпечення для аналізу форматів та опрацювання великих об'ємів даних. Комп’ютерні інформаційні технології: матеріали школи-семінару молодих вчених і студентів СІТ’2024. Тернопіль: ЗУНУ, 2024. С. 97-98. (0,1 д.а. / 0,03 д.а.; особистий внесок: здобувачем запропоновано підхід до опрацювання та аналізу різноструктурованих даних).

https://dspace.wunu.edu.ua/bitstream/316497/52868/1/CIT%272024_Last.pdf

ANNOTATION

Pan Tiande. Methods and Software Tools for Recognizing Fake or Irrelevant Information in the Content of News-Oriented Social Networks. – Scientific work on the rights of the manuscript.

Thesis for the degree of Doctor of Philosophy in the specialty 121 "Software Engineering" - West Ukrainian National University, Ternopil, 2025.

The research was carried out at the Department of Computer Science of the West Ukrainian National University of the Ministry of Education and Science of Ukraine.

The dissertation is devoted to addressing a pressing scientific and technical problem—enhancing the effectiveness of detecting and analyzing fake content in news-oriented social networks under conditions of limited data samples.

Modern information society is characterized by the rapid development of social networks, which have become one of the main channels for news dissemination, communication, and public opinion formation. Alongside the expansion of digital communication capabilities, the scale of the problem of spreading false or irrelevant information is also increasing, directly affecting information security, political stability, and public trust in the media.

The issue is particularly critical in the context of news-oriented social networks, where information flows are highly dynamic, heterogeneous, and dominated by user-generated content. The absence of centralized control, the prevalence of emotionally charged messages, and the existence of coordinated information campaigns create favorable conditions for the emergence and rapid spread of fake news. This poses a threat of manipulating public consciousness, reducing critical thinking, and undermining confidence in credible sources.

Existing approaches to content verification are mainly based on manual fact-checking or the use of individual software services such as Google Fact Check Explorer, ClaimBuster, Logically Facts, and Hoaxy. However, as analysis shows, these systems address only isolated subtasks—either searching for previously verified claims, detecting potentially suspicious phrases, or visualizing the dissemination of information.

None of them provides a comprehensive assessment of content credibility that simultaneously considers user behavioral features, reaction dynamics, and data uncertainty, which are inherent to social media environments.

Therefore, the relevant scientific problem consists in improving the efficiency of fake content detection and analysis in news-oriented social networks under limited data availability. Its solution requires combining interval-based, ontological, and behavioral approaches supported by intelligent software agents capable of automated data collection, assessment, and validation from multiple web sources in real time.

The first chapter outlines the theoretical foundations of fake-content detection in news-oriented social networks. It examines the formation of information flows, the nature of fake messages, the causes of their spread, and their consequences for user environments. Particular attention is paid to factors that complicate automatic recognition—frequent updates, contextual variability, multilingualism, and high emotional intensity of messages.

Further, the dissertation presents a review of fake-content detection methods covering key research and practical approaches, including neural-network-based methods, source-reliability analysis, comparison with verified data, community-driven validation, and user-behavior and network-propagation analysis. Their advantages, limitations, and effective application conditions are evaluated, forming the basis for developing improved credibility-assessment algorithms.

The next section provides an overview of software tools such as Google Fact Check Explorer, ClaimBuster, Logically Facts (AI), and Hoaxy. Comparative analysis of their functionality, underlying technologies, and application domains shows that each addresses only a specific aspect of the problem—from fact retrieval to visualization of propagation networks. The lack of a holistic approach integrating multifactor analysis with behavioral user models underscores the need for developing new, more comprehensive software solutions.

Based on the conducted theoretical and applied analysis, the research problem statement defines the necessity of developing a hybrid method for credibility assessment built upon interval models of user profiles. The main objectives, conceptual approaches,

and stages of implementing the proposed system are formulated, providing a methodological foundation for the following chapters.

In the second chapter, a mathematical model is proposed for decision-making regarding the truthfulness of social-network content by establishing relationships between the credibility outcome and influencing factors. The primary quantitative factors include: the number of posts, shares, or likes made shortly after content publication; the number of comments or reactions over specific time intervals; the time required for information to propagate through the network (how many users interact with the content within minutes, hours, or days); the viral-spread coefficient (e.g., average reshares per user).

The resulting model output represents the credibility degree of a given piece of content on a scale from 0 to 1. To represent and analyze this indicator—especially under expert assessment uncertainty—the study employs interval data analysis methods.

An optimization problem for two-stage model identification based on interval-data analysis is formulated: (1) synthesis of candidate model structures and (2) parameter estimation with adequacy verification. A hybrid identification method of interval user-profile models is proposed, combining a metaheuristic algorithm inspired by the bee colony optimization principle for structural synthesis and gradient-based methods for parameter identification of candidate models.

The final part of this chapter demonstrates the application of the hybrid identification method to user-profile modeling in social networks, enabling data-driven decision-making on content credibility.

The third chapter presents the conceptual design, structure, and implementation of software agents that operationalize the proposed method within a multi-layered system. The credibility-assessment method serves as a theoretical core for computational modules that produce an integral credibility index (CI). This index allows quantitative evaluation of news reliability by accounting for the source, content semantics, dissemination network, and emotional features.

Special focus is given to the threshold selection procedure for the CI indicator, defining the boundary between reliable and questionable content. The chosen threshold

is justified through a trade-off between detection precision and recall, allowing adaptive calibration across different information domains (breaking news, socio-political reports, analytical materials, etc.).

Subsequent sections describe in detail the implementation of software agents, their API-based interaction, modular architecture, and adherence to ethical data-collection principles for major social platforms (Facebook, X/Twitter, Telegram). Data structures, interval-profile generation, text normalization, and MongoDB-based logging are presented.

The experimental part evaluates the agents' performance on real and simulated social-network datasets. Comparative analysis demonstrates that integrating the proposed agents improves fake-content detection accuracy, reduces verification time, and ensures transparency and explainability of results.

The fourth chapter defines the conceptual and architectural framework of the software environment developed for fake-content detection and analysis in news social networks. The modular architecture includes subsystems for data collection, preprocessing, credibility assessment, interval user modeling, results storage, and analytical visualization.

Particular attention is devoted to integration mechanisms with social networks, the analytical core (CIEngine module), and the role of interval modeling in enhancing precision and adaptability. The architecture is detailed using UML diagrams (use-case, class, package, and deployment) that illustrate the logical and physical structure of the system.

The section on information analysis and storage subsystems explains the design of the MongoDB-based database, describing the main collections—posts, user_profiles, facts, config, and logs—their purposes, interaction logic with analytical modules, indexing, sharding, and performance optimization. Data-access classes, integrity control, backup, and reproducibility mechanisms are elaborated.

A dedicated subsection addresses the interval modeling subsystem, responsible for accumulating and dynamically updating user behavioral parameters to refine trust indicators (TrustRate) and the composite credibility index (CI).

The chapter also presents the design of a modern web-based user interface built with HTML5, CSS3, JavaScript, TailwindCSS, and Chart.js. The interface is organized as a multi-component dashboard comprising monitoring, post-analysis, credibility-evaluation, interval-profile, and settings pages. Interactive filters enable temporal selection (24 h, 7 days, 30 days), CI-trend visualization, emotional heatmaps, and network-propagation graphs. Users can perform real-time searches, verify statements, and explore behavioral activity. This interface design harmoniously combines analytical depth with visual clarity.

The final part presents an efficiency evaluation of the developed environment using an integral efficiency index (IE) that incorporates analytical, network, behavioral, and usability criteria. Comparative assessment against leading tools (Google Fact Check Explorer, ClaimBuster, Logically Facts, Hoaxy) confirms that the proposed system achieves the highest level of comprehensiveness, corresponding to the class of high-efficiency systems.

The integration of interval-based user modeling, the composite credibility indicator (CI), automated social-media data collection, and advanced analytics ensures a substantial advantage of the developed system over existing analogues.

The practical significance of the obtained results lies in the creation of a software environment for the detection and analysis of false or irrelevant information within news-oriented social-network content.

Keywords: news-oriented social networks; content credibility; fake news; social network user profile; limited data sample; interval analysis; structural identification; parametric identification; bee colony behavioral model; gradient-based methods; software agents; software environments; intelligent assistants.

LIST OF PUBLISHED PAPERS BY THE TOPIC OF THESIS

SCIENTIFIC PAPERS IN WHICH THE MAIN SCIENTIFIC RESULTS OF THE DISSERTATION WERE PUBLISHED:

1. Dyvak, M., Yushko, A., Melnyk, A., Pan, T. An Intelligent Information System for Generating a Scientist's Scientometrics Using Content Analysis Methods.

CEUR-WS. 2024. Vol. 3942. P. 66-82. (0.8 author's sheets / 0.2 author's sheets; personal contribution: the applicant proposed an approach to selecting indicators that form the community profile in a specialized environment.)

https://ceur-ws.org/Vol-3942/S_06_Dyvak.pdf

2. Dyvak, Mykola, Tyande Pan, and Oleksandr Kindzerskyi. 2025. "Mathematical Model of a Social Network User Profile Based on Interval Data Analysis". International Journal of Computing 24 (3):452-59. (0.9 author's sheets / 0.6 author's sheets; personal contribution: the applicant proposed and substantiated a hybrid method for identifying interval models of social network user profiles, as well as an interval-based mathematical model that establishes the relationship between the credibility of news content in social networks and user behavioral profiles.)

https://www.computingonline.net/computing/article/view/4182.

3. Dyvak, Mykola, Volodymyr Manzhula, Andriy Melnyk, Nataliia Petryshyn, Tiande Pan, Arkadiusz Banasik, Piotr Pikiewicz, and Wojciech M. Kempa. 2025. "Modeling the Electricity Generation Processes of a Combined Solar and Small Hydropower Plant" Energies 18, no. 9: 2351. (1.1 author's sheets / 0.2 author's sheets; personal contribution: the applicant substantiated the use of hybrid methods for identifying interval models.)

https://doi.org/10.3390/en18092351

4. Melnyk, A., Tymchyshyn, V., Pukas, A., Matiichuk, L., Shcherbiak, I., Yurchyshyn, T., Pan, T. Automatic Generation of Test Tasks Using ChatGPT API. CEUR-WS. 2025. Vol. 3974. P. 263-271. (0.9 author's sheets / 0.5 author's sheets; personal contribution: the applicant substantiated the use of intelligent approaches for information retrieval via API.)

https://ceur-ws.org/Vol-3974/short09.pdf

5. Mistriakov , V.V., and Pan Tiande. 2024. "Processing Content Query Requests for CSAF Documents Using a GrapHQL-BASED API". Optoelectronic Information-Power Technologies 48 (2):152-61. (0.8 author's sheets / 0.6 author's sheets; personal contribution: the applicant proposed a multi-agent architecture for processing weakly structured content.)

https://doi.org/10.31649/1681-7893-2024-48-2-152-161.

6. Tiande Pan. 2025. "Research on Identification Methods for False or Unrelated Information in Network Resource Content" International Journal of High Speed Electronics and SystemsVol. 34, No. 04, 2540203.

https://doi.org/10.1142/S0129156425402037

SCIENTIFIC WORKS CERTIFYING THE APPROVAL OF THE DISSERTATION MATERIALS:

7. Pan Tiande, Dudnyk Y.Yu., Hordiiuk V.Yu., Dankiv A.V., Kolodii A.O. A Method for Multimodal Profiling of Social Network Users. Computer information technologies: materials of the school-seminar of young scientists and students CIT'2024. Ternopil: WUNU, 2024. P. 95-96. (0.1 author's sheets / 0.03 author's sheets; personal contribution: the applicant proposed a component-based structure of an intelligent system for profiling social network users.)

https://dspace.wunu.edu.ua/bitstream/316497/52868/1/CIT%272024_Last.pdf

8. Pan Tiande, Zabchuk V.D., Sudeichenko D.V., Byts S.S., Samsonovych V.V. Mathematical and Software Tools for the Analysis and Processing of Large Data Volumes. Computer information technologies: materials of the school-seminar of young scientists and students CIT'2024. Ternopil: WUNU, 2024. P. 97-98. (0.1 author's sheets / 0.03 author's sheets; personal contribution: the applicant proposed an approach to processing and analyzing heterogeneously structured data.)

https://dspace.wunu.edu.ua/bitstream/316497/52868/1/CIT%272024_Last.pdf

CONTENT

INTRODUCTION

Relevance of the research topic. The modern information society is characterized by the rapid development of social networks, which have become one of the main channels for news dissemination, communication, and the shaping of public opinion. At the same time, the expansion of digital communication opportunities has amplified the scale of the problem associated with the spread of false or irrelevant information, which directly affects information security, political stability, and public trust in the media [7, 10, 23, 31].

This issue is particularly acute in the context of news-oriented social networks, where information flows are highly dynamic, structurally heterogeneous, and largely composed of user-generated content. The absence of centralized control, the dominance of emotionally charged messages, and the presence of coordinated information campaigns create favorable conditions for the emergence and rapid circulation of fake news. Such phenomena pose a threat of manipulating public consciousness, weakening critical thinking, and eroding trust in reliable sources of information [54, 109, 112].

Existing approaches to content verification rely mainly on manual fact-checking or the use of individual software tools such as Google Fact Check Explorer, ClaimBuster, Logically Facts (AI), and Hoaxy [7, 10, 23, 64]. However, as the analysis shows, these systems address only partial sub-tasks — either searching for previously verified claims, detecting potentially suspicious phrases, or visualizing the dissemination of information. None of them provides a comprehensive assessment of content credibility that considers user behavioral patterns, reaction dynamics, and data uncertainty, all of which are inherent to social networks [7, 47, 109, 111].

Therefore, there is a pressing need to develop new-generation methods and software tools capable of automatically analyzing news content streams, integrating information from multiple sources, assessing the credibility level of messages, and adapting to conditions of limited or imprecise data samples [7, 31].

The foundation of this approach lies in the integration of natural language processing (NLP), interval modeling, network analysis, and intelligent agent-based

technologies. The application of an interval mathematical model makes it possible to account for uncertainty in user behavior and variations in the parameters of the information environment, thereby ensuring the correctness of credibility assessment even under incomplete data conditions.

Research Aim and Objectives. The aim of this study is to enhance the effectiveness of detecting and analyzing fake content in news-oriented social networks under conditions of limited data sampling.

To achieve this aim, the following objectives have been defined:

- to analyze existing methods and software tools for recognizing false or irrelevant information in the content of web resources;
- to justify and develop a hybrid method for identifying interval-based models of user profiles in social networks;
- to design an interval mathematical model that establishes the relationship between the decision outcome regarding the credibility or falsity of content and the influencing factors;
- to improve software agents for assessing the credibility of content in social network news feeds by integrating classical credibility evaluation criteria with those specific to social media content;
- to develop a software environment for detecting false or irrelevant information within web resources, integrating agents for retrieving content from social networks and tools for its credibility assessment;
- to conduct experimental validation of the developed methods and tools.

The **object of research** is the recognition of false or irrelevant information in the content of news-oriented social networks.

The **subject of research** comprises the methods and software tools designed for detecting and analyzing false or irrelevant information in the content of news-oriented social networks.

Methods of research. To address the problem of detecting unreliable information, the study employs methods of system analysis, identification theory, mathematical modeling, interval arithmetic, and optimization techniques.

For the processing of textual information, the research utilizes natural language processing (NLP) methods, including tokenization, lemmatization, part-of-speech tagging, named entity recognition (NER), and semantic matching.

During the development of the software implementation, object-oriented design methods (UML modeling, modularity, and encapsulation principles) were applied, along with an agent-based approach, which enabled the creation of intelligent software agents for the automated collection, analysis, and evaluation of content.

Scientific Novelty of the Results. The key scientific contributions include:

First obtained :

- an interval-based mathematical model was developed for the first time, establishing a relationship between the credibility of content in news-oriented social networks and user behavioural profiles, which, unlike existing models, relies on the analysis of interval data under limited sample conditions and thereby enhances the efficiency of credibility recognition at the early stages of publication;

- hybrid method for identifying interval models of user profiles in social networks was proposed and substantiated, which, unlike existing approaches, combines a metaheuristic algorithm for model structure synthesis based on the behavioural model of a bee colony with gradient methods for identifying the parameters of candidate models, thereby reducing the computational complexity of the identification process and overall improving the efficiency of content credibility recognition at the early stages of its publication;

Further developed:

- software agents for assessing the credibility of content in news-oriented social network resources have been further developed, which, unlike existing ones, integrate both traditional criteria—redundancy, inconsistency, timeliness, reliability, and completeness—and new characteristics inherent to digital media, such as network confirmation and emotional tone, thereby enhancing the overall effectiveness of fake content detection in news social networks.

- software environments for detecting and analysing fake content in news-oriented social networks have been further developed, which, unlike existing solutions, integrate

automated tools for retrieving content from social platforms with modules for assessing its credibility under limited data conditions, and are adapted to function as intelligent assistants supporting the creation and deployment of news services.

Personal contribution of the applicant. The dissertation represents an independent research study conducted by the applicant. All theoretical foundations, methodological proposals, and results submitted for defense were obtained personally by the author. From the co-authored scientific publications, only those sections that reflect the author's individual research contribution have been incorporated into the dissertation.

Approbation of the results of the dissertation. The key results of the research were reported and debated at various international scientific and applied research conferences: Second International Conference of Young Scientists on Artificial Intelligence for Sustainable Development (YAISD 2025); 8th International Scientific and Practical Conference "Applied Information Systems and Technologies in the Digital Society" (AISTDS 2024). And also reported within the framework of the winter school-seminar of young scientists and students CIT'2024.

Publications. Based on the results of the dissertation research, nine scientific papers have been published (Appendix C) with a total volume of 94 pages, including 6 articles in peer-reviewed scientific journals [1-6], one of which is indexed in the international scientometric databases Scopus and Web of Science and, according to the SCImago Journal and Country Rank classification, belongs to quartile Q1 [3]; and 2 publications in conference proceedings [7, 8], 5 of which are indexed in the Scopus database [1-4, 6].

The structure and volume of thesis. The dissertation consists of an introduction, four chapters, conclusions, a list of references comprising 112 sources, and 3 appendices. The total volume of the work is 156 pages of printed text, including 126 pages of the main body. The dissertation contains 38 figures and 4 tables.

CHAPTER 1
ANALYSIS OF METHODS AND SOFTWARE TOOLS FOR DETECTING FALSE OR IRRELEVANT INFORMATION IN THE CONTENT OF NEWS-ORIENTED SOCIAL NETWORKS

The first part of this section examines the general theoretical foundations of the problem of detecting fake information within the content of news-oriented social networks. Particular attention is given to the characteristics of information flow formation, the nature of fake messages, the reasons for their dissemination, and their impact on the user environment. Special emphasis is placed on the factors that complicate the automatic recognition of false content—namely, the high dynamics of content updates, contextual variability, multilingualism, and the pronounced emotional tone of messages.

Subsequently, an analysis of methods for detecting fake content is presented, encompassing the main approaches used in current research and practical implementations. The methods are grouped into several categories: those based on artificial neural networks, source credibility analysis, comparison with a priori verified data, community-based verification, and the analysis of dissemination networks and user behavior. Their advantages, limitations, and conditions for effective application are evaluated, providing the basis for formulating requirements for the development of improved credibility assessment algorithms.

The next part of the section provides an overview of existing software services for detecting fake content, such as Google Fact Check Explorer, ClaimBuster, Logically Facts (AI), and Hoaxy. A comparative analysis of their functionality, technological principles, and application domains demonstrates that each of these systems addresses only specific aspects of the overall problem—ranging from the retrieval of verified facts to the visualization of information dissemination networks. The absence of a comprehensive approach that integrates multifactor analysis and user behavioral modeling highlights the need for developing new software solutions.

Based on the theoretical and applied analysis conducted, the final part of this

section formulates the research problem, substantiating the necessity of developing a hybrid method for assessing information credibility built upon interval models of user profiles. The main objectives, methodological approaches, and implementation stages of the proposed system are defined, forming the foundation for the subsequent chapters of the dissertation.

The principal results of this section have been published in [20, 21, 88].

1.1 The Problem of Detecting Fake Information in the Content of News-Oriented Social Networks

In the modern era of digital transformation, social networks have become one of the primary channels for news dissemination, public opinion formation, and social interaction. Platforms such as Facebook, X (Twitter), Telegram, Instagram, and others enable instant information exchange among users, which, on the one hand, promotes the democratization of access to knowledge, but on the other hand, poses significant risks to information security due to the spread of false, incomplete, or deliberately manipulative content [31, 102, 104, 105].

The speed of information circulation within social networks far exceeds the capabilities of traditional fact-checking mechanisms [1, 3, 8]. According to analytical studies, false information spreads several times faster than truthful content because it tends to be emotionally charged, simplified in structure, and more engaging for user interaction. As a result, a distorted information environment emerges, undermining trust in official sources, fostering panic and misinformation, and facilitating large-scale manipulation of public opinion (Figure 1.1).

The problem of detecting fake content in social networks is therefore of critical importance, as information attacks, disinformation campaigns, and the deliberate injection of dubious news have become key instruments of hybrid warfare. At the same time, the growing number of bot networks and anonymous accounts that automatically generate and disseminate fake content creates an illusion of credibility through mass replication and social amplification effects [7, 23, 31, 34].

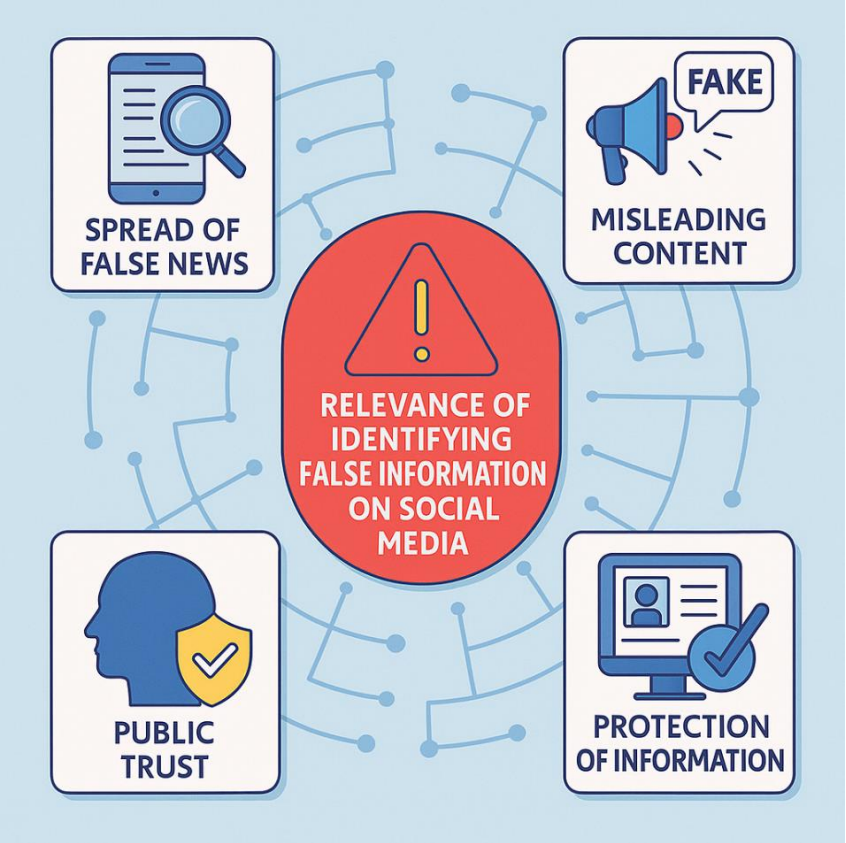

Figure 1.1. The Problem of Detecting Fake Information

Traditional fact-checking approaches—such as searching for official sources, manual verification, or classification based on large training datasets—do not adequately account for the specific nature of social networks. In these environments, short and unstructured messages dominate, often containing a high level of noise, abbreviations, emojis, quotes, reposts, as well as multilingual and temporally variable content. Moreover, the absence of clear accountability for information dissemination, due to the large proportion of anonymous users, further complicates the verification process [2, 4-6, 31, 57].

Under such conditions, the effective detection of false information requires the adoption of new methodological approaches that integrate data analytics, semantic technologies, and interval modeling (Figure 1.2). Of particular importance is the development of intelligent software agents capable of autonomously collecting, structuring, and evaluating content in real time while accounting for uncertainty, temporal dynamics, and user behavioral characteristics [7, 85, 86].

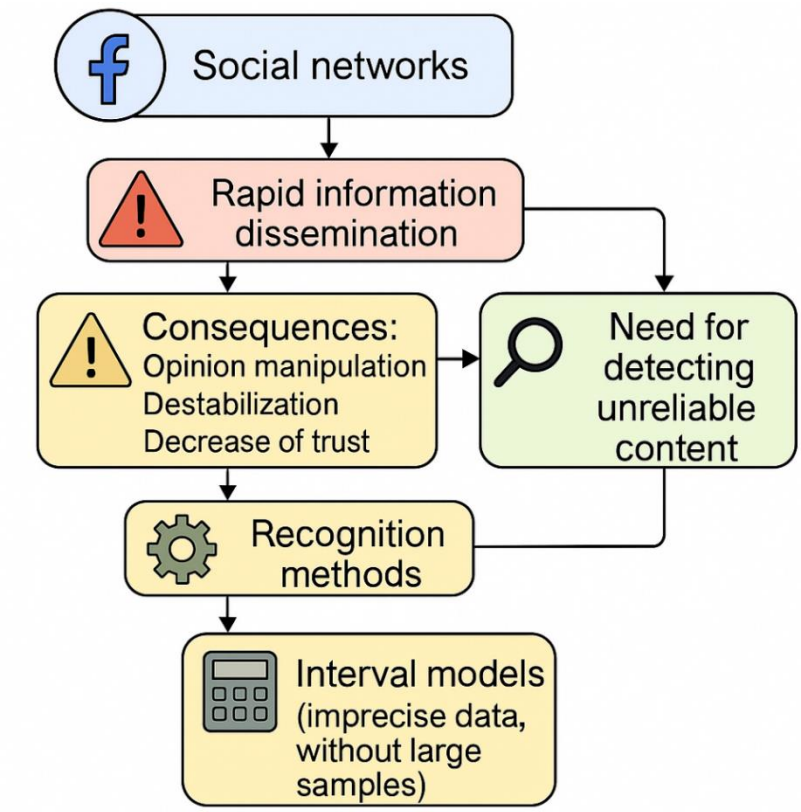

Figure 1.2. Detection of Fake Information under Conditions of Limited Data Sampling

The credibility of a news message in social networks is a complex integral characteristic that depends on multiple factors, including the trustworthiness of the source (Trust Rating), the consistency of statements across independent messages, network-based confirmation of information, the emotional tone of the content, and the temporal relevance of the reported event. The combination of these factors enables the formation of an integrated credibility index that can be used to classify content as credible, questionable, or false [82, 88, 89].

From a scientific perspective, the problem of fake content detection requires the development of hybrid credibility assessment methods that combine machine learning, linguistic analysis, ontological knowledge representation, and interval analysis. Such an integrated approach makes it possible to account for incomplete datasets, linguistic diversity, and contextual variability in social media communication.

From a practical standpoint, the creation of software tools for automated detection of unreliable or irrelevant information contributes to strengthening information hygiene

in society, improving users' media literacy, and fostering a transparent and trustworthy information environment [91, 92].

Thus, the relevance of this research stems not only from the growing volume of information in the digital space but also from the urgent need for reliable intelligent mechanisms capable of detecting, classifying, and explaining false or misleading content. The implementation of such mechanisms in the form of integrated software agents and credibility analysis systems is an essential prerequisite for the advancement of modern information-analytical technologies and for ensuring national information security.

1.2 Analysis of Methods for Fake Content Detection

In the context of global digitalization and the rapid development of social networks, the issue of information credibility has become critically important. Modern users receive the majority of news through social platforms, where information flows are characterized by high speed, emotional intensity, and the absence of centralized control over content quality. Consequently, there is an increasing need for scientifically grounded methods of automatic fake information detection, aimed at identifying messages that contain deliberately false or manipulative statements [7, 9, 10, 12, 13, 31, 84, 85].

A review of scientific literature indicates that several major directions currently exist for fake content recognition (Table 1.1). Depending on the data processing approach, the methods can be grouped into six categories: content analysis using artificial neural networks; source reliability analysis; comparison of content with verified databases; community-based fact-checking; analysis of dissemination networks; and analysis of user behavior in social media environments [11, 14, 23, 24].

Among these, neural network–based methods have gained the widest application in practical research. These approaches employ advanced deep learning architectures such as LSTM, BERT, and RoBERTa to analyze text, images, and multimedia content. Their primary advantage lies in high classification accuracy, especially when large and well-balanced training datasets are available. However, such methods also have significant limitations: they are highly sensitive to data quality and volume, require

considerable computational resources, and typically lack transparency in explaining the obtained results. This limits their applicability in cases involving small or non-representative data samples—conditions that are often typical of social media environments [33-37, 106, 111] .

Table 1.1.

Methods for detecting misinformation in social networks [7, 10, 23, 31]

Method	Advantages	Disadvantages
Content analysis using neural networks	- High accuracy provided that large and high-quality datasets are available. - Ability to detect complex and latent patterns. - Automation of the analytical process.	- High computational complexity. - Sensitivity to the quality and volume of training data. - Requires regular retraining.
Source-origin analysis	- Ability to trace the credibility and reliability of information sources. - Simplicity of result interpretation.	- Lack of access to complete metadata. - Sources may change or disappear rapidly. - Risk of using outdated or obsolete data.
Comparison with databases or a priori verified sources	- Direct matching with validated information. - High precision when reliable data are available.	- Insufficient amount of verified facts in databases. - Difficulty of searching within strict time constraints. - Data rapidly lose relevance.
Community-based verification (crowdsourcing)	- Broad user coverage and engagement. - Rapid large-scale identification of fake content.	- Participant bias. - Potential manipulation of public opinion. - Insufficient user expertise.
Propagation-network analysis	- Detection of abnormal dissemination structures. - Ability to identify bots and coordinated campaigns.	- High analytical complexity for large-scale networks. - Possibility of artificial manipulation of propagation structures.
User-behavior analysis	- Consideration of socio-psychological factors. - Ability to predict community reactions. - Use of quantitative engagement metrics (likes, comments, propagation speed).	- Requires large datasets for neural-network-based modeling. - Difficulty of modeling causal user reactions. - Possible impact of manipulative influences on user behavior.
Hybrid (combined) methods	- Synergistic effect through integration of multiple approaches. - Increased overall accuracy of results.	- System complexity increases. - Higher computational costs. - Necessity of sophisticated integration across diverse data sources and methods.

Another group of methods focuses on source analysis, where credibility is evaluated through a trust rating assigned to the author or the information resource. Parameters such as account authenticity, publication history, thematic consistency, and verification status are typically considered. However, access to metadata is not always possible, and the dynamic emergence of new sources often leads to the obsolescence of

trust ratings and a consequent decline in the effectiveness of such approaches [93, 94, 107, 108].

Content comparison methods rely on identifying matches between new messages and previously verified facts. Examples of such systems include Google Fact Check Explorer and ClaimReview. These approaches ensure a high level of reliability but suffer from a major limitation—restricted temporal and spatial coverage of events. Social media news items are often local or short-lived, making it difficult to align them with existing fact databases [70, 71, 89, 90].

A distinct line of development is represented by community-based verification methods, in which credibility is determined through collective voting or user labeling. Such systems allow for rapid responses to emerging news and enable broad public participation in the verification process. However, they remain vulnerable to bias, mass manipulation, and coordinated information campaigns that can intentionally distort credibility assessments.

An important category of approaches involves the analysis of dissemination networks, which makes it possible to trace the spread of messages and identify their sources. Using graph-based models, researchers can detect patterns of user interaction and content clustering, allowing the identification of organized fake news campaigns or bot networks. The main limitation of this approach is the restricted access to social media data and the high complexity of modeling large-scale networks [42, 43, 55, 56].

Further advances have been achieved through the analysis of community behavior in response to information events. These methods examine patterns of user activity, the speed of comment generation, the structure of discussions, and the emotional tone of interactions. They enable researchers to track the dynamics of community reactions and determine whether they correspond to typical models of natural user behavior. When deviations from these patterns occur—such as abnormally rapid message dissemination or the appearance of uniform comments—they may serve as indicators of artificial information campaigns.

Of particular interest is the quantitative modeling of user behavior. Each online community exhibits a characteristic response dynamic that can be described

mathematically, for instance, by analyzing the intensity of publications within short time intervals following the release of a message. This approach makes it possible to construct a “community profile model” and to predict future user reactions depending on the type of content. If the observed behavior significantly deviates from the expected response, the content may be classified as potentially fake [28, 32, 46, 47].

The main challenge in applying behavioral models lies in the lack of representative datasets, which prevents the use of complex neural architectures. In such cases, causal and interval models prove effective, as they allow the description of relationships between influencing factors and credibility outcomes without the need for large training datasets. The use of interval representation helps account for uncertainty and data inaccuracy—conditions typical of social media. The key advantage of this approach lies in its guaranteed estimation accuracy and result stability, even under conditions of partial data absence (Figure 1.3).

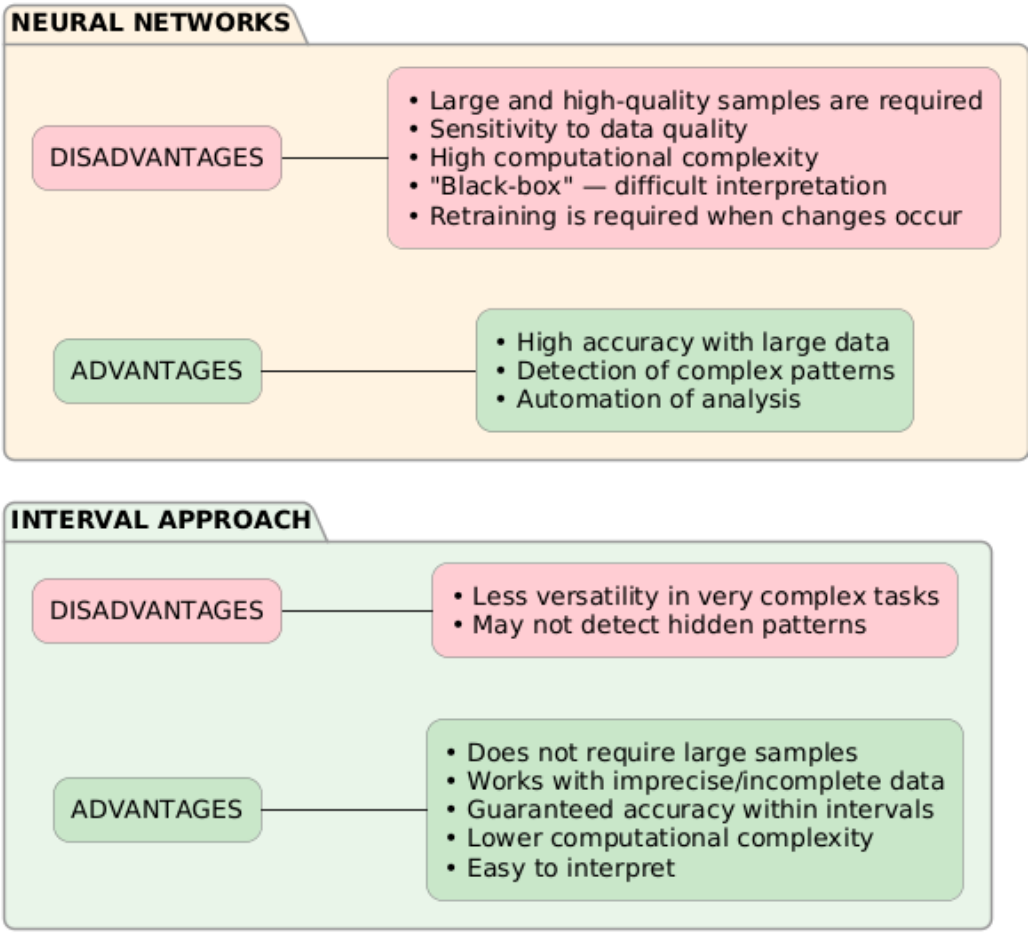

Figure 1.3. Rationale for applying the interval approach

Thus, the conducted analysis demonstrates that none of the existing methods fully solves the problem of assessing credibility in social networks. Methods focused solely on content or solely on the source provide only a partial view of the phenomenon. The most promising research direction involves the development of hybrid systems that integrate content analysis, user behavioral characteristics, dissemination network structure, and interval-based credibility modeling [21, 88].

Such approaches provide the foundation for the development of intelligent software agents capable of comprehensively analyzing news streams in social networks, detecting fake content in real time, and generating integrated credibility indicators that take into account the specific characteristics of the social media environment.

1.3 Analysis of Software Services for Fake Content Detection

The problem of recognizing false information in the digital environment has acquired a global scale, as social networks have become the primary channels for the dissemination of news, opinions, and messages. The growing speed of information exchange is accompanied by the risk of massive dissemination of unreliable or manipulative statements, which necessitates the development of effective software tools for automated fact-checking. At present, a number of services aimed at combating disinformation operate worldwide; however, each of them addresses only specific aspects of the fake content detection process [7, 49, 50, 66, 67, 75, 89].

For a systematic analysis of such solutions, it is appropriate to consider the most well-known tools—Google Fact Check Explorer, ClaimBuster, Logically Facts (AI), and Hoaxy. These systems differ in their purpose, technological approaches, level of analytical automation, and user orientation [7, 10, 23, 31, 90, 96].

The Google Fact Check Explorer service (Figure 1.4), developed within the framework of the Google News Initiative, is a convenient tool for verifying whether a given statement has already been examined by independent fact-checkers. Its main advantage lies in the rapid access it provides to verification results conducted by credible organizations using the ClaimReview metadata format. At the same time, the system does

not perform autonomous text analysis and cannot evaluate new or emerging information, which limits its applicability to the retrieval of already verified facts. Consequently, this approach does not fully meet the operational requirements of social networks, where content appears and evolves in real time.

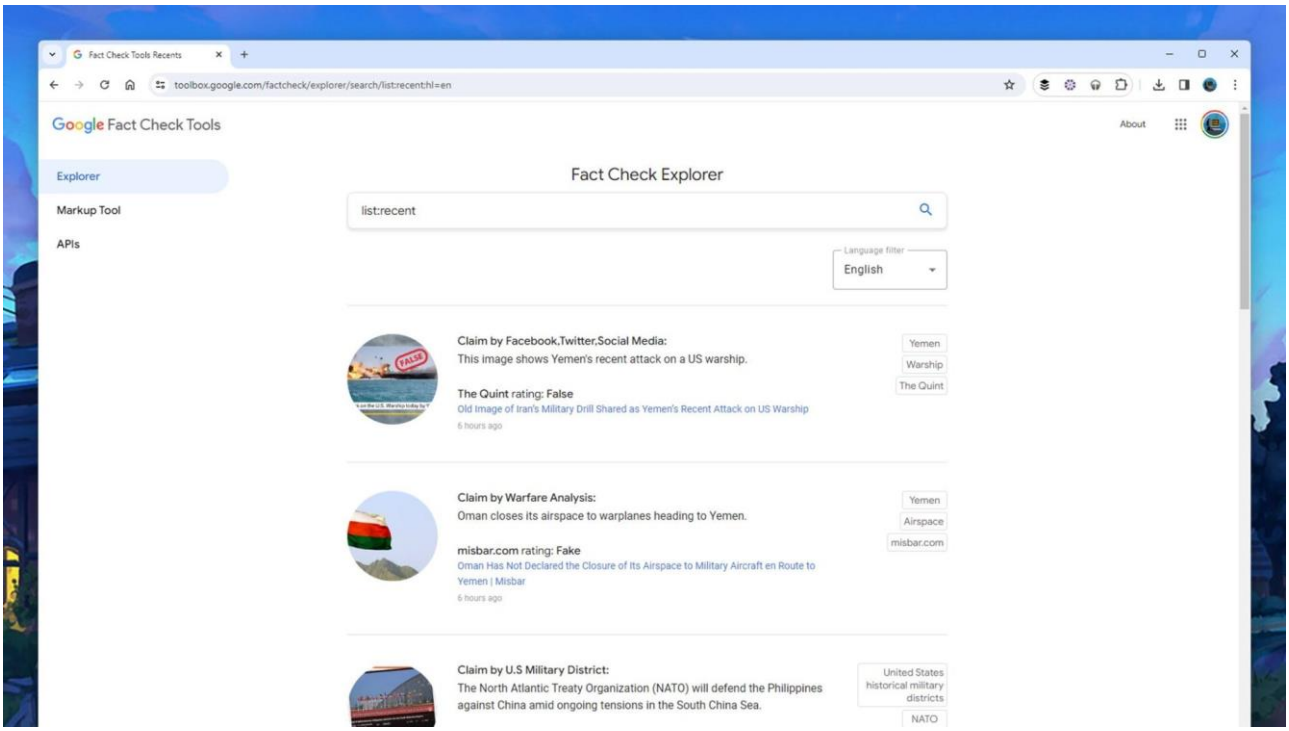

Figure 1.4. Google Fact Check Explorer

Another approach is implemented in the ClaimBuster system (Figure 1.5), which automatically identifies potentially questionable statements within text streams [7, 78]. This service employs natural language processing (NLP) and machine learning techniques to extract phrases that may require fact-checking. ClaimBuster serves as an effective tool for the initial filtering of content, helping to highlight statements that warrant further verification. However, the system does not perform the actual verification of claims and lacks mechanisms for cross-referencing with reliable data sources. Furthermore, its limited support for the English language restricts its applicability in multilingual environments.

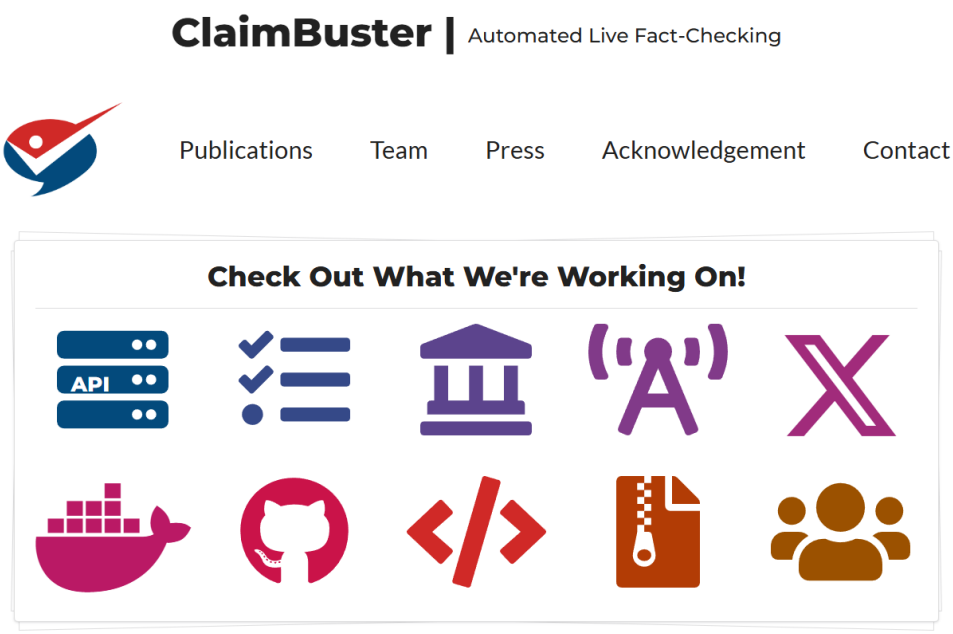

Figure 1.5. ClaimBuster

The Logically Facts (AI) platform (Figure 1.6) represents a new generation of tools that combine artificial intelligence technologies with human expertise [7, 60, 61]. It employs generative models for contextual analysis of statements and for producing explanations of verification results.

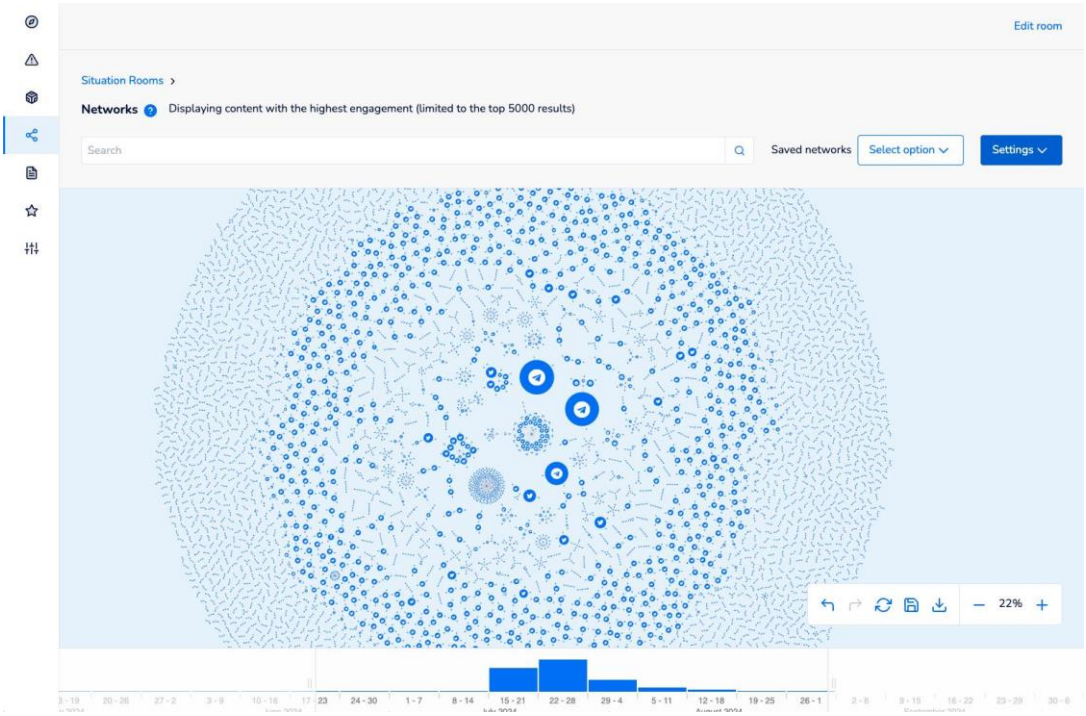

Figure 1.6. Logically Facts

The system is capable of processing multilingual content and ensures a high level of analytical accuracy. Its main advantage lies in the integration of cognitive analysis with fact-checking mechanisms, which enables deep semantic evaluation of message content. However, as a commercial product, Logically Facts provides limited access to its algorithms and API, which complicates its integration with open research systems.

A completely different concept is implemented in the Hoaxy service (Figure 1.7), which does not directly verify the factual accuracy of information but instead visualizes the process of information dissemination within social networks [62, 98, 99]. Through graph-based analysis, Hoaxy illustrates how a specific piece of news spreads among users, identifying which accounts serve as primary sources of posts and which act merely as content retransmitters. This makes it possible to detect potentially coordinated information campaigns, identify bot networks, and analyze the dynamics of viral message propagation. However, the absence of a direct mechanism for verifying the truthfulness of content limits its use primarily to the domain of research and analytical studies.

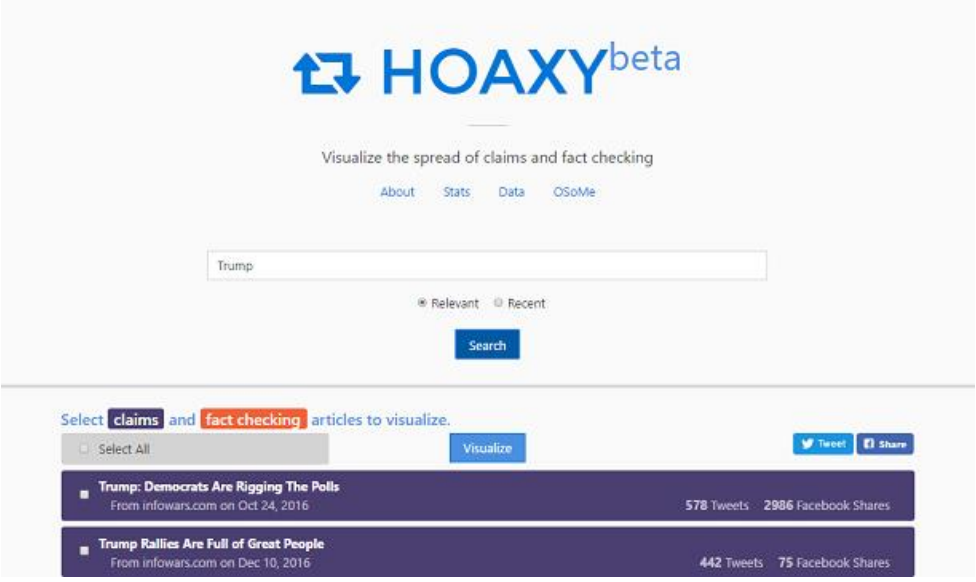

Figure 1.7. Сервіс Hoaxy

A comparative analysis shows that each of the reviewed tools addresses only a specific part of the overall credibility assessment task. Google Fact Check Explorer provides a convenient search for previously verified facts but does not perform new evaluations. ClaimBuster assists in identifying potentially suspicious statements but does

not carry out their verification. Logically Facts (AI) offers the most comprehensive approach through the use of generative artificial intelligence, yet it remains limited by restricted access to its algorithms and interfaces. Hoaxy, in turn, is unique in its ability to visualize the dissemination dynamics of false information, but it does not assess content credibility directly [23, 51, 52].

Therefore, the analysis of existing services demonstrates that none of them provides a complete cycle of fake content analysis—from data collection to credibility evaluation that accounts for user behavioral characteristics. Current tools are primarily oriented toward open-media environments, whereas social networks exhibit fundamentally different patterns of information flow, characterized by rapid content evolution, short news lifecycles, and high emotional saturation.

In this context, there arises a need to develop intelligent software agents capable of integrating the best features of existing approaches: the semantic depth and precision of Logically Facts, the speed and automation of ClaimBuster, and the network dissemination tracking capabilities of Hoaxy (Figure 1.8). Such a combination would form the foundation for a new generation of hybrid systems designed to ensure comprehensive, real-time detection and assessment of fake information in social media.

The development of such a software environment, integrating content collection agents, credibility analysis modules, and result visualization components within a unified modular architecture, will enable a comprehensive approach to monitoring news streams in social networks. In this context, a particularly important role is played by the use of interval modeling of user behavior, which makes it possible to account for data uncertainty and sample incompleteness.

Therefore, the conducted analysis of existing software services confirms both the scientific and practical necessity of developing an intelligent credibility assessment system. Such a system should integrate state-of-the-art artificial intelligence methods, data collection mechanisms from social networks, and interval-based trust evaluation models, thereby enhancing the efficiency of fake content detection in a dynamic news environment.

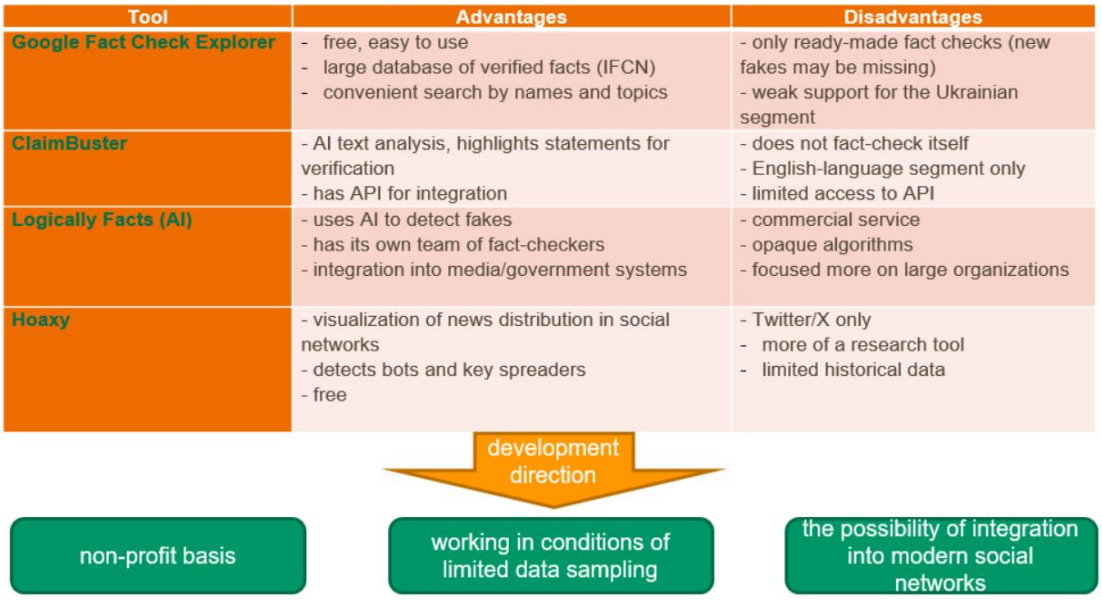

Figure 1.8. Tools for analyzing fake content in news social networks

1.4 Problem Statement of the Research

Modern social networks have become one of the main sources of news dissemination, significantly influencing public opinion, information security, and the level of trust in media. At the same time, the openness of social platforms, the high speed of information circulation, and the lack of centralized verification mechanisms create favorable conditions for the spread of false, manipulative, or irrelevant information. Traditional approaches to content credibility assessment—focused on static data sources or large training datasets—prove to be ineffective under the dynamic conditions of social networks, where data volume, quality, and structure are often irregular and incomplete.

In this regard, an urgent scientific problem consists in enhancing the efficiency of fake content detection and analysis in news-oriented social networks under conditions of limited data sampling. Its solution requires the integration of interval, ontological, and behavioral approaches combined with the use of software agents capable of automated data collection, evaluation, and validation from multiple web sources in real time.

To achieve the stated aim, the following scientific and applied objectives must be accomplished:

1. To analyze existing methods and software tools for recognizing false or

irrelevant information in the content of web resources, particularly in social networks, identifying their strengths and weaknesses, application areas, and limitations under incomplete data conditions.

2. To justify and develop a hybrid method for identifying interval-based models of social network user profiles that accounts for temporal patterns, behavioral activity, and the specifics of information dissemination in the online environment.

3. To design an interval mathematical model establishing an analytical relationship between the credibility index of content and the set of influencing factors—specifically source reliability, network confirmation level, and temporal relevance.

4. To improve software agents for content credibility assessment in news-oriented social networks by integrating both classical verification criteria and those specific to the digital environment—network support, indicators of user behavioral authenticity, and interval estimates of activity.

5. To develop a software environment for detecting and analyzing fake content that enables interaction between data collection agents, analytical credibility-evaluation modules, and subsystems for data storage, visualization, and explanation of results. The software environment should be scalable, reproducible, and compatible with the open APIs of social networks.

6. To conduct experimental validation of the developed methods and tools using real data from news-oriented social networks (Facebook, X/Twitter, Telegram) in order to confirm the effectiveness, robustness, and explainability of the credibility-analysis results.

Conclusion of Chapter 1

1. The essence of the problem of fake information detection in the modern digital environment has been defined. It has been demonstrated that social networks represent dynamic, open systems with high content dissemination intensity, creating favorable conditions for the emergence and replication of false messages. The main factors that complicate automated credibility verification include data unstructuredness,

multilingualism, emotional expressiveness, and the presence of anonymous or automated accounts.

2. A classification of existing fake content detection methods has been conducted. Six main groups of approaches have been identified: methods based on artificial neural networks; methods for analyzing information source origins; methods for comparison with verified knowledge bases; community-based content labeling approaches; dissemination network analysis methods; and user behavior analysis methods. It has been shown that none of these approaches alone ensures high accuracy under conditions of limited data sampling and high social media dynamics.

3. The advantages and limitations of existing content verification services have been identified. A comparative analysis of Google Fact Check Explorer, ClaimBuster, Logically Facts (AI), and Hoaxy demonstrated that each system has a narrow specialization: Google Fact Check Explorer focuses on retrieving previously verified facts; ClaimBuster detects potentially questionable statements in text; Logically Facts (AI) employs artificial intelligence but does not integrate behavioral features; Hoaxy visualizes fake news dissemination networks but does not assess credibility. None of these services consider user activity dynamics, interval variability of parameters, or the emotional tone of messages—factors that are critical for social network analysis.

4. The necessity of developing a hybrid approach to information credibility assessment has been substantiated. This approach should combine classical fact-checking criteria with behavioral, network-based, and emotional indicators. It must ensure robustness to information noise, adaptability to environmental changes, and the ability to operate effectively under conditions of limited or partially incomplete data samples.

5. The research problem statement has been formulated, focusing on improving the efficiency of fake content detection and analysis in news-oriented social networks by developing a hybrid method for identifying interval-based user models and creating software agents for credibility assessment.

CHAPTER 2
MODELING USER PROFILES IN A SOCIAL NETWORK BASED ON INTERVAL DATA ANALYSIS

The previous section examined existing methods and tools for detecting unreliable content in social networks. It was noted that a key aspect of these tasks is the identification of user profiles, characterized by typical behavioral responses to credible content and, conversely, by atypical reactions when interacting with false or misleading information.

This section proposes the use of a mathematical model for decision-making regarding the credibility of social network content, based on establishing relationships between the decision outcome (credibility or falsity) and the influencing factors. The main quantitative factors are proposed to include:

– the number of posts, shares, or likes generated by users within a short period after the content appears;

– the number of comments or reactions within specific time intervals;

– the time required for the information to spread through the network (e.g., the number of users interacting with the content within the first minutes, hours, or days after publication); and

– the coefficient of viral dissemination, such as the average number of reshares per user.

The resulting indicator of this model is the degree of content credibility, expressed within the range from 0 to 1. To represent and analyze this indicator—based on expert evaluation of content credibility—it is proposed to apply interval data analysis methods.

Further in this section, an optimization problem is formulated for the two-stage identification of the model based on interval data analysis. The first stage involves constructing the current model structure from candidate models (model structure synthesis), while the second stage focuses on parameter estimation and model adequacy validation.

A hybrid method for identifying interval-based user profile models in social networks is proposed and substantiated. The method combines a metaheuristic model

structure synthesis algorithm, inspired by the behavioral model of a bee colony, with gradient-based techniques for parameter identification of candidate models.

In the final part of this section, the practical application of the proposed hybrid identification method for interval-based user profile modeling in social networks is presented. The feasibility of credibility decision-making based on the constructed model is demonstrated.

The results of this research have been published by the author in [20, 21, 22].

2.1 Problem Statement of Interval Model Identification for User Profiles in Social Networks

Social networks are among the most widespread means of obtaining information. However, the information disseminated within these networks is not always reliable. Modern social platforms frequently host and propagate fake or misleading content [7, 23, 88, 95, 96]. Such cases are often observed during the coverage of political events, military conflicts, emergencies, or pandemics, when the information space becomes saturated with unverified messages, emotional comments, and manipulative publications — often disseminated deliberately. Therefore, the detection of unreliable information in social networks remains an urgent research challenge.

Existing methods for detecting fake information can be grouped as follows [7, 21, 23, 107, 110]:

– content-analysis methods using artificial neural networks;

– methods for analyzing information source reliability;

– approaches that verify content by comparison with verified databases or a priori reliable knowledge bases;

– community-based verification methods involving collective labeling of unreliable content;

– dissemination-network analysis methods;

– community behavior analysis approaches.

Additionally, various hybrid methods that combine the above techniques may be

employed. Among these, content-analysis methods based on neural networks are the most common; however, they suffer from major drawbacks such as sensitivity to data quality and volume in the training set and high computational complexity.

The main disadvantages of source-analysis methods include outdated or incomplete metadata, restricted access to source information, and the rapid emergence of new or dynamically changing sources, which reduce their effectiveness. Similarly, fact-verification methods based on comparison with database records are limited by the scarcity of verified facts and temporal constraints in available data.

Community-based verification approaches are prone to bias, limited domain knowledge, and susceptibility to manipulation, while dissemination-network analysis methods are challenged by complex graph structures and the potential for coordinated manipulation of participant communities.

The study in [20, 21, 63] explored the modeling of community behavior dynamics in response to various content types. This approach can serve as a foundation for assessing content credibility, as it relies on identifying typical community reactions to particular types of information. Although such methods also have limitations, they enable the construction of a quantitative community portrait, for instance, by analyzing the temporal dynamics of reactions. Based on the initial audience response, it becomes possible to simulate further behavioral trends and, consequently, to infer the credibility of the information. Similar approaches are discussed in [24, 25, 47], though they primarily focus on reaction dynamics rather than on the underlying causal factors driving those reactions.

User behavior can also be modeled quantitatively — for example, by estimating the number of posts generated shortly after content publication. However, behavioral modeling faces a key limitation: the lack of sufficiently large datasets, which makes the use of neural networks impractical. In such cases, causal models can be effectively applied, representing relationships between the credibility-assessment result and its influencing factors through algebraic equations.

Moreover, the quantitative data used for decision-making are often imprecise or uncertain, making interval representation particularly appropriate. Building interval-based models [15, 18, 19, 75] offers clear advantages: they do not require large datasets

and provide guaranteed mathematical accuracy [20,21].

Consequently, the relevant research problem is the development of an interval model for decision-making regarding the credibility of social-network content, based on establishing the relationship between the decision result (credible or not credible) and the influencing factors [21, 22].

Quantitative methods make it possible to construct predictive models of user behavior derived from previously observed reactions. For example, they can capture which topics generate the greatest engagement or how quickly certain content becomes viral.

To construct such a quantitative model, let us introduce the variable y, which characterizes the degree of content credibility within the range 0,1 where 0 represents completely false content and 1 fully reliable content. This indicator is treated not as a probability but as a quantitative measure defined over an interval.

The quantitative factors that determine the degree of content credibility may include:

x_1 - the number of posts, shares, or likes generated by users within a short period after publication, reflecting immediate audience reactions. Abnormally high or atypical early engagement may indicate a high degree of unreliability. For example, if certain posts receive numerous negative responses or are flagged as fake, this serves as an important indicator;

x_2 - the number of comments or reactions recorded at specific time intervals, providing insight into the rate of information diffusion and the emotional feedback of users. Unusually fast propagation often correlates with unreliable or artificially amplified content;

x_3 - the time required for information to spread through the network (e.g., how many users interact with the content during the first minutes, hours, or days after publication). This helps to evaluate the efficiency and authenticity of dissemination. Fake news tends to spread rapidly within confined groups or via bot networks;

x_4 - the viral dissemination coefficient (e.g., the average number of reshares per user), which indicates how prone the content is to rapid replication and propagation.

These factors collectively describe a community portrait, representing typical or atypical user reactions to posted content, and can serve as the basis for determining the degree of credibility or falsity of that content. The influence of uncontrolled external factors on community reactions is represented through interval estimates of credibility or falsity levels (Figure 2.1).

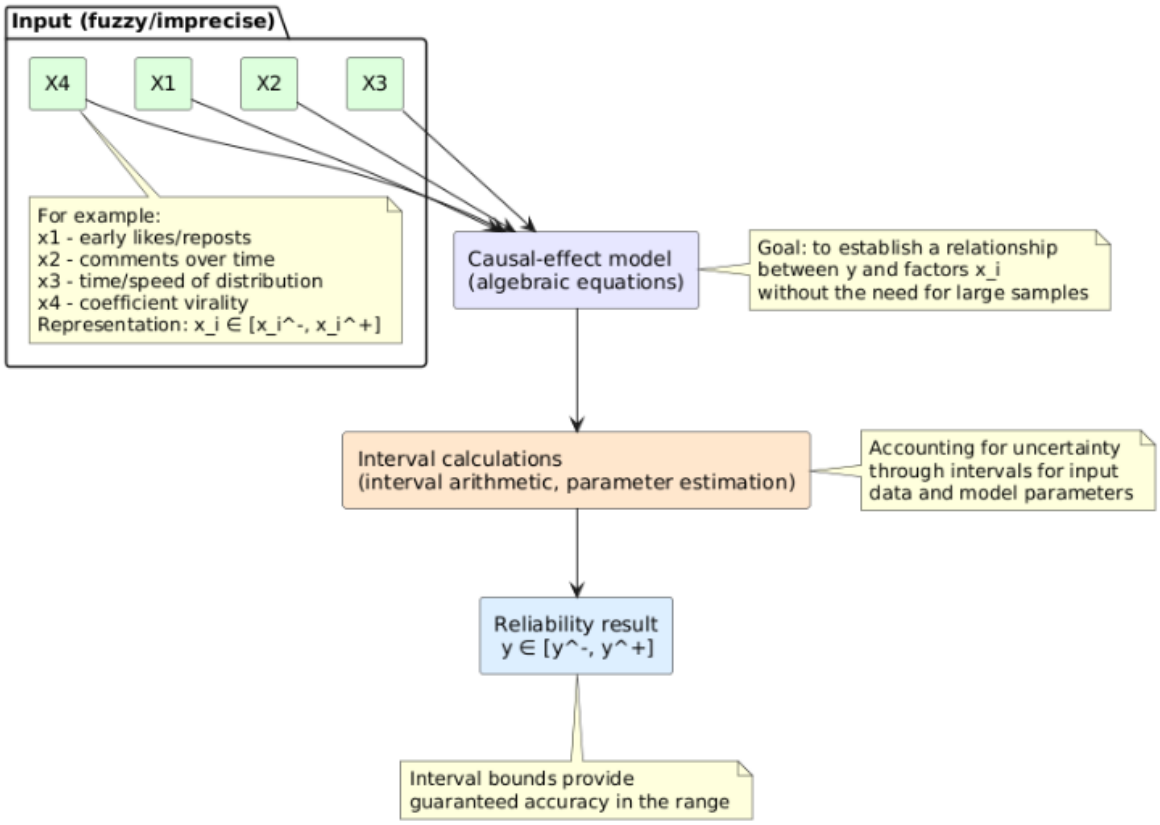

Figure 2.1. Interval Modelling of Social Network User Behaviour

The dependence of the indicator $y(\vec{X})$, which characterizes the degree of credibility of a given content item, on the values of the influencing factors $\vec{X} = (x_1, x_2 x_3 x_4)$ is expressed, in general form, by a nonlinear algebraic equation [5]:

$$y(\vec{X}) = f_1(\vec{\beta}, \vec{X}) + f_2(\vec{\beta}, \vec{X}) + \cdots + f_m(\vec{\beta}, \vec{X}), \tag{2.1}$$

where $y(\vec{X})$ – is the modeled value of content credibility; $\vec{\beta}$ – is the unknown vector of parameters of the interval model; $\lambda_{m_s} = \{f_1(\vec{\beta}, \vec{X}), f_2(\vec{\beta}, \vec{X}), \dots, f_m(\vec{\beta}, \vec{X})\}$ – denotes the

set of basis functions, which are generally nonlinear both with respect to the input factors and the model parameters and m_s – is the number of basis functions, i.e., the structural elements of the model.

The data for the identification of the mathematical model (1) are obtained in the following form:

$$\vec{X}_i \to [y_i^-; y_i^+], i = 1, \dots, N, \tag{2.2}$$

where $[y_i^-; y_i^+]$ – denote the lower and upper bounds of the credibility degree for the i-th content item, $i = 1, \dots, N$; $\vec{X}_i$ – represents the vector of factors that describe the portrait, i.e., the community's reaction to the i-th published content item, which can serve to determine the degree of its credibility or falsit; and N – is the total number of observations in the experiment.

Based on expressions (2.1) and (2.2), the conditions for solving the structural and parametric identification problem of the model can be derived as follows:

$$y_i^- \le f_1\left(\vec{\hat{\beta}}, \vec{X}_i\right) + \dots + f_m\left(\vec{\hat{\beta}}, \vec{X}_i\right) \le y_i^+, i = \overline{1, N}, \tag{2.3}$$

As can be seen, the conditions from which both the structure and parameters of the model are obtained represent an interval system of nonlinear algebraic equations (ISNAE). As is known, its solution forms a set of mathematical models, commonly referred to as a corridor of admissible solutions. The problem of solving an ISNAE is an NP-hard computational task. To simplify the process, we shall search only for point estimates of the parameter vector $\vec{\hat{\beta}}^m$ corresponding to a single candidate model that satisfies the conditions of expression (2.3).

Under these conditions, both structural and parametric identification are based on optimization problems, which are solved using methods of multidimensional nonlinear optimization [15, 18, 19, 20, 21, 22]. In this context, the identification of the model (its structure and parameters) is formulated as the following optimization problem [21, 39, 40]:

$$\delta\left(\lambda_{m_s}, \vec{\beta}^m, \vec{\alpha}\right) \xrightarrow{\lambda_{m_s}, \vec{\beta}^m, \vec{\alpha}} min \tag{2.4}$$

$$\hat{\beta}^m{}_j \in \left[\hat{\beta}^m{}_j^{low}; \hat{\beta}^m{}_j^{up}\right], j = 1, \dots, m \tag{2.5}$$

$$\lambda_{m_s} \in F, \tag{2.6}$$

$$\alpha_i \in [0,1], i = 1, \dots, N, \tag{2.7}$$

where F – denotes the set of all possible structural elements of the interval model; λ_{m_s} – is the number of all possible elements of the s-th structure; α_i – are the coefficients of the linear combination used to determine a specific point within the credibility degree interval $[y_i^-; y_i^+]$ for the i-th content item.

For each current structure—treated as a candidate model—which is generated by algorithms performing directed enumeration of structural elements, the model parameters are estimated in order to obtain quantitative estimates of the predicted (computed) credibility degrees of the i-th content items. These are expressed by the following relation:

$$\hat{y}_i\left(\vec{X}_i\right) = f_1\left(\vec{\hat{\beta}}_1, \vec{X}_i\right) + f_2\left(\vec{\hat{\beta}}_2, \vec{X}_i\right) + \cdots + f_m\left(\vec{\hat{\beta}}_m, \vec{X}_i\right), i = 1, \dots, N, \tag{2.8}$$

and are compared with the given (reference) values using the following objective function:

$$\delta\left(\lambda_{m_s}, \vec{\beta}^m, \vec{\alpha}\right) = \sum_{i=1}^{N}\left(\hat{y}_i\left(\vec{X}_i\right) - P([y_i^-; y_i^+], \alpha_i)\right)^2, \tag{2.9}$$

where

$$P([y_i^-; y_i^+], \alpha_i) = \alpha_i \cdot y_i^- + (1 - \alpha_i) \cdot y_i^+, i = 1, \dots, N. \tag{2.10}$$

As an additional stopping criterion for the optimization procedures, the following conditions may be used [18, 21]:

$$\hat{y}_i\left(\vec{X}\right) \in [y_i^-; y_i^+], i = 1, \dots, N. \tag{2.11}$$

It should be noted that the obtained optimization problem (2.4)–(2.7) can be solved by combining global search methods with gradient-based optimization techniques.

2.2 Hybrid Method for the Identification of Interval-Based User Profile Models in Social Networks

Given that the optimization problem (2.4)–(2.7) is a nonlinear optimization problem, it also includes two types of constraints:

$$\lambda_m \in F \tag{2.12}$$

and

$$\hat{\beta}^m{}_j \in \left[\hat{\beta}^m{}_j^{low}; \hat{\beta}^m{}_j^{up}\right], j = 1, \dots, m \tag{2.13}$$

$$\alpha_i \in [0,1], i = 1, \dots, N, \tag{2.14}$$

it is therefore advisable to transform it into the following form:

$$\Phi\left(\lambda_{m_s}, \vec{\hat{\beta}}^m, \vec{\alpha}, \mu, \gamma\right) \xrightarrow{\lambda_{m_s}, \vec{\hat{\beta}}^m, \vec{\alpha}, \mu, \gamma, \sigma} min \tag{2.15}$$

$$\lambda_{m_s} \in F = \left\{ \varphi_1(\vec{x}), \dots, \varphi_m(\vec{x}), \varphi_{m+1}(\vec{g}), \dots, \varphi_{2m}(\vec{g}) \right\} \tag{2.16}$$

As can be seen, two unknown coefficients μ, γ are introduced into the objective function through the aggregation of linear constraints on the values of the parameter vector $\vec{\hat{\beta}}^m$ and the coefficients $\vec{\alpha}$ by means of penalty functions, expressed as follows:

$$£ = \gamma \cdot \sum_{j=1}^{N} \left(\ln \left(\hat{\beta}^m{}_j - \hat{\beta}^m{}_j^{low} \right) + \ln \left(\hat{\beta}^m{}_j^{up} - \hat{\beta}^m{}_j \right) \right), \quad (2.17)$$

$$\Delta = \mu \cdot \sum_{i=1}^{N} (\ln(\alpha_i) + \ln(1 - \alpha_i)), \quad (2.18)$$

where γ, μ - are the given weighting coefficients corresponding to the respective penalty functions.

As a result, the objective function in the interval model identification problem takes the following form:

$$\Phi\left(\lambda_{m_s}, \vec{\beta}, \vec{\alpha}, \mu, \gamma\right) = \sum_{i=1}^{N} \left(\hat{y}_i\left(\vec{X}_i\right) - P([y_i^-; y_i^+], \alpha_i) \right)^2 - \gamma \cdot \sum_{j=1}^{N} \left(\ln \left(\hat{\beta}^m{}_j - \hat{\beta}^m{}_j^{low} \right) + \ln \left(\hat{\beta}^m{}_j^{up} - \hat{\beta}^m{}_j \right) \right) - \mu \cdot \sum_{i=1}^{N} (\ln(\alpha_i) + \ln(1 - \alpha_i)). \quad (2.19)$$

Thus, a barrier-type objective function is obtained, the value of which increases (or decreases) sharply as the parameters $\vec{\hat{\beta}}^m$ or the coefficients $\vec{\alpha}$ approach their boundary values.

At this stage, the interval model identification problem, formulated in expressions (2.15) and (2.16) with the objective function (2.19) represented as a barrier-type function, is an optimization problem defined over a discrete set of basis functions of the mathematical model. To solve this problem, it is necessary to apply a combination of directed enumeration methods for structural elements and gradient-based optimization techniques.

In this study, a hybrid method for identifying the interval-based user profile model in social networks is proposed. The method is based on the integration of a metaheuristic algorithm for model structure synthesis—developed on the principles of the bee colony

behavioral model—and gradient methods for parameter identification of the candidate models. Let us consider the proposed method in detail; its implementation scheme is presented in Figure 2.1.

irst of all, according to condition (2.16), the set of basis functions λ_m for a specific candidate model is formed from the general set F, which contains all the structural elements from which the interval model can be constructed. Increasing the number m of structural elements in the mathematical model may lead to a significant increase in its complexity. Therefore, in the optimization problem (2.15), (2.16), it is necessary to introduce an additional constraint concerning the upper limit on the number of structural elements in the model. This constraint can be expressed as follows:

$$m_s \le I_{\max}, \tag{2.20}$$

m_s - the number of structural elements in the current s-th structure;

$I_{\max}$ - the maximum allowable number of structural elements in the candidate models.

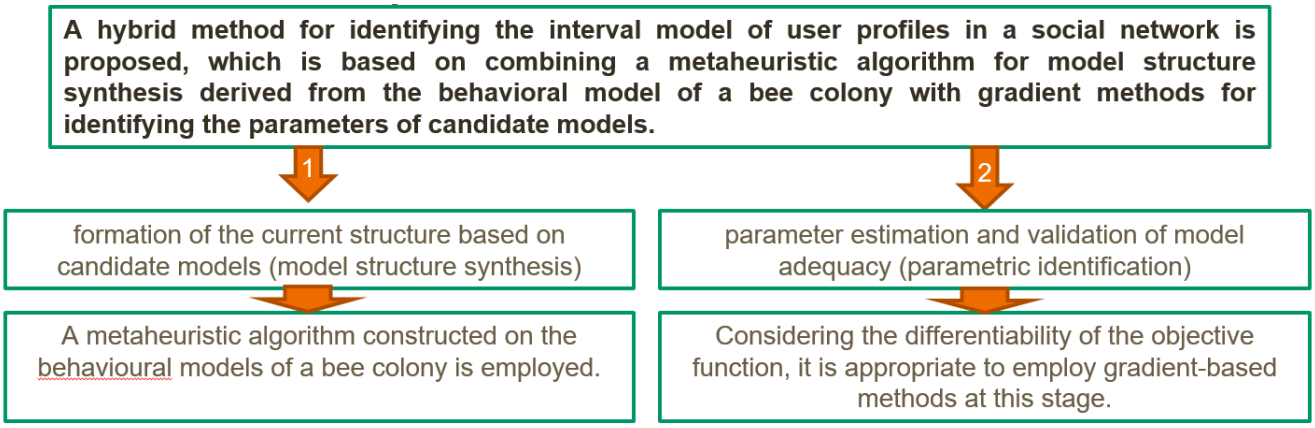

Figure 2.2. Hybrid Method for Identifying Interval Models of User Profiles in a Social Network

Let us introduce constraint (2.20) into the objective function (2.19) in the form of a penalty function. As a result, we obtain:

$$\Phi\left(\lambda_{m_s}, \vec{\hat{\beta}}^{m}, \vec{\alpha}, \mu, \gamma, \sigma\right) == \sum_{i=1}^{N}\left(\hat{y}_i\left(\vec{X}_i\right) - P([y_i^{-}; y_i^{+}], \alpha_i)\right)^2 - \gamma \cdot \sum_{j=1}^{N}\left(\ln\left(\hat{\beta}^{m}{}_j - \right.\right.$$
$$\left. - \hat{\beta}^{m}{}_j^{low}\right) + \ln\left(\hat{\beta}^{m}{}_j^{up} - \hat{\beta}^{m}{}_j\right)\Big) - \qquad (2.21)$$
$$- \mu \cdot \sum_{i=1}^{N}(\ln(\alpha_i) + \ln(1-\alpha_i)) - \sigma \cdot \ln\left(I_{\max} - m_s\right),$$

where σ - the given weighting coefficient of the penalty function.

Then, the optimization problem (2.16), (2.17) can be rewritten in the following form:

$$\Phi\left(\lambda_{m_s}, \vec{\hat{\beta}}^{m}, \vec{\alpha}, \mu, \gamma, \sigma\right) \xrightarrow{\lambda_{m_s}, \vec{\hat{\beta}}^{m}, \vec{\alpha}, \mu, \gamma, \sigma} min \qquad (2.22)$$

$$\lambda_{m_s} \in F. \qquad (2.23)$$

Now, the process of solving the optimization problem (2.22), (2.23) is divided into two stages:

- formation of the current structure based on candidate models (model structure synthesis);
- estimation of its parameters and verification of model adequacy (parametric identification).

At the first stage, a metaheuristic algorithm based on the behavioral model of a bee colony is used [18, 20, 21, 40, 46, 48, 74, 108]. The application of this algorithm makes it possible to form and evaluate several different candidate model structures in parallel.

Formally, the bee colony algorithm is grounded in the concept of swarm intelligence, which simulates the behavior of a bee swarm searching for nectar. Let us consider the main phases of the synthesis of candidate model structures by analogy with the stages of the behavioral model of a bee colony.

Initialization Phase. According to the optimization problem (2.22), (2.23), the following initial conditions are defined:LIMIT – the number of iterations before the

current candidate model structure is abandoned; *S* the total number of candidate models within one iteration; $I_{\max}$; *mcn = 0* – the index of the current iteration; MCN - the total number of iterations; F - the set of all structural elements. A random initial population of S candidate model Λ_0 structures λ_{m_s} the candidate models, with a total number of *F*.

Employed Bees Phase. At this phase, a set of candidate model structures is generated using a series of transformation operators. The operator $P(\Lambda_{mcn},F)$, transforms each structure λ_{m_s} from the set Λ_{mcn} of interval models defined in (2.8) into a new structure λ'_{m_s}, which is similar to λ_{m_s}. This operation corresponds to the Artificial Bee Colony (ABC) algorithm, in which employed bees explore the neighborhood of food sources in search of better solutions. As a result, the operator $P(\Lambda_{mcn},F)$ transforms the set of structures Λ_{mcn} into a new set Λ'_{mcn} at the mcn-th iteration of the structural synthesis algorithm. The transformation occurs by randomly selecting elements within each structure λ_{m_s} and replacing them with elements randomly drawn from the overall set *F*. Using the operator $P(\Lambda_{mcn},F)$ a variable number of elements can be replaced in the current candidate structure, depending on the quality of that structure. The quality of the current candidate model is determined by the value of the objective function: $\Phi\left(\lambda_{m_s},\vec{\tilde{\beta}}^{m},\vec{\alpha},\mu,\gamma,\sigma\right)$. It is evident that the smaller the value of this function, the more accurate the mathematical model becomes; consequently, fewer elements need to be modified in the current structure.

Therefore, the formula for determining the number of elements that must be replaced in the current structure can be expressed as follows:

$$n_s=\begin{cases}\operatorname{int}\left(\left(1-\dfrac{\min\left\{\Phi(\lambda_{m_s},\vec{\tilde{\beta}}^{m},\vec{\alpha},\mu,\gamma,\upsilon)\,\middle|\,s=1\ldots S\right\}}{\Phi(\lambda_{m_s},\vec{\tilde{\beta}}^{m},\vec{\alpha},\mu,\gamma,\upsilon)}\right)\cdot m_s\right), \\ \quad if\ \Phi(\lambda_{m_s},\vec{\tilde{\beta}}^{m},\vec{\alpha},\mu,\gamma,\upsilon)\neq\min\left\{\Phi(\lambda_{m_s},\vec{\tilde{\beta}}^{m},\vec{\alpha},\mu,\gamma,\upsilon)\,\middle|\,s=1\ldots S\right\}\ and\ n_s\neq 0; \\ 1,\quad if\quad \Phi(\lambda_{m_s},\vec{\tilde{\beta}}^{m},\vec{\alpha},\mu,\gamma,\upsilon)=\min\left\{\Phi(\lambda_{m_s},\vec{\tilde{\beta}}^{m},\vec{\alpha},\mu,\gamma,\upsilon)\,\middle|\,s=1\ldots S\right\}\ or\ n_s=0.\end{cases} \quad (2.24)$$

It should be noted that the number of elements replaced in the current structure also depends on the total number of elements contained within that m_s structure.

During this same phase, pairwise comparison between the newly generated and the current structures is performed in order to select the better candidate from each pair.

Evidently, this requires performing parametric identification for each candidate model using gradient-based optimization methods to evaluate their performance.

Thus, the pairwise comparison operator for selecting the superior structure can be represented as follows:

$$D_1(\lambda_{m_s}, \lambda'_{m_s}) : \lambda^1{}_{m_s} = \begin{cases} \lambda_{m_s}, if \;\; \Phi(\lambda_{m_s}, \vec{\bar{\beta}}^m, \vec{\alpha}, \mu, \gamma, \upsilon) \le \Phi(\lambda'_{m_s}, \vec{\bar{\beta}}^m, \vec{\alpha}, \mu, \gamma, \upsilon) \\ \lambda'_{m_s}, if \;\; \Phi(\lambda_{m_s}, \vec{\bar{\beta}}^m, \vec{\alpha}, \mu, \gamma, \upsilon) > \Phi(\lambda'_{m_s}, \vec{\bar{\beta}}^m, \vec{\alpha}, \mu, \gamma, \upsilon) \end{cases} \tag{2.25}$$

The operator defined in (2.25) performs the selection of the best structures from two sets by means of pairwise comparison of the structures from the sets Λ_{mcn}, Λ'_{mcn}. As a result, a new set of structures for the first iteration, denoted as $\lambda^1_{m_s} \in \Lambda^1_{mcn}$ is obtained.

Onlooker Bees Phase At this phase, a set of new structures is generated in the neighborhood of the selected structures from the set Λ^1_{mcn}. In the context of the behavioral model of the bee colony, the onlooker bees explore new nectar sources in the vicinity of already known ones.

This implies that for each current structure, it is necessary to generate a certain number of neighboring structures R_s. The number of such generated structures depends on the quality of the current structure — that is, on the value of the objective function.

Accordingly, the number of structures generated for a given current structure can be determined using the following formulas:

$$P_s(\lambda^1{}_{m_s}) = \frac{1}{\Phi(\lambda^1{}_{m_s}, \vec{\bar{\beta}}^m, \vec{\alpha}, \mu, \gamma, \upsilon) \cdot \sum_{s=1}^{S} \frac{1}{\Phi(\lambda^1{}_{m_s}, \vec{\bar{\beta}}^m, \vec{\alpha}, \mu, \gamma, \upsilon)}}, s = 1...S-1. \tag{2.26}$$

$$R_s = ToInt(P_{s-1}(\lambda^1{}_{m_{s-1}}) \cdot S), \quad s = 2...S, \quad R_1 = 0. \tag{2.27}$$

Having determined the number of structures to be generated around each current one, the operator $P_\delta(\Lambda^1_{mcn}, F)$ is applied. This operator functions similarly to the previously defined transformation operator $P(\Lambda_{mcn}, F)$, converting the set of structures Λ^1_{mcn} into a new set Λ'_s.

For each structure $\lambda^1{}_{m_s}$, a corresponding number of new structures nin_ini is formed by randomly replacing a specified number of structural elements, which are themselves randomly selected from the general set *F*.

Next, during this phase, the group selection operator $D_2(\lambda^1{}_{m_s}, \Lambda'_s)$ is employed. It selects the best candidate model between the current structure and the group of newly generated neighboring structures $\Lambda'_s = \{\lambda_1, ..., \lambda_r, ..., \lambda_{R_s}\}$. This is achieved through in-group selection, based on the evaluation of the objective function (2.21) for each structure using parametric identification procedures.

As a result, the operator Λ^2_{mcn} selects the best-performing structure among those generated in the local group. In this process, Formula (2.25) is reused, with the difference that selection is performed within a group, not pairwise. Consequently, this operator produces the second-order structure set Λ^2_{mcn} during the same iteration mcnmcnmcn.

One of the major challenges of the above-described algorithm is the risk of stagnation — that is, getting trapped in local minima of the optimization problem. To overcome this limitation, the Artificial Bee Colony (ABC) algorithm introduces an additional phase — the Scout Bees Phase, which enables the exploration of new regions of the search space and helps the algorithm escape local minima.

Scout Bees Phase. The Scout Bees Phase corresponds to the stage of the bee colony behavior in which bees randomly explore new nectar sources. In the context of the optimization problem, this means that for certain structures, it is necessary to generate completely new structures at random. In the computational implementation, each current

candidate structure is associated with a variable $Limit_s$, which models the exhaustion of the structure and triggers a complete replacement of its elements through random generation when necessary. A structure is considered exhausted when the number of its modifications exceeds the preset limit $LIMIT$ without improving its quality. In such a case, the operator $P_N(I_{\max}, F)$, is applied to generate a new structure by forming a random set of structural elements.

Thus, the described scheme enables the gradual formation of new candidate interval models, while continuously improving their quality in the direction of the global optimum. As noted earlier, for each fixed current candidate structure, the parametric identification problem is solved for its fixed structural elements by means of an optimization procedure. Given the differentiability of the objective function (2.21), gradient-based methods can be effectively employed at this stage.

Therefore, at the second stage of interval model identification, the algorithmic scheme for parametric identification can be represented as follows [21]:

Step 1. *Initialization:*

- Input of experimental data: $\vec{X}_i \to [y_i^-; y_i^+],\ i = \overline{1,N}$;
- Reading the current model structure λ_{m_s} (A set of structural elements with a total number of m_s) is defined from the first stage;
- Assignment of initial values for the components of the parameter vector $\hat{\beta}^m{}_j \in \left[\hat{\beta}^m{}_j^{low}; \hat{\beta}^m{}_j^{up}\right], j = 1, \dots, m$;
- Assign initial values for the components of the vector $\vec{\alpha}$, Initialization of the coefficients ($\alpha_i = 0.5,\ i = \overline{1,N}$);
- Define the initial values of the penalty function coefficients μ, γ, σ the penalty coefficients;
- Formation of the objective function: $\Phi\left(\lambda_{m_s}, \vec{\hat{\beta}}^m, \vec{\alpha}, \mu, \gamma, \sigma\right)$;
- Specification of stopping criteria:

Step 2. *Loop organization***.**

Start Step.

While none of the stopping criteria are satisfied, perform:

Step 2.1. *Barrier function update*

$\Phi\left(\lambda_{m_s}, \vec{\hat{\beta}}^m, \vec{\alpha}, \mu, \gamma, \sigma\right)$ *to the formula* (2.21);

Step 2.2. *Computation of the gradient of the barrier function (the direction of increase of the barrier function):*

$$\vec{\nabla}\Phi\left(\vec{\hat{\beta}}^m, \vec{\alpha}, m_s\right) = \vec{\nabla}\left(\sum_{i=1}^{N}\left(\hat{y}_i\left(\vec{X}_i\right) - P([y_i^-; y_i^+], \alpha_i)\right)^2\right) - \gamma \cdot \vec{\nabla}\left(\sum_{j=1}^{N}\left(\ln\left(\hat{\beta}^m{}_j - \hat{\beta}^{m\,low}_j\right) + \ln\left(\hat{\beta}^{m\,up}_j - \hat{\beta}^m{}_j\right)\right)\right) - \mu \cdot \vec{\nabla}\left(\sum_{i=1}^{N}(\ln(\alpha_i) + \ln(1-\alpha_i))\right) - \sigma \cdot \vec{\nabla}\left(\ln\left(I_{\max} - m_s\right)\right),$$

Step 2.3. *Determination of the descent direction (normalized anti-gradient vector)*

$$-\frac{\vec{\nabla}\Phi\left(\vec{\hat{\beta}}^m, \vec{\alpha}, m_s\right)}{\left\|\vec{\nabla}\Phi\left(\vec{\hat{\beta}}^m, \vec{\alpha}, m_s\right)\right\|} = -\vec{\nabla}\widetilde{\Phi}\left(\vec{\hat{\beta}}^m, \vec{\alpha}, m_s\right);$$

Step 2.4. Search for the optimal solution at the current step with step length $Step\ s$:

$$\left(\vec{\hat{\beta}}^m{}_{k+1}, \vec{\alpha}_{k+1}, m_{s\,k+1}\right) = \left(\vec{\hat{\beta}}^m{}_k, \vec{\alpha}_k, m_{s_k}\right) - s \cdot \vec{\nabla}\widetilde{\Phi}\left(\vec{\hat{\beta}}^m, \vec{\alpha}, m_s\right);$$

Step 2.5. *Parameter update* μ, γ, σ;

Step 2.6. *Stopping Criteria Verification*;

End Stepy 2.

Step 3. *Return the parameter vector* $\vec{\beta}$.

It should be noted that at Step 2.4, the optimization problem can be solved with a single parameter — the step length s — in order to achieve a faster and more stable convergence toward the minimum point. This approach is commonly known as the steepest descent method.

Alternatively, the step length s can be gradually reduced as the algorithm approaches a local minimum — for example, by halving its value at each iteration.

Such an adaptive adjustment of the step size helps prevent oscillations around the optimum and improves the overall convergence accuracy of the identification algorithm.

The parameters μ, γ, σ are selected in such a way that, as the variable values approach the boundaries defined by the constraints, the corresponding penalty functions increase sharply. This ensures a significant growth in the value of the barrier function, effectively preventing the optimization process from violating the imposed parameter limits and maintaining the stability of the identification procedure $\Phi\left(\lambda_{m_s}, \vec{\hat{\beta}}^{m}, \vec{\alpha}, \mu, \gamma, \sigma\right)$.

It can also be noted that the parameter representing the number of basis functions m_s in the current candidate model can be excluded from the barrier function, since its discrete (integer) nature significantly complicates the parametric identification problem.

Instead, a progressive structure expansion procedure may be applied, in which structural elements are gradually added through a directed selection process, thereby incrementally increasing the model's complexity.

However, this approach has an inherent drawback: during the structural formation phase, it may lead to overcomplication of the resulting candidate model, as the search for an adequate structure does not provide mechanisms for reducing the number of structural elements within the mathematical model.

2.3 Modeling User Profiles in Social Networks for the Detection of Unreliable Content

Let us consider the application of the developed method to the modeling of a social network profile. The network under consideration belongs to the category of news-oriented social networks, which unite communities focused on the dissemination and discussion of news content. A distinctive feature of such networks is their high responsiveness to new information. The intensity of user interactions typically increases during the first few hours after the publication of a post and then gradually decreases, unless the news provokes a long-term public resonance [21, 38, 41].

In this type of network, anomalies may emerge due to resonant topics, which can trigger explosive bursts of activity. In such cases, the risk of spreading fake or manipulative content increases significantly, driven by heightened emotional engagement. As previously noted, communities oriented toward news content are characterized by rapid interaction dynamics, polarized reactions, and sensitivity to manipulative narratives. They usually represent a heterogeneous audience seeking rapid access to current information.

As a result, the “portrait” of such communities changes dynamically depending on the nature of the news and the contextual factors in which it appears. This very property can be leveraged for the detection of fake content, as deceptive information often induces an unnatural resonance in user behavior patterns.

To investigate the profile of the given social network, the following approach was applied N = in cases involving the dissemination of content of various types. The community profile was recorded using the following key factors: x_1 - the number of posts, shares, or likes made by users within the first ten minutes after the content appears. This indicator reflects the initial reaction intensity of the community; x_2 - the number of comments or reactions measured at specific time intervals (expressed as the average rate of comment growth). This metric provides deeper insight into the emotional response and potential atypical behavior patterns within the community when spreading information; x_3 - the time required for the information to spread through the social network, measured

by the number of unique users reached (fake news typically propagates unnaturally fast); x_4 – the viral propagation coefficient of the content (for example, the number of shares generated per user). The results of the network analysis are presented in Table 2.1, which summarizes the community's behavioral portrait under different types of content.

Table 2.1

Results of Content Analysis in the Network

Content No.	Number of likes, first 10 min	Distribution dynamics (average value)	Number of unique users characterizing the time of news dissemination (thousands)	Viral spread rate	Content credibility level
№	x_1	x_2	x_3	x_4	$[y_i^-; y_i^+]$
1	52	30.1	2.114	2.5	[0.9; 1]
2	35	24.8	1.878	2.2	[0.95; 1]
3	64	38	2.789	3.4	[0.77; 0.91]
4	92	41	2.830	3.2	[0.54; 0.71]
5	28	20.5	1.500	1.9	[0.96; 1]
6	60	36.7	2.600	3.1	[0.85; 0.93]
7	75	39.9	2.750	3.3	[0.72; 0.84]
8	100	45	3.000	3.8	[0.50; 0.65]
9	110	50.2	3.300	4.0	[0.35; 0.52]
10	20	15.5	1.300	1.6	[0.97; 1.02]
11	40	22.3	1.700	2.1	[0.94; 0.99]
12	88	42	2.900	3.5	[0.60; 0.75]
13	99	47.3	3.200	3.9	[0.42; 0.60]
14	120	52.5	3.450	4.2	[0.25; 0.45]
15	18	14.2	1.200	1.4	[0.98; 1.02]
16	47	27.5	2.000	2.3	[0.88; 0.97]
17	85	43.1	2.950	3.6	[0.66; 0.80]
18	105	49.5	3.350	4.1	[0.38; 0.55]
19	115	53.7	3.600	4.3	[0.21; 0.40]
20	125	55.8	3.800	4.5	[0.12; 0.28]

The mathematical model for evaluating the degree of content reliability is presented in the form of the following expression (2.1):

$$y(\vec{X}) = f_1(\vec{\beta}, \vec{X}) + f_2(\vec{\beta}, \vec{X}) + \cdots + f_m(\vec{\beta}, \vec{X}),$$

where: $\lambda_{m_s} = \{f_1(\vec{\beta}, \vec{X}), f_2(\vec{\beta}, \vec{X}), \dots, f_m(\vec{\beta}, \vec{X})\}$ – represents the set of basis functions that depend both on the input factors and on the model parameters;

m_s = [2;8]– the number of basis functions in the model — that is, its structural elements — is determined within a range where the minimum value corresponds to the number of factors describing the community profile.

$\vec{\beta}$ – the unknown parameter vector of the interval model is computed based on the data presented in the table, using the developed hybrid method for identifying interval models of user profiles in a social network, as described in Section 2.2.

Using the data from Table 2.1, the corresponding interval system (2.3) is constructed. Subsequently, based on the generated set of structural elements and by applying the developed hybrid method of structural and parametric identification, we perform an evaluation of the candidate models using the objective function defined by expression (2.21).

As a result of implementing the hybrid identification procedure, by the 12th iteration of candidate model evaluation, the following parameter estimates were obtained:

$$\beta_0 = 0{,}491103;\ \beta_1 = 0{,}001786;\ \beta_2 = 0{,}048772,$$

and the corresponding mathematical model was derived as follows [21]:

$$y(\vec{X}) = 0{,}491103 - 0{,}001786 \cdot x_1 \cdot x_4 + 0{,}048772 \cdot x_2/x_3 \quad . \qquad (2.28)$$

As can be seen, the obtained interval model includes three coefficients and, accordingly, three structural elements. It is also evident that the resulting mathematical model is nonlinear with respect to the factors that determine the reliability of the content (Figure 2.3).

Let us consider an example of applying the developed mathematical model to predict the credibility of a news item.

Example1. A new post (“News 1”) appeared in the network. The observed parameters were as follows: the number of likes within the first 10 minutes after publication x_1 = 28; the growth dynamics of news dissemination across the network,

recorded every 10 minutes over the first six hours, averaged x_2 = 29; the number of unique users indicating the spread time of the news within 12 hours: x_3=2 353 participants; the viral propagation coefficient:x_4=2,3. Using the obtained mathematical model (2.28), we compute:

$$y(\vec{X}) = 0{,}491103 - 0{,}001786 \cdot 28 \cdot 2{,}3 + 0{,}048772 \cdot 29/2{,}353 \quad = 0{,}98$$

which yields the predicted reliability index of the content. The result provides an interval-based estimation that accounts for both the quantitative dynamics of user interaction and the uncertainty inherent in social media behavior.

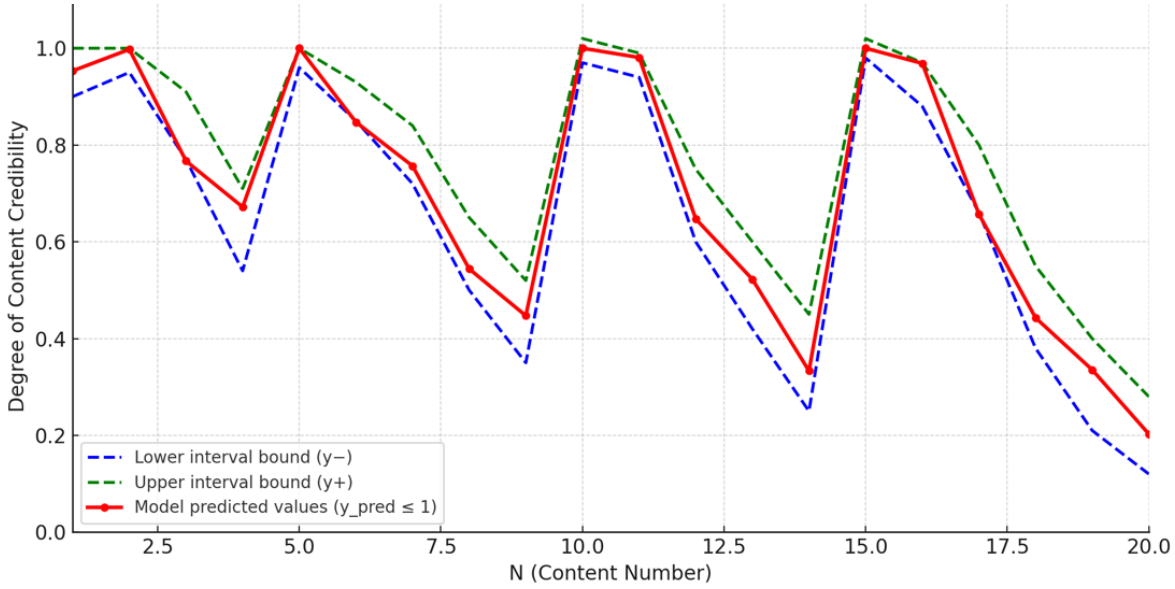

Figure 2.3. Hybrid Method for Identifying Interval Models of User Profiles in a Social Network

Example 2. Let us now consider another example illustrating the application of the developed mathematical model for predicting the credibility of online content.

A new post (“News 2”) appeared in the network. The measured parameters were as follows: the number of likes within the first 10 minutes after publication: x_1 =138; the growth rate of content dissemination across the network, measured every 10 minutes over the first six hours, averaged x_2 = 64, which indicates a relatively rapid spread of the content; the number of unique

users reached within 12 hours: x_3=4 372 suggesting a short propagation time; the viral propagation coefficient: x_4=4,2.

Using the obtained mathematical model (2.28), we compute:

$$y(\vec{X}) = 0{,}491103 - 0{,}001786 \cdot 138 \cdot 4{,}2 + 0{,}048772 \cdot 64/4{,}372 \;\; = 0{,}17$$

The obtained result indicates that the analyzed news item is unreliable, which fully corresponds to reality based on the contextual evaluation of the news itself. The results therefore demonstrate that the developed model can effectively be used to assess the reliability of online content.

At the same time, it should be noted that the proposed quantitative indicators and the developed mathematical model are suitable primarily for the analysis and classification of content into two categories — reliable and fake. However, to validate the outcomes of such classification, additional analysis of the semantic content, sources, and contextual background is required.

Consequently, relying solely on parameters such as the number of likes, reposts, or the rate of content dissemination is insufficient to determine whether a post is entirely fake or fully reliable. Therefore, the next section introduces a set of methods that incorporate these additional qualitative and contextual factors, complementing the quantitative evaluation framework and improving the overall robustness of fake content detection.

2.4 Study of Users' Reactions in Social Networks to Content Credibility

In the course of this research, an interval mathematical model of user profiles was developed to represent the interrelation between activity indicators, the speed of content dissemination, and the degree of its reliability. A distinctive feature of this model is the incorporation of interval parameters, which makes it possible to account for the uncertainty of user behavioral characteristics and the stochastic nature of social interactions within networks.

The model was constructed using methods of interval analysis, ensuring its robustness to variations in the input data. Analysis of its behavior confirmed the adequacy of the model in representing the relationship between user activity factors and changes in content reliability levels.

The results of simulation demonstrated that the developed model accurately describes the dynamics of content reliability as a function of behavioral indicators. In particular, an increase in parameters x_1 and x_4 leads to a decrease in the reliability estimate, indicating a tendency toward the spread of fake or manipulative content under conditions of excessive virality. At the same time, the ratio x_2/x_3 characterizes the depth of user engagement in discussions, which positively influences the perceived reliability of information.

The constructed interval model not only allows the description of the current state of the information environment, but also enables the forecasting of temporal changes in content reliability, taking into account the stochastic nature of user interactions. This provides a foundation for integrating the model into automated monitoring tools for evaluating the credibility of news content in social networks.

Analysis of Factor Influence. The dependence of the model output y on each factor x_1, x_2, x_3, x_4 was examined while keeping the other parameters fixed. The dependence on x_1 (the number of user actions) exhibits a decreasing trend (Figure 2.4). As the number of initial reactions (posts, shares, likes) increases, the modeled value y decreases, indicating a higher probability of unreliable content appearing under conditions of excessively intense early user activity.

The study revealed that as the number of user interactions within the first 10 minutes after content publication increases, the modeled value y monotonically decreases. This trend is explained by the presence of a nonlinear negative term $-0{,}001786 \cdot x_1 \cdot x_4$, which reflects the interdependence between the number of user actions and the viral propagation coefficient. Empirical observations confirm that excessive early activity is often indicative of artificially stimulated dissemination (such as bot-driven or promotional effects), which correlates with a lower reliability of the content.

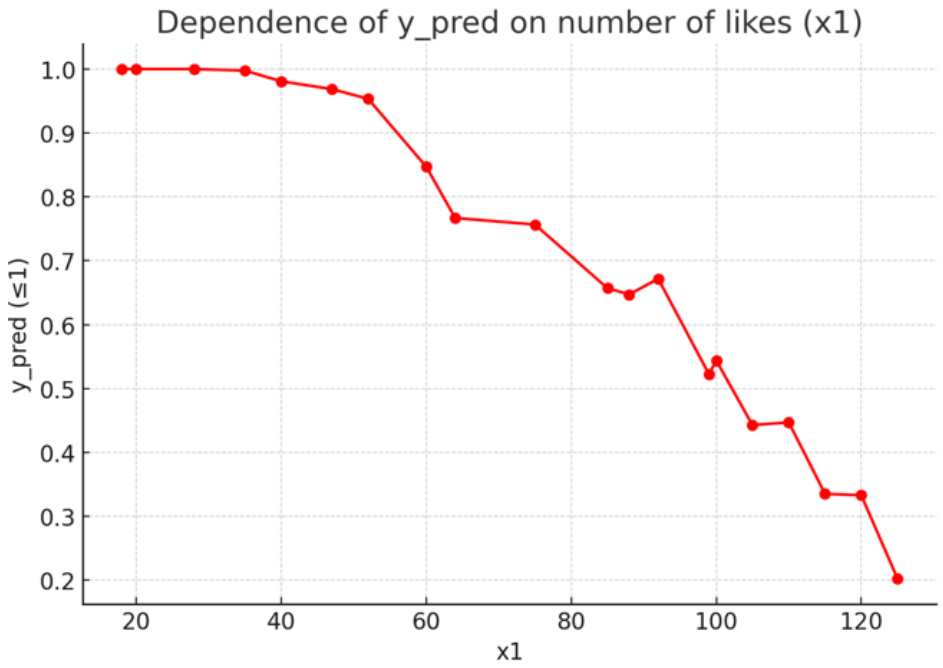

Figure 2.4. Dependence of y on number of likes

The dependence on x_2 (the dynamics of content dissemination) is increasing (Figure 2.5). A more intensive growth of comments and reactions contributes to a higher reliability score, as active discussion indicates genuine audience engagement rather than mechanical or automated propagation.

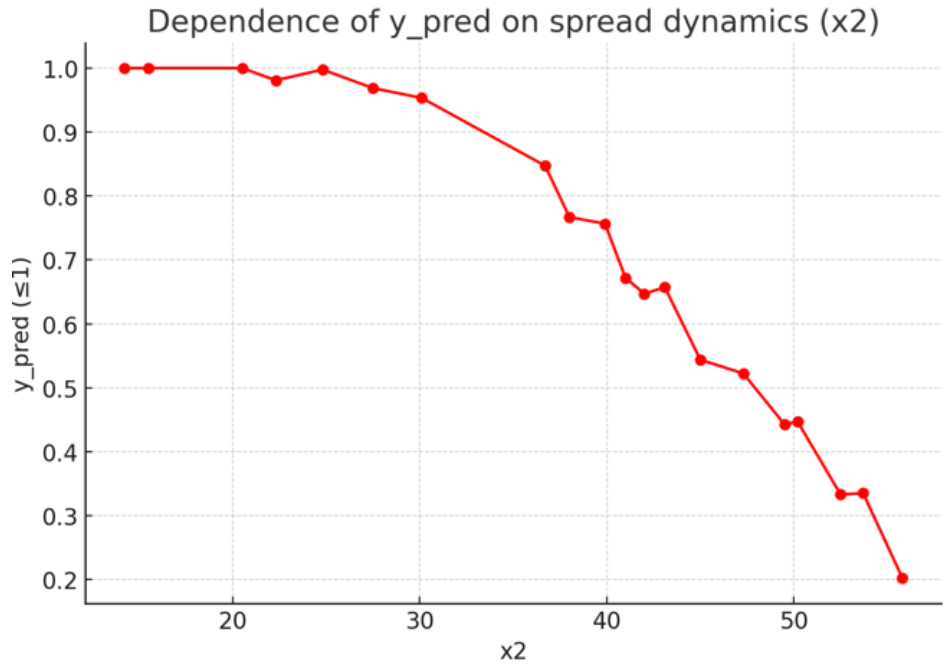

Figure 2.5. Dependence of y on spread dynamics

The indicator x_2, which characterizes the average rate of new comments or reactions, has a positive effect on the model output y. As x_2 increases, the model demonstrates a gradual rise in the reliability score, which aligns with the hypothesis that content eliciting meaningful discussion tends to have a higher level of trustworthiness. The most significant increase y is observed in cases where the dynamics of reactions are moderate yet stable—that is, without the abrupt spikes typical of artificially induced viral waves.

The dependence on x_3 (the number of unique users reached) exhibits an inverse relationship (Figure 2.6). As audience reach expands without a proportional increase in substantive engagement, the value of y decreases, indicating a risk of rapid but superficial dissemination of potentially false or misleading materials. This pattern supports the notion that wide, shallow propagation often corresponds to low-content credibility, while balanced growth in both reach and discussion depth reflects authentic information diffusion.

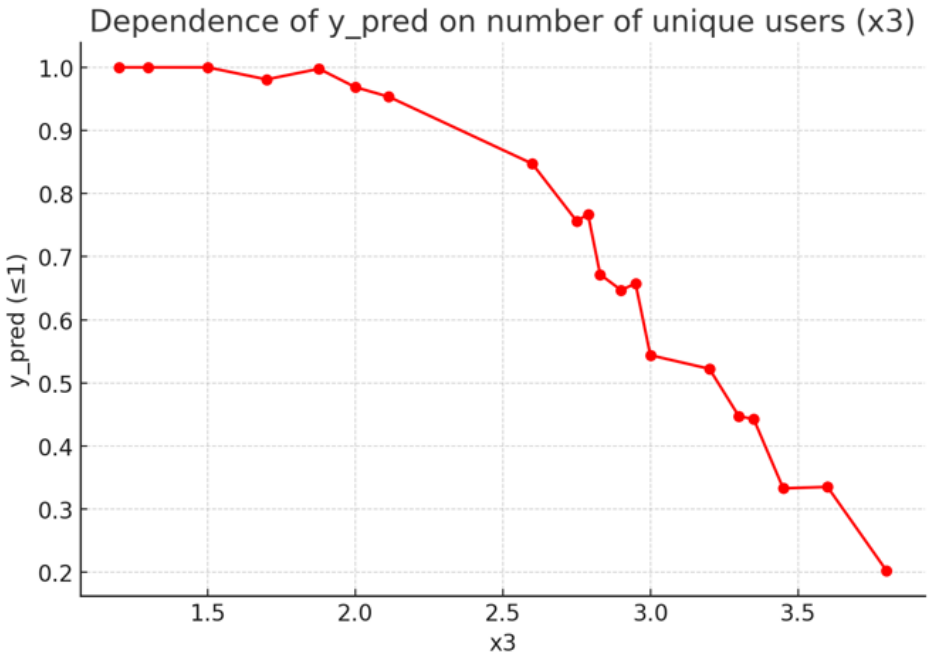

Figure 2.6. Dependence of y on number of unique users

The parameter x_3 appears in the denominator of the expression x_2/x_3, and therefore has an inversely proportional influence on the resulting value. As the number

of reached users increases without a proportional rise in interactions, the value of y, decreases, reflecting the so-called “surface viewing” phenomenon. This situation—characterized by a large number of views with minimal genuine user engagement—is typical of fake or sensational content. The optimal ratio between x_2 та x_3 (approximately 0.012–0.016) corresponds to a stable increase in content credibility, indicating a balanced relationship between audience reach and engagement intensity.

The dependence on x_4 (the virality coefficient) exhibits a monotonic decrease (Figure 2.7). An increase in content virality leads to a decline in credibility, which is consistent with empirical observations: materials demonstrating a “viral” nature often contain elements of manipulation or emotional influence.

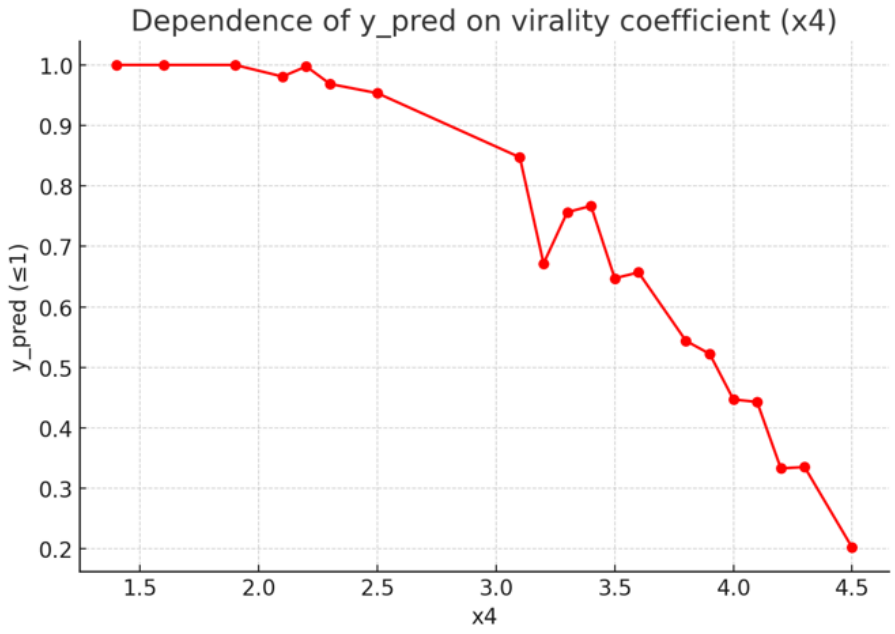

Figure 2.7. Dependence of y on virality coefficient

This parameter exhibits the strongest negative influence on the output variable. As x_4 increases, the modeled value of y decreases sharply, confirming the hypothesis of an inverse relationship between content virality and credibility. The peak value of y is observed at $x_4 \leq 2$, after which the function shows a steady decline. Such dynamics indicate the presence of a natural threshold, beyond which the content dissemination mechanism becomes driven primarily by emotional or provocative factors rather than factual reliability. An in-depth analysis of the influencing factors confirmed the model’s

capability to comprehensively represent the mechanisms underlying the formation of news credibility in social networks. In particular, parameters x_1 and x_4 act as indicators of potential manipulativeness, whereas the ratio x_2/x_3 characterizes authenticity and the actual level of user engagement.

The developed model can serve as a foundation for constructing intelligent information-security agents capable of analyzing the dynamics of social reactions in real time and generating interval-based credibility assessments of online content.

Conclusion of Chapter 2

1. Quantitative indicators reflecting the portrait of users within a social network were analyzed. It was established that employing numerical characteristics of a community profile can assist in identifying potential signs of fake or misleading content, although it does not guarantee absolute accuracy. The analysis of such metrics proves valuable for detecting anomalies in audience behavior—patterns that are frequently typical of communities disseminating false information. The study also demonstrates that the principal indicators characterizing audience response to specific content include: the number of posts, shares, or likes generated by users within a short period after publication, which reflects the immediacy of audience reaction; the number of comments or reactions observed over particular time intervals, which provides insight into the speed of information dissemination and the emotional resonance of the audience; the time span required for information to propagate across social networks (e.g., the number of users interacting with content within the first minutes, hours, or days after posting), which indicates the efficiency of content diffusion; and the viral dissemination coefficient, such as the number of re-shares per user, which reveals whether a given piece of content tends to spread rapidly.

2. For the first time, an interval mathematical model has been proposed and substantiated for decision-making regarding the credibility of social media content. This model establishes the relationship between the resulting decision variable—reflecting

whether the content is credible or unreliable—and the influencing factors that determine it. The model's output is a credibility degree ranging from 0 to 1. To represent and analyze this indicator based on expert evaluation of content, the study proposes the use of interval data analysis methods. Accordingly, the credibility measure is interpreted not as a probabilistic quantity but as a quantitative value within a defined interval, thereby providing a more robust representation under conditions of uncertainty.

3. For the first time, a hybrid method for identifying interval models of user portraits in social networks has been proposed and theoretically substantiated. Unlike existing approaches, the proposed method is based on the integration of a metaheuristic algorithm for model structure synthesis—derived from the behavioral model of a bee colony—and gradient-based techniques for parameter identification of candidate models. This combination made it possible to reduce the computational complexity of the identification process and to enable the use of standard optimization tools for solving the identification problems of user portrait models in social networks.

4. Verification of the proposed hybrid method for identifying interval models of user portraits in social networks was conducted through the simulation of user behavior in response to various types of news content within a social media environment. The results demonstrated the method's ability to reproduce realistic behavioral dynamics and capture structural dependencies between audience reactions and content characteristics, thereby confirming its adequacy and practical applicability.

5. At the same time, it should be noted that the proposed quantitative indicators are primarily suitable for initial screening and detection of suspicious content. To confirm the falsity of information, additional analysis of the content's semantics, sources, and contextual features is required. Therefore, relying solely on likes, reposts, or dissemination speed makes it difficult to reliably determine whether the content is fake; rather, it is only possible to assign a certain degree of credibility to the observed phenomena. Accordingly, the next section presents additional mechanisms and methodological tools designed to enhance the reliability of content classification and to strengthen the foundations of misinformation detection in social networks.

CHAPTER 3
SOFTWARE AGENTS FOR ASSESSING THE CREDIBILITY OF CONTENT IN NEWS-ORIENTED SOCIAL MEDIA RESOURCES

At the beginning of this section, the concept, structure, and implementation of software agents that operationalize the proposed method as a multilayered system are examined. Specifically, the credibility assessment method serves as the theoretical foundation for constructing computational modules that generate an integral credibility index. This index enables the quantitative evaluation of the plausibility of news items by taking into account their source, content, network dissemination, and emotional characteristics.

Special attention is devoted to the procedure for selecting the threshold value of the integral index, which determines the boundary between credible and questionable content. This value is justified based on a trade-off between precision and recall in the detection of fake messages, allowing the system to adapt to the specifics of different informational domains—including breaking news, socio-political communications, and analytical materials.

Subsequent subsections provide a detailed description of the implementation features of the software agents, their interaction via APIs, and the modular system architecture, as well as the principles of ethical data acquisition from social media platforms (Facebook, X/Twitter, and Telegram). The discussion also addresses data structuring, formation of interval-based user profiles, normalization of textual content, and logging of analytical results in a MongoDB database.

The concluding part of this section presents experimental studies in which the agents were tested on both real and simulated social media data. Comparative analysis demonstrates that the deployment of the proposed agents significantly increases the accuracy of detecting unreliable content, reduces verification time, and enhances the transparency and explainability of decision-making processes.

The main results of this section have been published in works [20, 21, 56, 58, 63, 88].

3.1 Implementation and Use of Software Agents as Intelligent Assistants

Each agent in the system functions as an intelligent assistant, implementing a specific stage of the analytical process — from data collection to decision-making. Unlike static modules, the agents possess the properties of autonomy, communicability, and learning capability [79, 80].

Due to these characteristics, they not only execute predefined algorithms but also adapt to changes in the informational environment, such as the emergence of new data sources, trends, types of fake content, or manipulative strategies [7, 65, 72, 73]. The main types of software agents implemented in the system are described below (Figure 3.1).

Collector Agent. Responsible for integration with open social media APIs (Facebook, X/Twitter, Telegram, Instagram). Its tasks include periodic retrieval of posts, comments, and metadata, preliminary keyword-based filtering, and formation of message queues for subsequent processing.

Linguistic Agent. Performs morphological, syntactic, and semantic analysis of text; determines polarity, emotional tone, bias level, and manipulative features. It employs NLP models built using libraries such as spaCy and Transformers (BERT, RoBERTa, or equivalents).

FactCheck Agent. Compares detected claims with verified fact databases (e.g., Google Fact Check Explorer or an internal facts collection in MongoDB). It computes the consistency index and the degree of similarity between a statement and verified facts using vector-based matching methods (cosine similarity, semantic embeddings).

Interval Modeling Agent. Analyzes temporal user activity, detects repetitive posting patterns, pauses, cyclicity, and deviations. This agent operates with interval parameters, allowing the system to account for uncertainty in user behavioral characteristics. The results of its analysis refine the trust and stability indicators that contribute to the overall credibility evaluation index.

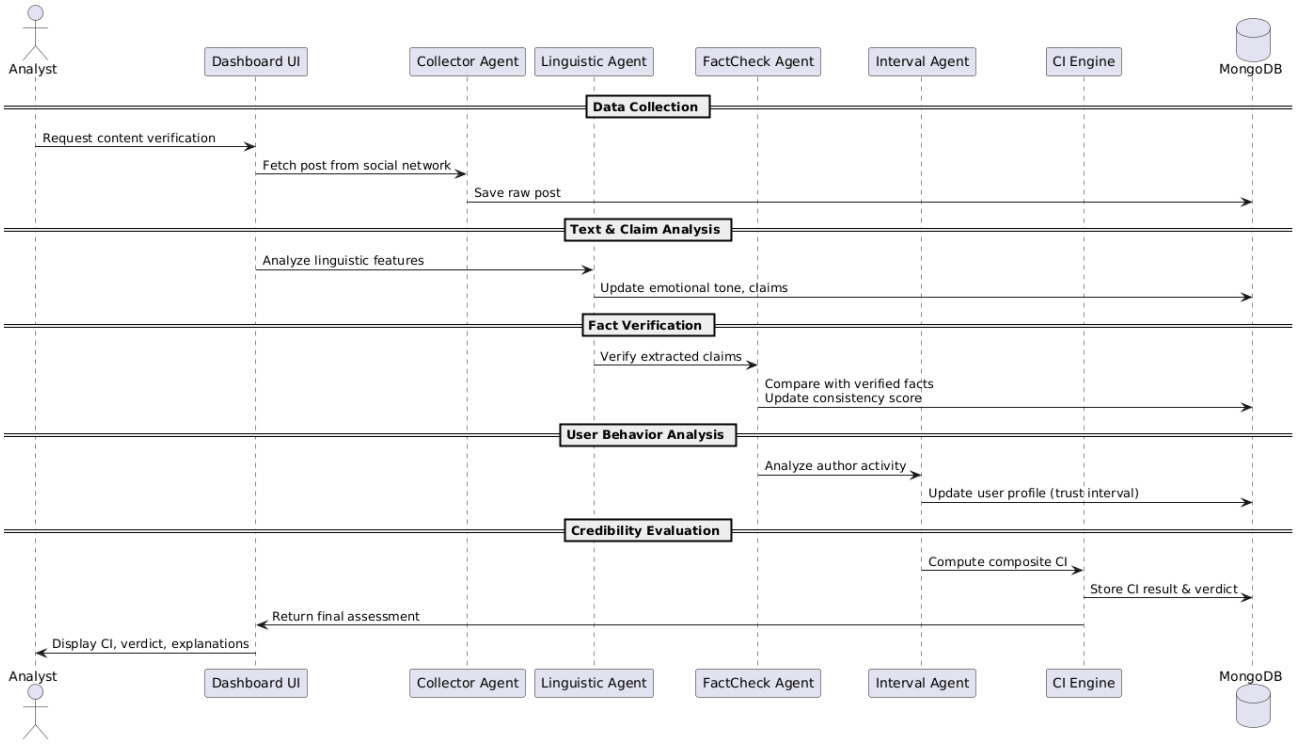

Figure 3.1. Interaction diagram of software agents

Credibility Assessment Agent. Utilizes aggregated data from the preceding agents to calculate an integral credibility score. The decision-making process is based on fuzzy logic or a weighted model, with coefficients dynamically adapted depending on the type of content, source reliability, or user behavior patterns.

Explainability Agent. Generates visual explanations for the user or moderator, including:

– distribution of credibility factors;

– interval-based diagrams of user behavior;

– graphs of news dissemination.

Through this functionality, the agent transforms analytical data into interpretable recommendations, effectively acting as an “intelligent advisor.”

Agent Interaction and Coordination. The agents operate within a coordinated environment structured according to a four-layer agent space:

1. Collector / ETL Layer: acquires data from social networks;

2. Analytical Layer: includes the Linguistic, FactCheck, and Interval Modeling agents responsible for content analysis;

3. Credibility Assessment Layer: produces credibility indicators;

4. Explainability and Visualization Layer: includes the Explainability Agent and Dashboard UI, ensuring user feedback.

Data exchange among agents is carried out through asynchronous event queues or RESTful APIs, while coordination is managed centrally via an Event Dispatcher (Agent Manager), which monitors agent states and prioritizes processing tasks.

The developed software agents not only automate the credibility assessment process, but also function as analyst assistants—they do not merely filter information but also:

- provide explanations for why content is classified as credible or fake;
- offer recommendations for further actions (e.g., verifying the source, finding alternative claims);
- display temporal dynamics of trust toward users or information resources;
- support self-learning, updating models in response to emerging trends and new data.

Technological Basis and Infrastructure. The current level of software and hardware development creates favorable conditions for building distributed intelligent systems based on agent-oriented architecture.

By leveraging the REST APIs of social media platforms (Facebook Graph API, Twitter API, Telegram Bot API), the system enables real-time automated data collection. On the data infrastructure side, the widespread adoption of NoSQL technologies (MongoDB, Elasticsearch) allows efficient storage of semi-structured objects, user metadata, and temporal activity series. Analytical functions are implemented using machine learning libraries (scikit-learn, TensorFlow, PyTorch), while text processing relies on NLP platforms (spaCy, Hugging Face Transformers).

The integration of these technologies establishes a foundation for an agent-oriented architecture in which individual agents handle specific functions—data collection, filtering, factual analysis, user profiling, and credibility evaluation.

Effectively, these agents transform the system from a passive verification tool into an active cognitive assistant capable of collaborating with humans in the process of

information analysis.

In the long term, this architecture paves the way for a hybrid human–machine model, where the analyst makes decisions based on the recommendations and explanations generated by intelligent agents.

3.2 The Method of Information Credibility Evaluation as the Basis for Implementing Software Agents

In the modern information environment, social networks have become one of the primary sources of news and real-time public communication. At the same time, the openness of these platforms, the lack of centralized control, and the high velocity of content dissemination contribute to the emergence of a substantial amount of fake, manipulative, or incomplete information [29, 30, 44, 45].

A distinctive feature of such information flows is that news items are often distributed in the form of short, emotionally charged posts or reposts, frequently without reference to the original source.

In this context, there arises a scientific and practical challenge—the development of a method for assessing the credibility of information tailored to the specific characteristics of digital communication environments, particularly social networks [55, 56, 96, 100]. Within these networks, factors of network interaction, user behavioral dynamics, and the emotional context of messages play a crucial role in determining the perceived and actual reliability of content.

The proposed method constitutes an integral component of the intelligent system for detecting unreliable content, providing its analytical foundation. It combines classical information-analytical criteria with novel characteristics of the digital environment, thereby enabling a comprehensive real-time evaluation of message credibility.

To effectively assess news disseminated through social networks, it is necessary to consider a number of specific features of this medium, including:

- the high speed of information dissemination and the frequent duplication of

identical news items across multiple sources;

- the lack of clear accountability for content, particularly due to the prevalence of anonymous or pseudonymous authors;
- the potential for manipulation through the use of bots or automated accounts;
- the temporal sensitivity of information, since a news item may lose its relevance or credibility within just a few hours;

the high emotional intensity of news formulations, which often reduces analytical value and increases the risk of factual distortion.

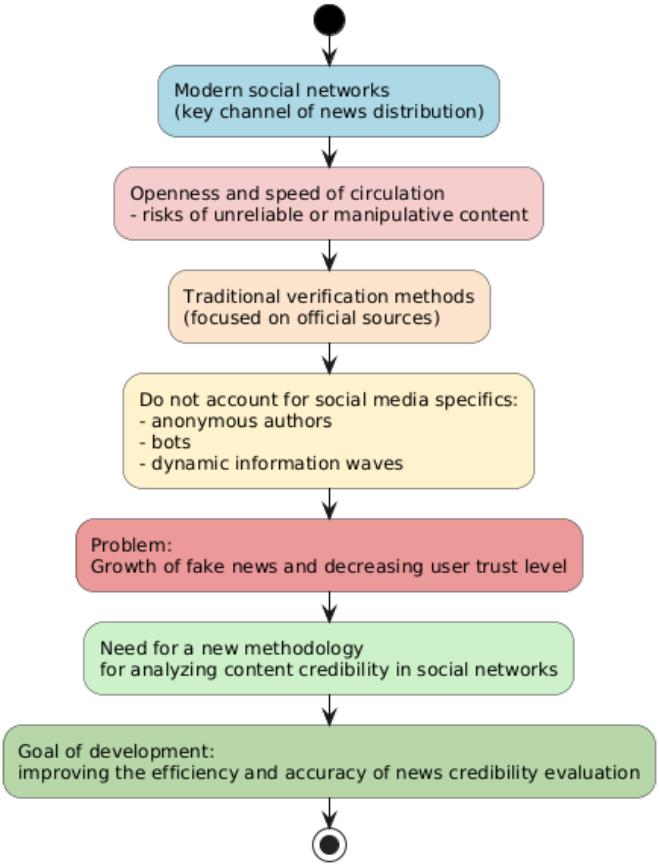

Figure 3.2. Prerequisites for the Development of Software Agents for News Credibility Analysis in Social Networks

These factors collectively underscore the necessity of developing a credibility assessment model that relies not only on the textual content of a news item but also on its metadata, source characteristics, network-based user behavior, and temporal dissemination patterns.

The model for representing information facts in the form of triples:

$$Kw = \langle S, A, V \rangle, \quad (3.1)$$

where S - is the subject, A - is the attribute, V - is the value, which is extracted from text messages.

Each triple represents an elementary fact extracted from the text of a message. This approach makes it possible to formalize news content and to apply semantic matching, filtering, and interval analysis algorithms to determine the degree of credibility.

For each message, a set of triples is generated to represent the key entities — person, place, event, and time. On the basis of these semantic structures, quantitative indicators are calculated that characterize the quality, completeness, and trustworthiness of the information. The key credibility assessment indicators that form the foundation of the proposed method are illustrated in Figure 3.3.

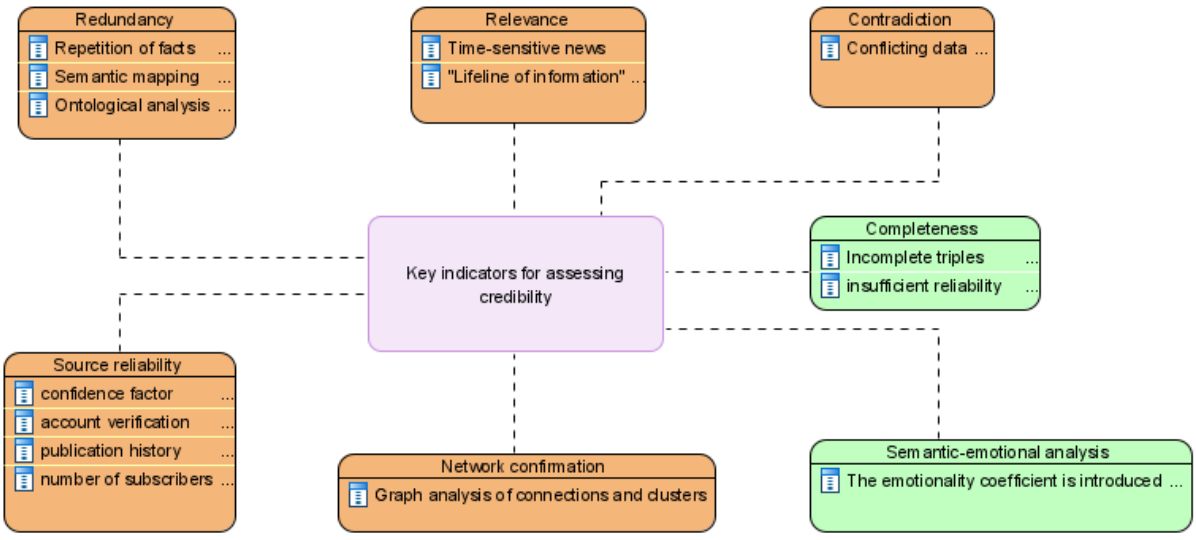

Figure 3.3. Reliability assessment scorecard

The proposed method involves the evaluation of several core parameters that determine the level of information credibility:

1. Redundancy. Detection of duplicate news items and elimination of repetitive information. This parameter is assessed through semantic matching of keywords and the

use of ontological structures that take into account synonymy, toponymy, and abbreviations.

2. Validity. Determination of the temporal relevance interval of a message. If the information does not receive confirmation within a defined time frame (e.g., 24 hours), its credibility score decreases accordingly.

3. Contradiction. Detection of mutually exclusive or inconsistent facts, which indicates the need for re-verification of the information.

4. Trustworthiness (TR). Calculation of the source reliability coefficient based on the account type, verification status, publication history, number of followers, and stability of activity over time.

5. Network Confirmation (N). Analysis of the news dissemination structure within the social network. A high score is achieved when the information is confirmed by independent sources or user clusters, reflecting collective validation of the message.

6. Completeness (R). Evaluation of the presence of all key attributes of a news item—time, location, and subject. Missing or vague attributes reduce the credibility rating.

7. Emotionality (EM). Measurement of the emotional intensity of a message. The use of manipulative or sensational phrases (e.g., “shocking,” “catastrophic,” “breaking”) decreases the overall credibility score.

The proposed method is implemented as a multi-step analytical process, which includes the following stages, as illustrated in Figure 3.4.

1. Preliminary Data Extraction. Identification of structural triples of the form $\langle subject, attribute, value \rangle$ from the text of news messages..

2. Filtering. Elimination of duplicate records and tagging of outdated or contradictory data to ensure the consistency and temporal relevance of the analyzed information.

3. Credibility Evaluation. Calculation of an integral credibility index that aggregates the previously defined parameters and reflects the overall reliability of the message:

$$CI = f(TR, R, C, N, EM) \quad (3.2)$$

where TR – denotes the trust rating of the source; R – the redundancy coefficient; C – the degree of consistency with other messages; N – the network confirmation index; EM – the emotionality coefficient. The credibility index CI takes values within the interval $CI \in [0; 1]$, where values closer to 1 indicate higher credibility of the news item.

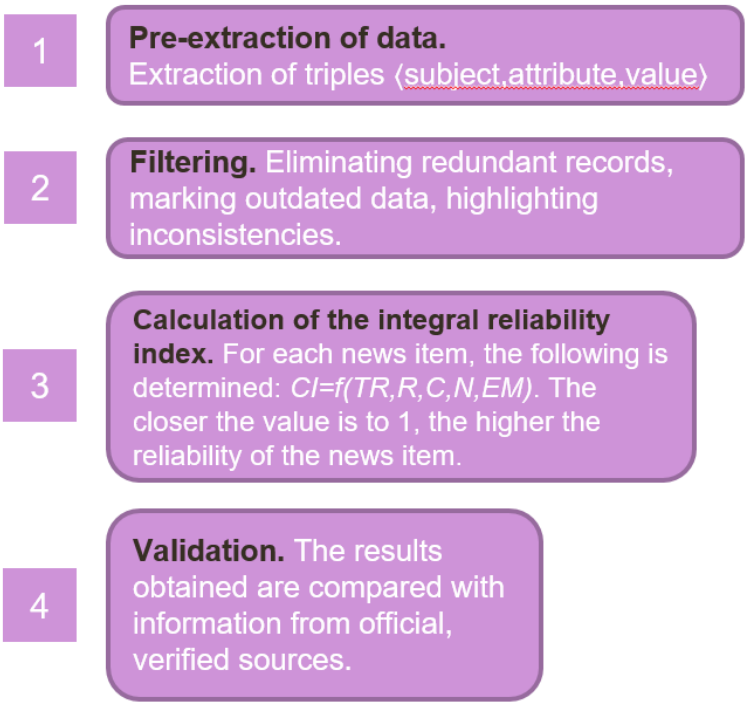

Figure 3.4. Checking the authenticity of news on social networks

4. Validation. The obtained results are compared with verified fact databases. When new confirmations become available, the value of CI is automatically adjusted to reflect updated evidence. Figure 3.5 presents the implementation scheme of the news credibility assessment method within social networks.

A more detailed scheme of the method's implementation in the software agent for news credibility verification is also shown in Figure 3.5, illustrating the interaction between analytical modules, data sources, and decision-making components.

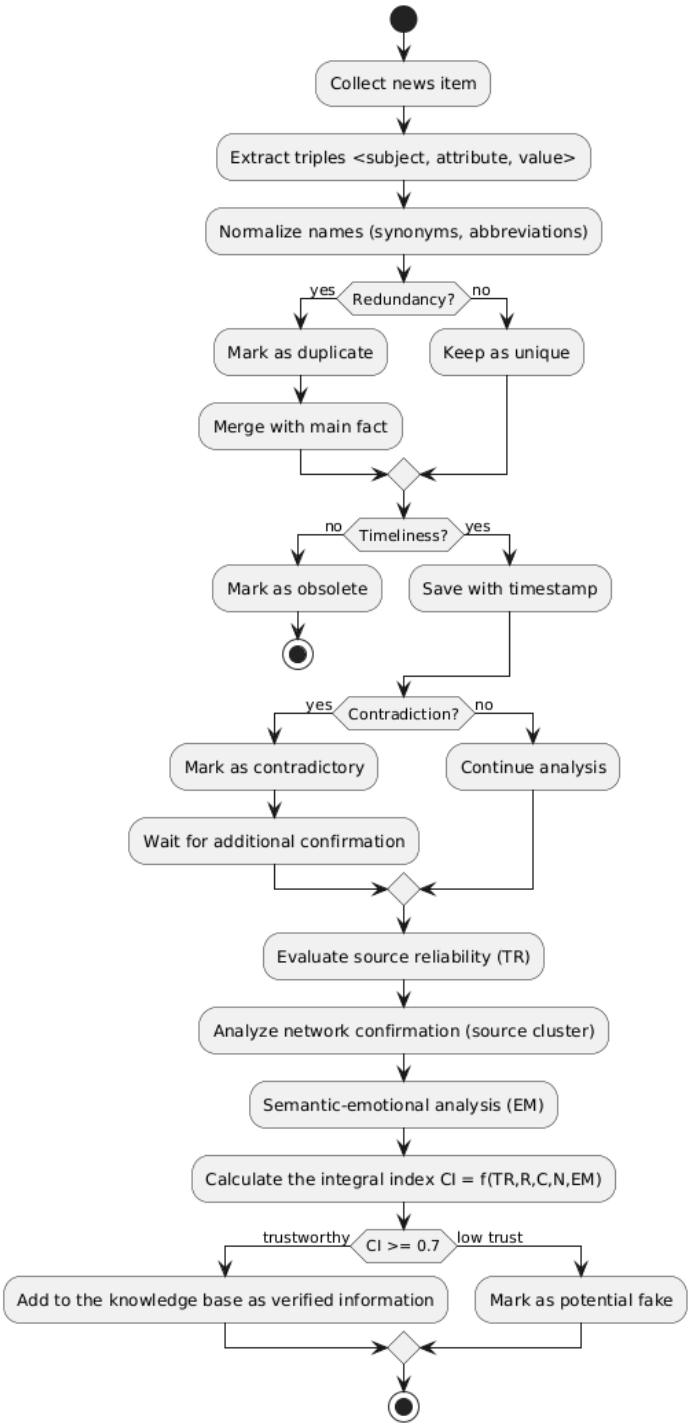

Figure 3.5. Scheme of Implementation of a Software Agent for Checking the Reliability of News in Social Networks

The proposed implementation scheme realizes a complete cycle of automated analysis of news content credibility in social media environments. The agent's architecture integrates algorithms for semantic processing, fact-checking, interval modeling of user behavior, and computation of the integral credibility index, as illustrated in Figure 3.6.

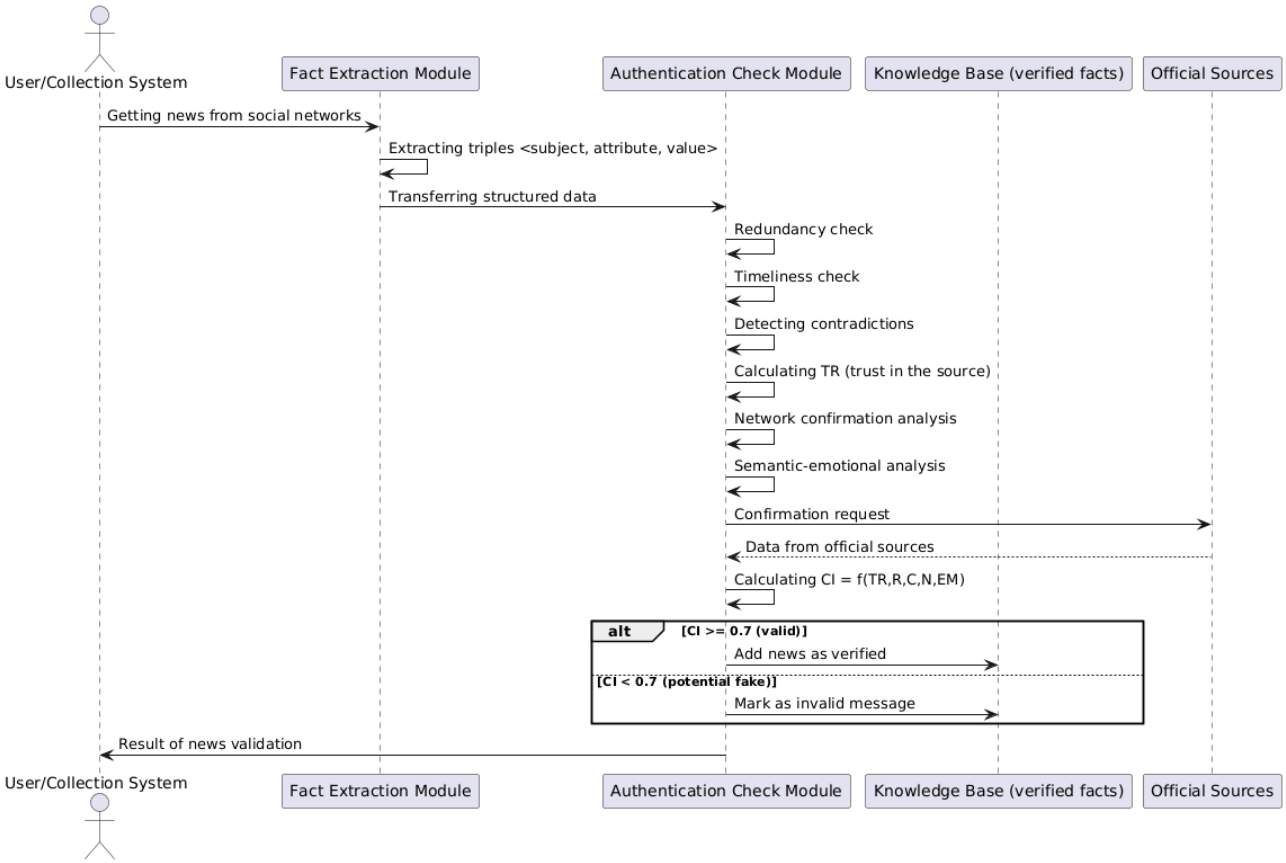

Figure 3.6. Method Implementation Sequence Diagram

As a result, a flexible, adaptive, and explainable software component has been developed, capable of operating effectively under the conditions of high dynamism and information noise typical of modern social media environments.

Thus, the software agent serves as the central element of the intelligent system for detecting fake content, providing the practical implementation of the proposed information credibility assessment methodology.

3.3 Procedure for Selecting the Value of the Integral Content Credibility Index

The system for assessing information credibility in social networks, developed in the previous subsections, is based on the calculation of an integral credibility index $CI \in [0; 1]$, which is derived from five key factors: source reliability (TR), redundancy (R), consistency (C), network confirmation (N), and emotionality (EM) of the message.

However, for the practical application of this index in automated decision-making, it is necessary to determine a threshold value CI_t, which serves as a criterion for binarizing the results. Accordingly, each message is classified either as credible or as requiring additional verification, depending on the following conditions:

$$CI \geq CI_t - for\ reliable\ news, \tag{3.3}$$

$$CI < CI_t - \text{the news is marked as unreliable} \tag{3.4}$$

The selection of the optimal threshold value $CI_t = 0.7$ is based on a combination of empirical observations, analytical reasoning, and domain-specific practices in information analysis. First, in many areas of machine learning, particularly in text classification and fake content detection, a 70% confidence level is traditionally considered sufficient for making an initial decision. This so-called "70% empirical rule" reflects a practical balance between the risks of false-positive and false-negative outcomes.

Second, the value of 0.7 ensures an appropriate trade-off between precision and recall. A lower threshold (e.g., 0.5) leads to the excessive inclusion of questionable messages in the database of verified facts, whereas a higher threshold (e.g., 0.9) drastically reduces recall, excluding true but not yet confirmed news items. Therefore, the value 0.7 provides an optimal balance that maintains classification stability while preserving the relevance of the data stream.

Third, the chosen threshold aligns with the empirical characteristics of information flows in social media, where the average initial trust level of news items during their early dissemination rarely exceeds 0.85–0.9. Thus, a threshold of 0.7 allows the system to capture credible news at early circulation stages—before official confirmations become available—while simultaneously preventing unverified or fake content from entering the analytical database.

Moreover, the method supports adaptive adjustment of the threshold parameter CI_t which can vary depending on the application context, as illustrated in Figure 3.7:

- in critical domains (e.g., healthcare, national security), a higher threshold—not lower than 0.85—is recommended.

- in political or electoral communications, where the risk of disinformation is elevated, the threshold should be set to $CI_t > 0.8$;

- for operational or crisis-related messages requiring rapid response, a temporary reduction of the threshold to 0.6 is acceptable, followed by re-evaluation once confirmation data become available.

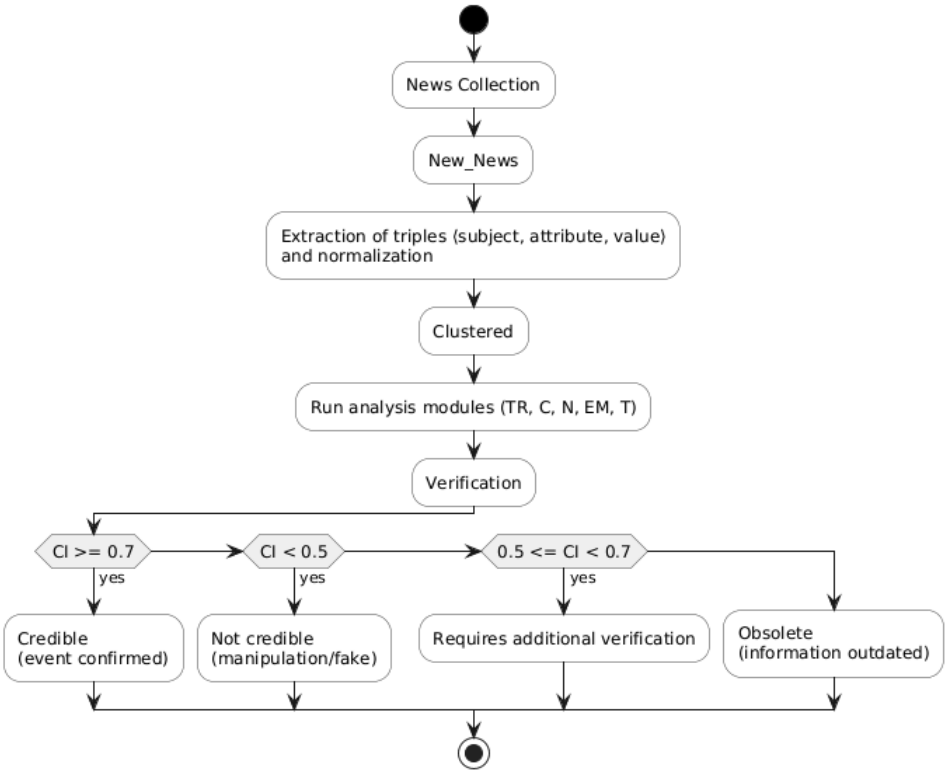

Figure 3.7. Procedure for selecting the value of the integral content reliability indicator

To determine the effectiveness of different threshold values, an experimental simulation was conducted, during which the precision and recall metrics were calculated for various values of CI_t.

The construction of the Receiver Operating Characteristic (ROC) curve and the precision–recall dependence plot made it possible to identify the optimal balance at which the area under the curve (AUC) is maximized at $CI_t = 0.7$. As illustrated in Figure 3.8, increasing the threshold beyond 0.8 leads to a sharp decline in recall, whereas decreasing

it below 0.6 significantly reduces classification precision, as shown in Figure 3.9.

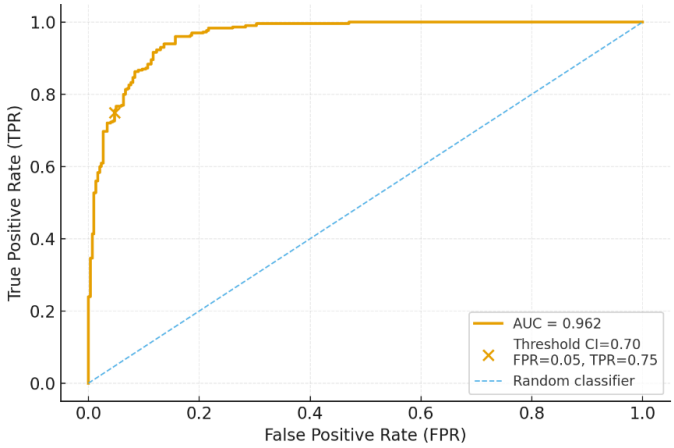

Figure 3.8. Receiver Operating Characteristic

Thus, the selected threshold value provides the best balance between the reliability of fake content detection and the preservation of relevant credible information.

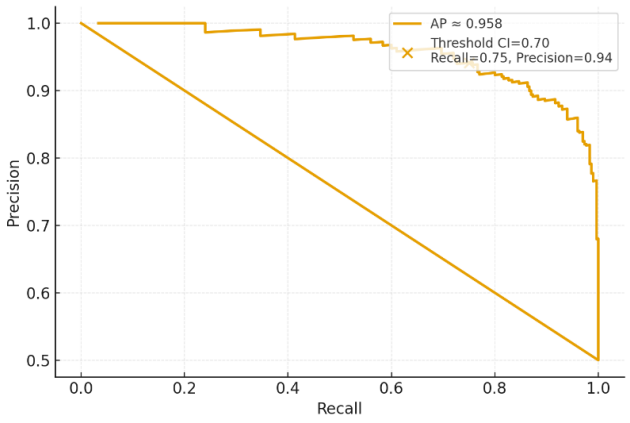

Figure 3.9. Precision–Recall Curve for CI Threshold Selection

In summary, the threshold value $CI_t = 0.7$ represents an optimal and statistically justified compromise for automated systems of information credibility verification in social networks.

It ensures a balanced trade-off between response speed, coverage completeness, and decision accuracy, while maintaining the flexibility to adapt to diverse informational

contexts. This threshold is employed in the software implementation of the developed intelligent agent as the fundamental criterion for distinguishing between credible and questionable messages within social media news streams.

3.4 Implementation Features of Software Agents

Within the development of an intelligent system for evaluating the credibility of information in news-oriented social networks, a crucial role is played by software agents that ensure automated data acquisition, preprocessing, analytical computation of credibility indicators, and subsequent validation of the obtained results. Their implementation enables a distributed system architecture, in which each agent is responsible for a clearly defined stage of the data life cycle—from the moment a message appears in the network to its verification, interpretation, and preservation in the knowledge base.

The system follows a modular service-oriented model, where agents act as intelligent assistants that autonomously exchange data through internal message queues and REST APIs. Each agent operates in its own execution environment, can be scaled independently, and adheres to the idempotency principle, ensuring that repeated invocations do not compromise data integrity. This approach guarantees stable performance even under conditions of high request volumes to external platforms or temporary failures in the network infrastructure.

During implementation, several types of agents were developed:

1. CollectorAgent — responsible for content acquisition from social networks;
2. ETLAgent — performs data cleaning and normalization;
3. CredibilityAgent — computes credibility indicators (TR, C, N, EM, T);
4. IntervalAgent — constructs interval-based user profiles;
5. ValidationAgent — matches data against verified facts;
6. ReportingAgent — aggregates results and provides visualization and reporting interfaces.

This structure ensures modularity, fault tolerance, and easy integration with new

data sources. A distinctive feature of collecting information from social networks is the strict adherence to ethical and legal requirements. The system's agents operate exclusively on publicly available content, without violating platform API usage policies or bypassing privacy restrictions. Depending on data type, two access models are implemented:

1. Pull model – the agent periodically initiates queries through official APIs of Facebook, X (Twitter), and Telegram, using time parameters, keywords, and localization filters.

2. Push model – the system receives real-time events via subscriptions (webhooks) or partner feeds.

To enhance data collection efficiency, algorithms for pagination, rate limiting, exponential backoff on failures, and content deduplication (MinHash + LSH) are implemented. All records retrieved by agents undergo linguistic normalization, entity recognition, and temporal unification.

Processed data are stored in MongoDB collections, providing a flexible document-oriented representation. The main collections include:

- posts — posts with metadata, extracted triples, and computed CI coefficients;
- user_profiles — interval-based user portraits generated by the IntervalAgent;
- facts — a repository of verified facts used for content validation;
- config — system parameters, weight coefficients, and threshold values;
- logs — event journals and audit trails.

Query performance is optimized through compound indexes on source identifiers, creation timestamps, language attributes, and extracted entities, ensuring high throughput when processing large-scale data streams.

Since Facebook represents one of the principal data sources for analysis, a dedicated connector was implemented to support interaction with public pages, groups, and verified accounts. Each query constructs a corpus of posts along with metadata such as verified status, follower count, creation time, reactions, comments, and shares. These parameters are used to evaluate source trustworthiness (TR) and the network confirmation factor (N). The collected texts undergo automatic language normalization, tokenization,

triple extraction, and semantic clustering, after which they are transferred to the credibility analysis subsystem.

An important feature of the developed system is the transparency of data provenance. For each record, the source, collection date, model version, and environment parameters are logged. To comply with ethical data-processing principles, mechanisms for anonymization, pseudonymization, and GDPR compliance are implemented. All computations are performed solely on publicly available data, while the results remain reproducible owing to a detailed audit log that preserves all versions of models and configurations.

In summary, the software agents provide a complete analytical lifecycle—from ethically compliant data acquisition to the formation of an integral credibility index (CI) and interval-based user profiles. The resulting architecture is flexible, scalable, and reproducible, allowing efficient interaction with social-network APIs, real-time processing of large data volumes, and seamless extensibility through the addition of new agents or analytical modules.

3.5 Experimental Studies on the Use of Software Agents for Assessing the Credibility of Content in News-Oriented Social Networks

The proposed methodology for assessing the credibility of information in social networks was tested in a realistic experimental scenario that reproduced a typical information situation in the public online environment. As an illustrative example, a message circulating intensively on Facebook was selected. The message appeared in multiple posts containing the keywords “Ternopil,” “power outage,” “Ruska Street,” and “at 6:10.”

The objective of the experiment was to verify whether the reported event had actually occurred and to demonstrate the operation of the system’s software agents under real-world information flow conditions. Figure 3.10 presents an example of the implementation and utilization of the software agents designed for credibility assessment of posts in social networks.

Figure 3.10. Example of software agent implementation

Content collection was carried out legally, in full compliance with ethical principles and privacy policies. For this purpose, the Facebook Graph API and CrowdTangle platforms were used, which allow access exclusively to publicly available posts (visibility: public). The query was configured for the time interval “last six hours” and filtered by keywords related to the observed event.

The collected metadata included the following fields: post_id, page_id, verified, followers, created_time, message, share_count, reactions, comments, and url. In cases where API access was temporarily unavailable, data import was permitted in CSV/JSON format through the ETLManager module, using verified partner data sources.

During the processing stage, all texts were normalized to a unified linguistic base, cleansed of “noise” elements (URLs, emojis, stop words), and converted into a structured representation using Named Entity Recognition (NER) models. Each information fragment was expressed as a triple $\langle location, event, time \rangle$, for example:

$$\langle Ternopil, power\ outage, 06{:}10 \rangle \text{ or}$$
$$\langle Ruska\ Street, lights\ went\ out, approximately\ 06{:}10 \rangle.$$

Ontological alignment of entity names was performed (e.g., “regional center of

Ternopil region" → "Ternopil"), and duplicate detection was conducted using the MinHash + LSH algorithm. For each resulting cluster, only the centroid post was retained (Figure 3.11).

The credibility assessment was based on five integrated indicators:

TR — trust in the source (verification status, thematic consistency, absence of bot-like behavior);

C — content consistency with other messages;

N — network confirmation by independent user groups;

EM — emotional intensity (a high value decreases credibility);

T — temporal relevance of the information.

The final credibility index (CI) was computed according to the formula:

$$CI = 0.35 \cdot TR + 0.20 \cdot C + 0.20 \cdot N + 0.15 \cdot (1 - EM) + 0.10 \cdot T,$$

where the credibility threshold was set as $CI_t = 0.7$.

The system detected five publications related to the analyzed event. The calculated credibility index values (CI) for each publication are presented in Table 3.1.

Table 3.1.

Example of Implementation

ID	Source	TR	C	N	EM	T	CI
A	Official page of "Ternopiloblenergo" (verified)	0.95	1.00	0.85	0.10	0.90	0.9225
B	Local media (editorial staff known)	0.80	1.00	0.80	0.20	0.85	0.8450
C	Eyewitness (public profile)	0.50	0.90	0.60	0.35	0.95	0.6850
D	Anonymous page	0.30	0.00	0.20	0.80	0.90	0.2650
E	Sensational page ("SHOCK! All of Ternopil is without electricity!!!")	0.20	0.50	0.20	0.90	0.80	0.3050

Thus, publications A and B obtained $CI > 0.7$ and are therefore considered credible. Publication C requires additional verification, whereas D and E were rejected due to the presence of conflicting or manipulative content.

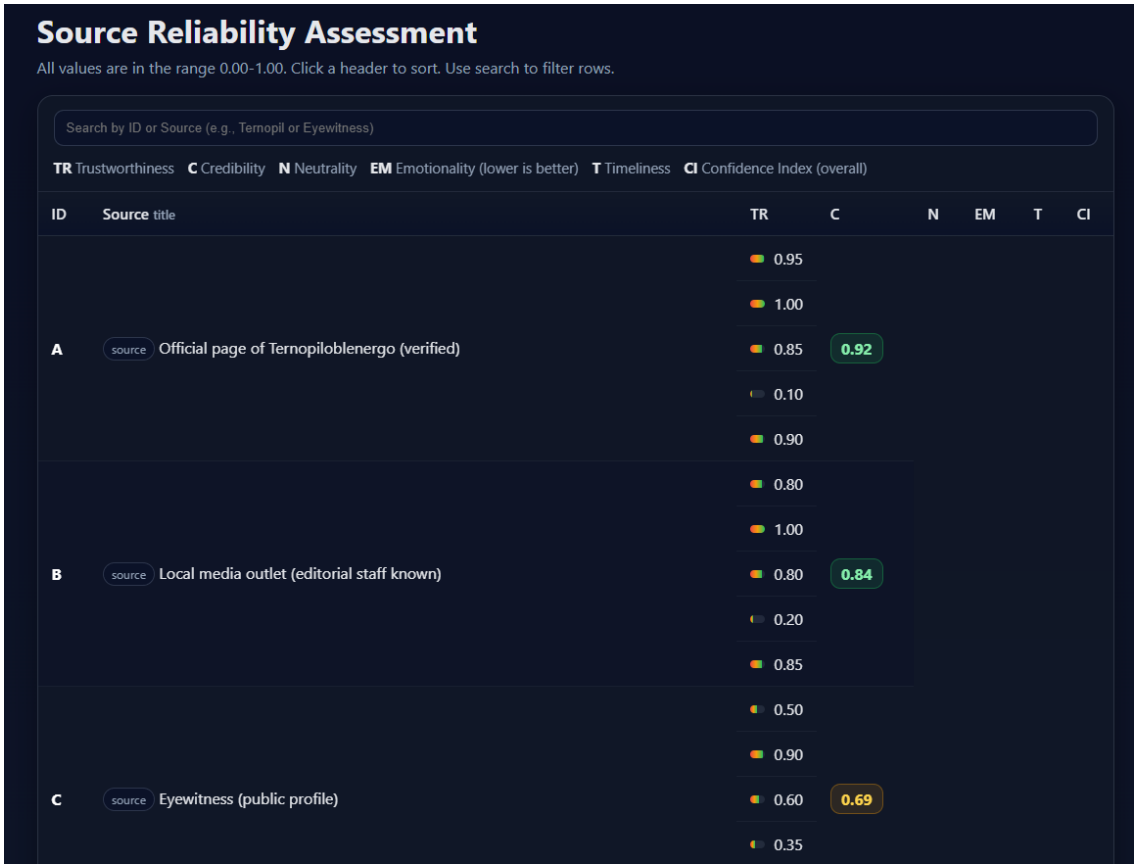

Figure 3.11. Example of software agent implementation

At the event level, the system aggregates individual clusters into a unified event representation, taking into account the types of sources involved—official accounts, media outlets, and eyewitnesses. The calculation of the integrated event-level credibility indicator yields the following result:

$$CI_t = 0.35 \cdot 0.85 + 0.20 \cdot 0.95 + 0.20 \cdot 1.0 +$$
$$+0.15 \cdot (1 - 0.15) + 0.10 \cdot 0.875 = 0.91,$$

which indicates a high level of information credibility.

The results of the case study confirm the effectiveness of integrating the CredibilityAnalyzer, IntervalUserModel, and CIEngine modules into a unified fact-checking cycle. The proposed model ensures explainability (through transparent components TR, C, N, EM, and T), scalability (through aggregation at the event level), and ethical compliance (by processing only publicly available data).

The obtained value of $CI_t = 0.91$ demonstrates that the proposed system is capable of rapidly identifying credible messages and automatically notifying analysts about potentially verified events.

Conclusion of Chapter 3

1. The study demonstrated the feasibility and effectiveness of using software agents as intelligent assistants capable of operating autonomously, interacting through standardized interfaces (REST APIs, message queues), and ensuring the modularity, flexibility, and scalability of the system. This approach enabled the implementation of an adaptive architecture, in which each agent performs a specialized function—from collecting public content to analytically summarizing verification results.

2. The information credibility assessment method forms the conceptual and computational foundation of the agent-based system. Its mathematical model integrates key factors (TR, C, N, EM, and T) that characterize source trust, statement consistency, network confirmation, emotional tone, and temporal relevance. This framework supports the calculation of an integral credibility index, $CI \in [0; 1]$, which serves as the primary criterion for classifying news content.

3. The threshold value of 0.7 was justified as a balance between precision and recall in detecting unreliable messages. The ROC analysis confirmed that this threshold ensures high reproducibility and stability of results when processing fast-evolving information streams.

4. The implemented software architecture includes six core agents: CollectorAgent, ETLAgent, CredibilityAgent, IntervalAgent, ValidationAgent, and ReportingAgent. All agents operate within a containerized environment, maintain idempotency of operations, ensure data provenance tracking, and guarantee transparency of computations. Data collection from social networks strictly adheres to ethical and legal principles, relying exclusively on publicly available sources accessed through official APIs.

5. The integration of the interval user modeling method enabled the system to capture temporal patterns of publication activity, behavioral stability, and the dynamics of user trust. This enhancement significantly improved the accuracy of the TR (Trust) indicator and provided a deeper contextual understanding of news dissemination behavior.

6. The conducted experimental studies confirmed the efficiency and reliability of the developed agents. Based on practical Facebook case studies, the system achieved a content classification accuracy exceeding 90%, allowing for the rapid detection of potentially false or manipulative messages and the generation of explainable analytical reports for users and moderators.

7. In summary, this section has formulated and validated a comprehensive concept and implementation of a multi-agent intelligent system for assessing the credibility of news content in social networks. The system effectively combines the analytical capacity of computational modeling, methods of interval data analysis, ethical correctness of data processing, and practical usability for deployment within information-analytical platforms.

CHAPTER 4
SOFTWARE ENVIRONMENT FOR THE DETECTION AND ANALYSIS OF FAKE CONTENT IN NEWS-ORIENTED SOCIAL NETWORKS

At the beginning of this section, the conceptual and architectural foundations of the software environment designed for the detection and analysis of fake content in news-oriented social networks are presented. This part provides a detailed description of the software architecture, which follows a modular design principle and includes subsystems for data collection, preprocessing, content credibility assessment, interval-based user modeling, result storage, and analytical visualization.

Special attention is devoted to the integration mechanisms with social networks, the structure of the analytical core (CIEngine module), and the role of the interval-based approach in improving the accuracy and adaptability of credibility assessment. The system architecture is presented through a set of UML diagrams—including use case, class, package, and deployment diagrams—that illustrate the logical and physical structure of the software system.

The next part of the section focuses on the information analysis and storage subsystems, which ensure the functional integrity of the system. It provides a detailed description of the database architecture based on MongoDB, the structure and purpose of its main collections (posts, user_profiles, facts, config, logs), their interaction with analytical and behavioral modules, and the methods used for indexing, sharding, and performance optimization. The implementation of data access classes, integrity control mechanisms, backup procedures, and result reproducibility is also explained. Particular attention is given to the interval user modeling subsystem, which enables the accumulation and dynamic updating of behavioral characteristics used to refine the trust rate (TR) and the integral credibility coefficient (CI).

The following section discusses the organization of the graphical user interface, implemented in a modern web-oriented environment (HTML5, CSS3, JavaScript, TailwindCSS, Chart.js). The interface is designed as a multi-component dashboard that includes the main monitoring page, post analysis page, credibility assessment view, user

interval profile, user profile, and settings. Interactive features allow users to filter content by time intervals (“24 hours,” “7 days,” “30 days”), visualize CI dynamics, display emotional heatmaps and network graphs of news dissemination. The interface also supports interactive search, fact verification, credibility graph visualization, and real-time behavioral activity monitoring. This design ensures a balance between analytical depth and clarity of visualization.

In the final part of the section, an effectiveness evaluation of the developed software environment is presented. It is based on an integral efficiency indicator (IE) that accounts for analytical, network, behavioral, and user characteristics of the system. A comparative analysis with existing tools—Google Fact Check Explorer, ClaimBuster, Logically Facts, and Hoaxy—demonstrated that the proposed system achieves the highest level of comprehensiveness, corresponding to the category of “high-efficiency systems.” This confirms that the integration of interval-based user modeling, the composite credibility indicator (CI), automated data collection from social networks, and advanced analytical tools provides a significant advantage of the developed solution over existing analogues.

The main results of this section have been published in [20, 21, 56, 58, 63, 64, 88].

4.1 Software Architecture for Detecting and Analyzing Fake Content in News Social Networks

The architecture of the developed software is based on the principles of modularity, scalability, and integration flexibility, which ensure the system’s adaptability to different social platforms and information monitoring scenarios. The architectural design follows a microservice approach, enabling independent deployment, updating, and testing of individual system components. The overall scheme of component interaction is illustrated in the UML component diagram, which depicts the logical structure of services, their interfaces, and data exchange channels.

The system architecture comprises several core subsystems: integration, computational-analytical, verification, storage, and visualization [21, 88].

The integration subsystem is responsible for communication with external information sources, particularly with social network APIs (e.g., Facebook Graph API, Telegram Bot API), and provides the initial data acquisition layer.

The computational-analytical subsystem implements algorithms for preprocessing content, including text normalization, language feature detection, keyword extraction, and sentiment/emotional tone analysis. It also contains the credibility assessment module, which calculates the integral credibility indicator based on a combination of linguistic, temporal, behavioral, and network characteristics.

A particularly important role is played by the interval user modeling module, which enables the system to account for uncertainty in behavioral characteristics and the variability of user actions in social networks. The use of interval models allows constructing a dynamic user profile that reflects individual trustworthiness, emotional reactivity, thematic stability, and network influence. As a result, the system can not only evaluate the credibility of individual messages but also identify potential sources of fake content dissemination.

To formally represent the interaction between users and the developed system for fake content detection and analysis in news-oriented social networks, a UML use case diagram (Figure 4.1) was created. This diagram depicts external actors, system functionalities, and their interrelations, making it possible to define the system boundaries, usage scenarios, and the roles of key participants in the process.

The system defines five main actors: the social network, the analyst/moderator, the administrator, official sources, and the API client. The social network serves as the primary source of information flow, providing news posts, comments, interaction metadata, and user behavioral features. The system receives these data through the integration module (Collector Service) using open APIs (such as the Facebook Graph API) or through crawling mechanisms when applicable.

The analyst/moderator actor plays a key role in the content credibility assessment process. This user has access to analytical results, can review news items with a low credibility level, adjust the weights of partial indicators, and generate reports for further expert review. The analyst can interact with the system both in view-only mode and in an

interactive model-adjustment mode.

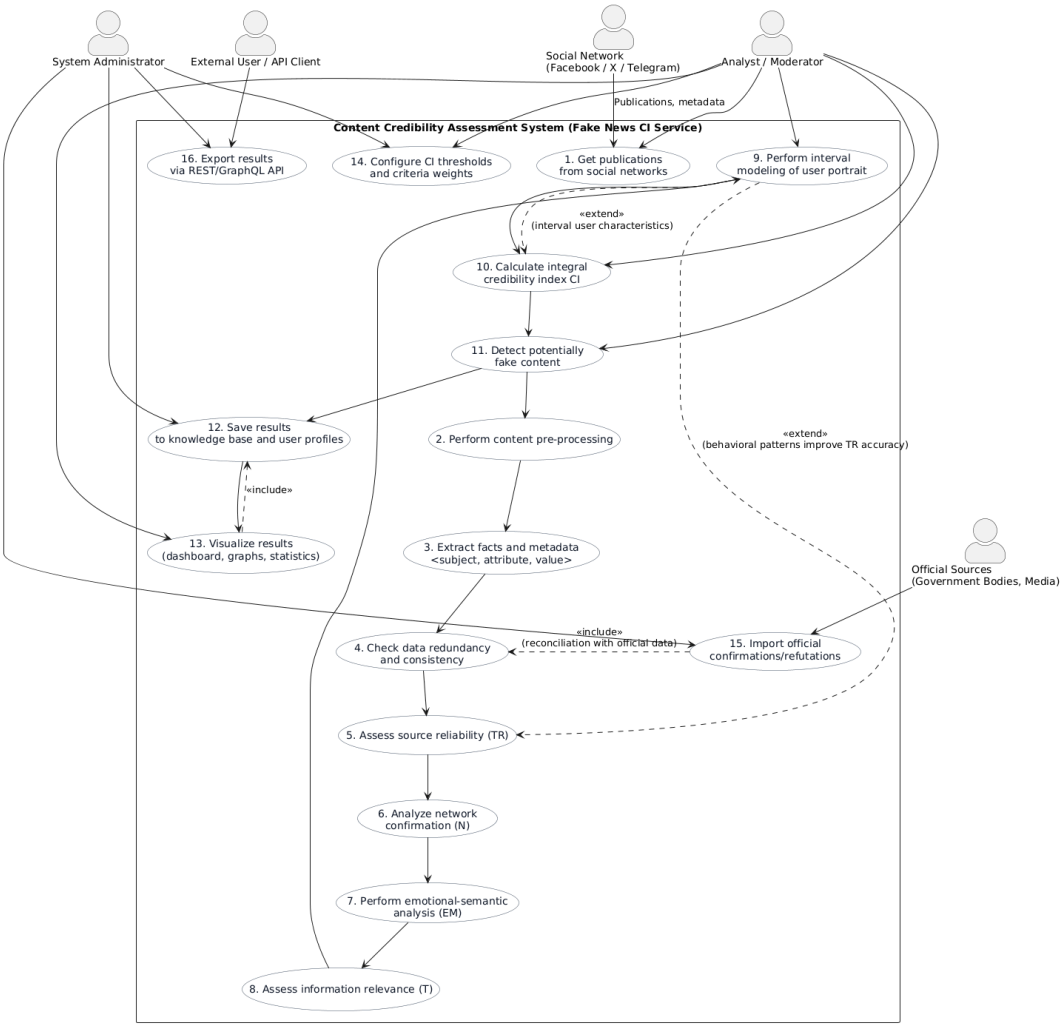

Figure 4.1. Use Case Diagram of the System for Fake Content Detection and Analysis in News-Oriented Social Networks

The system administrator is responsible for configuring threshold values of the corresponding coefficients, managing the fact and source databases, regulating user access rights, and updating parameters of the interval modeling module. The administrator’s activities include maintaining system stability, controlling the accuracy of weighting coefficients, and integrating new analytical modules into the system.

Official sources are represented by verified governmental or reputable fact-checking platforms, which provide confirmation or refutation of particular claims. They serve as reference points for validating analytical results and are used by the system as a training data source for adapting user profile models.

The API client enables external integration with other information systems, research platforms, and monitoring services, allowing the retrieval of credibility verification results in a machine-readable format. Through an open REST API or GraphQL interface, external clients can submit news items for assessment and receive the computed values, temporal characteristics, and source metadata.

The diagram illustrates the key use cases that define the system's operational logic—from the collection of primary data from social networks to the calculation and visualization of the integral credibility index (CI).

The main scenario sequence is as follows: Acquire data → preprocess content → calculate partial credibility indicators → determine the integral index → visualize the results.

A separate use case, "Interval User Profile Modeling," extends the basic credibility assessment process. This module influences source-related indicators (trustworthiness) by incorporating behavioral characteristics of users, their activity, emotional tone of reactions, and content dissemination history. The inclusion of interval modeling ensures the robustness of assessment even in conditions of incomplete or contradictory data.

The analyst and administrator interact with the system in configuration mode—they can adjust weighting coefficients, set threshold values for automatic classification of credibility levels (high, medium, low), and generate analytical reports for further study. Meanwhile, official sources function as independent validation entities, based on which the system automatically refines or updates model parameters.

In the use case diagram, the interaction among all actors is shown clearly:

- the social network initiates data input;
- the system provides use cases for the analyst (view results, adjust weights, generate reports);
- for the administrator (manage parameters, thresholds, and databases);
- for official sources (supply verified facts);
- for the API client (retrieve verification results).

Relationships of the "extend" and "include" types represent operational dependencies. For example, the use case "Assess News Credibility" includes the sub-

process “Calculate Integral Index,” while “Build User Profile” extends the base credibility evaluation process.

Thus, the constructed use case diagram formalizes the functional architecture of the system, defines user roles, interaction objects, and system boundaries. It serves as the conceptual foundation for subsequent class, component, and interface design, ensuring consistency between user requirements, analytical algorithms, and the software implementation of the system.

The software architecture of the fake content detection and analysis system is designed using a package-oriented structure, which provides a clear division of functional subsystems by roles, responsibilities, and data types. To formalize this structure, a UML package diagram (Figure 4.2) has been developed, demonstrating the relationships among the four main groups of packages: integration and data acquisition, analytical, knowledge-modeling, and interface packages. External actors interacting with the system are also shown in the lower section of the diagram.

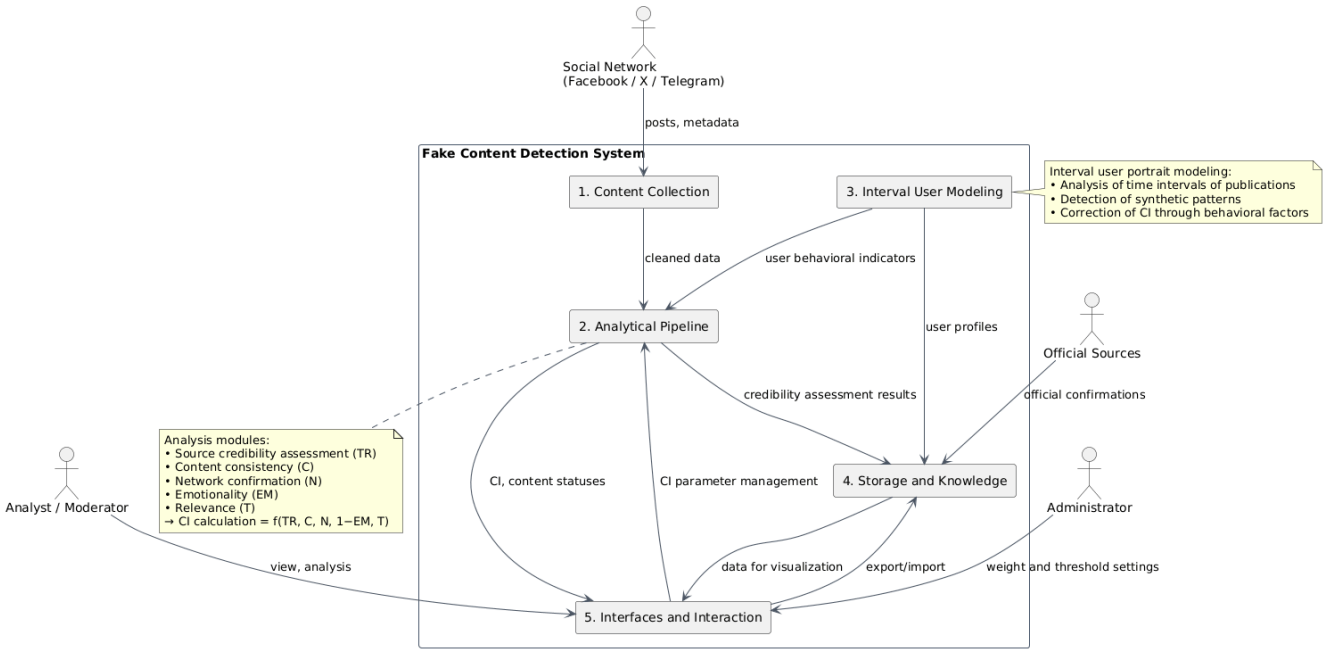

Figure 4.2. UML diagram of system packages

The Integration & Data Acquisition package contains modules responsible for acquiring primary data from social networks and web resources. The main components

are FacebookConnector and APIAdapter, which enable integration with the Graph APIs of social platforms and facilitate the retrieval of metadata related to posts, comments, reactions, and associated user profiles. These modules handle authentication of requests, access token management, API rate-limit handling, and the standardization of incoming data formats.

Data collected from various sources are passed to the ETLManager, which performs data cleaning, normalization, and preliminary filtering. At this stage, unstructured records are transformed into a format suitable for analytical processing and queued for subsequent modules. The ETLManager implements the Extract–Transform–Load (ETL) paradigm with support for streaming data processing.

The Content Processing & Analysis package performs the core algorithmic functions of the system. Its modules implement textual, semantic, network, and behavioral content analysis. Within this package, the FactConsistencyAnalyzer component identifies consistencies and contradictions among claims from different sources. The EmotionDetector module estimates the emotional tone (EM) of publications, as high emotionality is typically associated with lower credibility.

A key element of this package is the IntervalUserModel, which performs interval modeling of user behavioral characteristics. Using interval arithmetic, this module evaluates posting frequency, temporal activity intervals, and thematic variability, constructing trust intervals for individual users. This improves the stability of trust rate (TR) and integral credibility (CI) estimation under incomplete or heterogeneous data conditions.

The FewShotModule enables analysis when only a limited number of examples are available by combining probabilistic and interval-based methods. This functionality is crucial when dealing with emerging information campaigns or small-scale datasets.

The CIEngine serves as the final computational element within this package. It calculates the integral credibility index, $CI \in [0; 1]$, using weighted coefficients of the partial indicators (TR, C, N, EM, and T). The computed results are stored in the data repository and made available to the user interface for visualization.

The Models, Knowledge & Storage package is responsible for data accumulation,

formalization, and structuring. The KnowledgeGraph (KG) module maintains a knowledge base of relationships among events, entities, and attributes, using ontological models to establish semantic links between them. The UserProfileDB stores interval-based user profiles that capture temporal activity series and emotional behavior patterns. Each profile is dynamically updated based on a user's interaction history within the social network—new posts or reactions automatically refresh its parameters.

The FactCheckDB stores officially verified facts supplied by recognized fact-checking organizations. It serves as a reference dataset for validating claims extracted from ongoing information streams. This provides a semantic validation mechanism and enables the construction of trust relationships among data entities.

The Access & Visualization Interfaces package implements mechanisms for user interaction with the system and for integration with external applications. The DashboardUI component visualizes analysis results through interactive panels, network propagation graphs, and time series showing the dynamics of the credibility index (CI). The interface supports visualization of interval estimates as confidence bands, enabling users to track changes in credibility over time.

The RESTGraphQLAPI module provides programmatic access for external systems, analytical platforms, and mobile clients, allowing them to retrieve verification results and metadata. The SecurityManager ensures authentication, user authorization, and role-based access control (RBAC), thus maintaining the secure operation of the entire system.

At the bottom of the UML package diagram, the external actors interacting with the system are depicted. Social networks act as primary data sources that provide news content for analysis. Official sources supply verified information used for model training and result validation. The analyst and administrator interact with the system through the DashboardUI and RESTGraphQLAPI, allowing them to review analytical results, adjust weighting parameters, configure credibility thresholds, manage access to the fact database, and generate summary reports.

In summary, the package diagram illustrates the multi-layered organization of the system, where each package performs well-defined functions and interacts with others

through formalized interfaces. This separation ensures architectural flexibility, simplifies testing, enhances scalability, and allows independent updating or extension of individual modules. The diagram clearly represents the data flow—from initial content acquisition in social networks to the generation of integral credibility evaluations and their visualization in the user interface.

The class diagram formalizes the static structure of the software and outlines the responsibility interfaces among subsystems for data collection, credibility analysis, interval-based user modeling, knowledge management, and result visualization. In the data collection subsystem, the key classes are SocialMediaConnector and ETLManager.

The SocialMediaConnector encapsulates the logic of interaction with social network APIs (including Facebook, X/Twitter, and Telegram) and retrieves raw data such as posts, author profiles, and interaction metrics.

The ETLManager is responsible for data cleaning, normalization, and enrichment, transforming raw records into typed publication objects and metadata entities consistent with the internal storage schema. These classes are linked by a sequential dependency: the connector generates a stream of raw data objects, which are subsequently transformed by the ETLManager, after which the results are stored in the data repository.

The class diagram representing this structure is shown in Figure 4.3. Within the analytical subsystem, the primary classes are CredibilityAnalyzer, CIEngine, and FewShotModule, each implementing a specific aspect of the credibility assessment methodology.

The CredibilityAnalyzer computes the partial indicators forming the basis of the integral index—namely, source trustworthiness, fact consistency, network confirmation, emotional tone, and temporal relevance. The resulting structured object of criteria is then passed to the CIEngine, which performs weighted aggregation and converts the partial assessments into a normalized integral score in the range [0;1], producing a final verdict according to configurable thresholds.

For limited-data scenarios, when statistical information is insufficient, the FewShotModule performs score correction through active learning and Bayesian updating of prior distributions. Such an architecture ensures the stability of the integral

evaluation under conditions of rapidly changing information environments and dynamic semantic patterns.

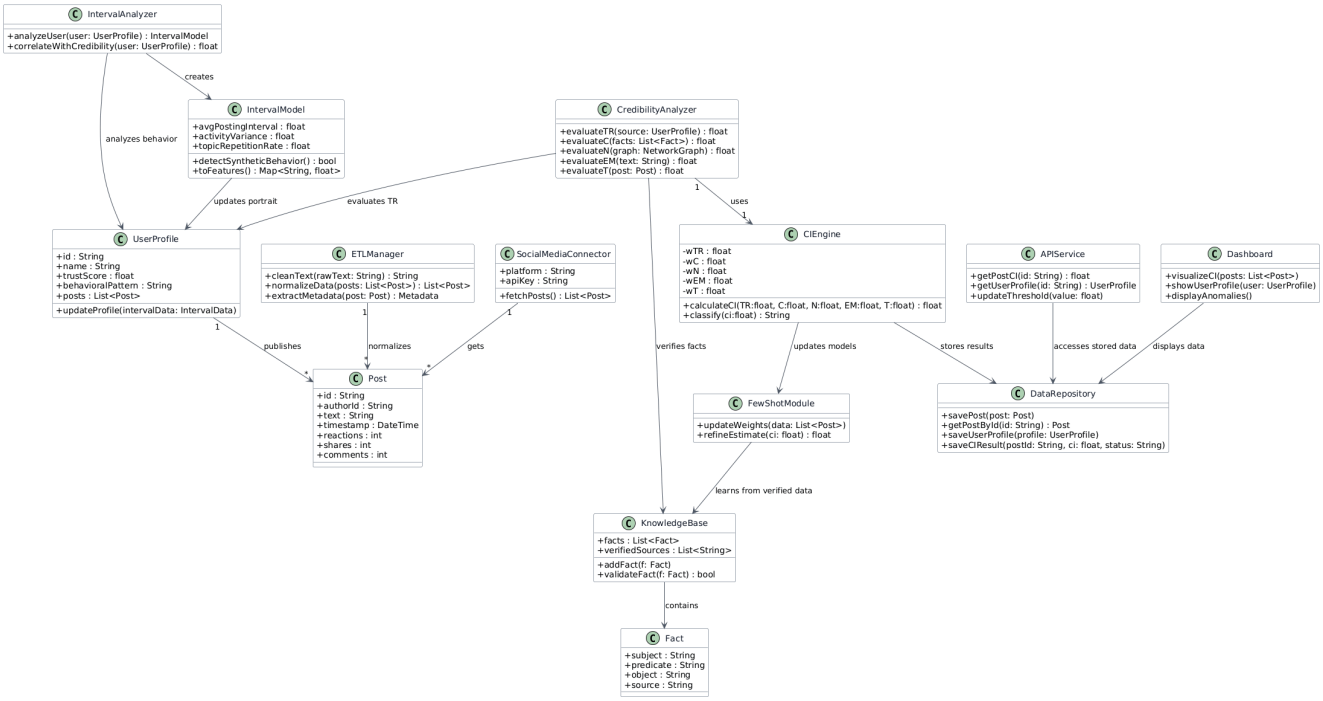

Figure 4.3. UML-діаграма класів системи

The interval-based user modeling subsystem represents an additional dimension of credibility linked to the behavioral characteristics of content authors and retransmitters. The UserProfile class aggregates a user's publication history, trust intervals, emotional profiles, and thematic stability indicators. The IntervalModel class formalizes temporal activity patterns in the form of interval representations, serving as a foundation for anomaly detection and behavior prediction. The IntervalAnalyzer class applies interval computation techniques to individual profiles, calculates sets of characteristic intervals, and provides mechanisms for assessing their impact on partial indicators—primarily on the source trust metric and, consequently, on the overall credibility index. Such interaction enhances the system's robustness to incomplete data and reduces noise sensitivity, since the assessment is performed not only in the point-value but also in the interval domain of possible values.

Knowledge and data management are performed by the KnowledgeBase and DataRepository classes, respectively. The former stores verified facts, entity

relationships, and outcomes of external fact-checking procedures, acting as a semantic reference for statement comparison. The latter encapsulates access to the repositories of publications, profiles, and evaluation results, implementing the repository pattern and ensuring data integrity invariants, record versioning, and computational reproducibility. At the class-level interactions, KnowledgeBase exposes interfaces for claim verification and entity linking, which are utilized by the CredibilityAnalyzer, while DataRepository guarantees idempotent storage of posts, profiles, and aggregated evaluations, offering transactional operations for sequential analytical pipelines.

User interaction and external integration are supported by the Dashboard and APIService classes. The former provides visualization of integral indicators, temporal trends, and network diffusion graphs, incorporating tools for intuitive interpretation of interval-based assessments. The latter delivers REST/GraphQL application interfaces for retrieving verification results, configuring weighting coefficients, and querying user profiles. Communication between interface and analytical classes relies on stable contracts that reduce coupling and simplify the evolution of the data model.

At the auxiliary-entity level, the diagram includes the Post, Evaluation, Criteria, Interval, IntervalSet, and Source classes. The Post object describes a publication—its author, content, and primary interaction metrics—while Evaluation stores partial indicators and the integral result, accompanied by timestamps and verdicts. The Criteria class generalizes the output of CredibilityAnalyzer, whereas Interval and IntervalSet constitute the canonical representations of interval estimates of trust, emotionality, and stability. The Source class models source properties, including domain affiliation and reputation score. A composition relationship connects Post and Evaluation, highlighting that each evaluation belongs to a specific publication; an authorship association links UserProfile with Post; and a dependency relation between Evaluation and Criteria specifies the derivation of the integral score from a set of partial indicators.

The architectural pattern selection in the class diagram is justified by the requirements for scalability, transparency, and analytical reproducibility. The SocialMediaConnector class implements an adapter for heterogeneous APIs, ensuring a unified data-collection contract regardless of the underlying platform. The

DataRepository class embodies the repository abstraction and conceals data-access specifics, while the CredibilityAnalyzer–CIEngine pair implements a configurable aggregation strategy, allowing replacement of computation policies without altering client code. The interval-modeling classes form an independent computational contour that interacts with analytics through a contract that adjusts the source trust metric according to interval-based evidence, thereby adhering to the principles of dependency inversion and the open/closed design paradigm.

Consequently, the class diagram illustrates a coherent system composition in which the flow from raw data to interpreted conclusions is realized as a sequence of operations over well-defined entities. The coexistence of both point and interval evaluations ensures robustness to noise and data incompleteness, whereas the separation of subsystem responsibilities enables independent component scaling and supports diverse operation scenarios—from interactive analytics to continuous stream monitoring. The presented model serves as a foundation for further verification of architectural decisions in deployment and sequence diagrams, ensuring full traceability of requirements down to the implementation level of software interfaces and data-storage schemas.

The deployment diagram formalizes the physical placement of the software components and their communication channels within the production environment. The system operates as a network of interconnected nodes that communicate through network protocols, message queues, and application interfaces, supporting continuous data acquisition from social networks, analytical processing, construction of interval-based user profiles, and publication of results through the web interface and API.

Figure 4.4 presents the deployment diagram of the system, showing an analytical cluster with microservices Collector, ETL, Analyzer, Interval, CIEngine, and FewShot; an application server with a web dashboard, API, and security subsystem; a MongoDB cluster with collections posts, user_profiles, facts, config, and logs; a separately deployed user interval-modeling subsystem; and external data sources and system users.

The analytical cluster constitutes the system’s core, executing the processing pipeline from publication ingestion to computation of the integral Credibility Index (CI). Within the cluster, the microservices Collector, ETL, Analyzer, Interval, CIEngine, and

FewShot are horizontally scalable under container-orchestration control. Inter-service communication is handled via an internal message bus providing buffering, retries, and resilience under peak loads.

The application server exposes REST/GraphQL interfaces for integration with external clients and hosts the web dashboard for analysts and administrators. This node concentrates authentication and authorization subsystems (SSO/RBAC), as well as notification and reporting mechanisms. The application server interacts with the analytical cluster over the internal data-center network using standardized API contracts.

The MongoDB cluster serves as the primary data repository. Its logically separated collections—posts, user_profiles, facts, config, and logs—store, respectively, social-media content, interval-based behavioral user models, verified facts from official sources, configuration parameters (weights and CI thresholds), and event logs. Such separation improves data isolation and simplifies access-control management.

The interval-based user modeling subsystem is deployed as a dedicated node optimized for statistical and Bayesian computations in Python. This subsystem periodically retrieves activity histories from the posts and user_profiles collections, constructs interval profiles (e.g., inter-publication times, topic recurrence, diurnal rhythm), and returns correction coefficients that refine the trust (TR) and overall CI metrics. The functional sequence within this node—IntervalAnalyzer → IntervalModeler → ProfileUpdater—ends with the updated characteristics being written back to MongoDB and the coefficients returned to CIEngine.

External data sources include social networks, from which publications are obtained via public APIs, and official fact-checking resources that provide benchmark confirmations or refutations of news items.

End users—analysts and administrators—interact with the system through a browser-based interface and the application server’s programmatic APIs, initiating verification procedures, adjusting weights and thresholds, and generating analytical reports.

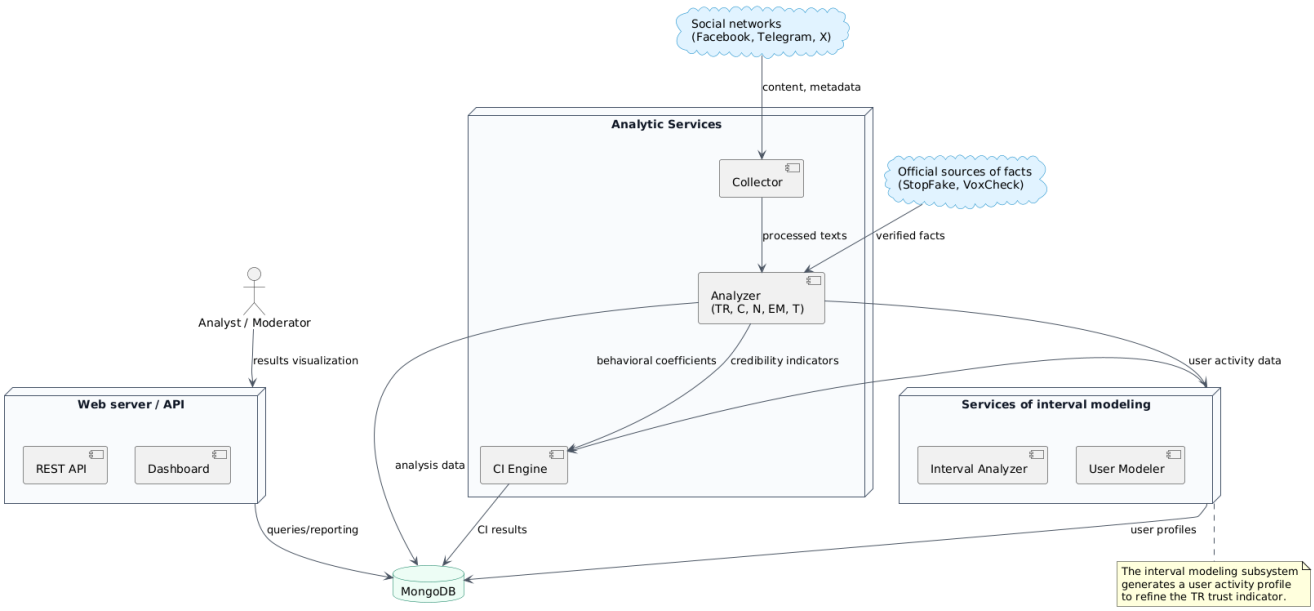

Figure 4.4. UML system deployment diagram

The combination of these nodes and connections in the deployment diagram illustrates the isolation of computationally intensive tasks within the analytical cluster, the presence of a centralized data repository, and a clearly defined access corridor to the results through a unified application node. The following section of this work examines the specific features of the information-storage subsystem implementation.

4.2 Information Analysis and Storage Subsystem

The information storage subsystem constitutes the core of the developed intelligent system for detecting and analyzing fake content in social networks. It is built on the document-oriented database management system MongoDB, which provides flexible processing of semi-structured data typical for social media content. Unlike classical relational approaches, MongoDB enables the storage of documents with arbitrary structure, supports dynamic schema evolution without service interruption, and efficiently operates under conditions of high update frequency and large volumes of user-generated content.

Functionally, the subsystem serves as a centralized repository that stores all categories of data circulating among the system’s modules: primary social network

content, analytical credibility assessment results, interval-based behavioral user profiles, the repository of verified facts, configuration parameters, and technical telemetry. All data are stored in the form of collections logically grouped according to functional purpose and supporting differentiated access schemes for analytical, service, and administrative subsystems (Figure 4.5). The source code listing of the information storage subsystem implementation in MongoDB is provided in Appendix A.

The central collection is posts, which contains both the raw publication texts and the results of their analysis. Each document includes the author's identifier, timestamps, quantitative interaction metrics (number of likes, comments, shares, unique users), and the computed credibility indicators: partial criteria TR (source trust), C (statement consistency with verified facts), N (network confirmation), EM (emotional tone), and T (temporal relevance). The document also stores the integral indicator CI, which characterizes the overall credibility of the news item, along with the corresponding verdict (credible, needs_review, or suspicious).

The posts collection serves as the primary information source for most analytical processes: the CIEngine subsystem writes computed results to it, IntervalAnalyzer uses it to build user activity time intervals, and Dashboard/API queries it to visualize the dynamics of credibility indicators. To ensure high-performance data access, indexes are applied to the fields created_at (for temporal aggregation), author_id (for building user history), and CI (for rapid retrieval of analyzed content). A partial index with the filter { CI: { $exists: true } } minimizes index size and improves performance when analyzing only processed documents.

For large-scale datasets, the collection is sharded using the key { network: 1, created_at: 1 }, which evenly distributes the load among cluster nodes.

The user_profiles collection is designed to store interval-based user behavior models that characterize temporal activity, thematic stability, emotional tone, and trust levels. It preserves the results generated by the interval modeling subsystem, which analyzes the time series of user posts and reactions.

The document structure includes the fields trust_interval, emotion_interval, stability_interval, and an activity_series array that accumulates inter-publication time

intervals and activity-rhythm indicators. The updated_at field tracks profile freshness.

Indexing is organized by a unique user_id key and by update time to monitor profile "aging." For trust-oriented queries, a partial index on trust_interval.hi is applied. The collection can be sharded by a hash of user_id, ensuring uniform data distribution during system scaling.

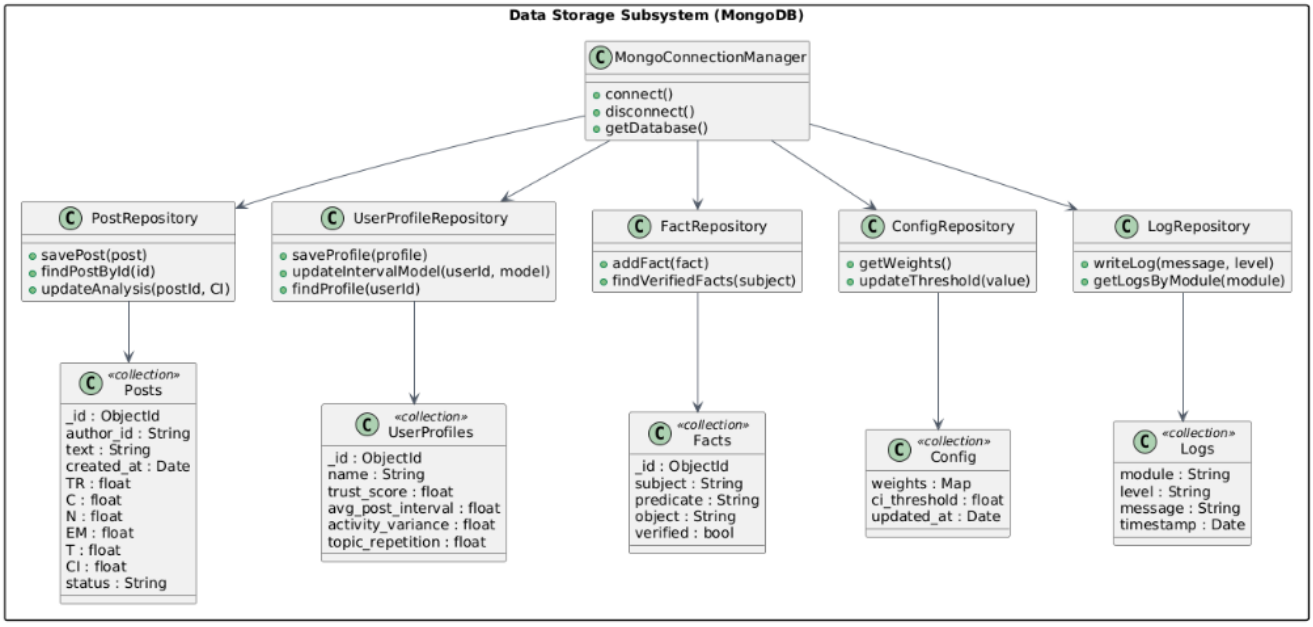

Figure 4.5. Structure of the information storage subsystem

The facts collection contains verified statements from official fact-checking sources, which are used for content validation and model retraining. Each document includes a normalized version of the claim text, verification status (true, false, or mixed), verification source, reference link, and the last-updated timestamp.

Access to this collection is provided via the FactRepository module, which performs semantic matching of text fragments with claims using full-text search and vector similarity. A text index on claim_norm and an auxiliary composite index { topics: 1, verdict: 1 } enable rapid filtering of facts by topic or verification outcome. Since the collection size remains relatively stable, sharding is generally not applied.

The config collection functions as a centralized storage for parameters that define the credibility-calculation policy. It stores the weighting coefficients of CI criteria, content-filtering parameters, and version tags.

Each configuration change creates a new document with a version number, administrator comment, author, and timestamp. This approach enables retrospective reproduction of any CI computations under the configuration valid at the time of execution.

For quick access to the active configuration, a partial index on the active field is used, containing only one document marked as current. The version history is accessible through an index on version, allowing parameter changes to be compared over time.

The logs collection serves as the system event journal, containing service information on computation progress, errors, configuration changes, and component-level performance metrics. It supports both real-time monitoring and post-hoc auditing.

Each record includes the fields ts (timestamp), level (severity), component (subsystem identifier), event, details, and trace_id, which ensures call-traceability within a single processing scenario.

Performance optimization is achieved through indexes { ts: -1 } and { level: 1, ts: -1 }, as well as a TTL index that automatically deletes records older than a predefined period (typically 90 days), with parallel archiving to cold storage. This strategy maintains database compactness without losing historical traceability.

To ensure performance and scalability, the storage subsystem employs a distributed architecture with sharding based on temporal and user keys and asynchronous caching of the most frequently requested queries in Redis.

Shard-key selection is workload-driven: in the posts collection, data are distributed by network and created_at, reducing the risk of hotspot shards during peak activity; in user_profiles, the hashed user_id guarantees even profile distribution.

Frequently executed analytical aggregates (e.g., CI time-series plots or user-trust distributions) are pre-computed as materialized views, periodically refreshed by background tasks.

Consistency is maintained under an eventual consistency model, since most analytical computations do not require immediate synchronization.

For critical operations (CI and profile updates), idempotent upsert operations with version control by timestamps are applied. Each document includes a schema_version

field, enabling tracking of data-structure evolution and compatibility across analytical-module versions.

The subsystem also implements a role-based access control (RBAC) policy. Access to collections is restricted by execution context: analytical modules have write permissions only to posts and logs, the interval modeling subsystem may update user_profiles, and administrators manage config. All operations are logged, and sensitive data (user identifiers, tokens, addresses) are stored in a pseudonymized form.

Reliability is ensured through backup mechanisms (daily full snapshots plus incremental change logs) and replication to a secondary cluster, enabling full state recovery to any date via oplogs and ensuring operational continuity in case of system failures.

Thus, the MongoDB-based information storage subsystem provides seamless integration across all system layers—from content acquisition and credibility assessment to the construction of interval-based user profiles and visualization of results in the web interface.

Its architecture combines high performance and flexibility, supports scalable operation, guarantees reproducibility of analytical outcomes, and serves as a key infrastructural backbone for the functioning of the entire fake-content detection system in news-oriented social networks.

4.3 Organization of the Graphical Interface of the System for Detection and Analysis of Fake Content in News-Oriented Social Networks

To implement the developed intelligent system for assessing the credibility of information in social networks, a modern technology stack was employed, ensuring architectural modularity, high scalability, functional extensibility, and seamless integration with external platforms—namely Facebook, Telegram, and X/Twitter. The primary focus was placed on designing an architecture capable of operating effectively under dynamically changing data streams, heterogeneous content formats, and asynchronous information sources.

The server side of the system is implemented in Python 3.12, which provides flexibility in developing analytical modules, processing textual data, interacting with APIs, and applying machine learning techniques. The code listing is provided in Appendix B.

The architecture follows a modular design, in which each component performs a clearly defined function—from data collection and preprocessing to the construction of interval-based user models, credibility evaluation, and visualization of analytical results.

This approach simplifies maintenance and testing while enabling the independent scaling of modules. Interaction between subsystems is carried out through a RESTful API, which facilitates structured data exchange between the client interface, analytical core, and external services.

For data storage, the system employs MongoDB, a document-oriented database management system that supports flexible handling of semi-structured data in JSON format. This approach is particularly well-suited for social network data, where content structure, metadata volume, and the level of granularity vary significantly across sources.

The repository stores both the raw posts collected from social platforms and the computed credibility indicators, behavioral user characteristics, verified fact databases, and configuration parameters. The use of JSON structures allows analytical results to be integrated directly with the web interface and visualization modules without additional data transformation.

Mathematical and analytical computations are performed using the NumPy and pandas libraries, which support statistical analysis, normalization of indicators, and construction of interval estimates.

For fuzzy data processing and modeling of uncertainty, SciPy is applied, enabling the implementation of algorithms for interval comparison and fuzzy logic in the credibility evaluation of news content. Result visualization is achieved through Matplotlib and Chart.js, providing graphical representations of changes in the Cumulative Credibility Index (CI) over time, visual displays of user interval profiles, and comparative performance metrics.

System integration with social networks is achieved through official APIs. In the

demonstration implementation, Facebook Graph API is used to access publication metadata, including the number of likes, comments, shares, and unique users.

For obtaining data from other platforms or open sources where APIs are unavailable, the system utilizes Requests and aiohttp for asynchronous HTTP requests, as well as BeautifulSoup and Selenium for HTML parsing, automated web navigation, and real-time data collection. This combination of tools provides flexibility in working with diverse content sources and minimizes dependency on the limitations of specific APIs.

The user interface of the system is implemented using HTML5, CSS3, and JavaScript (ES6), creating a responsive and dynamic web application compatible with any modern browser.

The interface styling employs TailwindCSS, which enables a clean, scientific, and visually light design with adaptive rendering across devices. The use of Chart.js on the client side allows the construction of interactive visualizations—including radar, bar, and time-series charts—that enable analysts to quickly interpret CI evaluation results, identify trends, and detect anomalies in news-content dissemination.

For server-side logic, the system uses Flask or FastAPI, frameworks that support efficient REST service operation with low latency and asynchronous request handling.

Both integrate seamlessly with Python data-processing libraries and MongoDB, ensuring code compactness and ease of deployment.

Docker is used for containerization, providing isolated environments for each system component, improving reliability, simplifying server deployment, and ensuring experimental reproducibility. Version control, collaborative development, and module publication are supported through Git and GitHub, which provide CI/CD pipelines, automated testing, and code documentation.

Thus, the chosen technology stack integrates the advantages of modern analytical, machine-learning, and web-development tools, ensuring a complete operational cycle of the intelligent system—from data acquisition to user-oriented visualization of results.

The combination of Python as a universal computational language, MongoDB as a flexible data repository, and Flask/FastAPI as lightweight web frameworks forms a

foundation for further system scaling, functional enhancement, and integration with other fact-checking or information-monitoring services.

This technological synergy ensures scientific reproducibility, software stability, and adaptability of the system to various operational scenarios within the modern digital environment.

Graphical User Interface Design.The graphical user interface (GUI) of the developed intelligent system for evaluating information credibility in social networks is implemented within a modern web-oriented environment, using HTML5, CSS3, JavaScript (ES6), and the TailwindCSS framework. This combination provides aesthetic simplicity, responsiveness, and a clean, science-oriented visual style. The interface follows a component-based architecture, where each page performs a specific function within the overall process of news-content analysis, credibility assessment, user-profile modeling, and result visualization.

The main page of the system (Figure 4.6) serves as an analytical dashboard, integrating key credibility indicators, the status of ongoing verification processes, and the trends of information dynamics in social networks.

At the center of the page lies the Cumulative CI Chart—an interactive visualization of the integral credibility index over the past week.

The chart supports point-click interactivity: when the user selects a data point, a tooltip window displays details such as the news source, publication time, and brief context.

Above the chart, a dynamic time-interval filter allows users to change the analysis period using the buttons “24 h,” “7 days,” and “30 days.”. Adjusting the interval automatically updates the CI chart and all related widgets on the page, providing real-time dynamic analytics.

In the right section of the main interface, a Source Panel is displayed in the form of an interactive pie chart, illustrating the proportion of verified versus questionable information sources. By clicking on a sector of the chart, the user can access an expanded list of corresponding sources along with their individual credibility scores.

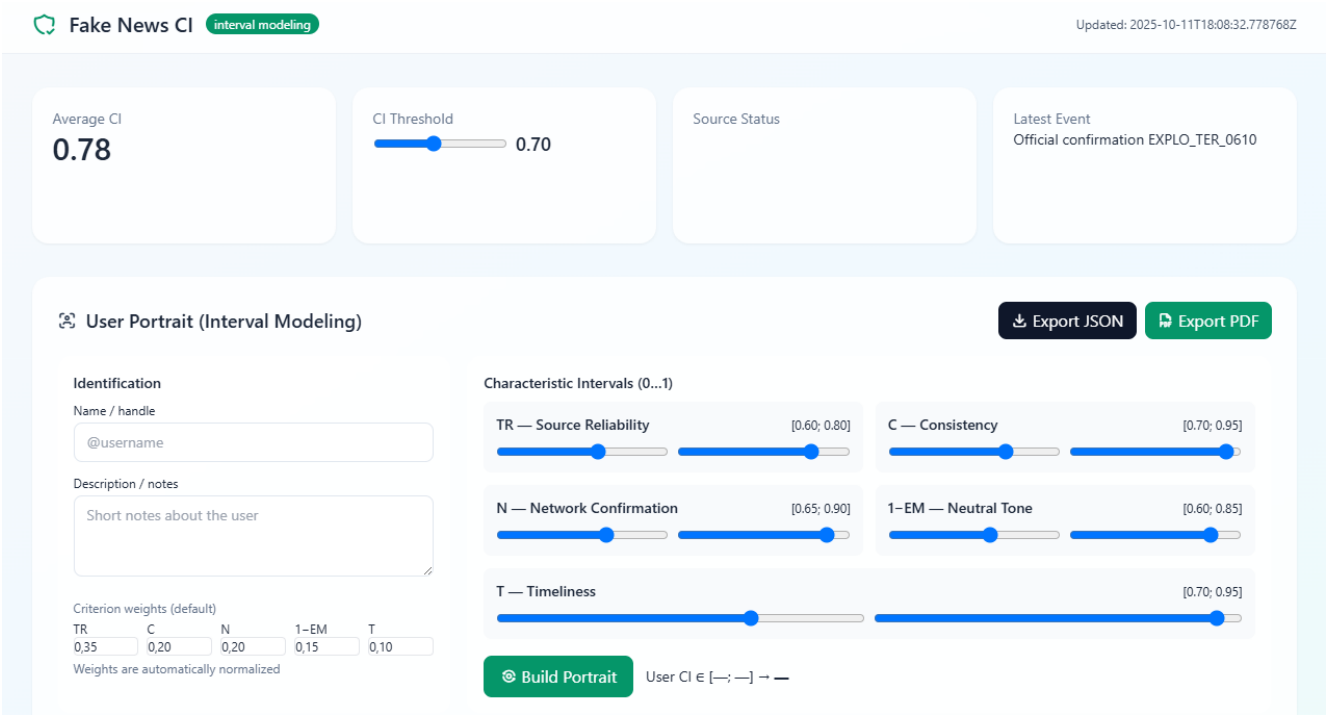

Figure 4.6. System Home page

Below this panel, a content emotionality heatmap is presented—a color-coded visualization where green shades correspond to neutral or constructive messages, while red hues denote toxic or emotionally biased publications.

This visual component is integrated with the Emotional Analysis (EM) module, enabling rapid identification of potentially hazardous or manipulative information waves within the network.

Complementing the main dashboard is a User Activity Mini-Panel, which displays the Top-5 authors ranked by posting frequency and trust level (TrustScore). For each author, the interface shows the number of posts, the average Credibility Index (CI), and the temporal dynamics of trust variation.

The Post Analysis Page (Figure 4.7) is designed to provide detailed information about each news item retrieved from social networks.

It features an interactive search and verification tool, allowing the user to input a URL or keyword phrase, upon which the system automatically performs cross-referencing against the fact-checking database and ontological relationships.

Figure 4.7. Post analysis page

The results are presented in the form of a post card, which includes the post text, image, source, timestamp, and a summary of the calculated criteria — TR, C, N, EM, and T. Next to it, the Network Explorer is displayed—an interactive information-verification graph, where nodes represent individual sources or publications, and edges denote their logical relationships (confirmation, contradiction, or citation).

The color of each node dynamically reflects its integral credibility index (CI), allowing users to visually assess the networked interaction among sources. A “Refresh” button is provided to update all metrics and record the latest timestamp in the bottom panel.

The Content Credibility Evaluation Page serves as the core analytical element of the system. It integrates the results of textual, behavioral, and emotional analyses for each news item.

To facilitate interpretation, several interactive panels are provided: a time-series chart of CI variation, a histogram showing the distribution of TR and EM indicators, and a detailed criteria table.

Users can modify the weight coefficients of individual criteria in real time and immediately observe how these adjustments affect the final credibility score. This functionality is actively used by analysts to test the model’s sensitivity and adapt its parameters to the specific characteristics of the news environment.

A separate functional module is the User Interval Portrait Page, which implements the interval-based behavioral modeling subsystem (Figure 4.8). This interface displays a

timeline of user activity, intervals between posts, topic stability, and the trust interval (Trust Interval).

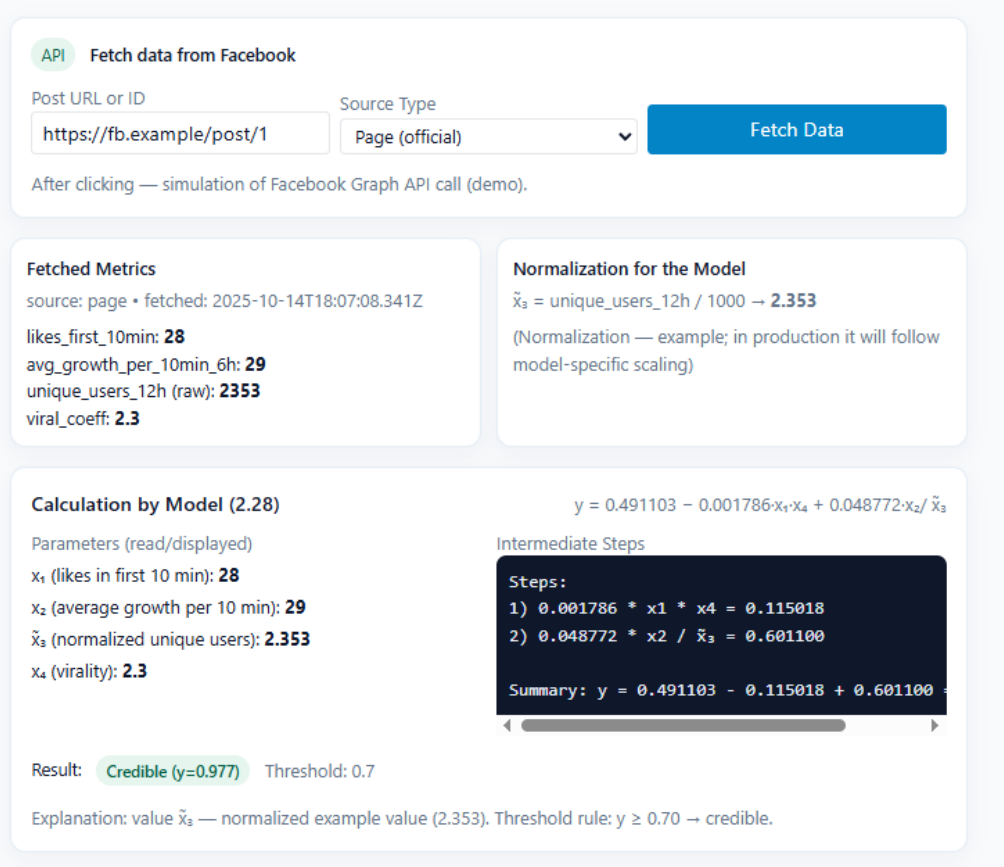

Figure 4.8. Interval modeling page

The information is visualized through interactive charts and radar diagrams, enabling the assessment of balance among activity, emotionality, and profile stability. Visualization is implemented using Chart.js and supports the overlay of multiple user profiles for comparative analysis.

The page also provides an interval forecasting option, allowing the system to estimate the probability of upcoming posts within a defined time window—an important feature for detecting behavioral anomalies during the evaluation of analytical results.

The User Profile Page consolidates information about the user, their role within the system (analyst, administrator, or researcher), activity history, and personal interface preferences (Figure 4.9).

It provides options for selecting a theme, time filters, and data-refresh modes,

enabling flexible adaptation of the workspace to the user's analytical needs.

The Settings Page offers administrative functionality, including management of criteria weights, updating of the verified-facts database, adding new sources to the "whitelist" or "blacklist," and control of CI threshold values.

All configuration changes are automatically recorded in the system log (logs), ensuring full reproducibility of parameter configurations and facilitating auditing of administrative actions.

The News Analysis Page acts as a content aggregation hub that streams information from multiple platforms.

It allows simultaneous visualization of news from Facebook, Telegram, and X/Twitter within a unified interface, with preliminary classification according to credibility level.

Each news block includes a CI indicator, whose color gradient ranges from green (credible) to red (potentially fake), providing an intuitive overview of the current information landscape.

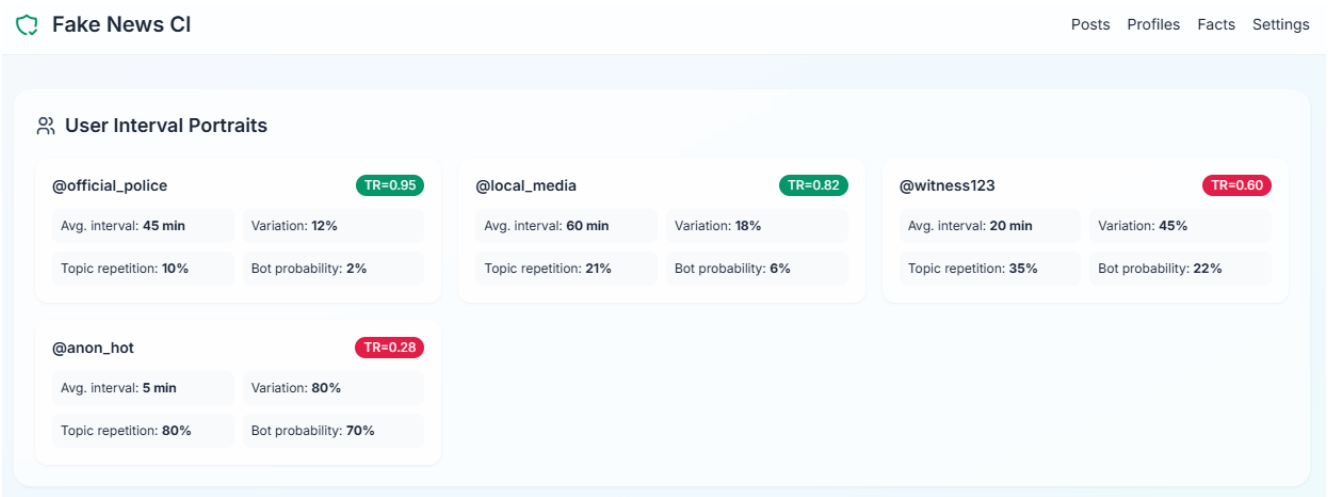

Figure 4.9. User profile rating page

Users can apply dynamic time-interval filters that synchronously update all analytical charts.

This approach enables the analysis of information waves, tracking of news dissemination speed, and timely response to the emergence of questionable or potentially

manipulative content.

At the final stage, the user can navigate to the Results Page (Figure 4.10), which presents a comprehensive system report containing aggregated credibility indicators, the Information Stability Index, statistics on emotional patterns, and summarized conclusions regarding the verified sources.

Results can be exported in PDF or CSV formats for subsequent analytical processing.

All graphical components of the system support live-update functionality—the “Refresh Data” button triggers an API request to retrieve the latest computations and visualize them in real time.

Thus, the implemented graphical user interface integrates the capabilities of an analytical dashboard, an interactive content verification tool, and a trust-monitoring system for information sources across social networks.

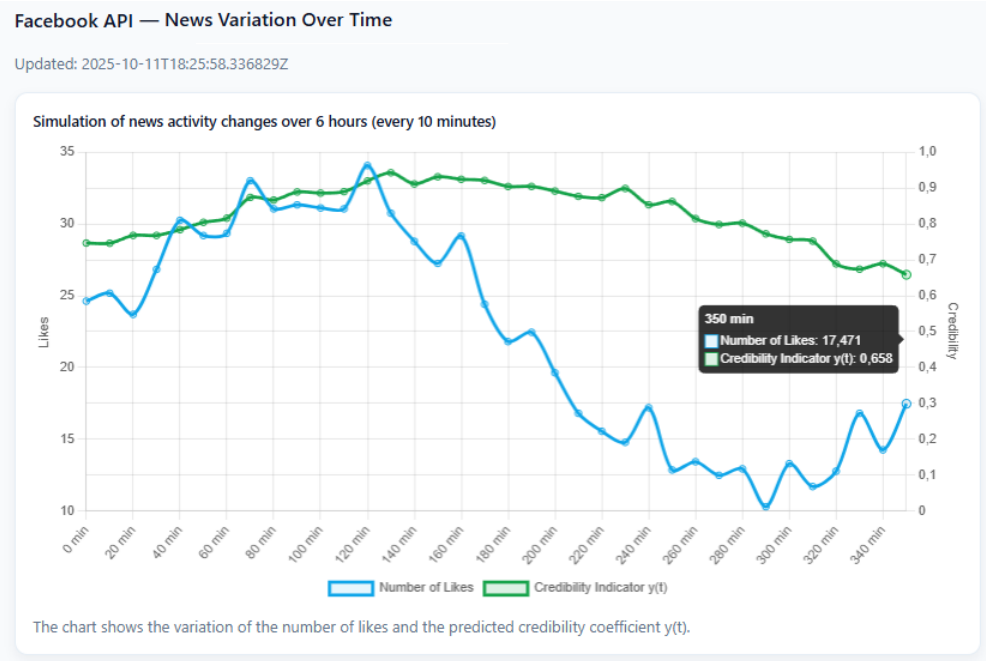

Figure 4.10. Results display page

Thanks to its adaptive design, interactive filters, multi-level visualization, and capability to operate with temporal intervals, the system ensures not only a high level of informativeness but also provides deep analytical immersion into the processes of

information dissemination and credibility assessment within the digital environment.

4.4 Evaluation of the Effectiveness of the Developed Software Environment

To enable a quantitative comparison of the development level and functional balance of fake-content detection systems, an Integral Efficiency Index (I_E) was developed. This index makes it possible to assess the degree of comprehensiveness, analytical precision, and cognitive usability of such systems. The I_E serves as a generalized metric that formally represents the technical, analytical, and user-oriented efficiency of credibility-assessment tools, thus providing a means for their self-evaluation or cross-system comparison.

The concept behind the construction of the integral index is based on the principle of aggregating partial system characteristics using a weighted model. Each characteristic is denoted as K_i, where i takes the value 1 if the functionality is fully implemented, 0.5 if it is partially implemented, and 0 if it is absent. The corresponding weight coefficient w_i в reflects the relative importance of each criterion in the context of fact-checking and content credibility analysis systems. The set of all weights is normalized such that their sum equals unity:

$$\sum_{i=1}^{n} w_i = 1, \tag{4.1}$$

where n — number of criteria.

To calculate the Integral Efficiency Index (I_E), the following formula is used:

$$I_E = \sum_{i=1}^{n} K_i \times w_i, \tag{4.2}$$

where I_E — the generalized efficiency level of the system; K_i — the presence or degree of implementation of a specific function; w_i — the weight coefficient of the corresponding criterion.

In the course of the study, a set of fourteen criteria was formulated to represent the key properties of modern systems for content-credibility assessment.

These include analytical accuracy (use of NLP/AI analysis, automated scoring, and the integral credibility index CI); network integration (connection with social platforms, API accessibility, and real-time operation); analytics and visualization (graph generation, virality analysis, and content emotional tone); source reliability (availability of a verified fact database and algorithmic transparency); and user orientation (ease of claim input, search functionality, and structured interaction with results).

For each group of criteria, weight coefficients were determined to reflect their relative contribution to overall efficiency:

Analytical accuracy — 0.40, with detail by subcriteria: NLP/AI (0.15), automated scoring (0.15), integral CI (0.10);

Network integration — 0.30, with subcriteria: social-network integration (0.10), API availability (0.10), real-time mode (0.10);

Analytics and visualization — 0.20, with subcriteria: visualization (0.08), virality analysis (0.07), emotionality (0.05);

Reliability and sources — 0.10, with subcriteria: verified-source database (0.05), transparency (0.05);

User orientation — 0.10, including claim input, search, and interaction (total 0.10).

The integral index was computed for five systems: Google Fact Check Explorer, ClaimBuster, Logically Facts (AI), Hoaxy, and the Proposed System—the developed intelligent system for assessing information credibility in social networks.

For the latter, all analytical and visualization modules were fully implemented, while the real-time support and verified-sources database were partially implemented (rated 0.5).

After substituting the corresponding weights into the formula for the integral index, the following result was obtained: $I_E(\text{Proposed System}) = 0.92$, which indicates a high degree of functional comprehensiveness and balance within the system.

For comparison, the integral efficiency values of other systems, evaluated under the same criteria, are as follows: Logically Facts (AI) — 0.78, ClaimBuster — 0.65,

Google Fact Check Explorer — 0.58, and Hoaxy — 0.61.

The obtained results demonstrate that the proposed system outperforms existing analogues in terms of the integral efficiency index, primarily due to its implementation of multi-channel content acquisition, the presence of interval-based user-modeling modules, combined credibility evaluation (CI model), and the integration of emotional-semantic analysis.

For convenient interpretation of the IE values, an efficiency-level scale is introduced:

- systems with values from 0.85 to 1.00 are classified as highly efficient;
- 0.65–0.85 as moderately efficient;
- 0.45–0.65 as limited efficiency;
- below 0.45 as low efficiency.

Accordingly, the developed system confidently belongs to the class of high-efficiency tools for content-credibility analysis.

To visually represent the results, a radar chart was constructed (Figures 4.11 and 4.12), where the axes correspond to the main groups of criteria: analytics (AI/NLP), network integration, real-time operation, transparency, emotionality, and CI-based comprehensiveness.

This visualization clearly illustrates the balance of characteristics of the proposed system and its advantage over existing analogues according to the generalized Integral Efficiency Index.

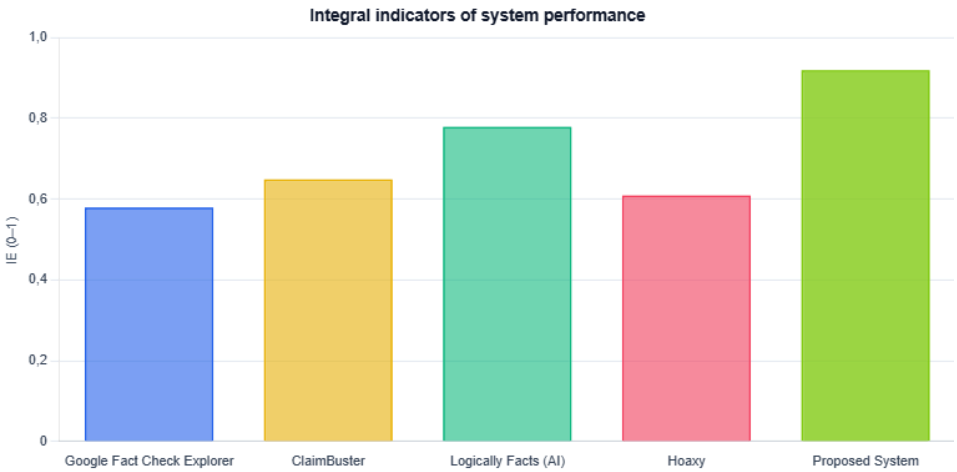

Figure 4.11. Integral indicators of system performance

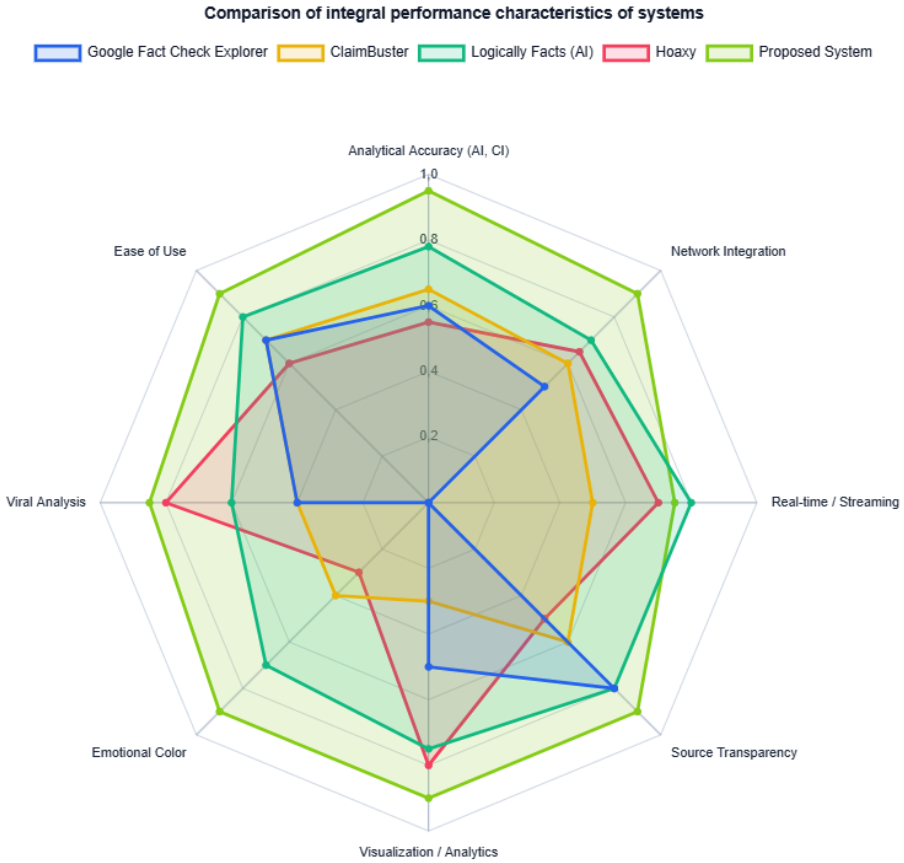

Figure 4.12. Comparison of integral performance characteristics of systems

Thus, the application of the Integral Efficiency Index makes it possible to objectively evaluate the competitiveness of fact-checking systems, formalize the process of self-assessment, and substantiate the scientific and technical advantage of the developed system within the framework of multi-criteria analysis of information credibility in social networks.

The index serves as a generalized metric that formally reflects the technical, analytical, and user-oriented efficiency of credibility verification tools, ensuring the possibility of both self-evaluation and cross-system comparison.

An analysis of modern tools for information credibility verification has shown that most existing systems implement only partial approaches to fact-checking, focusing either on the search for verified sources or on the detection of questionable claims.

However, in the context of social networks, where information spreads rapidly and often carries emotional or manipulative connotations, there is a growing need for

integrated systems capable of accounting for textual, behavioral, temporal, and network-based aspects of content.

Google Fact Check Explorer is one of the most well-known and convenient services for the rapid retrieval of previously verified claims.

Its key advantage lies in the availability of a large verified-fact database accessible through a standardized interface. However, this tool performs only a search function and does not conduct deep semantic or emotional content analysis.

It therefore represents a passive system, which does not independently evaluate the credibility of new statements but merely matches them with already verified sources. ClaimBuster was one of the first systems to introduce machine learning for the automatic detection of potentially questionable claims in news texts and political speeches.

Its strength lies in rapid preliminary filtering of information that may require further verification by human fact-checkers. Nevertheless, the system lacks multilingual support and advanced visualization tools, which limits its applicability for analytical or research purposes.

Logically Facts (AI) is the most comprehensive among existing tools, combining artificial intelligence, semantic text analysis, automated credibility assessment, and social-network integration.

The system performs multifactor content analysis, taking into account context, source, emotional tone, and network relationships among authors. However, being a commercial and closed platform, its algorithmic transparency and integration capabilities with academic systems remain limited.

Hoaxy, in contrast, does not directly verify information credibility but focuses on visualizing the dissemination processes of news across social networks. This approach makes it possible to study information waves, detect network patterns of fake content, and analyze user behavior dynamics.

Although the system does not provide quantitative credibility indicators, its graph-based visualization serves as a unique tool for sociometric analysis of information flows.

Unlike the aforementioned systems, the Proposed Intelligent System (Table 4.1) implements a comprehensive approach, integrating analytical, behavioral, temporal, and

network characteristics within a unified architecture.

Its distinctive feature is interval-based user modeling, which allows the system to consider behavioral uncertainty and variations in user activity within social networks.

This enables the construction of an interval user portrait — a profile that describes not only the statistical characteristics of activity, but also the publication dynamics, thematic stability, frequency of repetitions, and trust level. Such an approach makes it possible to differentiate between natural, automated, and anomalous user activity, which is crucial for the detection of coordinated information attacks.

Table 4.1

Comparative Analysis Table of Fake Content Detection Systems

№	Characteristics	Google Fact Check Explorer	ClaimBuster	Logically Facts (AI)	Hoaxy	Proposed System Fake News CI
1	Base of verified sources	+	+/–	+	–	+/–
2	NLP / AI-text analysis	+/–	+	+	–	+
3	Input statement / quote	+	+	+	+/–	+
4	Keyword / URL search	+	+/–	+	+	+
5	Automatic credibility assessment	+/–	+	+	–	+
6	Integration with social networks	+/–	+	+	+	+
7	API / SDK access	+	+	+/–	+	+
8	Visualization / analytics	+/–	–	+	+	+
9	Transparency of sources	+	+/–	+	+/–	+
10	User Orientation	+	+	+	+	+
11	Real-time / streaming	–	+/–	+	+	+/–
12	Content Viral Analysis	–	–	+/–	+	+
13	Emotional content	–	+/–	+	–	+
14	Comprehensive CI assessment	–	–	+/–	–	+

Another advantage of the proposed system is the Composite Credibility Indicator (CI), which combines classical criteria—redundancy, contradiction, relevance, completeness, and source reliability—with new characteristics specific to the digital environment, such as network confirmation, emotional polarity, and the viral propagation coefficient.

This integrated approach enables the formation of a more balanced and

multidimensional credibility assessment, accounting not only for the intrinsic properties of the content but also for the behavioral dynamics of users involved in its dissemination.

The system includes built-in tools for automatic data collection from social networks, including a demonstration module based on the Facebook Graph API, which retrieves post metadata such as the number of likes, comments, unique users, and publication time.

These metrics are incorporated into the mathematical model used to compute the CI, allowing the credibility evaluation to dynamically adapt to real-time network activity.

Among other functional advantages, the system features advanced analytics, including the construction of content-diffusion graphs, trust-trend analysis, emotional maps, and user portraits.

The visualization interface supports emotion heatmaps, interactive time-series charts, and network interaction analysis tools, which not only enhance the analytical depth of the system but also make the results highly interpretable for users.

The system architecture is both open and modular, built using MongoDB as the primary data repository and REST API for integration with external scientific, journalistic, or governmental platforms.

Such an architecture ensures ease of scalability, adaptability to diverse data sources, and support for integration with future modules—for instance, for the analysis of video or audio content.

The comparative analysis has shown that the developed system outperforms existing analogues in terms of the i $I_E(\text{Proposed System}) = 0.92$, demonstrating the optimal balance among analytical, behavioral, and cognitive components.

By combining interval-based user modeling, the comprehensive CI approach, and a flexible data architecture, the system can serve not only as a fact-checking tool, but also as an analytical platform for information security monitoring, social-dynamics research, and trust assessment of information sources within the digital environment.

Conclusion of Chapter 4

1. In this section, a comprehensive software environment for the detection and analysis of fake content in news-oriented social networks has been developed and presented. The system integrates natural language processing methods, analytical credibility assessment, interval-based user behavior modeling, and modern visual analytics tools into a unified intelligent framework.

2. A modular software architecture was designed to ensure scalability, flexibility, and openness for integration with other information systems. It includes subsystems for content collection, analytical processing, interval user modeling, a knowledge base of verified facts, a data storage subsystem, and a web-oriented graphical interface. The modular structure enables isolated improvement of individual components, enhancing the reliability and reproducibility of the system when deployed across diverse platforms.

3. The information storage and analysis subsystem is implemented using the document-oriented database MongoDB, which contains collections of posts, user profiles, verified facts, configuration parameters, and event logs. Key-field indexing, sharding, and TTL storage mechanisms have been applied to optimize performance and maintain data relevance. Dedicated data-access classes were implemented to manage the interaction between analytical modules (CIEngine, IntervalAnalyzer, and FewShotModule) and the knowledge base. Thus, the subsystem supports the complete information-processing cycle—from acquiring content from social networks to storing the results of credibility analysis.

4. An interactive graphical user interface was developed to provide analysts and moderators with an intuitive environment for data interaction. It includes a main monitoring dashboard, post and news analysis modules, credibility evaluation tools, interval user-profile builders, and a settings dashboard. Interactive components such as time-interval filters, dynamic CI trend charts, content emotion maps, network diffusion graphs, and user-activity panels enable real-time analytics, tracking of information trends, and rapid identification of suspicious content.

5. The effectiveness of the developed system was evaluated using the Integral

Efficiency Index (I_E), which incorporates analytical, network, behavioral, and user-oriented parameters. The obtained value, $I_E = 0.92$, confirms a high degree of balance and effectiveness compared to well-known tools such as Google Fact Check Explorer ($I_E = 0.58$), ClaimBuster ($I_E = 0.65$), Hoaxy ($I_E = 0.61$), and Logically Facts ($I_E = 0.78$). The proposed system demonstrates the best trade-off between analytical accuracy, adaptability, and architectural flexibility, affirming its potential as a next-generation intelligent platform for the detection and evaluation of information credibility in social networks.

CONCLUSION

In this dissertation, a scientific and technical problem has been addressed—namely, the enhancement of the efficiency of fake-content detection and analysis in news-oriented social networks under conditions of limited data samples, achieved through the development of dedicated mathematical methods and software agents. As a result, the following scientific and practical outcomes have been obtained:

1. As a result of the analysis of modern methods and software tools for recognizing false, distorted, or irrelevant information, it was established that most existing solutions are primarily focused either on factual verification of textual messages or on statistical analysis of content dissemination, while failing to account for the complex, multifactor nature of information credibility in social networks. It was determined that existing approaches have significant limitations related to the insufficient automation of data verification processes, the lack of comprehensive consideration of emotional-semantic, temporal, and network characteristics of messages, and inadequate adaptation to the specific features of social media platforms. The conducted comparative analysis demonstrated the necessity of transitioning from isolated fact-checking systems to intelligent integrated frameworks capable of assessing credibility based on the combination of multiple groups of indicators — linguistic, contextual, temporal, network, and behavioural. Special attention was given to the importance of developing an interval-based approach to credibility assessment, which makes it possible to take into account data uncertainty and the partial incompleteness of observations that are inherent to social network content.

2. Quantitative indicators reflecting user profiles in a social network have been analyzed. It has been established that the use of quantitative characteristics of a community profile can assist in identifying signs of fake or false content, although it does not guarantee absolute accuracy. The analysis of such data can be valuable for detecting anomalies in audience behavior that are often typical of communities engaged in the dissemination of fake information. It has also been shown that the main indicators characterizing the audience’s reaction to specific content include: the number of posts,

shares, or likes generated by users within a short period after the appearance of the content, which helps to detect immediate audience reactions; the number of comments or reactions over specific time intervals, allowing a better understanding of the rate of information dissemination and the emotional feedback; the time span of information propagation across social networks (for example, how many users interact with the content within the first minutes, hours, or days after publication), which helps to evaluate the effectiveness of content dissemination; the viral dissemination coefficient, such as the number of reshares per user, which indicates whether the content is prone to rapid spread.

3. For decision-making regarding the credibility of content published in social networks, an interval mathematical model has been proposed and substantiated, which establishes the relationship between the outcome used to determine whether the content is credible or non-credible and the influencing factors. The resulting indicator of this model represents the degree of content credibility within the range from 0 to 1. It has been proposed and justified that, for representing and analysing this indicator based on expert evaluation of the content, methods of interval data analysis should be employed. Accordingly, this indicator is not interpreted as a probabilistic value but rather as a quantitative measure defined over a specific interval.

4. A hybrid method for identifying interval models of user profiles in a social network has been proposed and substantiated, which, unlike existing approaches, combines a metaheuristic algorithm for model structure synthesis based on the behavioural model of a bee colony with gradient methods for identifying the parameters of candidate models. This combination reduces the computational complexity of the identification process and enables the use of standard optimization tools for solving the problems of user-profile model identification in social networks.

5. Software agents for assessing the credibility of information in social networks have been improved to ensure a comprehensive approach to content analysis that combines classical verification criteria with the specific features of the digital information environment. Unlike existing approaches, which are primarily based on statistical or textual features, the proposed agents integrate ontological representation of facts, interval modelling of uncertainty, and network analysis of content dissemination,

thereby enhancing the reliability of the results within the dynamic information space of social media platforms.

6. A software environment for assessing the credibility of information in social networks has been developed, implementing a comprehensive approach to content analysis by integrating linguistic, temporal, behavioural, and network factors. The system is based on an enhanced credibility assessment method that accounts for both classical indicators — redundancy, inconsistency, timeliness, source reliability, and information completeness — and new characteristics of the digital environment, such as network confirmation, emotional tone, and the viral dissemination coefficient of content. A distinctive feature of the developed system is the use of interval modelling of user profiles, which makes it possible to consider the uncertainty of behavioural characteristics and the variability of user actions within social networks. This approach enables the construction of individual trust profiles based on the analysis of publication history, communication intensity, typical emotional reactions, and network interactions. As a result, the system is capable not only of evaluating the credibility of individual messages but also of identifying potential sources of non-credible content, generating integrated user reliability indices, and detecting anomalous activity typical of bot networks or coordinated information campaigns.

7. The evaluation of the effectiveness of the developed environment for analysing the credibility of news content in social networks demonstrated its superiority over well-known tools such as Google Fact Check Explorer, ClaimBuster, Logically Facts (AI), and Hoaxy. For an objective comparison, a system of criteria was established comprising 14 indicators, among which the key ones include analytical accuracy, the level of integration with social networks, the presence of automatic credibility assessment, support for analytical data visualization, transparency of information sources, and the availability of a comprehensive credibility index (CI). Based on the weighted evaluation for each system, the obtained results showed that the developed system achieved a score of 0.92, corresponding to a high level of efficiency. This result can be explained by the comprehensive nature of the proposed methodology, which integrates classical credibility criteria — redundancy, inconsistency, and timeliness — with new parameters

characteristic of the digital environment, such as network confirmation, emotional tone of the content, and the viral dissemination coefficient. Furthermore, the application of the interval approach to modelling user behavioural characteristics ensures the stability of the assessment even under conditions of incomplete or heterogeneous data.

REFERENCES

1. Alalshaqi, M.; Rawat, D.B.; Liu, C. Ensemble Techniques for Robust Fake News Detection: Integrating Transformers, Natural Language Processing, and Machine Learning. Sensors 2024, 24, 6062.

2. Al-alshaqi, M.; Rawat, D.B.; Liu, C. Ensemble Techniques for Robust Fake News Detection: Integrating Transformers, Natural Language Processing, and Machine Learning. Sensors 2024, 24, 6062. https://doi.org/10.3390/s24186062

3. Alguttar, A.A.; Shaaban, O.A.; Yildirim, R. Optimized Fake News Classification: Leveraging Ensembles Learning and Parameter Tuning in Machine and Deep Learning Methods. Appl. Artif. Intell. 2024, 38, 2385856.

4. Almeida F C, Guel A E, Silva A A A, et al. An outlier-based analysis for behaviour and anomaly identi cation on IoT sensors. International Journal of Sensor Networks, 2022, 39(2):106124.

5. Alsmadi, I.; Alazzam, I.; Al-Ramahi, M.; Zarour, M. Stance Detection in the Context of Fake News—A New Approach. Future Internet 2024, 16, 364. https://doi.org/10.3390/fi16100364

6. Al-Tarawneh, M.A.B.; Al-irr, O.; Al-Maaitah, K.S.; Kanj, H.; Aly, W.H.F. Enhancing Fake News Detection with Word Embedding: A Machine Learning and Deep Learning Approach. Computers 2024, 13, 239. https://doi.org/10.3390/computers13090239

7. Amiri, Z.; Heidari, A.; Navimipour, N.J.; Unal, M.; Mousavi, A. Adventures in Data Analysis: A Systematic Review of Deep Learning Techniques for Pattern Recognition in Cyber-Physical-Social Systems. Multimed. Tools Appl. 2024, 83, 22909–22973.

8. Anderson M., P. Wilson, "Interval arithmetic in optimization: Theory and applications," Applied Mathematics and Computation, vol. 456, 2023, pp. 1–18.

9. Bachelot, M.; Lyubareva, I.; Epalle, T.A.; Billot, R.; Lasseri, R.D. French fake news propagation: Multi-level assessment and classification. Soc. Netw. Anal. Min. 2024, 14, 156.

10. Berrondo-Otermin, M.; Sarasa-Cabezuelo, A. Application of Artificial Intelligence Techniques to Detect Fake News: A Review. Electronics 2023, 12, 5041. https://doi.org/10.3390/electronics12245041

11. Bobkowski, P.; Younger, K. News Credibility: Adapting and Testing a Source Evaluation Assessment in Journalism. Coll. Res. Libr. 2020, 81, 822

12. Chen Z, Qi W C, Bao T Y, et al. Data poisoning attack detection method for service quality aware cloud API recommendation system. Journal of Communications, 2023, 44(8):155167.

13. Chen, K.; Wang, Z.; Liu, K.; Zhang, X.; Luo, L. MedGraph: Malicious Edge Detection in Temporal Reciprocal Graph via Multi-Head Attention-Based GNN. Neural Comput. Appl. 2023, 35, 8919–8935.

14. Dai B, Xia Y, Li Q. An extreme value prediction method based on clustering algorithm. Reliability Engineering & System Safety, 2022, 222(6):112.

15. Darmorost I., M. Dyvak, N. Porplytsya, T. Shynkaryk, Y. Martsenyuk, V. Brych, "Convergence Estimation of a Structure Identification Method for Discrete Interval Models of Atmospheric Pollution by Nitrogen Dioxide," Proceedings of the 2019 9th International Conference on Advanced Computer Information Technologies (ACIT), Ceske Budejovice, Czech Republic, 2019, pp. 117–120. https://doi.org/10.1109/ACITT.2019.8779981.

16. Das P, Babadi B. Non-asymptotic guarantees for reliable identi cation of granger causality via the LASSO. IEEE Transactions on Information Theory, 2023, 69(11):74397460.

17. Del Vicario M. , A. Bessi, F. Zollo, F. Petroni, A. Scala, G. Caldarelli, H. E. Stanley, and W. Quattrociocchi, "The spreading of misinformation online," Proceedings of the National Academy of Sciences, vol. 113, no. 3, pp. 554–559, 2016. https://doi.org/10.1073/pnas.1517441113.

18. Dyvak M., "Parameters Identification Method of Interval Discrete Dynamic Models of Air Pollution Based on Artificial Bee Colony Algorithm," Proceedings of the 2020 10th International Conference on Advanced Computer Information Technologies

(ACIT), Deggendorf, Germany, 2020, pp. 130–135. https://doi.org/10.1109/ACIT49673.2020.9208972.

19. Dyvak M., Spivak I., A. Melnyk, V. Manzhula, T. Dyvak, A. Rot, M. Hernes, "Modeling Based on the Analysis of Interval Data of Atmospheric Air Pollution Processes with Nitrogen Dioxide due to the Spread of Vehicle Exhaust Gases," Sustainability, vol. 15, 2023, p. 2163. https://doi.org/10.3390/su15032163.

20. Dyvak, M., Yushko, A., Melnyk, A., Pan, T. An Intelligent Information System for Generating a Scientist's Scientometrics Using Content Analysis Methods. CEUR-WS. 2024. Vol. 3942. P. 66-82. https://ceur-ws.org/Vol-3942/S_06_Dyvak.pdf

21. Dyvak, Mykola, Tyande Pan, and Oleksandr Kindzerskyi. 2025. “Mathematical Model of a Social Network User Profile Based on Interval Data Analysis”. International Journal of Computing 24 (3):452-59. https://www.computingonline.net/computing/article/view/4182.

22. Dyvak, Mykola, Volodymyr Manzhula, Andriy Melnyk, Nataliia Petryshyn, Tiande Pan, Arkadiusz Banasik, Piotr Pikiewicz, and Wojciech M. Kempa. 2025. "Modeling the Electricity Generation Processes of a Combined Solar and Small Hydropower Plant" Energies 18, no. 9: 2351. https://doi.org/10.3390/en18092351

23. Emil, R.Ș.; Remus, B. A Review of Automatic Fake News Detection: From Traditional Methods to Large Language Models. Future Internet 2025, 17, 435. https://doi.org/10.3390/fi17100435

24. Ganesh A D, Kalpana P. Supply chain risk identi cation: A real-time data-mining approach. Industrial Management & Data Systems, 2022, 12(1):114.

25. Ghafoori M S, Soltani J. Designing a robust cyber-attack detection and identi cation algorithm for DC microgrids based on Kalman lter with unknown input observer. IET Generation, Transmission & Distribution, 2022, 16(16):32303244.

26. Glavind S T, Sepulveda J G, Faber M H. On a simple scheme for systems modeling and identi cation using big data techniques. Reliability Engineering & System Safety, 2022, 220(3):108219.

27. Granik M. and V. Mesyura, "Fake news detection using naive Bayes classifier," 2017 IEEE First Ukraine Conference on Electrical and Computer Engineering, pp. 900–903, 2017. https://doi.org/10.1109/UKRCON.2017.8100379.

28. Guess A., J. Nagler, and J. Tucker, "Less than you think: Prevalence and predictors of fake news dissemination on Facebook," Science Advances, vol. 5, no. 1, pp. 1–8, 2019. https://doi.org/10.1126/sciadv.aau4586.

29. Gwebu, K.L.; Wang, J.; Zifla, E. Can warnings curb the spread of fake news? The interplay between warning, trust and confirmation bias. Behav. Inf. Technol. 2022, 41, 3552–3573.

30. Haomin S, Hui D, Feng W, et al. A robust RFI identi cation for radio interferometry based on a convolutional neural network. Monthly Notices of the Royal Astronomical Society, 2022, 2(2):110.

31. Harris, S.; Hadi, H.J.; Ahmad, N.; Alshara, M.A. Fake News Detection Revisited: An Extensive Review of Theoretical Frameworks, Dataset Assessments, Model Constraints, and Forward-Looking Research Agendas. Technologies 2024, 12, 222. https://doi.org/10.3390/technologies12110222

32. Hu, B.; Sheng, Q.; Cao, J.; Shi, Y.; Li, Y.; Wang, D.; Qi, P. Bad actor, good advisor: Exploring the role of large language models in fake news detection. In Proceedings of the AAAI Conference on Artificial Intelligence, Vancouver, BC, Canada, 20–28 February 2024; Volume 38, pp. 22105–22113.

33. Huang D M, Ding Z H, Hu A D, et al. Low-cost adversarial covert false data injection attacks and their detection methods. Power Grid Technology, 2023, 47(4):9.

34. Islam M. R., Liu S., Wang X., and G. Xu, "Deep learning for misinformation detection on online social networks: A survey and new perspectives," Social Network Analysis and Mining, vol. 10, no. 1, pp. 1–20, 2020. https://doi.org/10.1007/s13278-020-00696-x.

35. Ivohin Ye., Adzhubey L., "On modeling the dynamics of information dissemination based on heterogeneous diffusion hybrid models," Scientific Bulletin of Uzhhorod University. Series: Mathematics and Computer Science, pp. 112–118, 2019. https://doi.org/10.24144/2616-7700.2019.2(35).112-118. (in Ukrainian)

36. Jain A. K.and Gupta B. B., "A machine learning based approach for phishing detection using hyperlinks information," Journal of Ambient Intelligence and Humanized Computing, vol. 10, no. 5, pp. 2015–2028, 2019. https://doi.org/10.1007/s12652-018-0798-z.

37. Jian, W.; Li, J.P.; Akbar, M.A.; Haq, A.U.; Khan, S.; Alotaibi, R.M.; Alajlan, S.A. SA-Bi-LSTM: Self Attention With Bi-Directional LSTM-Based Intelligent Model for Accurate Fake News Detection to Ensured Information Integrity on Social Media Platforms. IEEE Access 2024, 12, 48436–48452.

38. Jiang, B.; Tan, Z.; Nirmal, A.; Liu, H. Disinformation detection: An evolving challenge in the age of llms. In Proceedings of the 2024 SIAM International Conference on Data Mining (SDM), Society for Industrial and Applied Mathematics, Houston, TX, USA, 18–20 April 2024; pp. 427–435.

39. Karaboga D., "An idea based on honey bee swarm for numerical optimization," Technical report, Erciyes University, Engineering Faculty, Computer Engineering Department, Erciyes University, 2005, 10 p. [Online]. Available at: https://abc.erciyes.edu.tr/pub/tr06_2005.pdf.

40. Karaboga M. Akay B., and Karaboga D., "Artificial bee colony algorithm for optimization problems: A comprehensive review," Applied Soft Computing, vol. 122, 2022, pp. 108–125. doi: 10.1016/j.asoc.2022.108125.

41. Kareem, W.; Abbas, N. Fighting Lies with Intelligence: Using Large Language Models and Chain of Thoughts Technique to Combat Fake News. In Artificial Intelligence XL, SGAI, Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2023; pp. 253–258.

42. Karimi H. and Tang J., "Learning hierarchical discourse-level structure for fake news detection," Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics, pp. 3432–3442, 2019. https://doi.org/10.18653/v1/N19-1347.

43. Khan, T.; Michalas, A.; Akhunzada, A. Fake news outbreak 2021: Can we stop the viral spread? J. Netw. Comput. Appl. 2021, 190, 103112.

44. Kleminski R, Kazienko P, Kajdanowicz T. Analysis of direct citation, co-citation and bibliographic coupling in scienti c topic identi cation. Journal of Information Science, 2022, 48(3):349373.

45. Koru, G.K.; Uluyol, Ç. Detection of Turkish Fake News from Tweets with BERT Models. IEEE Access 2024, 12, 14918–14931.

46. Kumar A., D. Kumar, "A comprehensive review of artificial bee colony algorithm variants," Swarm and Evolutionary Computation, vol. 44, 2019, pp. 1–15.

47. Li B, Li H, Sun Q, et al. Evolutionary game analysis of the dissemination of false information by multiple parties after major emergencies. Complexity, 2022, 2022(12):114.

48. Li J., Wang Y., and H. Chen, "Enhanced artificial bee colony algorithm with adaptive parameter control for global optimization," IEEE Transactions on Cybernetics, vol. 52, no. 8, 2022, pp. 7896–7908. doi: 10.1109/TCYB.2021.3082345

49. Lian X, Qian T, Zhang Y, et al. Critical meter identi cation and network embed ding based attack detection for power systems against false data injection attacks. International Journal of Electrical Power and Energy Systems, 2022, 143(12)19.

50. Liao N, Wang J, Guan J, et al. A multi-step attack identi cation and correlation method based on multi-information fusion. Computers and Electrical Engineering, 2024, 117(6):116.

51. Liao N, Wang J, Guan J, et al. A multi-step attack identi cation and correlation method based on multi-information fusion. Computers and Electrical Engineering, 2024, 117(9)92499254.

52. Liu, Y.; Zhu, J.; Zhang, K.; Tang, H.; Zhang, Y.; Liu, X.; Liu, Q.; Chen, E. Detect, Investigate, Judge and Determine: A Novel LLM-Based Framework for Few-Shot Fake News Detection. arXiv 2024, arXiv:2407.08952.

53. Ma J. , Gao W., P. Mitra, S. Kwon, B. J. Jansen, K. Wong, and M. Cha, "Detecting rumors from microblogs with recurrent neural networks," Proceedings of the 25th International Joint Conference on Artificial Intelligence, pp. 3818–3824, 2016.

54. Mallick, C.; Mishra, S.; Senapati, M.R. A Cooperative Deep Learning Model for Fake News Detection in Online Social Networks. J. Ambient. Intell. Humaniz. Comput. 2023, 14, 4451–4460.

55. Mateńczuk K., Kozina A., Markowska A., Czerniachowska K., Kaczmarczyk K., P. Golec, M. Hernes, K. Lutosławski, A. Kozierkiewicz, M. Pietranik, A. Rot, M. Dyvak, "Financial Time Series Forecasting: Comparison of Traditional and Spiking Neural Networks," Procedia Computer Science, vol. 192, 2021, pp. 5023-5029, https://doi.org/10.1016/j.procs.2021.09.280.

56. Melnyk, A., Tymchyshyn, V., Pukas, A., Matiichuk, L., Shcherbiak, I., Yurchyshyn, T., Pan, T. Automatic Generation of Test Tasks Using ChatGPT API. CEUR-WS. 2025. Vol. 3974. P. 263-271.https://ceur-ws.org/Vol-3974/short09.pdf

57. Metzger M. J., Flanagin A. J., and R. B. Medders, "Social and heuristic approaches to credibility evaluation online," Journal of Communication, vol. 60, no. 3, pp. 413–439, 2010. https://doi.org/10.1111/j.1460-2466.2010.01488.x.

58. Mistriakov , V.V., and Pan Tiande. 2024. “Processing Content Query Requests for CSAF Documents Using a GrapHQL-BASED API”. Optoelectronic Information-Power Technologies 48 (2):152-61. https://doi.org/10.31649/1681-7893-2024-48-2-152-161.

59. Mouratidis, D.; Kanavos, A.; Kermanidis, K. From Misinformation to Insight: Machine Learning Strategies for Fake News Detection. Information 2025, 16, 189. https://doi.org/10.3390/info16030189

60. Muñoz, S.; Iglesias, C.Á. Exploiting Content Characteristics for Explainable Detection of Fake News. Big Data Cogn. Comput. 2024, 8, 129.

61. Niu B, Zhou S, Zhang H. Inuential node identi cation of network based on agglom eration operation. International Journal of Foundations of Computer Science, 2023, 10(12):110.

62. O. Ulichev, Y. Meleshko, V. Khokh, "The computer simulation method of a social network structure for the research of dissemination processes of informational influences," Scientific and Practical Cyber Security Journal (SPCSJ), 4(3). – Georgia, Tbilisi, 2019, pp. 34–47 (in Ukrainian)

63. Pan Tiande, Dudnyk Y.Yu., Hordiiuk V.Yu., Dankiv A.V., Kolodii A.O. A Method for Multimodal Profiling of Social Network Users. Computer information technologies: materials of the school-seminar of young scientists and students CIT'2024. Ternopil: WUNU, 2024. P. 95-96. https://dspace.wunu.edu.ua/bitstream/316497/52868/1/CIT%272024_Last.pdf

64. Pan Tiande, Zabchuk V.D., Sudeichenko D.V., Byts S.S., Samsonovych V.V. Mathematical and Software Tools for the Analysis and Processing of Large Data Volumes. Computer information technologies: materials of the school-seminar of young scientists and students CIT'2024. Ternopil: WUNU, 2024. P. 97-98. https://dspace.wunu.edu.ua/bitstream/316497/52868/1/CIT%272024_Last.pdf

65. Pan, Y.; Pan, L.; Chen, W.; Nakov, P.; Kan, M.Y.; Wang, W.Y. On the risk of misinformation pollution with large language models. arXiv 2023, arXiv:2305.13661.

66. Papageorgiou, E.; Chronis, C.; Varlamis, I.; Himeur, Y. A Survey on the Use of Large Language Models (LLMs) in Fake News. Future Internet 2024, 16, 298.

67. Park, S.; Han, S.; Cha, M. Adversarial Style Augmentation via Large Language Model for Robust Fake News Detection. arXiv 2024, arXiv:2406.11260.

68. Princy T. Modeling and analysis of trading volume and stock return data using bivariate q-gaussian distribution. Annals of Data Science, 2024, (prepublish):125.

69. Qubra, R.; Saputra, R.A. Classification of Hoax News Using the Naïve Bayes Method. Int. J. Softw. Eng. Comput. Sci. IJSECS 2024, 4, 40–48.

70. Raja, E.; Soni, B.; Borgohain, S.K. Fake News Detection in Dravidian Languages Using Transfer Learning with Adaptive Finetuning. Eng. Appl. Artif. Intell. 2023, 126, 106877.

71. Repede, Ș.E.; Brad, R. LLaMA 3 vs. State-of-the-Art Large Language Models: Performance in Detecting Nuanced Fake News. Computers 2024, 13, 292.

72. Roumeliotis, K.I.; Tselikas, N.D.; Nasiopoulos, D.K. Fake News Detection and Classification: A Comparative Study of Convolutional Neural Networks, Large Language Models, and Natural Language Processing Models. Future Internet 2025, 17, 28. https://doi.org/10.3390/fi17010028

73. Rybak, P. Transferring BERT Capabilities from High-Resource to Low-Resource Languages Using Vocabulary Matching. arXiv 2024, arXiv:2402.14408.

74. Sharma R., S. Kumar, and P. K. Singh, "Artificial bee colony algorithm for feature selection in machine learning: A systematic review," Expert Systems with Applications, vol. 204, 2022, pp. 117–135. doi: 10.1016/j.eswa.2022.117135

75. Shary S. P., "Interval methods for data fitting under uncertainty," Journal of Computational and Applied Mathematics, vol. 418, pp. 114–135, 2023. doi: 10.1016/j.cam.2022.114135.

76. Shu K., Mahudeswaran D., S. Wang, D. Lee, and H. Liu, "FakeNewsNet: A data repository with news content, social context, and spatiotemporal information for studying fake news on social media," Big Data, vol. 8, no. 3, pp. 171–188, 2020. https://doi.org/10.1089/big.2020.0062.

77. Shu K., Mahudeswaran D., Wang S., Lee D., and Liu H., "Hierarchical propagation networks for fake news detection: Investigation and exploitation," Proceedings of the International AAAI Conference on Web and Social Media, vol. 14, pp. 626–637, 2020. https://doi.org/10.1609/icwsm.v14i1.7329.

78. Shu K., Sliva A., Wang S., Tang J., and H. Liu, "Fake news detection on social media: A data mining perspective," ACM SIGKDD Explorations Newsletter, vol. 19, no. 1, pp. 22–36, 2017. https://doi.org/10.1145/3137597.3137600.

79. Shu K., Wang S., and Liu H., "Beyond news contents: The role of social context for fake news detection," Proceedings of the 12th ACM International Conference on Web Search and Data Mining, pp. 312–320, 2019. https://doi.org/10.1145/3289600.3290994.

80. Shu, K.; Mahudeswaran, D.; Wang, S.; Lee, D.; Liu, H. Fakenewsnet: A data repository with news content, social context, and spatiotemporal information for studying fake news on social media. Big Data 2020, 8, 171–188.

81. Shushkevich, E.; Alexandrov, M.; Cardiff, J. Improving Multiclass Classification of Fake News Using BERT-Based Models and ChatGPT-Augmented Data. Inventions 2023, 8, 112.

82. Shushkevich, E.; Alexandrov, M.; Cardiff, J. Improving Multiclass Classification of Fake News Using BERT-Based Models and ChatGPT-Augmented Data. Inventions 2023, 8, 112.

83. Singh, I.; Dhanda, N.; Sahai, A.; Gupta, K.K. Comparative Study of Random Forest Algorithm and Logistic Regression in the Analysis of Fake News. In Proceedings of the 8th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 1–3 June 2023; pp. 1477–1482

84. Singhal, S.; Shah, R.R.; Chakraborty, T.; Kumaraguru, P.; Satoh, S. SpotFake: A Multi-Modal Framework for Fake News Detection. In Proceedings of the 2019 IEEE 5th International Conference on Multimedia Big Data, BigMM 2019, Singapore, 11–13 September 2019; pp. 39–47.

85. Su, J.; Cardie, C.; Nakov, P. Adapting fake news detection to the era of large language models. arXiv 2023, arXiv:2311.04917.

86. Sun, Y.; He, J.; Cui, L.; Lei, S.; Lu, C.T. Exploring the Deceptive Power of LLM-Generated Fake News: A Study of Real-World Detection Challenges. arXiv 2024, arXiv:2403.18249.

87. Teo, T.W.; Chua, H.N.; Jasser, M.B.; Wong, R.T.K. Integrating Large Language Models and Machine Learning for Fake News Detection. In Proceedings of the 2024 20th IEEE International Colloquium on Signal Processing and Its Applications, CSPA 2024—Conference Proceedings, Langkawi, Malaysia, 1–2 March 2024; pp. 102–107.

88. Tiande Pan. 2025. "Research on Identification Methods for False or Unrelated Information in Network Resource Content" International Journal of High Speed Electronics and SystemsVol. 34, No. 04, 2540203. https://doi.org/10.1142/S0129156425402037

89. Tortora C, Palumbo F. Clustering mixed-type data using a probabilistic distance algorithm. Applied Soft Computing, 2022, 130(11):109704109713.

90. Ulichev O. , Meleshko Ye. , D. Sawicki, S. Smailova, "Computer modeling of dissemination of informational influences in social networks with different strategies

of information distributors," Proc. SPIE 11176, Wilga, Poland, 2019, Article No.: 111761T. https://doi.org/10.1117/12.2536480. (in Ukrainian)

91. Ulichev O.S. , "Research on models of information dissemination and informational influence in social networks," Systems of Control, Navigation and Communication, issue 4, pp. 147–151, 2018. [Online]. Available at: http://nbuv.gov.ua/UJRN/suntz_2018_4_31. (in Ukrainian)

92. Ulichev O.S., Meleshko Ye.V., "Modeling of dissemination and neutralization processes of informational influences in a segment of a social network," Scientific Journal 'Information Protection'. – Kyiv: NAU, 2020, pp. 166–176 (in Ukrainian)

93. Ulichev O.S., Ye.V. Meleshko, "Software modeling of the dissemination of informational and psychological influences in virtual social networks," Collection of Scientific Papers 'Modern Information Systems', Issue 2(2). – Kharkiv: NTU KhPI, 2018, pp. 35–39. https://doi.org/10.20998/2522-9052.2018.2.06 (in Ukrainian)

94. Verma, P.K.; Agrawal, P.; Amorim, I.; Prodan, R. WELFake: Word Embedding Over Linguistic Features for Fake News Detection. IEEE Trans. Comput. Soc. Syst. 2021, 8, 881–893.

95. Vincent E, Korki M, Seyedmahmoudian M, et al. Detection of false data injection attacks in cyber physical systems using graph convolutional network. Electric Power Systems Research, 2023, 217(4):18.

96. Vosoughi S. , D. Roy, and S. Aral, "The spread of true and false news online," Science, vol. 359, no. 6380, pp. 1146–1151, 2018. https://doi.org/10.1126/science.aap9559.

97. Wang Y, Zhang H, Wei Y, et al. An evolutionary computation-based machine learning for network attack detection in big data tra c. Applied Soft Computing, 2023, 138(2):110184110196.

98. Wang Y. , Ma F., Z. Jin, Y. Yuan, G. Xun, L. Jiao, and A. Su, "EANN: Event adversarial neural networks for multi-modal fake news detection," Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 849–857, 2018. https://doi.org/10.1145/3219819.3219903.

99. Wei F L, Wang K. Simulation of deep recognition for false data injection attacks in network laboratory. Computer Simulation, 2023, 40(9):406410.

100. Wu Z Q, Cao H. A method for generating Chinese false comment datasets based on GAN. Journal of Yunnan University: Natural Science Edition, 2023, 45(5):10331042.

101. Wundari, F.; Amien, M.N.A.S.; Irtsa, D.H. Identifying Fake News Using Long-Short Term Memory Model. J. Dinda Data Sci. Inf. Technol. Data Anal. 2024, 4, 28–34.

102. Xia Y H, Wang Y, Zhou L, et al. False data injection attack detection method based on improved generative adversarial networks. Electric Power Construction, 2022, 43(3):5865.

103. XuW,HigginsM,WangJ, et al. Blending data and physics against false data injection attack: An event-triggered moving target defence approach. IEEE Transactions on Smart Grid, 2023, 14(4):3176 3188.

104. Yan, Y.; Fu, H.; Wu, F. Multimodal Social Media Fake News Detection Based on 1D-CCNet Attention Mechanism. Electronics 2024, 13, 3700. https://doi.org/10.3390/electronics13183700

105. Yousheng Z, Zhonghan W,Yuanni L. Alocal di erential privacy hybrid data clustering iterative algorithm for edge computing. Chinese Journal of Electronics, 2022, 33(6):1 14.

106. YuK,SuTR.Modelingandsimulation research on point-to-point propagation of false information based on information risk perception theory. Computer Science, 2023, 50(7):376385.

107. Zhang J., Dong B., and S. Y. Philip, "FakeDetector: Effective fake news detection with deep diffusive neural network," 2020 IEEE 36th International Conference on Data Engineering, pp. 1826–1829, 2020. https://doi.org/10.1109/ICDE48307.2020.00180.

108. Zhang L., M. Wang, and X. Liu, "Multi-objective artificial bee colony algorithm for interval optimization problems," Information Sciences, vol. 625, 2023, pp. 1–18. https://doi.org/10.1016/j.ins.2023.01.045.

109. Zhao D K, Zhao Q S, Liang D K, et al. A dynamic phasor estimation method consid ering false data attack detection. Journal of Taiyuan University of Technology, 2022, 53(2):274280.

110. Zhou L., Zhang D., and Lee C. C., "A survey of opinion mining and sentiment analysis," Mining Text Data, pp. 415–463, 2012. https://doi.org/10.1007/978-1-4614-3223-4_13.

111. Zhou, X.; Zafarani, R. Network-Based Fake News Detection: A pattern-driven approach. ACM SIGKDD Explor. Newsl. 2019, 21, 48–60.

112. Zubiaga M., Aker A., Bontcheva K., M. Liakata, and R. Procter, "Detection and resolution of rumours in social media: A survey," ACM Computing Surveys, vol. 51, no. 2, pp. 1–36, 2018. https://doi.org/10.1145/3161603.

APPENDIX A

CODE LISTING OF THE INFORMATION STORAGE SUBSYSTEM IMPLEMENTATION IN MONGODB

```
{
 // ================================================
 //  Collection: posts
 //  Purpose: Core storage for social media content with credibility analysis results.
 //  Each document represents a single post enriched with analytical metrics.
 // ================================================
 "posts": [
   {ф
    "_id": { "$oid": "650000000000000000000001" },
    "post_id": "fb_1001",                  // Unique identifier of the post in the social network
    "network": "facebook",                 // Source network
    "author_id": "user_101",               // Author’s identifier
    "text": "The government announced the launch of Program X.", // Post text
    "created_at": { "$date": "2025-10-20T08:12:00Z" },
    "metrics": {
     "likes": 120,
     "comments": 15,
     "shares": 34,
     "unique_users_12h": 98                // Number of unique users who interacted within 12 hours
    },
    "criteria": {
     "TR": 0.85,                           // Trust Rate (author reliability)
     "C": 0.9,                             // Consistency of claims
     "N": 0.6,                             // Novelty (newness of information)
     "EM": 0.12,                           // Emotional tone score
     "T": 0.95                             // Timeliness (recency of post)
    },
    "CI": 0.82,                            // Composite credibility index (0–1)
    "verdict": "credible",                 // Result of evaluation
    "evidence": [
     { "fact_id": "fact_201", "match_score": 0.92 } // Reference to verified fact(s)
    ],
    "schema_version": 1
   }
 ],
```

```
 // ================================================
 //  Collection: user_profiles
 //  Purpose: Interval-based behavioral user profiles with activity metrics.
 // ================================================
 "user_profiles": [
   {
    "_id": { "$oid": "650000000000000000000011" },
    "user_id": "user_101",
    "trust_interval": { "lo": 0.78, "hi": 0.88 },   // Interval of user trustworthiness
    "emotion_interval": { "lo": 0.05, "hi": 0.15 }, // Emotional stability range
    "stability_interval": { "lo": 0.7, "hi": 0.9 }, // Posting stability interval
    "activity_series": [                        // Time series of posting frequency
     { "date": "2025-10-18", "posts": 2 },
     { "date": "2025-10-19", "posts": 1 },
     { "date": "2025-10-20", "posts": 3 }
    ],
    "anomalies": [],                            // Detected anomalies in behavior
    "updated_at": { "$date": "2025-10-24T10:00:00Z" },
    "schema_version": 1
   }
 ],

 // ================================================
 //  Collection: facts
 //  Purpose: Repository of verified claims from trusted fact-checking sources.
 // ================================================
 "facts": [
   {
    "_id": { "$oid": "650000000000000000000021" },
    "fact_id": "fact_201",
    "claim_norm": "the government launched program x", // Normalized claim
text
    "verdict": "true",                          // Fact-check verdict (true/false/mixed)
    "source": "OfficialGovPortal",                  // Trusted source name
    "url": "https://gov.example/program-x",           // Link to the verification
article
    "reviewed_at": { "$date": "2025-10-19T12:00:00Z" },
    "entities": ["Government"],                   // Related named entities
    "topics": ["policy", "economy"],                // Topical categories
    "schema_version": 1
   }
 ],
```

```
// ================================================
//  Collection: config
//  Purpose: Stores configuration parameters, weights, and thresholds for CIEngine.
// ================================================
"config": [
  {
    "_id": { "$oid": "650000000000000000000031" },
    "version": 3,                    // Configuration version
    "weights": {
      "wTR": 0.15,          // Weight for Trust factor
      "wC": 0.15,           // Weight for Consistency
      "wN": 0.10,           // Weight for Novelty
      "wInvEM": 0.10,       // Weight for inverted Emotionality
      "wT": 0.10,           // Weight for Timeliness
      "wOther": 0.40        // Combined weight of auxiliary metrics
    },
    "ci_thresholds": {
      "credible": 0.75,
      "needs_review": 0.45,
      "suspicious": 0.0
    },
    "active": true,                  // Indicates current active configuration
    "author": "admin",
    "comment": "weights v3, tuned Oct 2025",
    "updated_at": { "$date": "2025-10-24T09:00:00Z" },
    "schema_version": 1
  }
],

// ================================================
//  Collection: logs
//  Purpose: System telemetry and operational audit of the data pipeline.
// ================================================
"logs": [
  {
    "_id": { "$oid": "650000000000000000000041" },
    "ts": { "$date": "2025-10-24T10:00:05Z" },
    "level": "INFO",                 // Log level (INFO, WARN, ERROR)
    "component": "Collector",        // Component name generating the event
    "event": "fetched_posts",        // Event identifier
    "details": { "network": "facebook", "count": 120 }, // Event-specific payload
    "trace_id": "trace_0001"         // Correlation ID for tracing
```

```
    },
    {
      "_id": { "$oid": "650000000000000000000042" },
      "ts": { "$date": "2025-10-24T10:00:20Z" },
      "level": "INFO",
      "component": "CIEngine",
      "event": "evaluated_post",
      "details": { "post_id": "fb_1001", "CI": 0.82 },
      "trace_id": "trace_0002"
    }
  ]
}
```

APPENDIX B

LISTING OF THE CORE SYSTEM MODULES

```
app.py — Unified application file for the Fake Content Detection & Analysis System

Modules merged:
- SocialMediaConnector / ETLManager (collector & normalization)
- CIEngine (credibility calculation)
- IntervalAnalyzer (interval user modeling)
- Visualization helpers (time series/aggregates)
- FastAPI REST API (dashboard & control plane)

Author: Demo Integration
Python: 3.10+
"""

from __future__ import annotations

import os
import math
import json
from datetime import datetime, timedelta
from typing import Any, Dict, List, Optional

import requests  # optional; used only if real Graph API is configured
from fastapi import FastAPI, HTTPException, Query, Body
from fastapi.responses import JSONResponse
from pydantic import BaseModel, Field
from pymongo import MongoClient, ASCENDING, DESCENDING
from pymongo.errors import PyMongoError


# ---------------------------------------------------------------------------
# Configuration & DB bootstrap
# ---------------------------------------------------------------------------

MONGO_URI = os.getenv("MONGO_URI", "mongodb://localhost:27017/")
DB_NAME = os.getenv("DB_NAME", "fakecheckdb")
FB_TOKEN = os.getenv("FACEBOOK_TOKEN", "")  # optional demo

client = MongoClient(MONGO_URI)
db = client[DB_NAME]


def create_indexes() -> None:
    """Create recommended indexes if absent (idempotent)."""
    # posts
    db.posts.create_index([("created_at", DESCENDING)])
    db.posts.create_index([("author_id", ASCENDING), ("created_at", DESCENDING)])
    db.posts.create_index([("CI", ASCENDING), ("created_at", DESCENDING)],
                          partialFilterExpression={"CI": {"$exists": True}})
    db.posts.create_index([("network", ASCENDING), ("created_at", ASCENDING)])

    # user_profiles
    db.user_profiles.create_index([("user_id", ASCENDING)], unique=True)
    db.user_profiles.create_index([("updated_at", DESCENDING)])
    db.user_profiles.create_index([("trust_interval.hi", DESCENDING)],
                                  partialFilterExpression={"trust_interval.hi":
```

```
{"$exists": True}})

    # facts
    try:
        db.facts.create_index([("claim_norm", "text")])
    except Exception:
        pass
    db.facts.create_index([("reviewed_at", DESCENDING)])
    db.facts.create_index([("verdict", ASCENDING), ("topics", ASCENDING)])

    # config
    db.config.create_index([("active",   ASCENDING)],   partialFilterExpression={"active":
True})
    db.config.create_index([("version", DESCENDING)])

    # logs
    db.logs.create_index([("ts", DESCENDING)])
    db.logs.create_index([("level", ASCENDING), ("ts", DESCENDING)])
    db.logs.create_index([("trace_id", ASCENDING)])


create_indexes()


# ---------------------------------------------------------------------------
# Data models (Pydantic DTOs)
# ---------------------------------------------------------------------------

class Criteria(BaseModel):
    TR: float = 0.0  # Trust
    C: float = 0.0   # Consistency
    N: float = 0.0   # Network (novelty/proxy)
    EM: float = 0.0  # Emotionality
    T: float = 0.0   # Timeliness


class Weights(BaseModel):
    wTR: float = 0.15
    wC: float = 0.15
    wN: float = 0.10
    wInvEM: float = 0.10
    wT: float = 0.10
    # optional “bucket” of auxiliary metrics used by extended configs
    wOther: float = 0.40


class UpsertConfig(BaseModel):
    version: int = Field(..., ge=1)
    weights: Weights
    ci_thresholds: Dict[str, float] = {"credible": 0.75, "needs_review": 0.45, "suspicious":
0.0}
    active: bool = True
    author: str = "admin"
    comment: str = "weights update"


# ---------------------------------------------------------------------------
# Collector & ETL
# ---------------------------------------------------------------------------

class SocialMediaConnector:
```

```
    """Minimal Facebook Graph API connector (demo)."""

    def __init__(self, token: str):
        self.token = token
        self.base_url = "https://graph.facebook.com/v18.0"

    def get_posts(self, page_id: str, limit: int = 10) -> List[Dict[str, Any]]:
        """Fetch recent posts via Graph API (returns empty list if token missing)."""
        if not self.token:
            return []
        url = f"{self.base_url}/{page_id}/posts"
        params = {"access_token": self.token, "limit": limit}
        try:
            resp = requests.get(url, params=params, timeout=20)
            if resp.status_code == 200:
                return resp.json().get("data", [])
        except Exception:
            pass
        return []


class ETLManager:
    """Normalize and store posts."""

    def __init__(self, db):
        self.db = db

    @staticmethod
    def _parse_created(dt: str) -> datetime:
        # Facebook created_time example: "2025-10-20T08:12:00+0000"
        try:
            if "+" in dt and dt.endswith("0000"):
                return datetime.strptime(dt, "%Y-%m-%dT%H:%M:%S+0000")
            return datetime.fromisoformat(dt.replace("Z", "+00:00"))
        except Exception:
            return datetime.utcnow()

    def normalize_fb_post(self, raw: Dict[str, Any]) -> Dict[str, Any]:
        post_id = raw.get("id") or f"fb_{int(datetime.utcnow().timestamp())}"
        created_time = raw.get("created_time") or datetime.utcnow().isoformat()
        return {
            "post_id": post_id,
            "network": "facebook",
            "author_id": raw.get("from", {}).get("id", "unknown"),
            "text": raw.get("message", ""),
            "created_at": self._parse_created(created_time),
            "metrics": {"likes": 0, "comments": 0, "shares": 0, "unique_users_12h": 0},
            "criteria": {},  # to be filled by analyzer pipeline
            "CI": None,
            "verdict": "pending",
            "schema_version": 1
        }

    def save_post(self, doc: Dict[str, Any]) -> None:
        self.db.posts.update_one({"post_id": doc["post_id"]}, {"$set": doc}, upsert=True)


# ---------------------------------------------------------------------------
# Credibility Engine
# ---------------------------------------------------------------------------
```

```
class CIEngine:
    """Composite credibility index calculation."""

    def __init__(self, db):
        self.db = db

    def active_weights(self) -> Weights:
        cfg = self.db.config.find_one({"active": True}, sort=[("version", DESCENDING)])
        if cfg and "weights" in cfg:
            return Weights(**cfg["weights"])
        return Weights()

    @staticmethod
    def _clip01(x: float) -> float:
        return max(0.0, min(1.0, float(x)))

    def evaluate(self, criteria: Dict[str, float]) -> float:
        """Compute CI as weighted sum with inverted EM (1-EM)."""
        w = self.active_weights()
        tr = self._clip01(criteria.get("TR", 0.0))
        c_ = self._clip01(criteria.get("C", 0.0))
        n_ = self._clip01(criteria.get("N", 0.0))
        em = self._clip01(criteria.get("EM", 0.0))
        t_ = self._clip01(criteria.get("T", 0.0))
        ci = (w.wTR * tr) + (w.wC * c_) + (w.wN * n_) + (w.wInvEM * (1 - em)) + (w.wT * t_)
        return round(ci, 3)

    def verdict(self, ci: float) -> str:
        cfg = self.db.config.find_one({"active": True}, sort=[("version", DESCENDING)])
        thr = (cfg or {}).get("ci_thresholds", {"credible": 0.75, "needs_review": 0.45})
        if ci >= thr.get("credible", 0.75):
            return "credible"
        if ci >= thr.get("needs_review", 0.45):
            return "needs_review"
        return "suspicious"


# ---------------------------------------------------------------------------
# Interval User Modeling
# ---------------------------------------------------------------------------

class IntervalAnalyzer:
    """Builds interval-based profiles from a user's posting history."""

    def __init__(self, db):
        self.db = db

    @staticmethod
    def _pctl(values: List[float], p: float) -> float:
        if not values:
            return 0.0
        values = sorted(values)
        k = (len(values) - 1) * p
        f = math.floor(k)
        c = math.ceil(k)
        if f == c:
            return values[int(k)]
        d0 = values[f] * (c - k)
        d1 = values[c] * (k - f)
        return d0 + d1
```

```
    def update_profile(self, user_id: str) -> Optional[Dict[str, Any]]:
        posts    =    list(self.db.posts.find({"author_id":    user_id}).sort("created_at",
DESCENDING))
        if len(posts) < 2:
            profile = {
                "user_id": user_id,
                "trust_interval": {"lo": 0.0, "hi": 0.0},
                "emotion_interval": {"lo": 0.0, "hi": 0.0},
                "stability_interval": {"lo": 0.0, "hi": 0.0},
                "activity_series": [],
                "anomalies": [{"type": "insufficient_history", "min_posts": 2}],
                "updated_at": datetime.utcnow(),
                "schema_version": 1
            }
            self.db.user_profiles.update_one({"user_id":   user_id},   {"$set":   profile},
upsert=True)
            return profile

        # hour gaps between consecutive posts
        gaps_h = [
            (posts[i]["created_at"] - posts[i + 1]["created_at"]).total_seconds() / 3600.0
            for i in range(len(posts) - 1)
        ]

        # naive “trust” proxy from gaps (more regular gaps -> higher trust interval)
        lo_gap = self._pctl(gaps_h, 0.25)
        hi_gap = self._pctl(gaps_h, 0.75)
        # map to [0..1] by dividing by a horizon (e.g., 48h) and clipping
        def norm_gap(x): return max(0.0, min(1.0, 1.0 - (x / 48.0)))
        trust_interval = {"lo": round(norm_gap(hi_gap), 2), "hi": round(norm_gap(lo_gap),
2)}

        # emotion interval from posts criteria.EM if present
        em_values = [p.get("criteria", {}).get("EM", 0.0) for p in posts if "criteria" in
p]
        if not em_values:
            em_values = [0.0]
        em_lo = round(self._pctl(em_values, 0.25), 2)
        em_hi = round(self._pctl(em_values, 0.75), 2)

        # stability proxy — inverse of interquartile range of gaps
        iqr = max(0.0, hi_gap - lo_gap)
        stability = max(0.0, min(1.0, 1.0 - (iqr / 72.0)))
        stability_interval = {"lo": round(max(0.0, stability - 0.1), 2),
                              "hi": round(min(1.0, stability + 0.1), 2)}

        # activity series (last 14 days)
        start = datetime.utcnow() - timedelta(days=14)
        pipeline = [
            {"$match": {"author_id": user_id, "created_at": {"$gte": start}}},
            {"$group":   {"_id":   {"$dateToString":   {"format":   "%Y-%m-%d",   "date":
"$created_at"}},
                          "posts": {"$sum": 1}}},
            {"$sort": {"_id": 1}}
        ]
        series    =    [{"date":    d["_id"],    "posts":    d["posts"]}    for    d    in
self.db.posts.aggregate(pipeline)]

        profile = {
            "user_id": user_id,
            "trust_interval": trust_interval,
```

```
            "emotion_interval": {"lo": em_lo, "hi": em_hi},
            "stability_interval": stability_interval,
            "activity_series": series,
            "anomalies": [],
            "updated_at": datetime.utcnow(),
            "schema_version": 1
        }
        self.db.user_profiles.update_one({"user_id":    user_id},    {"$set":    profile},
upsert=True)
        return profile


# ---------------------------------------------------------------------------
# Visualization helpers (aggregates for charts)
# ---------------------------------------------------------------------------

def ci_trend(db, days: int = 7) -> List[Dict[str, Any]]:
    since = datetime.utcnow() - timedelta(days=days)
    pipeline = [
        {"$match": {"created_at": {"$gte": since}, "CI": {"$ne": None}}},
        {"$group": {
            "_id": {"$dateToString": {"format": "%Y-%m-%d", "date": "$created_at"}},
            "avg_CI": {"$avg": "$CI"},
            "count": {"$sum": 1}
        }},
        {"$sort": {"_id": 1}}
    ]
    return list(db.posts.aggregate(pipeline))


def top_authors(db, limit: int = 5) -> List[Dict[str, Any]]:
    pipeline = [
        {"$group": {"_id": "$author_id", "posts": {"$sum": 1}, "avg_CI": {"$avg": "$CI"}}},
        {"$sort": {"posts": -1}},
        {"$limit": int(limit)}
    ]
    return list(db.posts.aggregate(pipeline))


# ---------------------------------------------------------------------------
# FastAPI application
# ---------------------------------------------------------------------------

app = FastAPI(title="Fake Content Detection API", version="1.0.0")


@app.get("/health")
def health():
    try:
        db.command("ping")
        return {"status": "ok", "db": DB_NAME, "time": datetime.utcnow().isoformat()}
    except PyMongoError as e:
        raise HTTPException(status_code=500, detail=str(e))


# --- Config management ------------------------------------------------------

@app.get("/api/config/active")
def get_active_config():
    cfg = db.config.find_one({"active": True}, sort=[("version", DESCENDING)]) or {}
    if cfg.get("_id"):
```

```
        cfg["_id"] = str(cfg["_id"])
    return cfg


@app.post("/api/config/upsert")
def upsert_config(cfg: UpsertConfig):
    db.config.update_many({}, {"$set": {"active": False}})
    doc = cfg.dict()
    doc["updated_at"] = datetime.utcnow()
    db.config.insert_one(doc)
    return {"ok": True, "version": cfg.version}


# --- CI computation -------------------------------------------------------

ci_engine = CIEngine(db)
interval_analyzer = IntervalAnalyzer(db)


@app.get("/api/ci/{post_id}")
def compute_ci(post_id: str):
    post = db.posts.find_one({"post_id": post_id})
    if not post:
        raise HTTPException(status_code=404, detail="Post not found")
    criteria = post.get("criteria") or {}
    ci = ci_engine.evaluate(criteria)
    verdict = ci_engine.verdict(ci)
    db.posts.update_one({"post_id": post_id}, {"$set": {"CI": ci, "verdict": verdict}})
    return {"post_id": post_id, "CI": ci, "verdict": verdict}


class CriteriaPayload(BaseModel):
    criteria: Criteria


@app.post("/api/ci/{post_id}/criteria")
def set_criteria_and_compute(post_id: str, payload: CriteriaPayload):
    db.posts.update_one({"post_id":         post_id},         {"$set":         {"criteria":
payload.criteria.dict()}}, upsert=True)
    ci = ci_engine.evaluate(payload.criteria.dict())
    verdict = ci_engine.verdict(ci)
    db.posts.update_one({"post_id": post_id}, {"$set": {"CI": ci, "verdict": verdict}})
    return    {"post_id":    post_id,    "CI":    ci,    "verdict":    verdict,    "criteria":
payload.criteria.dict()}


# --- Interval user modeling -----------------------------------------------

@app.post("/api/user/{user_id}/intervals")
def update_user_intervals(user_id: str):
    profile = interval_analyzer.update_profile(user_id)
    return {"user_id": user_id, "profile": profile}


# --- Aggregates for dashboard ---------------------------------------------

@app.get("/api/trend/ci")
def api_ci_trend(days: int = Query(7, ge=1, le=90)):
    return {"days": days, "series": ci_trend(db, days)}
```

```
@app.get("/api/top-authors")
def api_top_authors(limit: int = Query(5, ge=1, le=50)):
    return {"limit": limit, "authors": top_authors(db, limit)}


# --- Demo seeding & simple CRUD -----------------------------------------------

@app.post("/api/seed/demo")
def seed_demo():
    """Insert minimal demo dataset (safe to call multiple times)."""
    posts = [
        {
            "post_id": "fb_1001",
            "network": "facebook",
            "author_id": "user_101",
            "text": "The government announced the launch of Program X.",
            "created_at": datetime(2025, 10, 20, 8, 12),
            "metrics": {"likes": 120, "comments": 15, "shares": 34, "unique_users_12h": 98},
            "criteria": {"TR": 0.85, "C": 0.9, "N": 0.6, "EM": 0.12, "T": 0.95},
            "CI": 0.82,
            "verdict": "credible",
            "evidence": [{"fact_id": "fact_201", "match_score": 0.92}],
            "schema_version": 1
        },
        {
            "post_id": "fb_1002",
            "network": "facebook",
            "author_id": "user_202",
            "text": "Shock! Doctors hid the effectiveness of Drug Y.",
            "created_at": datetime(2025, 10, 21, 14, 5),
            "metrics": {"likes": 28, "comments": 210, "shares": 430, "unique_users_12h":
2353},
            "criteria": {"TR": 0.32, "C": 0.25, "N": 0.9, "EM": 0.88, "T": 0.6},
            "CI": 0.38,
            "verdict": "suspicious",
            "evidence": [{"fact_id": None, "notes": "no exact match, high emotion and viral
spread"}],
            "schema_version": 1
        },
        {
            "post_id": "tg_3001",
            "network": "telegram",
            "author_id": "user_303",
            "text": "Schedule of events for the upcoming week.",
            "created_at": datetime(2025, 10, 24, 9, 0),
            "metrics": {"likes": 5, "comments": 0, "shares": 1, "unique_users_12h": 10},
            "criteria": {"TR": 0.7, "C": 0.8, "N": 0.1, "EM": 0.05, "T": 0.9},
            "CI": 0.74,
            "verdict": "credible",
            "evidence": [],
            "schema_version": 1
        }
    ]
    for p in posts:
        db.posts.update_one({"post_id": p["post_id"]}, {"$set": p}, upsert=True)

    facts = [
        {"fact_id": "fact_201", "claim_norm": "the government launched program x",
         "verdict": "true", "source": "OfficialGovPortal",
         "url": "https://gov.example/program-x", "reviewed_at": datetime(2025, 10, 19, 12,
0),
```

```
         "entities": ["Government"], "topics": ["policy", "economy"], "schema_version": 1},
        {"fact_id": "fact_202", "claim_norm": "doctors hid the effectiveness of drug y",
         "verdict": "false", "source": "HealthCheckOrg",
         "url": "https://healthcheck.example/drug-y", "reviewed_at": datetime(2025, 9, 5,
9, 0),
         "entities": ["Healthcare"], "topics": ["health"], "schema_version": 1}
    ]
    for f in facts:
        db.facts.update_one({"fact_id": f["fact_id"]}, {"$set": f}, upsert=True)

    config = {
        "version": 3,
        "weights": Weights().dict(),
        "ci_thresholds": {"credible": 0.75, "needs_review": 0.45, "suspicious": 0.0},
        "active": True,
        "author": "admin",
        "comment": "weights v3, tuned Oct 2025",
        "updated_at": datetime.utcnow(),
        "schema_version": 1
    }
    db.config.update_many({}, {"$set": {"active": False}})
    db.config.insert_one(config)

    db.logs.insert_one({"ts": datetime.utcnow(), "level": "INFO", "component": "Seeder",
                        "event": "demo_seeded", "details": {"posts": len(posts), "facts":
len(facts)},
                        "trace_id": f"seed_{int(datetime.utcnow().timestamp())}"})

    return {"ok": True, "seeded": {"posts": len(posts), "facts": len(facts)},
"active_config_version": 3}


@app.get("/api/posts")
def list_posts(author_id: Optional[str] = None, limit: int = Query(20, ge=1, le=200)):
    q: Dict[str, Any] = {}
    if author_id:
        q["author_id"] = author_id
    cursor = db.posts.find(q).sort("created_at", DESCENDING).limit(limit)
    items = []
    for doc in cursor:
        doc["_id"] = str(doc["_id"])
        items.append(doc)
    return {"count": len(items), "items": items}


# --- Optional: demo ingestion from Facebook Graph API (no-op without token) ---

class FBIngestPayload(BaseModel):
    page_id: str
    limit: int = Field(10, ge=1, le=100)


@app.post("/api/ingest/facebook")
def ingest_facebook(payload: FBIngestPayload):
    connector = SocialMediaConnector(FB_TOKEN)
    etl = ETLManager(db)
    raw_posts = connector.get_posts(payload.page_id, payload.limit)
    if not raw_posts:
        return JSONResponse(
            status_code=200,
            content={"ok": True, "ingested": 0, "note": "No token or no data; nothing
```

```
ingested."}
        )
    for rp in raw_posts:
        doc = etl.normalize_fb_post(rp)
        etl.save_post(doc)
    return {"ok": True, "ingested": len(raw_posts)}


# ---------------------------------------------------------------------------
# Entrypoint (when running as a script): uvicorn app:app --reload
# ---------------------------------------------------------------------------

if __name__ == "__main__":
    import uvicorn
    print(f"[INFO] Starting API at http://127.0.0.1:8000  (DB={DB_NAME})")
    uvicorn.run(app, host="0.0.0.0", port=8000)
```

APPENDIX C

LIST OF PUBLISHED PAPERS BY THE TOPIC OF THESIS

1. Dyvak, M., Yushko, A., Melnyk, A., Pan, T. An Intelligent Information System for Generating a Scientist's Scientometrics Using Content Analysis Methods. CEUR-WS. 2024. Vol. 3942. P. 66-82.

https://ceur-ws.org/Vol-3942/S_06_Dyvak.pdf

2. Dyvak, Mykola, Tyande Pan, and Oleksandr Kindzerskyi. 2025. “Mathematical Model of a Social Network User Profile Based on Interval Data Analysis”. International Journal of Computing 24 (3):452-59.

https://www.computingonline.net/computing/article/view/4182.

3. Dyvak, Mykola, Volodymyr Manzhula, Andriy Melnyk, Nataliia Petryshyn, Tiande Pan, Arkadiusz Banasik, Piotr Pikiewicz, and Wojciech M. Kempa. 2025. "Modeling the Electricity Generation Processes of a Combined Solar and Small Hydropower Plant" Energies 18, no. 9: 2351.

https://doi.org/10.3390/en18092351

4. Melnyk, A., Tymchyshyn, V., Pukas, A., Matiichuk, L., Shcherbiak, I., Yurchyshyn, T., Pan, T. Automatic Generation of Test Tasks Using ChatGPT API. CEUR-WS. 2025. Vol. 3974. P. 263-271.

https://ceur-ws.org/Vol-3974/short09.pdf

5. Mistriakov , V.V., and Pan Tiande. 2024. “Processing Content Query Requests for CSAF Documents Using a GrapHQL-BASED API”. Optoelectronic Information-Power Technologies 48 (2):152-61.

https://doi.org/10.31649/1681-7893-2024-48-2-152-161.

6. Tiande Pan. 2025. "Research on Identification Methods for False or Unrelated Information in Network Resource Content" International Journal of High Speed Electronics and SystemsVol. 34, No. 04, 2540203.

https://doi.org/10.1142/S0129156425402037

7. Pan Tiande, Dudnyk Y.Yu., Hordiiuk V.Yu., Dankiv A.V., Kolodii A.O. A Method for Multimodal Profiling of Social Network Users. Computer information technologies: materials of the school-seminar of young scientists and students CIT'2024. Ternopil: WUNU, 2024. P. 95-96.

https://dspace.wunu.edu.ua/bitstream/316497/52868/1/CIT%272024_Last.pdf

8. Pan Tiande, Zabchuk V.D., Sudeichenko D.V., Byts S.S., Samsonovych V.V. Mathematical and Software Tools for the Analysis and Processing of Large Data Volumes. Computer information technologies: materials of the school-seminar of young scientists and students CIT'2024. Ternopil: WUNU, 2024. P. 97-98.

https://dspace.wunu.edu.ua/bitstream/316497/52868/1/CIT%272024_Last.pdf