Echo Trace:OSINT-Based Trend Analyzer Extension for X

WAHAJ AHMED
01-134221-086
MUHAMMAD UMAIR
01-134221-054

Supervisor: Dr. Muhammad Asfand E Yar

Bachelor of Science in Computer Science

Department of Computer Science
Bahria University, Islamabad

June 2025

Certificate

We accept the work contained in the report titled "OSINT-Based Twitter Trend Analyzer Extension," written by Wahaj Ahmed and Muhammad Umair, as meeting the standards for the partial fulfillment of the degree of Bachelor of Science in Computer Science.

Approved by . . . :

Supervisor: Dr. Muhammad Asfand E Yar

_____

Internal Examiner:

_____

External Examiner:

_____

Project Coordinator:

_____

Head of the Department:

_____

Abstract

The project of the OSINT-Based Twitter Trend Analyzer Extension provides an in-depth solution to the social media intelligence gathering based on applying sophisticated analytical frameworks to the free widely-used and updated browsers. Using the potential of natural language processing models, network analysis algorithms, and machine learning methods to identify the formation of trends, the system is able to dynamically and, in real time, process Twitter data streams on trending topic every day. The extension helps in improving intelligence collection by automating sentiment classification, influencer identification procedures and bot activity identification systems built directly into the Chrome browser interface. This combined method goes directly to the issue that most analysts, researchers and organizations have with handling massive amounts of social media data, and it makes more complex trend analysis and intelligence extraction far more efficient and available.

Moreover, it is possible to retrieve and analyze tweet patterns automatically and in an easy manner, and hence integrate sophisticated visualization features in a seamless context, which renders the OSINT Twitter Analyzer an immersive analytical presentation, specifically designed to meet the wide range of requirements of different professionals across various fields. The technical structure and execution is both on scalability and performance optimization, where infrastructure of a project is a viable way to process data efficiently and strong backend integration is done using Python/Flask structures and frontend using next.js implementation. Thus, the system does not only contribute to the enhancement of social media monitoring opportunity greatly, but also proactively enables guided decision-making, coordination of crisis responses, and collection of strategic intelligence in political analysis, journalism, brand and management. By so doing, the project will be able to contribute positively towards the democratization of sophisticated social media analytics as a background towards encouraging the practice of intelligence that is accessible and in real-time to democratic organization and research environments.

Acknowledgments

All praise is to Almighty Allah who bestowed upon us a minute portion of His infinite knowledge, by virtue of which we were able to complete this meaningful and challenging task.

We are deeply thankful to our supervisor, Dr. Muhammad Asfand E Yar, for his unwavering guidance, support, and insightful feedback throughout the development of our project. Without his continuous supervision and encouragement, the successful completion of this work would not have been possible.

We also extend our heartfelt gratitude to our parents and families, whose constant support, prayers, and belief in us have been a source of motivation and strength during this academic journey.

WAHAJ AHMED
MUHAMMAD UMAIR
Bahria University Islamabad, Pakistan

June 2025

Contents

Abstract i

1 Introduction 1
- 1.1 Background . . . 2
- 1.2 Problem Description . . . 2
- 1.3 Objectives . . . 3
- 1.4 Project Scope . . . 4
- 1.5 Solution Application Areas . . . 4

2 Introduction 6
- 2.1 Background . . . 7
- 2.2 Problem Description . . . 7
- 2.3 Objectives . . . 8
- 2.4 Project Scope . . . 8
- 2.5 Solution Application Areas . . . 8

3 Requirement Specifications 10
- 3.1 Existing System . . . 10
- 3.2 Proposed System . . . 10
- 3.3 Requirement Specifications . . . 11
  - 3.3.1 User requirements . . . 11
- 3.4 Functional Requirements . . . 12
- 3.5 Non-Functional Requirements . . . 14
- 3.6 Use Case Diagram . . . 15
- 3.7 Use Cases . . . 15
  - 3.7.1 Initiate New Trend Analysis . . . 15
  - 3.7.2 Perform Sentiment Analysis . . . 17
  - 3.7.3 Detect Influential Users . . . 17
  - 3.7.4 Identify Bot Activity . . . 17
  - 3.7.5 Search by Hashtag . . . 17
  - 3.7.6 Visualize Engagement Graphs . . . 17
  - 3.7.7 Geolocation Analysis . . . 22
  - 3.7.8 Top Entities . . . 22
  - 3.7.9 Find Trend in History . . . 22

4 System Architecture and Design **26**
4.1 Overview . . . 26
4.2 System Flow . . . 26
4.3 System Architecture . . . 27
4.4 Activity Diagram . . . 29
4.4.1 Context Diagram (Level 0 DFD) . . . 30
4.5 Block Diagram . . . 30
4.6 Entity Relationship (ER) Diagram . . . 32
4.7 Sequence Diagram . . . 32
4.8 GUI Design . . . 32
4.8.1 GUI Overview . . . 35
4.8.2 Interactive Behavior . . . 38

5 System Implementation **39**
5.1 System Architecture . . . 39
5.2 Tools and Technologies Used . . . 41
5.3 Database Management and Integration . . . 42
5.4 Sentiment Analysis and Influencer identification Modules . . . 42
5.4.1 Sentiment Analysis Engine . . . 42
5.4.2 Influencer Identification and Ranking . . . 43
5.5 Bot Activity Identification . . . 43
5.6 Deployment and Integration . . . 44

6 System Testing and Evaluation **45**
6.1 Test Approach . . . 45
6.1.1 Test Plan . . . 46
6.1.2 Testing tools and Requirements . . . 46
6.1.3 Test Cases . . . 47
6.2 Summary . . . 47

7 Conclusions **54**
7.1 Key Learnings . . . 54
7.2 Project Achievements . . . 55
7.3 Future Developments . . . 55

A User Manual **57**
A.1 Introduction . . . 57
A.2 System Requirements . . . 57
A.3 Installation Guide . . . 57
A.4 Launching the Extension . . . 58
A.5 User Interface Overview . . . 58
A.5.1 Header Section . . . 58
A.5.2 Filter Panel . . . 58
A.5.3 Main Analytics Panel . . . 59
A.6 Performing a Trend Analysis . . . 59
A.7 Understanding Sentiment Analysis . . . 59
A.8 Influencer Identification . . . 60
A.9 Bot Activity Detection . . . 60

A.10 Engagement Metrics Visualization 60
A.11 Geolocation Analysis 61
A.12 Historical Trend Analysis 61
A.13 Error Handling and Notifications 61
A.14 Best Practices for Users 61
A.15 Limitations 62
A.16 Conclusion 62

References **63**

List of Figures

3.1 Usecase Diagram 16
3.2 Use Case Diagram: Initiate New Trend Analysis 18
3.3 Use Case Diagram: Sentiment Analysis 19
3.4 Use Case Diagram: Detect Influential Users 19
3.5 Use Case Diagram: Identify Bot Activity 20
3.6 Use Case Diagram: Search by Hashtag 21
3.7 Use Case Diagram: Visualize Engagement Graphs 21
3.8 Use Case Diagram: Geolocation Analysis 23
3.9 Use Case Diagram: Top Entities 24
3.10 Use Case Diagram: Find Trend in History 25

4.1 System Flow Diagram 28
4.2 High-Level System Architecture 31
4.3 System Processing Flowchart 31
4.4 Level 0 Data Flow Diagram (Context Diagram) 33
4.5 Block Diagram of System Components 33
4.6 ER Diagram 34
4.7 Sequence Diagram for “Analyze Hashtag” Workflow 36
4.8 GUI Mockup – Left hand sidebar 36
4.9 GUI Mockup – Sentiment Analysis 36
4.10 GUI Mockup – Top Users 37
4.11 GUI Mockup – Location Analysis 37

List of Tables

3.1 Use Case: Initiate New Trend Analysis 16
3.2 Use Case FR2: Perform Sentiment Analysis 18
3.3 Use Case FR3: Detect Influential Users 19
3.4 Use Case FR4: Identify Bot Activity 20
3.5 Use Case FR5: Search by Hashtag 20
3.6 Use Case: Visualize Engagement Graphs 21
3.7 Use Case FR8: Geolocation Analysis 23
3.8 Use Case FR9: Top Entities 24
3.9 Use Case FR10: Find Trend in History 25

6.1 Dashboard Navigation 48
6.2 Hashtag Search and Analysis 48
6.3 Sentiment Analysis Classification 49
6.4 Influencer Detection and Ranking 49
6.5 Bot identification 50
6.6 Engagement Metrics Visualization 50
6.7 Time-Series Trend Graphs 51
6.8 User Profile Details Display 51
6.9 Cache Performance and Hit Rates 52
6.10 API Response Time 52

Acronyms and Abbreviations

DSA	Data Structure and Algorithms
OOP	Object Oriented Programming
PF	Programming Fundamentals
SE	Software Engineering
SQL	Structured Query Language
UNESCO	United Nations Educational, Scientific and Cultural Organization
UNICODE	Unique, Universal, and Uniform Character enCoding
XML	Extensible Markup Language

Chapter 1

Introduction

The emergence of social media sites has radically altered the information distribution and discussion among populations in the digital era. X is one of these platforms and it is a crucial communication medium where real time discussions affect the opinion of the people, political movements, and market trends in various industries. The platform produces an otherwise unmatched quantity of information in a single day and there are millions of tweets daily that hold invaluable information on social feelings, tempos and trends of trustworthiness in information. This has made the task of isolating meaningful intelligence amidst this mass data ecosystem an even more complicated issue especially given the speed at which information propagates as well as the intricacy of the modern disinformation operation. Conventional methods of analysis cannot match the size, pace and intricacy of X data streams. Manual ways of analysis are not only time consuming but also prone to human error in terms of bias and errors in interpretation, hence they are not applicable in cases where real time decisions are supposed to be made. The Open Source Intelligence (OSINT) practices introduce a formal structure and framework of gathering, processing and analyzing publicly accessible data on social media platforms. OSINT frameworks can convert raw social media data into useable intelligence using the latest computational methods such as natural language processing, machine learning algorithms, among others. It is an invaluable intelligence in various fields such as political campaign analysis, crisis management, checking the fact of journalistic facts, and market research projects. With the advent of browser extensions as easily available analysis tools, the capability of advanced data analysis has been made democratic so that non-technical users can employ highly sophisticated intelligence collection methods. The combination of this accessibility feature with the increasing demand to integrate into real-time social media analysis functions preconditions the creation of the solutions based on the complete analysis of X trends to be integrated into the existing workflows of users.

1.1 Background

The use of social media sites has completely transformed the flow of information and the life cycle of mass communication in the modern digital age. X is one of these platforms, which is a potential key driver of mass opinion, mobilization of politics, and market trends because of real-time discussion format and extensive use of the application globally.

Every day X users provide an enormous amount of information, summarizing essential information about the tendencies in society, changes in mood, and trends in information credibility. This has made the quality of analysis of such huge stream of information very difficult because of the high rate at which information diffuses and the advanced techniques used in the modern day misinformation campaigns. Manual analysis procedures, which are human interpreted and stagnant in the approach, are very inefficient and subject to biases. Such methods cannot reflect the dynamic, real-time changes that the discussion on social media has, and thus such approaches are not suitable in making quick and informed decisions in different areas.

The Open-Source Intelligence (OSINT) practices provide a solid answer to these problems. OSINT is the systematic gathering, handling, and examination of openly accessible information in social media with the aid of the state-of-the-art computational units like Natural Language Processing (NLP) and machine learning algorithms. Such approaches are needed to transform huge volumes of raw data into useful information, which can be applied to a wide variety of domains such as political strategy, journalism, crisis management, cybersecurity, and market analytics.

The current advancements in technology, particularly the use of browser-based analytical websites, have made data analysis a democratic process since the non-technical users can now access complex OSINT functionalities. This high-order analysis coupled with their daily utilization through browser extensions enables the user to incorporate strong analytical functionality with ease in his/her working process. As a result, there exists a great and increasing need of the ready-to-use, real-time analysis solutions that will be capable of tracking and understanding the trends in social media, thereby providing the justification behind the formation of all-inclusive X trend analysis solutions.

1.2 Problem Description

The data of X grows exponentially making it very difficult to analyze the data effectively. The current methodologies are not adequate to handle the large scale, fast, and multifaceted content of social media due to its manual or semi-automated nature. The inability to draw timely, precise, and viable insights out of Twitter information creates a problem for analysts, journalists, marketers, and policymakers with respect to responding to emerging trends, misinformation, and public sentiment.

Old data of X rather than of live data is the primary foundation of traditional analytical tools, which are not as deep to capture the sentiment subtleties, distinguish influential users and find advanced campaigns of misinformation. Such shortcomings lead to insufficient knowledge, slow decision-making and higher vulnerability to the negative impact of misinformation especially in situations that are critical such as elections, emergencies or health emergencies. Therefore, there is an urgent demand of automated and high-end analytical systems to provide real-time sentiment analysis, identification of influencers, identification of bots, and overall trend analytics.

1.3 Objectives

The main goal of the project is to come up with a strong, easy-to-use OSINT-based extension of X trend analysis that is capable of significantly improving social media analytics through offering real-time and actionable insights. The project has the detailed objectives:

- **X Data Collection and Processing:** Implement an automated data collection and processing system that will be able to effectively process the collected X data: hashtags, mentions, user interactions, and engagement metrics etc. This goal makes a smooth processing of structured datasets.
- **Complex Sentiment Analysis Implementation:** Train complex sentiment analysis algorithms based on trained NLP models on X in order to classify Twitter conversations with high accuracy into positive, negative, or neutral. This feature enables users to be able to quantify what people think and can quickly define the trends in opinion.
- **Influencer Identification and Ranking:** Develop all-purpose procedures to identify and rank influential X users in terms of network ties, followers, activity, and credibility measures. This helps to identify key opinion leaders that influence the discussions about certain trends.
- **Strong Bot identification Structures:** Develop powerful structures of identification of synchronized bot actions and content distribution structure exceptions. This is to make sure that man-made inflated stories are identified fast.
- **Intuitive Operational Dashboard:** Design and Implement an interactive rich operational dashboard that will display analytical output in a visualized form, such as charts and graphs. This goal guarantees access to and clarity of obscure data understanding to various users.

- **Customizable Analysis parameter:** Introduce convenience-level features to allow the end user to adjust analytical parameter, such as selection by hashtags. This elasticity guarantees focused analysis in line with certain user needs
- **Historical Trend Analysis and Reporting:** Build historical trend analysis functionality whereby customers can see how the X conversations have changed over time. Also, add reporting tools to allow easy exporting of reports and additional analysis of analytical findings.
- **Scalable and Efficient System Architecture:** User will get a scalable efficient architecture that would support future expansion, future higher data process needs, and incorporation of other features without affecting the performance and reliability.

1.4 Project Scope

The project will include the entire development of an OSINT based Chrome extension that will be focused on the analysis of X trends. The scope involves creation of advanced sentiment analysis, identification of influencers, and real-time analytics dashboards consisting of interactive visualization and easy usability. It also has functionality of historical trend analysis, whereby it allows users to explore how a discussion has changed with time.

The project will not directly integrate with Twitter APIs, create automatic forms of interaction or responding to analysis outcomes, and/or third-party tools, other than basic visualization and reporting. The project will be dependent on external data providers to supply the datasets and this would need to be coordinated to facilitate consistency in quality and timely data. The scope is used to demarcate the limits of the system and is concerned with analytical service and not integration into the platform or automated response, which may breach platform terms of service or lead to ethical objections.

1.5 Solution Application Areas

The X Trend Analyzer Extension, which is based on OSINT, has a great deal of actual applications in the various professional fields:

- **Political Analysis and Campaign Management:**
  Facilitates real time tracking of voter sentiment, who are influential commentators and early detection of hot areas around the time of elections.
- **Crisis Management and Emergency Response:**
  This crisis management and response service is designed to assist in managing and responding to any crisis that might impact the nation.Supports emergency services

and crisis management teams to monitor urgent messages, the development of sentiments, and geolocation information in case of an emergency or disaster.

- **Journalism and Media Analysis:**
  Supports journalists and fact-checkers in verifying information authenticity, identifying emerging stories, and combating misinformation.
- **Market Research and Brand Monitoring:**
  Enables the marketers and brand managers to gauge brand sentiments, monitor competitor moves, evaluate consumer perceptions, and gauge marketing success.
- **Cybersecurity and Threat Intelligence:**
  Offers cybersecurity experts with instruments to discover bot-based stories, and online activities.

Chapter 2

Introduction

The emergence of social media sites has radically altered the information distribution and discussion among populations in the digital era [1]. X is one of these platforms and it is a crucial communication medium where real time discussions affect the opinion of the people, political movements, and market trends in various industries. The platform produces an otherwise unmatched quantity of information in a single day and there are millions of tweets daily that hold invaluable information on social feelings, tempos and trends of trustworthiness in information [2]. This has made the task of isolating meaningful intelligence amidst this mass data ecosystem an even more complicated issue especially given the speed at which information propagates as well as the intricacy of the modern disinformation operation [3]. Conventional methods of analysis cannot match the size, pace and intricacy of X data streams. Manual ways of analysis are not only time consuming but also prone to human error in terms of bias and errors in interpretation, hence they are not applicable in cases where real time decisions are supposed to be made. The Open Source Intelligence (OSINT) practices introduce a formal structure and framework of gathering, processing and analyzing publicly accessible data on social media platforms [4]. OSINT frameworks can convert raw social media data into useable intelligence using the latest computational methods such as natural language processing, machine learning algorithms, among others. It is an invaluable intelligence in various fields such as political campaign analysis, crisis management, checking the fact of journalistic facts, and market research projects. With the advent of browser extensions as easily available analysis tools, the capability of advanced data analysis has been made democratic so that non-technical users can employ highly sophisticated intelligence collection methods. The combination of this accessibility feature with the increasing demand to integrate into real-time social media analysis functions preconditions the creation of the solutions based on the complete analysis of X trends to be integrated into the existing workflows of users.

2.1 Background

The use of social media sites has completely transformed the flow of information and the life cycle of mass communication in the modern digital age [1]. X is one of these platforms, which is a potential key driver of mass opinion, mobilization of politics, and market trends because of real-time discussion format and extensive use of the application globally.

Every day X users provide an enormous amount of information, summarizing essential information about the tendencies in society, changes in mood, and trends in information credibility [2]. This has made the quality of analysis of such huge stream of information very difficult because of the high rate at which information diffuses and the advanced techniques used in the modern day misinformation campaigns [3]. Manual analysis procedures, which are human interpreted and stagnant in the approach, are very inefficient and subject to biases. Such methods cannot reflect the dynamic, real-time changes that the discussion on social media has, and thus such approaches are not suitable in making quick and informed decisions in different areas.

The Open-Source Intelligence (OSINT) practices provide a solid answer to these problems [4]. OSINT is the systematic gathering, handling, and examination of openly accessible information in social media with the aid of the state-of-the-art computational units like Natural Language Processing (NLP) and machine learning algorithms. Such approaches are needed to transform huge volumes of raw data into useful information, which can be applied to a wide variety of domains such as political strategy, journalism, crisis management, cybersecurity, and market analytics.

The current advancements in technology, particularly the use of browser-based analytical websites, have made data analysis a democratic process since the non-technical users can now access complex OSINT functionalities. This high-order analysis coupled with their daily utilization through browser extensions enables the user to incorporate strong analytical functionality with ease in his/her working process. As a result, there exists a great and increasing need of the ready-to-use, real-time analysis solutions that will be capable of tracking and understanding the trends in social media, thereby providing the justification behind the formation of all-inclusive X trend analysis solutions.

2.2 Problem Description

The data of X grows exponentially making it very difficult to analyze the data effectively [2]. The current methodologies are not adequate to handle the large scale, fast, and multifaceted content of social media due to its manual or semi-automated nature. The inability to draw timely, precise, and viable insights out of Twitter information creates a problem for analysts, journalists, marketers, and policymakers with respect to responding to emerging trends, misinformation, and public sentiment [5].

Old data of X rather than of live data is the primary foundation of traditional analytical tools, which are not as deep to capture the sentiment subtleties, distinguish influential users and find advanced campaigns of misinformation [3]. Such shortcomings lead to insufficient knowledge, slow decision-making and higher vulnerability to the negative impact of misinformation especially in situations that are critical such as elections, emergencies or health emergencies. Therefore, there is an urgent demand of automated and high-end analytical systems to provide real-time sentiment analysis, identification of influencers, identification of bots, and overall trend analytics.

2.3 Objectives

The main goal of the project is to come up with a strong, easy-to-use OSINT-based extension of X trend analysis that is capable of significantly improving social media analytics through offering real-time and actionable insights.

2.4 Project Scope

The project will include the entire development of an OSINT based Chrome extension that will be focused on the analysis of X trends. The scope involves creation of advanced sentiment analysis, identification of influencers, and real-time analytics dashboards consisting of interactive visualization and easy usability. It also has functionality of historical trend analysis, whereby it allows users to explore how a discussion has changed with time.

The project will not directly integrate with Twitter APIs, create automatic forms of interaction or responding to analysis outcomes, and/or third-party tools, other than basic visualization and reporting. The scope is used to demarcate the limits of the system and is concerned with analytical service and not integration into the platform or automated response, which may breach platform terms of service or lead to ethical objections [6].

2.5 Solution Application Areas

The X Trend Analyzer Extension, which is based on OSINT, has a great deal of actual applications in the various professional fields:

- **Political Analysis and Campaign Management**
- **Crisis Management and Emergency Response**
- **Journalism and Media Analysis**
- **Market Research and Brand Monitoring**

- **Cybersecurity and Threat Intelligence**

Chapter 3

Requirement Specifications

3.1 Existing System

The existing state of the social media intelligence collection, especially of such service as X, is based on a manual control of the processes, separate analytics tools, or few third-party services which can provide only superficial insights. Some of them offer trend tracking or sentiment analysis but fail to provide the essential elements like influencer identification, geolocation analysis, and detailed engagement analytics on a single and easily accessible platform. The solutions currently available are frequently limited to more superficial data analysis and do not have a contextual understanding or a more sophisticated feature such as network centrality computation, bot behavior detection, or real-time sentiment tracking.

Moreover, such systems tend to be technical to operate, and usually rely on stale X data, and thus cannot be utilized by non-technical users like journalists, political analysts, campaign managers and emergency response personnel. Moreover the majority of platforms lacks modular architecture, scalable processing, and cloud based storage required to process high volumes of data in real time. Consequently, experts in different fields are experiencing a major challenge of deriving timely, reliable, and actionable intelligence on the social media. These tools are discontinuous, and have minimal levels of analyses and ease of use, which complicates the ability to satisfy modern and data-driven decision-making.

3.2 Proposed System

The OSINT-Based X Trend Analyzer Extension is a Chrome browser extension which offers an all-in-one social media intelligence tool by offering an in-built analytical framework. The system will process X data using external sources of data, run a sophisticated set of analytical algorithms, such as sentiment and influencer identification, and display results in a user-friendly dashboard interface.

The system architecture is in the form of a client-server where the Chrome extension acts as a frontend component and back-end services carry out data processing, data analysis, and data storage. The extension secures data and provides integrity of the data sent to the analytics system by communicating with backend services via secure API endpoints.

3.3 Requirement Specifications

Requirement specifications are accounts of the anticipated behavior and performance of the proposed system. These are normally classified into two:

3.3.1 User requirements

Individuals who use OSINT-Based X Trend Analyzer Extension such as journalists, political commentators, emergency services, marketers, and researchers will desire the system to be user-friendly, quick and informative. They desire the extension to enable them to follow real-time X trends, sentiment analysis, and influential users in topics. The interface must have the results displayed in an engaging dashboard and give users the option to filter the information based on the time and topics.

Users also demand that the system should be compatible with their browser, and provide precise and context-driven insights without the need to have high-order technical knowledge. They also want customized systems analysis of new threats or trend events or sentiment changes. The following user requirements define the expected features and behaviors of the system in terms of functionality, usability, and integration.

- **UR-01: Real-Time Trend Detection** User should be capable of identifying trending topics on X in real-time, including hashtags, mentions as well as keywords that are pertinent to their interests or areas of interest.
- **UR-02: Sentiment Analysis with NLP** Advanced NLP models should be used to classify the sentiment of a tweet as positive, negative, or neutral, with score of confidence.
- **UR-03: Influencer Identification** Influencer account users are expected to be able to distinguish between influential accounts by such parameters as engagement rate, number of followers, and frequency of retweets.
- **UR-04: Dashboard Interface** The users should be given an interactive dashboard with filters which can be customized (e.g., time) to analyze accurately.
- **UR-05: Semantic Search and Querying** The system should support semantically querying and search through datasets with the use of natural language queries.

- **UR-06: Cross-Platform Compatibility** The browser extension should be receptive and compatible in various platforms such as desktop, laptops, and tablets.
- **UR-08: Historical and Comparative Analysis** The user is expected to see trends over time and compare sentiment, activity, or influence of various periods of time or events.
- **UR-09: Visual Representation of Data** Analytical results must be presented with interactive graphs, charts, and network diagrams using libraries such as D3.js and Chart.js.
- **UR-10: Error Handling** When the system is unable to handle a request or no result is provided, the system should display alternative error messages to help the user.

3.4 Functional Requirements

FR1: Data Acquisition Interface

- **Input:**
  - User initiates analysis by searching a trending topic.
- **Output:**
  - Clean dataset ready for processing, including tweet content, metadata, and user information. .
- **Process:**
  1. Connect to external data provider API. .
  2. Fetch a batch of tweets per topic per day .
  3. Parse and store relevant data fields.
- **Description:** Empowers the system in gathering X data in a structured format, so that it is ready to downstream analytics and pre-processed and complete.

FR2: Sentiment Analysis Engine

- **Input:**
  - Cleaned tweet content.
- **Output:**
  - Sentiment classification (positive, negative, neutral) with confidence score.

- **Process:**
  1. Apply pre-trained NLP models.
  2. Return labeled sentiment and store results.
- **Description:** Analyzes user opinions expressed in tweets to detect overall public sentiment about a topic or event.

FR3: Influencer identification and Ranking

- **Input:**
  - User interactions, follower count, retweet metrics, and engagement stats.
- **Output:**
  - Ranked list of top influencers with influence scores.
- **Process:**
  1. Calculate engagement ratios, and follower quality.
  2. Sort and return the most influential accounts.
- **Description:** Identifies opinion leaders and influential users who significantly shape discussions on specific topics.[7].

FR4: Bot identification

- **Input:**
  - Tweet metadata, account behavior logs, and engagement patterns.
- **Output:**
  - Suspicion flags, risk scores, and potential network activity labels .
- **Process:**
  1. Analyze anomalies in behavior, timing, and content.
  2. Identify bot-like behaviors using different patterns.
- **Description:** Detects coordinated inauthentic behavior using statistical and behavioral analysis. [3].

FR5: Interactive Analytics Dashboard

- **Input:**
  - User-selected filters (e.g., hashtag).
- **Output:**
  - Real-time visualizations of trends, sentiment, influencers, locations and more.
- **Process:**
  1. Query analytics database based on filters.
  2. Render charts and graphs using D3.js, Chart.js.
- **Description:**

  Provides a user-friendly interface to explore complex data through visual summaries.

FR6: Advanced Search

- **Input:**
  - User-entered keywords or queries.
- **Output:**
  - Matched tweets, users, or trends with semantic relevance.
- **Process:**
  1. Perform semantic search using MONGODB.
  2. Return high-relevance results.
- **Description:** Allows users to find the most contextually relevant data without relying on exact keyword matches.

3.5 Non-Functional Requirements

- **NFR1: Internet Connectivity**
  The system must require a stable internet connection to access external X data providers, cloud-based APIs, MongoDB database, and analytics services.
- **NFR2: Secure Database Access**
  The system must connect to MongoDB Atlas or Firebase Realtime Database via secure channels (TLS/SSL).

- **NFR3: RESTful API Communication**
  Any unfolding between the browser extension and the services on the back-end should be through RESTful API and secure endpoints.
- **NFR4: next-based Chrome Extension Frontend**
  The user-interface of the extension should be built with next.js and be styled with Bootstrap CSS in a manner that it is responsive to changes in screen size and browser setting.
- **NFR5: Python + Flask Backend Server**
  The extension has to be implemented as backend services written in Python and Flask that offer scalable services on data ingestion, NLP processing, and analytics.
- **NFR6: Python 3.10+ Compatibility**
  The system should also operate on Python version 3.10 or more, which is compatible with NLP libraries.

3.6 Use Case Diagram

There are detailed descriptions of each use case with preconditions, main flows, alternative flows, and postconditions to guarantee the full understanding of the system by any kind of user.The OSINT-Based X Trend Analyzer Extension can be used by all types of users.

The extension can also help users to explore trending Twitter topics, sentiment analysis, and view simple influencer data, as well as public visualizations. They are able to play with major use cases like “Search Trends, View Sentiment Summary and Explore Hashtag Insights. They however, lack depth and export capabilities in data. These interactions are the ones that include relationships with core modules like real-time data processing and visualization rendering.

3.7 Use Cases

3.7.1 Initiate New Trend Analysis

The Initiate New Trend Analysis option lets general users start to analyze a new trending topic or hashtag. Upon activation, it erases the current analysis session and asks the user to input/choose a new keyword/hashtag, and kicks off the backend, to fetch and process coincidental X(Twitter) data in real time. This guarantees a keen and unspoilt condition of every new trend analysis.

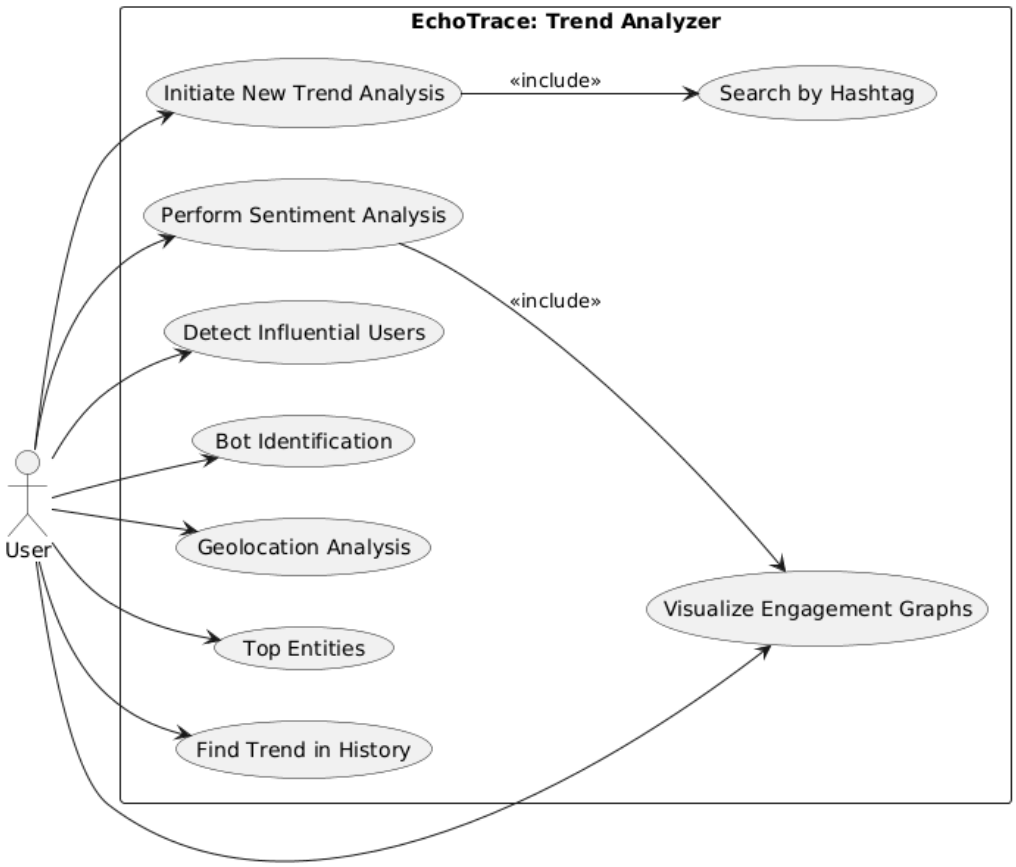

Figure 3.1: Usecase Diagram

Table 3.1: Use Case: Initiate New Trend Analysis

Use Case ID	FR1
Use Case Name	Initiate New Trend Analysis
Actors	General User, Registered Analyst
Data	Selected or entered keyword/hashtag, previous session data (if any)
Trigger Stimulus	User search the new trend
Precondition	User is on the extension interface and authenticated (if required)
Assumptions	System backend is online; X(Twitter) data source is accessible
Normal Flow of Events	1. User enters a new topic to analyze. 2. System clears any previous session data from memory/storage. 3. A new analysis is generated. 4. Backend fetches new Twitter data for the selected keyword. 5. Initial insights (volume, sentiment, engagement) are displayed.
Alternative Flow of Events	1. If the Twitter data source is unavailable, display an error message. 2. If trend fetching fails, notify the user and allow re-entry. 3. If initialization fails, prompt the user to retry later.
Main Success Scenario	A new analysis session is successfully started and initial trend data is visualized.
Postcondition	Previous analysis session is cleared; new session data is saved.

3.7.2 Perform Sentiment Analysis

X Trend Analyzer Extension is the tool that enables users to determine the sentiment in the social media conversations employing Natural Language Processing (NLP). The system is a query processor that parses user typed queries in the natural language and extracts significant entities, feelings, and a deep-seated purpose. It then parses through the tweets and provides topical tweets, sentiment classifications using advanced classification algorithms, and shows the results in interactive graphics.

3.7.3 Detect Influential Users

X Trend Analyzer Extension recognizes influential users through the use of engagement metrics. Relevant information on Twitter is retrieved and then processed by the system that identifies key opinion leaders, trend effectors and follower connections, and the level of engagement to guarantee that insights are correct and current.

3.7.4 Identify Bot Activity

The **X Trend Analyzer Extension** employs bot identification rules to assess the authenticity of trending content. It analyzes tweet metadata, engagement patterns, and linguistic features, tagging suspicious content based on anomaly identification rule based techniques. The system integrates bot credibility scoring to filter out coordinated inauthentic behavior, ensuring users receive accurate and reliable insights.

3.7.5 Search by Hashtag

X Trend Analyzer Extension will allow one to follow trending topics and discussions using a search query of particular hashtags. The system pulls the corresponding tweets that relate to the chosen hashtag, giving the sentiment analysis, engagement rate, and influential individual. The user is able to look through historical data on the trend, compare the activity of a hashtag over time, and refine their searches based on location, time, and content relevance in order to gain a better understanding of the changing.

3.7.6 Visualize Engagement Graphs

The X trend analyzer extension offers interactive engagement graph visualization to assist the user to study the interaction of tweets, retweets, replies, and likes over a period of time. The system extracts the data about Twitter interactions, converts network relationships, and produces dynamic graphs to represent the measure of user influence, sentiment trends, and content reach. The graphs provide information on patterns of discussions enabling users to know how trends develop.

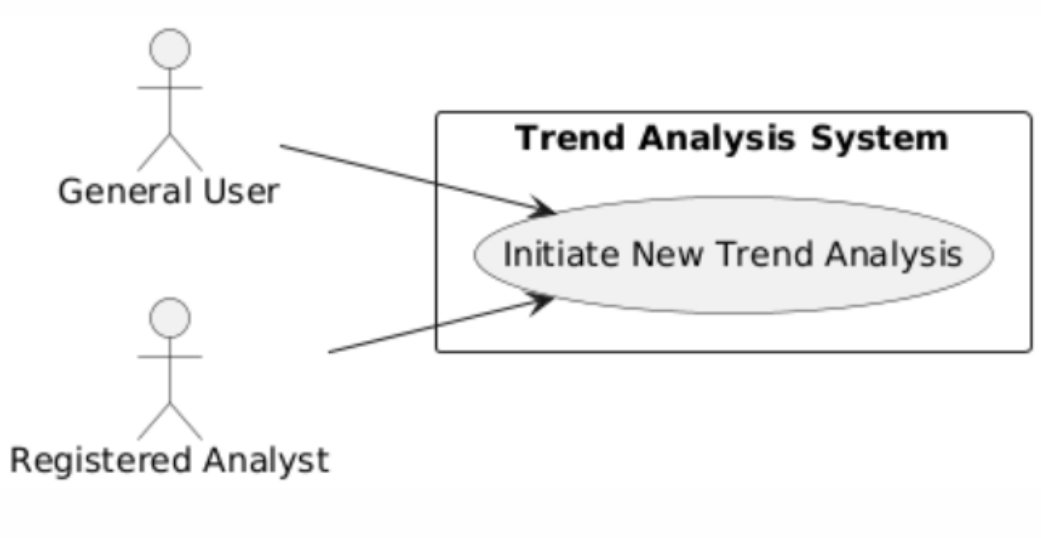

Figure 3.2: Use Case Diagram: Initiate New Trend Analysis

Table 3.2: Use Case FR2: Perform Sentiment Analysis

Use Case ID	FR2
Use Case Name	Perform Sentiment Analysis
Actors	General User, Registered Analyst
Data	Pre-processed tweet text for the selected trend
Trigger Stimulus	Users will scroll to “Analyze Sentiment” section.
Precondition	Valid trend data is available in the current session
Assumptions	Sentiment engine is functional
Normal Flow of Events	1. User requests sentiment analysis. 2. Tweets are passed to the NLP model. 3. Each tweet is classified as positive, negative, or neutral. 4. Aggregated sentiment is visualized.
Alternative Flow of Events	1. If tweet data is missing, notify the user. 2. If NLP model fails, log error and show fallback result.
Main Success Scenario	Sentiment breakdown is displayed with percentages and charts
Postcondition	Sentiment tags are stored and used in dashboards

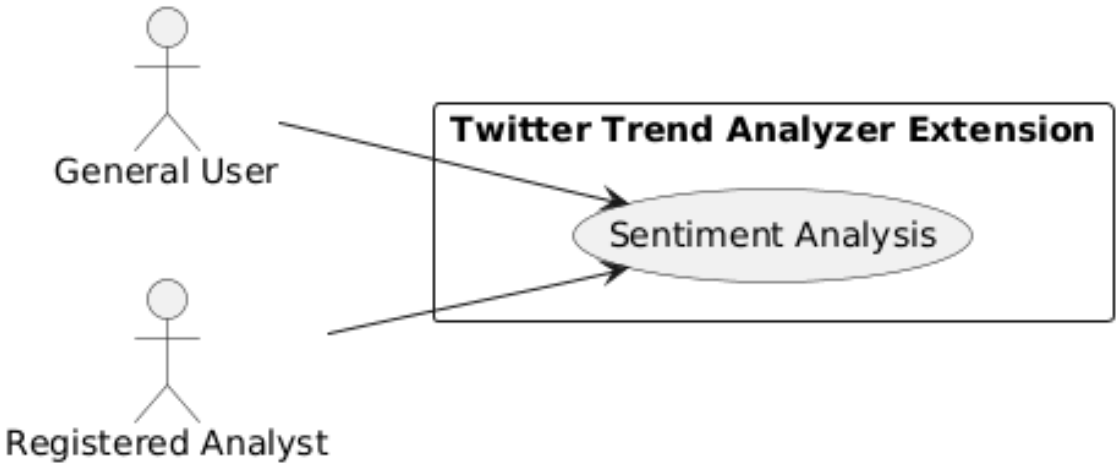

Figure 3.3: Use Case Diagram: Sentiment Analysis

Table 3.3: Use Case FR3: Detect Influential Users

Use Case ID	FR3
Use Case Name	Detect Influential Users
Actors	General User, Registered Analyst
Data	User engagement metrics (retweets, followers, mentions).
Trigger Stimulus	User selects Any trend or search any trend to view the influential users
Precondition	Trend data is loaded with user metadata.
Assumptions	Influence scoring algorithm is accessible.
Normal Flow of Events	1. User data is sorted based on engagement. 2. Influence score is computed for each account. 3. Top influencers are shown with metrics.
Alternative Flow of Events	1.If user metadata is missing, show warning. 2. If computation fails, return default ranked list.
Main Success Scenario	List of ranked influencers is shown with impact scores.
Postcondition	Influencer data is cached for reuse.

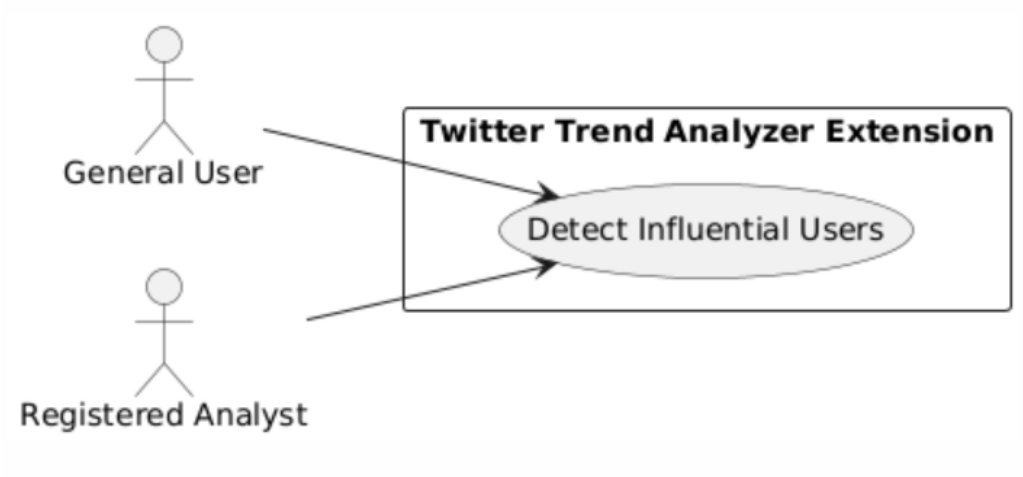

Figure 3.4: Use Case Diagram: Detect Influential Users

Table 3.4: Use Case FR4: Identify Bot Activity

Use Case ID	FR4
Use Case Name	Identify Bot Activity
Actors	General User, Registered Analyst
Data	Tweet content, user metadata, interaction patterns.
Trigger Stimulus	User clicks "on User Profile"
Precondition	Valid trend data has been processed.
Assumptions	Bot identification scoring models are accessible.
Normal Flow of Events	1. System analyzes tweet patterns and sources 2. Flags suspicious behavior or inauthentic engagement
Alternative Flow of Events	1.If analysis fails, display error and offer retry 2.If data is insufficient, warn user to again search query.
Main Success Scenario	Bot flagged tweet and account is labelled with red marker.
Postcondition	Analysis can be used for future so that user will be aware of bot acitivities .

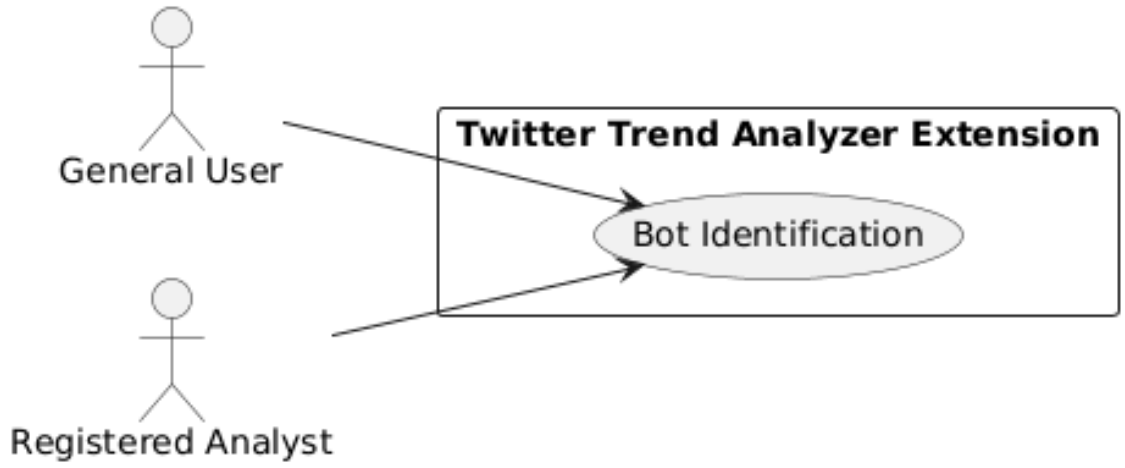

Figure 3.5: Use Case Diagram: Identify Bot Activity

Table 3.5: Use Case FR5: Search by Hashtag

Use Case ID	FR5
Use Case Name	Search by Hashtag
Actors	General User, Registered Analyst
Data	Twitter search query (hashtag, keyword, or @username)
Trigger Stimulus	User enters query into search bar.
Precondition	Extension is installed and initialized.
Assumptions	Backend search engine is running.
Normal Flow of Events	1. User enters search term (e.g., "AI" or "@elonmusk"). 2. System queries backend and retrieves relevant tweet data.
Alternative Flow of Events	1.If no results found, prompt user to modify query. 2.If backend fails, show fallback message.
Main Success Scenario	Query results and trend metrics are shown to the user.
Postcondition	Session data is saved and used for future recommendations.

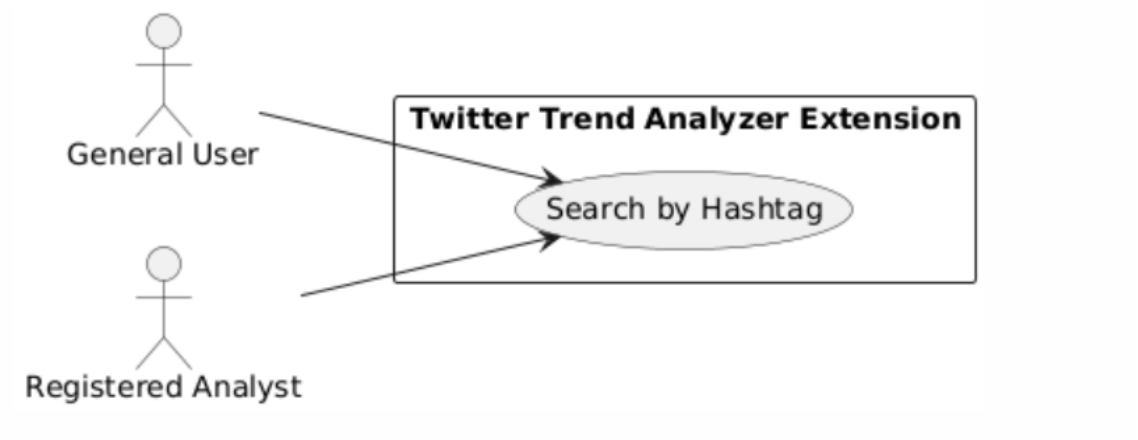

Figure 3.6: Use Case Diagram: Search by Hashtag

Table 3.6: Use Case: Visualize Engagement Graphs

Use Case ID	FR7
Use Case Name	Visualize Engagement Graphs
Actors	General User, Registered Analyst
Data	Retweet count, like count, reply count, follower growth
Trigger Stimulus	User scrolls to “Engagement Metrics” section on dashboard
Precondition	A valid trend session is active with engagement data.
Assumptions	Chart rendering libraries (D3.js/Chart.js) are integrated
Normal Flow of Events	1. System retrieves tweet-level engagement data. 2. Aggregates and prepares data for graphing. 3. Displays line/bar/pie charts interactively
Alternative Flow of Events	1. If data is incomplete, warn the user. 2. If rendering fails,ask user to wait.
Main Success Scenario	Charts are rendered showing how users engaged with the trend.
Postcondition	Graphs remain available for comparison.

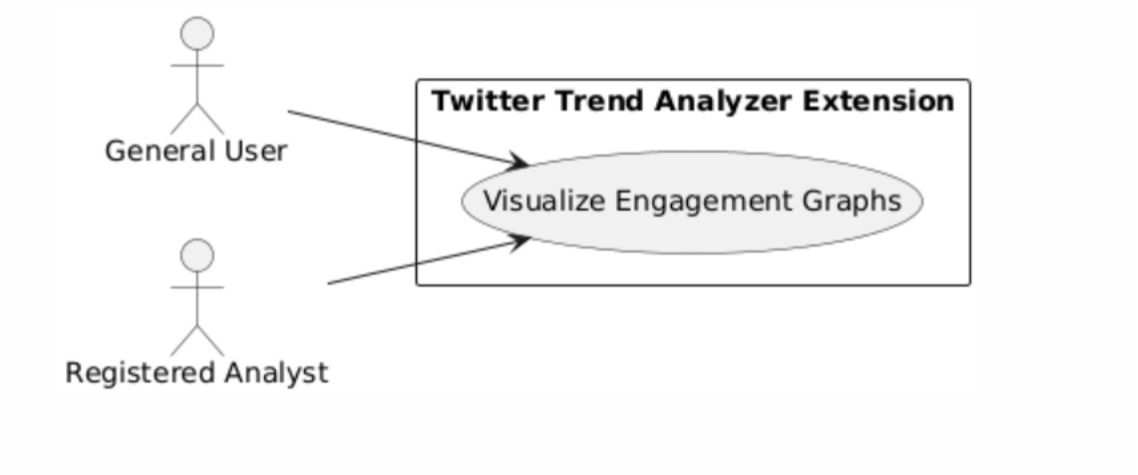

Figure 3.7: Use Case Diagram: Visualize Engagement Graphs

3.7.7 Geolocation Analysis

The **X Trend Analyzer Extension** performs geolocation analysis to examine the regional distribution of trending X content. It uses publicly available metadata such as user-declared location, language, and time-zone information to infer the geographic origin of tweets. This analysis helps identify region-wise engagement patterns and supports OSINT investigations by providing contextual insight into how trends spread across different locations.

3.7.8 Top Entities

The **X Trend Analyzer Extension** identifies top entities in Twitter trends to provide insights into the most influential users, hashtags, and mentions. It analyzes tweet metadata, engagement patterns, and interaction frequency to rank entities, enabling users to understand key participants in trending discussions.

3.7.9 Find Trend in History

The **X Trend Analyzer Extension** allows users to explore historical Twitter trends. It analyzes archived tweet data, engagement metrics, and temporal patterns to provide insights into how trends evolved over time. Users can filter trends by date ranges, keywords, or regions to gain meaningful historical insights.

Table 3.7: Use Case FR8: Geolocation Analysis

Use Case ID	FR8
Use Case Name	Geolcoation Analysis
Actors	General User, Registered Analyst
Data	Tweet content, user metadata, location coordinates, interaction patterns.
Trigger Stimulus	User clicks "on folium map and labels marked on it"
Precondition	Trend data with location information has been collected and processed.
Assumptions	Location extraction services/APIs are accessible and accurate.
Normal Flow of Events	1. System extracts geolocation from tweets and user metadata. 2. Maps tweet activity and trends on a geographic visualization. 3. Highlights regions with highest activity or unusual trends. 4. Provides insights about trend origins and regional engagement.
Alternative Flow of Events	1.If location data is unavailable, display a warning to the user. 2.If mapping fails, show error and offer retry.
Main Success Scenario	Trend geolocation is successfully visualized, allowing users to identify regions of high engagement and emerging trend hotspots.
Postcondition	Users and analysts can leverage geolocation insights to understand regional trend patterns and make informed decisions for OSINT investigations.

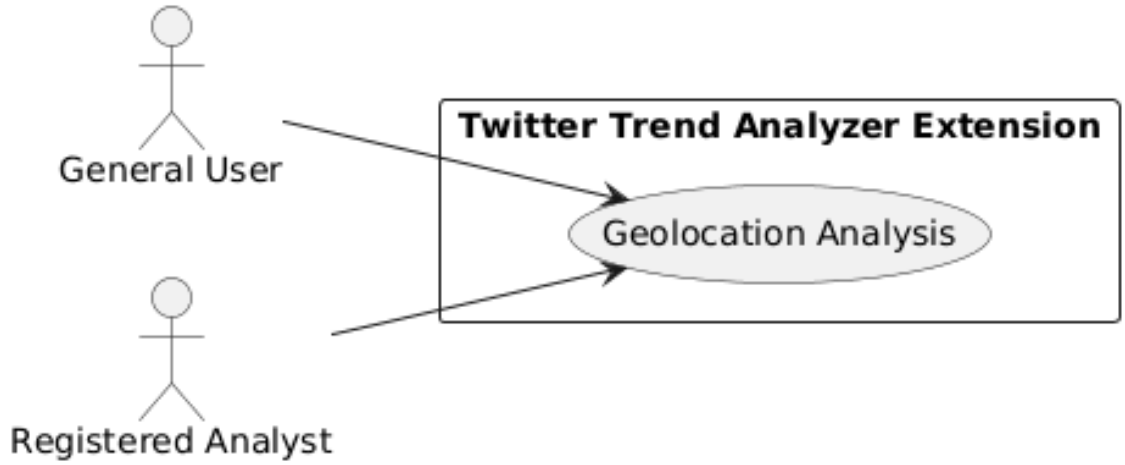

Figure 3.8: Use Case Diagram: Geolocation Analysis

Table 3.8: Use Case FR9: Top Entities

Use Case ID	FR9
Use Case Name	Identify Bot Activity
Actors	General User, Registered Analyst
Data	Tweet content, user metadata, hashtags, mentions, interaction patterns.
Trigger Stimulus	User search query.
Precondition	Valid trend data has been processed.
Assumptions	Entity ranking algorithms and metrics are available.
Normal Flow of Events	1. System extracts users, hashtags, and mentions from tweets. 2. Calculates engagement metrics (likes, retweets, mentions). 3. Ranks entities based on frequency and influence. 4. Displays top entities to the user in a clear format (e.g., table or chart).
Alternative Flow of Events	1. If entity extraction fails, display error and offer retry. 2. If insufficient data, warn user to refine query.
Main Success Scenario	Top entities are successfully identified and displayed, providing actionable insights for trend analysis.
Postcondition	Users can leverage top entity information to track influential users, trending hashtags, and important mentions for OSINT investigations.

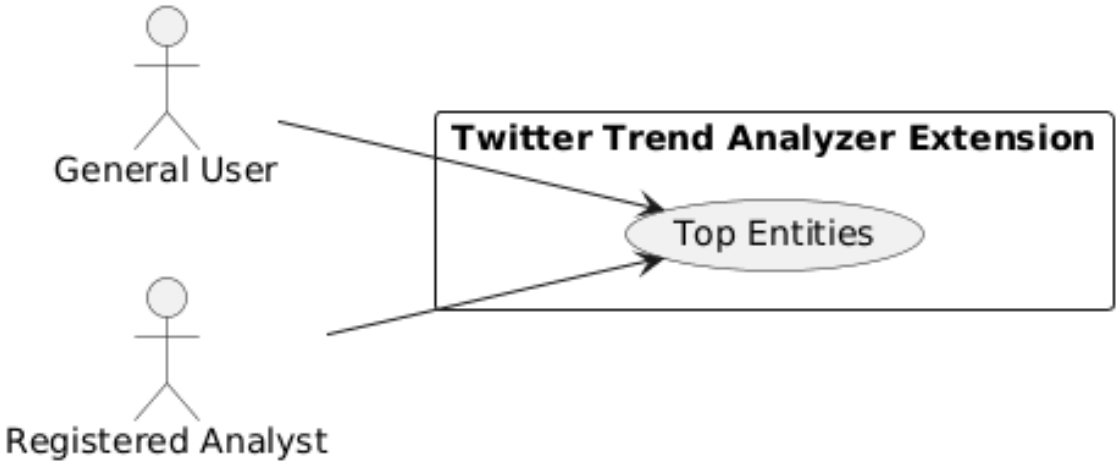

Figure 3.9: Use Case Diagram: Top Entities

Table 3.9: Use Case FR10: Find Trend in History

Use Case ID	FR10
Use Case Name	Identify Bot Activity
Actors	General User, Registered Analyst
Data	Historical tweet content, metadata, engagement patterns, timestamps.
Trigger Stimulus	User clicks "View Historical Trends" with a date range.
Precondition	Historical trend data has been collected and stored in the system.
Assumptions	Access to archived X data and trend analysis algorithms is available.
Normal Flow of Events	1. User selects a date range or keyword for historical analysis. 2. System fetches relevant historical tweet data. 3. Analyzes engagement patterns, mentions, and hashtags over time. 4. Displays trends on a timeline or chart, highlighting key events or spikes.
Alternative Flow of Events	1. If historical data is unavailable, display a warning to the user. 2. If analysis fails, display an error and offer retry.
Main Success Scenario	Historical trends are successfully analyzed and visualized, allowing users to understand past patterns and trend evolution.
Postcondition	Users can leverage historical trend insights for research, reporting, or predictive analysis of social media trends.

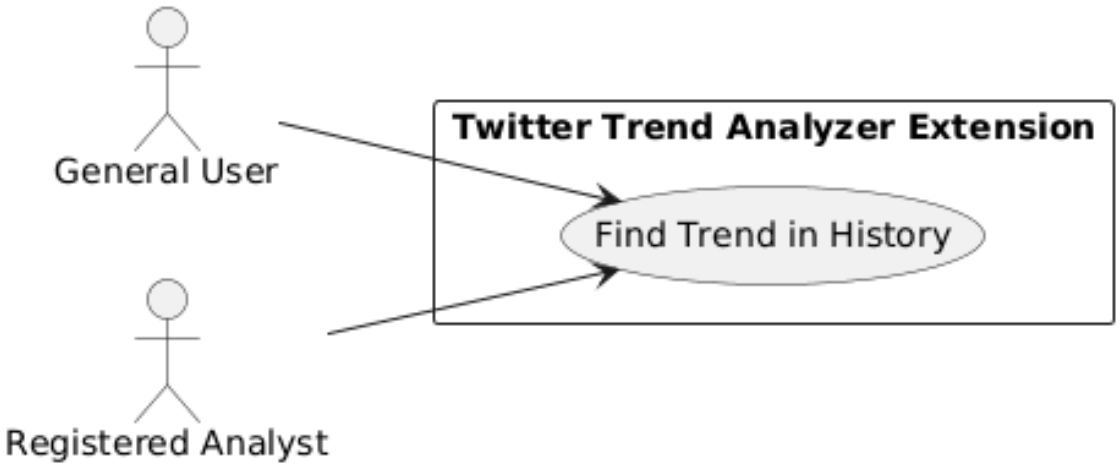

Figure 3.10: Use Case Diagram: Find Trend in History

Chapter 4

System Architecture and Design

4.1 Overview

The development of a smart and trustworthy X trend analyzer demanded a lot of planning, to make sure that every module communicates with the others, data crunching is done effectively and that the feedback presented to the users is clear and easy to act upon. Modularity, to allow independent development and testing of components, maintainability, and the ability to enhance as requirements change are some of the design goals that we have established. We created a system comprising of separate layers (frontend, backend, data storage, and analytics), and specified the interfaces between those layers. This division of responsibilities permits teams to be simultaneously working on UI elements, data ingestion, natural language processing, and visualization without leading to dependencies that slacken down development. Further, the performance needs (caching efficiency and rate-limiting) were included in the design, which ensured responsiveness with hundreds of tweets in a query. Other guiding principles were security and extensibility, by decoupling the user interface and data retrieval and processing logic, new enhancements in the future such as adding new machine-learning models or geographic filters can be implemented with little disturbance. In the process, we had a strong focus on user experience, making sure that the design allows easy workflow, fast insights, and scalable analytics as X volumes of data increase.[8].

4.2 System Flow

In the case when a user clicks on the Chrome extension and wants to start a trend analysis, the frontend receives input parameters (e.g., hashtag, date range, location) and transfers a JSON request to the API of the backend. The backend checks server first to verify if it has already served the results to avoid rate limits and decrease unnecessary processing. In case of the unavailability of the cache, the corresponding tweet entries are read and

analyzed in MongoDB, processed in terms of text (cleaning, tokenization), and fed into the analytics pipeline. This pipeline does engagement analysis (adding likes, retweets, replies) and sentiment classification (using a pretrained NLP model) and influencer ranking (aggregating follower counts, engagement metrics and network centrality) and bot detection (applying heuristic rules). When all modules are finished, they combine their results into one big JSON payload that is stored in a cache (e.g. Redis) to be used again and displayed to the front end. The extension subsequently transforms time-series charts, influencer tables, and lists of flagged-tweets.

4.3 System Architecture

The overall architecture is organized into three layers:

- **Frontend (Chrome Extension):** Built with Next, JavaScript, HTML, and CSS. Takes user input (hashtag, date range, location, advanced filters) and makes interactive charts (D3.js or Chart.js) of sentiment, engagement, and influencer data. Posts to the backend through RESTful HTTPS requests.
- **Backend (API and Processing Engine):**
  - API Controller: Provides a trend analysis and input validation endpoints, rate limiting (specific number of tweets per day), and directs requests to processing modules.
  - Cache Layer: Uses Redis to store the results of a query in the form of a JSON with a key consisting of query parameters (hashtag, date range). Every one of the cache entries has an expiration TTL.
  - Processing Modules:
    * Data Ingestion: Loads packaged tweet data sets of MongoDB, which is filtered by hashtag, and timestamp.
    * Preprocessing: Cleans tweet text (removes URLs, punctuation, stop-words), tokenizes, and normalizes. Optionally filters out non-English tweets.
    * Sentiment Engine: Uses a pretrained transformer or lightweight classifier (e.g., BERT or LSTM) to classify tweets as positive, negative or neutral.
    * Engagement Analyzer: The data will be aggregated into likes, retweets, replies and views per tweet and time bin.

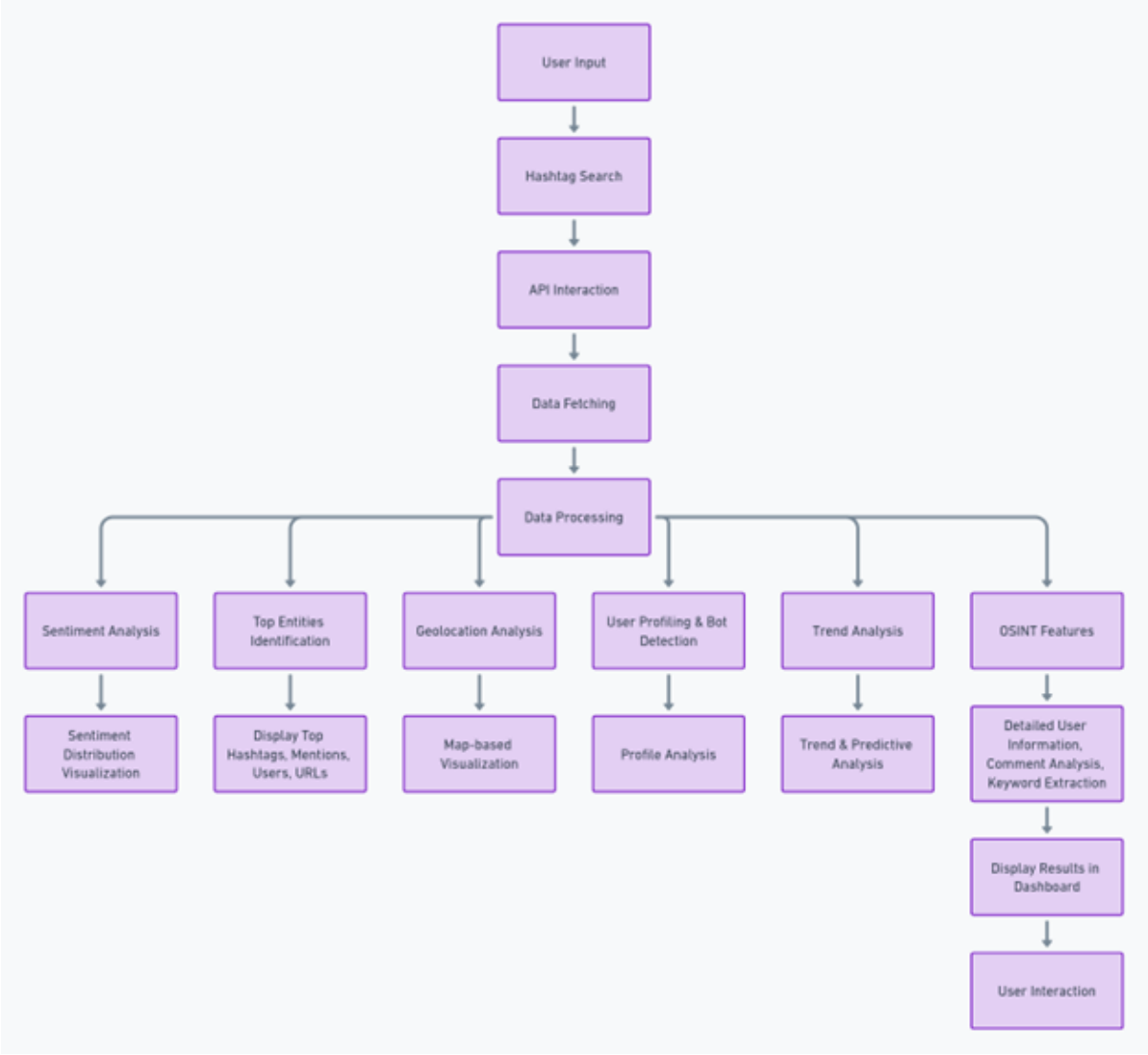

Figure 4.1: System Flow Diagram

    * Influencer Ranker: This section computes a mark of every user depending on elements such as the number of followers, the average amount of engagement (likes, retweets, etc.) of their content, and their verified status. It then categorizes users to come up with the best influencers.

- **Data Layer:**
  - Primary Database (MongoDB): It contains tweets data (text, date, user metadata, likes and retweets) and user profiles (followers, verified status). Sorted by hashtag, time, and location to be easily accessed.
  - Cache (Redis): Caches the aggregated results of repeated queries in the form of JSON, which removes load on the backend.

4.4 Activity Diagram

The activity diagram shows the process of workflow since the user initiation to the ultimate data visualization. The search parameters are given to the extension by the user, which causes the following chain:

1. **User Inputs and Validation:** The user specifies a hashtag or keyword, and selects a date range. If any required field is missing, an error message appears.
2. **Cache Check:** The backend searches the Redis and when a result is found (hashtag, date range), it is expected to be a cached result.
   - When a cached record is discovered, then it is fetched and sent to the front end to be displayed instantly.
   - Otherwise, it flows to the step of data retrieval.
3. **Data Retrieval and Preprocessing (Concurrent Activities):**
   - Fetch Tweets: Query MongoDB for all tweets matching the criteria.
   - Clean and Tokenize: In parallel, run text-cleaning routines (URL removal, lowercase, stopword filtering) and tokenize each tweet.
4. **Analytics Pipeline (Sequential or Pipelined):**
   - Sentiment Classification: Every preprocessed tweet is categorized as positive, negative or neutral.

- Retweets, Likes, replies of a tweet and time bin.
- Influencer Ranking: Calculate the influence score of users with the help of the number of followers, verified status, and engagement.
- Bot Flags: Provide the bot probability each tweet.

5. **Result Aggregation and Caching:** The results of all modules are combined to a structure of JSON and placed into Redis with a TTL to speed up the similar requests.
6. **Response and Visualization:** The extension is provided with the JSON response, and displays charts (time-series sentiment, engagement heatmap), tables (top influencers, flagged tweets), and maps (where available, geolocation).
7. **User Interaction and Refinement:** When the user refines filters or clicks on a particular piece of data (e.g. a timestamp or row of influencers), the extension will either update the view on the device or make another request in case there is a change in parameters.

4.4.1 Context Diagram (Level 0 DFD)

The context diagram (Level 0 Data Flow Diagram) shows all external entities and the primary data flows between them and the OSINT X Trend Analyzer Extension.

4.5 Block Diagram

The block diagram abstracts the system into logical components (blocks) and shows the data flow between them:

- **Chrome Extension (Frontend)**
  - UI Components: Form fields for hashtag, date range, location, advanced filters; placeholder panels for charts and tables.
  - API Client: Handles all RESTful calls, including input validation on the client side and JSON parsing.
- **Backend (API Server)**
  - Request Router: Directs incoming HTTP requests to the appropriate controller endpoint (e.g., analysis endpoint).
  - Cache Layer: Interfaces with Redis to retrieve or store analysis results.

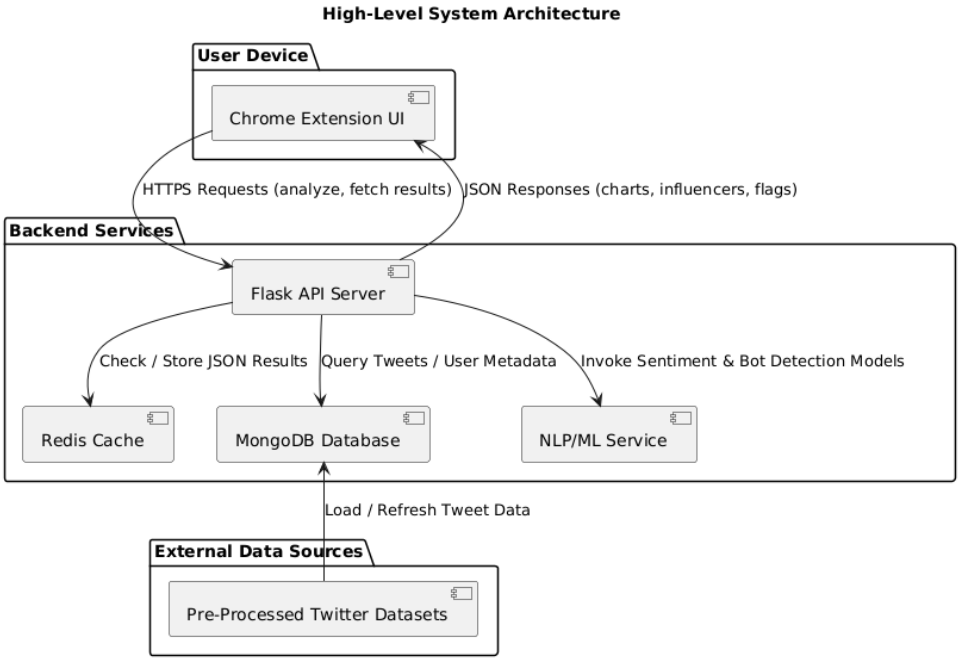

Figure 4.2: High-Level System Architecture

figures/activity_flowchart.png

Figure 4.3: System Processing Flowchart

  - Controller Logic: Orchestrates calls to each processing block and composes the final JSON response.

- **Processing Modules**
  - Data Ingestion: Fetches and filters tweets from MongoDB.
  - Preprocessing: Cleans and tokenizes tweet text.
  - Sentiment Engine: Classifies tweet sentiment using a pretrained model.
  - Engagement Analyzer: Sums up likes, retweets and replies.
  - Influencer Ranker: Calculates influence scores and ranks the users. [7].
  - Bot Detector: Marks as suspicious accounts and tweets.
  - Aggregator: Consolidates the outputs into a single JSON.
- **Data Layer**
  - Database (MongoDB): Tweets, user profiles, history of engagement.
  - Cache (Redis):It is a storage of recently accessed JSON responses.
  - External Services (Optional): Fact-checking APIs, geo-lookup Services.

4.6 Entity Relationship (ER) Diagram

The Entity Relationship (ER) diagram describes the logical structure of the database, showing key entities such as User, Tweet, Trend, and their relationships. This helps in understanding how data is stored, accessed, and related across the system.

4.7 Sequence Diagram

This sequence diagram presents a runtime scenario for the “Analyze Hashtag” operation, showing interactions over time:

4.8 GUI Design

The Chrome extension has a GUI that is structured into a header, a sidebar which is the filter, a main body with analytics panels. All of the components are conducive to natural navigation and understandable display of findings.

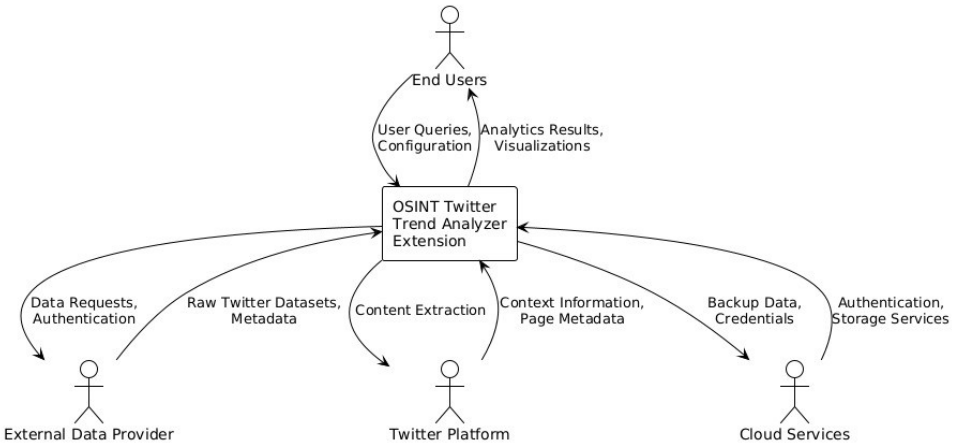

Figure 4.4: Level 0 Data Flow Diagram (Context Diagram)

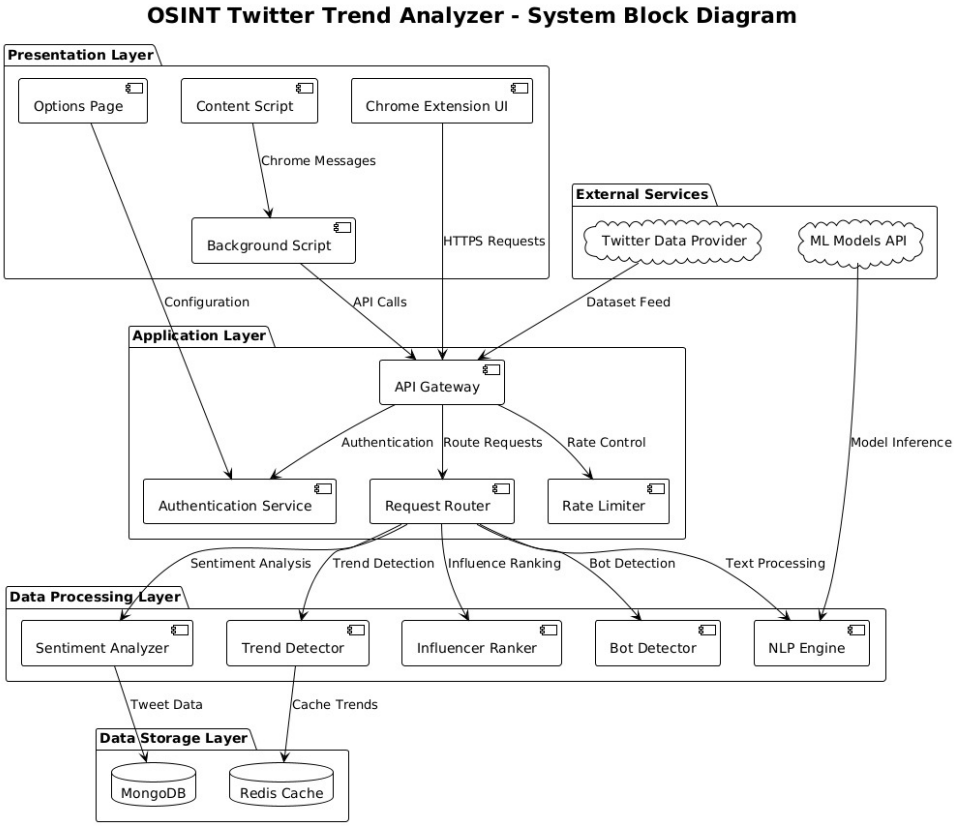

Figure 4.5: Block Diagram of System Components

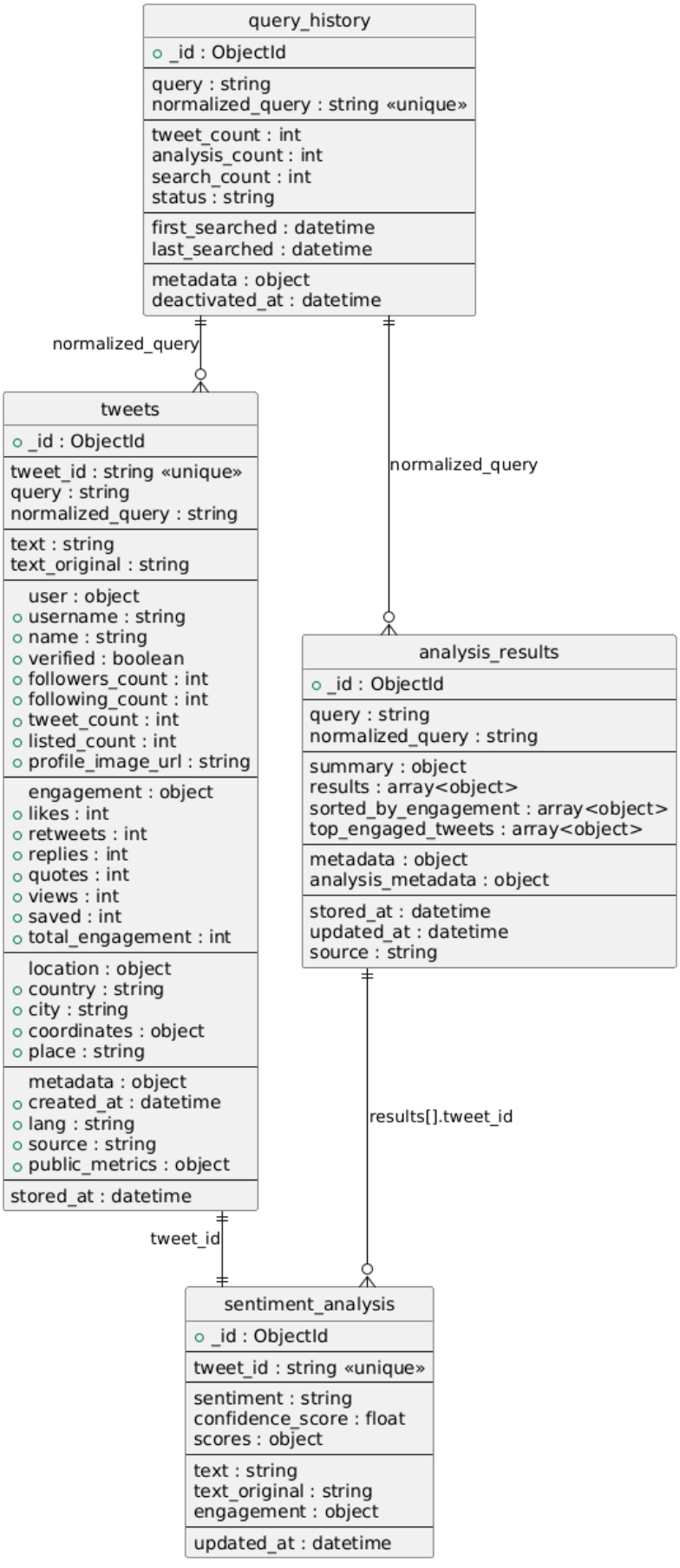

Figure 4.6: ER Diagram

4.8.1 GUI Overview

- **Header Section:**
  - Displays the extension logo and title (“OSINT X Trend Analyzer”).
  - Includes a settings icon () for preferences (theme toggle, API key info, rate-limit status).
- **Filter Panel:**
  - Hashtag / Keyword Input: A text field with placeholder text “Enter hashtag or keyword” and autocomplete suggestions based on recent searches.
  - Analyze Button: A prominent button labeled “Analyze,” disabled until valid inputs are provided.
- **Main Content Area:**
  - Real-Time Analytics Header: Shows the current analysis parameters (e.g., MyTopic with date range).
  - Sentiment Chart: This chart shows the proportion of positive, negative, and neutral tweets
  - Engagement Metrics Overview: Statistic cards on the top indicating the number of tweets analyzed, the total number of likes, the total number of retweets, and the total number of replies analyzed during the chosen period.
  - Influencer Ranking Table: This is a sortable table that contains the top 10 influencers, where the table includes columns like username, number of followers, average interaction per tweet, influence score and a verified-status icon. All the rows can be clicked to display additional information in a modal (sample tweets, follower/following ratios).
  - Bot Flags: Two side-by-side panels:
    - Bot Likelihood Distribution: Flags showing the proportion of tweets classified as low, medium, or high bot likelihood.

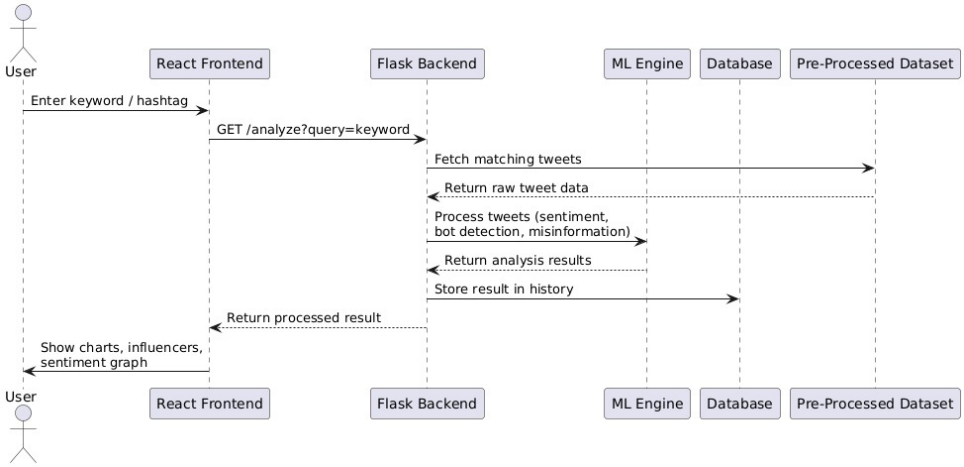

Figure 4.7: Sequence Diagram for “Analyze Hashtag” Workflow

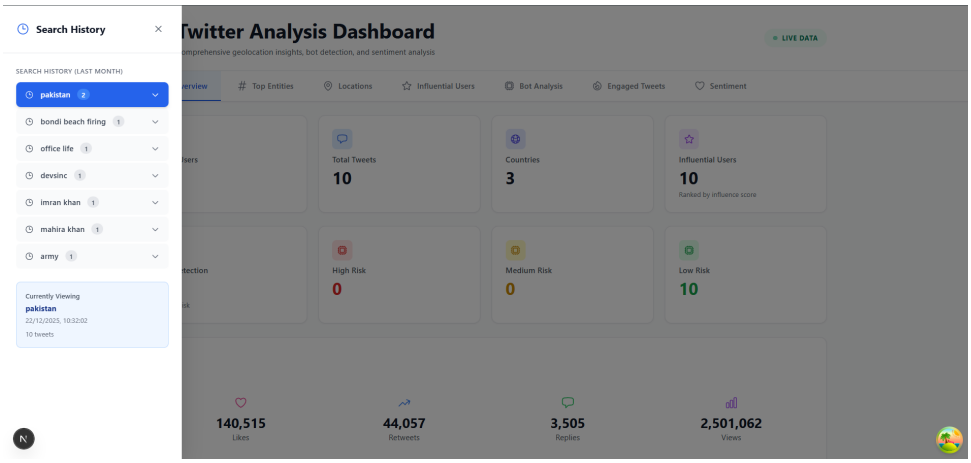

Figure 4.8: GUI Mockup – Left hand sidebar

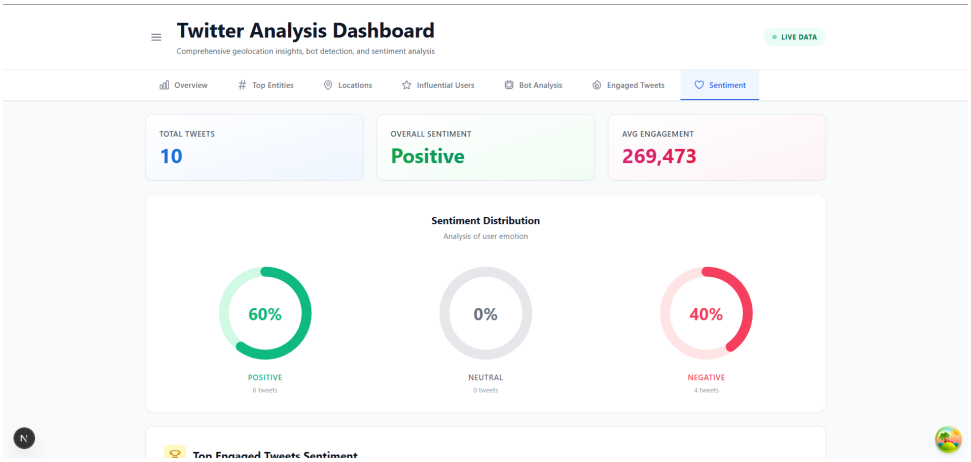

Figure 4.9: GUI Mockup – Sentiment Analysis

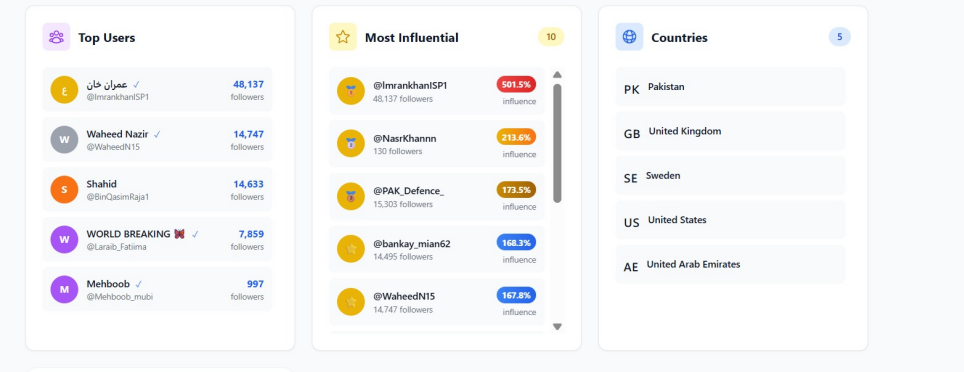

Figure 4.10: GUI Mockup – Top Users

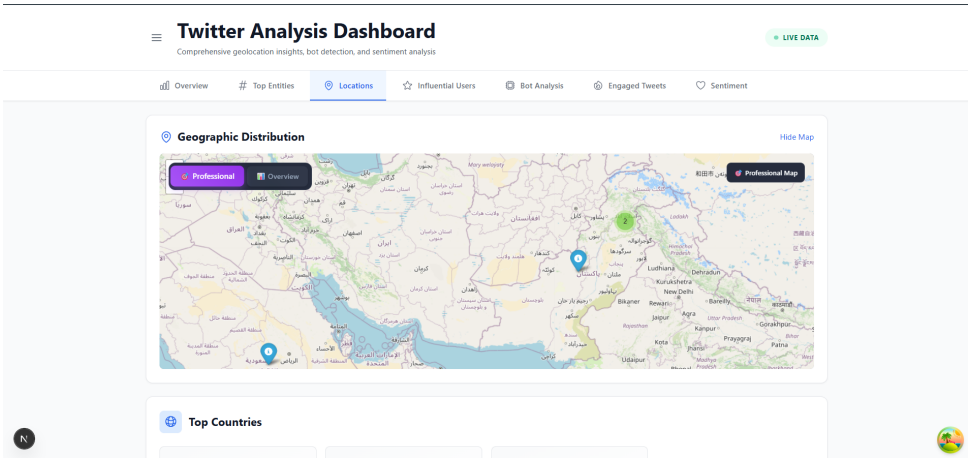

Figure 4.11: GUI Mockup – Location Analysis

4.8.2 Interactive Behavior

Interactive Data Visualization: EchoTrace Twitter Trend Analyzer offers interactive visual components that allow users to observe Twitter trends, sentiment distribution, and engagement metrics. The displayed charts update dynamically based on the selected trend or dataset, helping users understand overall patterns effectively.

Loading States: As long as backend processing occurs, individual analytics panel displays a spinning wheel that displays the Loading... please wait. In case the number of tweets per day is met, a banner is displayed on the top: “Daily tweet limit met. Please try again tomorrow.”

Chapter 5

System Implementation

This chapter provides a comprehensive account of the implementation of the Echo Trace: OSINT-Based X Trend Analyzer Extension. The system has been developed with a modular, scalable architecture to achieve effective real-time social media intelligent collection and analysis. It describes the architecture of the system, the tools and technologies used, database management including the integration of MongoDB and Redis, the sentiment analysis and identification of the influencers process, the mechanism of misinformation identification, and strategies of deployment with constant integration pipelines. [5, 9]

5.1 System Architecture

The OSINT-Based Twitter Trend Analyzer Extension has a layered design that ensures that different elements of the system can easily communicate. The structure is structured into three major layers which include; Presentation Layer, Application Layer and Data Layer. Each layer is designed in the strictest way so as to make it efficient, scalable, and maintainable. This multifaceted strategy goes along with the accepted best practices of browser-based analytics systems and social media intelligence applications.

The **Presentation Layer** is made up of the Chrome browser extension developed with the help of Next.js, JavaScript, HTML, and CSS. This extension is the main point of interaction with users with the system. Analysts have the opportunity to search trending topics, filter hashtags, real-time analytics, and sentiment distributions. The user interface is responsive and intuitive and can be easily accessed on different devices and screen sizes. Among the design features, one should not exclude interactive visualization based on D3.js and Chart.js, collapsible filtering panels with

advanced settings, and the clarity of various complex analytics displayed in cards, tables, and graphs.

The **Application Layer** is the processing environment of the OSINT Twitter Trend Analyzer. This layer is implemented in Python and Flask framework and oversees many back-end functions such as coordination of data retrieval, preprocessing and tokenization of text, classification of sentiments with pre-trained NLPs, aggregation of engagement metrics and ranking of influencers. Application Layer also has an efficient caching system using Redis to improve performance and minimize unnecessary processing. To stay within the limits of data usage policy and system resource limits, it implements rate limiting of specific number of tweets per trending topic per day. Also, this layer contains API request route and input validation, orchestration of analytics and the composition of JSON responses modules.

The **Data Layer** handles all forms of data storage,retrieval and caching. MongoDB is the single NoSQL database, which handles structured collections such as documents on tweets containing text information, timestamps, user metadata, and engagement metrics. The profiles of users with follower numbers, verified status, and the influence scores are also saved in this place. The Redis is a high-performance caching layer that is used to store aggregated JSON data based on query parameters like hashtag and date range. In each entry in the cache, there is a configurable Time-To-Live (TTL) between freshness and performance. A combination of these services provides real-time analytics needs, fast, scalable, and secure data operations.

The overall workflow begins when a user inputs search parameters into the interface of the Chrome extension. These parameters go through the Application Layer whereby the system first checks Redis cache to find out the already calculated results. In the event that no stored information is provided, the backend makes an inquiry with MongoDB to get the information in the form of tweets data sorted on hashtag, date, and location. The recovered data are preprocessed through text cleaning, tokenization and normalization. Then the data is processed with several analytics modules simultaneously: sentiment analysis determines the level of emotion polarity, engagement data determines metrics of interaction, key opinion leaders are identified in the influencer ranking, and suspicious content is marked with misinformation. After all modules finish their processing, the outcomes are combined into a single payload in the form of a JSON, which is stored in Redis memory to be used later and sent to the frontend where interactive visualization takes place.

5.2 Tools and Technologies Used

The development of the OSINT-Based X Trend Analyzer Extension required the integration of several advanced tools and technologies, as they are reliable, high-performing, and fit the social media analytics application. This stack of technology is an indication of what is happening in the industry when it comes to browser-based intelligence collection systems. [**?**].

next.js/JavaScript: The Chrome extension frontend is written with next.js, a component based architecture, and thus, ensures maintainability and reusability of UI elements. JavaScript processing deals with customer side logic, API interaction, and dynamic content generation. The virtual DOM allows efficient updates due to arrival of new analytics data and ensures the seamless user experience despite the intricate visualizations.

Python/Flask: The backend services are implemented in the Python 3.10+ language and Flask framework. The flexibility and light weight of Flask will be suitable in developing RESTful APIs to serve the requests of trend analysis. The NLP models, statistical analysis and network algorithms can easily be integrated to Python due to its rich ecosystem of data processing and machine learning libraries.

MongoDB Atlas: This is the primary NoSQL database that is used to save a collection of tweets and user profile documents. The schema of MongoDB is flexible and can support different metadata of tweets. It structures, its indexing and query features on hashtag, timestamp and fields are very fast. Atlas service on the cloud has automatic scaling, backup and monitoring facility which is required in production deployment.

Natural Language Processing Libraries: The sentiment analysis engine makes use of pre-trained transformer models like BERT (Bidirectional Encoder Representations from Trans- formers) or social media-screen optimized lightweight models. These models are combined using Python libraries such as Hugging Face Transformers or TensorFlow to correctly classify the sentiment of the tweets as a positive, negative or neutral tweet with a confidence score.

D3.js/Chart.js: This offers extremely good data visualization directly inside the extension in Chrome. D3.js supports more complex custom visualizations such as time-series sentiment trends and network influence graphs whereas Chart.js displays the standard charts such as bar graphs to represent the bot likelihood distribution and pie charts to display the engagement metrics. Both libraries provide support of the interactive features such as the tooltips, drill-down and dynamic filtering.

RESTful API Architecture: The communication between the back-end services and the Chrome extension is done in accordance with the principles of the REST

and is built on the basis of the HTTP methods (GET, POST) and the JSON data interchange format. The API endpoints are encrypted with the HTTPS/TLS.

5.3 Database Management and Integration

The OSINT X Trend Analyzer uses database management as the means of processing and analyzing a large amount of data on social media in an efficient way. It uses a dual database approach which uses MongoDB as a persistent storage and Redis as a high performance caching system to guarantee that the data is durable and at the same time the response time to the query is very high. MongoDB manages the data as clear collections that reflect the system analysis needs. Tweets collection holds records of each specific tweet, including the content of tweet text, date and time of publication, user identifier, user name, verified or not, number of followers, a record of engagement, likes, retweets, replies, views, an array of hashtags, and optional geolocation. Hashtag, timestamp indexing are extensively indexed to allow effective filtering of the data retrieval operations. The Users collection stores aggregated profile data on accounts that are found within analyzed datasets, such as the number of followers, the number of people followed, verified status, average engagement rate, the influence score, and the estimation of the likelihood of being a bot. The Trend History collection is used to store previous analysis outcomes, allowing the comparison of the results in the past and tracking changes in the trends over long durations of time. The document-oriented design of MongoDB is suitable in the case of Xr data because not all tweets have the same metadata elements. The loose schema enables the system to add more attributes when it is available without database migrations. MongoDB Atlas offers cloud deployment which offers automatic sharding as a method of horizontal scale, replica sets as a form of high availability and point-in-time backup of data as a means of data protection. Inbuilt monitoring displays follow. database query performance and storage usage, and connection pooling parameters, so that database activity can be preemptively optimized.

5.4 Sentiment Analysis and Influencer identification Modules

5.4.1 Sentiment Analysis Engine

The sentiment analysis engine applies sophisticated NLP methods to categorize tweets into positive, negative or neutral. Every tweet has been processed (URLs, mentions, and special characters are removed, emojis are converted to text) and then a transformer-based model that has been trained on social media text is used to analyze

it. The sentiment labels developed in the model have confidence values ranging between 0 and 1. Tweets are batch processed to be efficient and the outcomes are also aggregated to display distributions of sentiment, trends of sentiment and correlation to engagement measurements. The frontend is a visualization that enables insight by displaying charts interactively overtime a time-series graph of sentiment and a pie chart of portion that allows users to get a glimpse of the tweet-level information. These visualizations can assist journalists, analysts and brand managers to monitor the real-time opinion and reaction of the population.

5.4.2 Influencer Identification and Ranking

The influencer detector module detects and sorts users that cause significant responses in a discussion.Instead of relying solely on follower counts, it uses a multi-factor scoring algorithm that considers follower base, engagement rate, retweet amplification, verified status, and network centrality to measure true influence. The system uses these weighted factors to compute a normalized influence score to each user and ranks them in descending order. The frontend presents the results in a table that can be sorted in an interactive manner and containing key metrics users, profile picture, number of followers, engagement rates, influence score and being verified. The user has access to in-depth information on the insights, such as sample X posts, engagement, and direct links. This aspect assists journalists, marketers, and analysts in realizing key opinion leaders and the pattern of influence that traverses across social networks.

5.5 Bot Activity Identification

The OSINT X Trend Analyzer considers the implementation of the sophisticated mechanisms that will identify bot-like behavior and provide data that is considered authentic. The bot identification module involves heuristic and statistical analysis to identify automated accounts with regard to the frequency of activity, repetitive content, regularity proposed, and incomplete profile. The bot likelihood score is provided and each entry is classified as low, medium, or high and aggregate statistics informs about the general prevalence of suspected bots in the data. These findings are also displayed through interactive visualizations in the frontend. A combination of all these tools enables journalists, analysts, and other researchers to check the authenticity, identify coordinated actions, and concentrate on actual public discourse.

5.6 Deployment and Integration

The X Trend Analyzer Extension OSINT is currently in use as a locally executing Chrome extension, which is easy to install and operate off-line without external cloud-based infrastructure. The frontend (Chrome extension) and the backend (Flask API) are decoupled in the system, which means that the frontend and the backend may be developed and tested separately.

The backend is also implemented locally with a Python Flask environment. It will be connected to MongoDB to store the data and to Redis to cache this information if it has access to it, and a virtual environment will be used to handle all the dependencies. The Flask server reveals RESTful endpoints that receive trend analysis requests of the extension and return processed results.

The frontend of the Chrome extension is developed with next, JavaScript, HTML, and CSS and packaged together to include all the required files into the package that includes the manifest, popup interface, and dashboard pages. It connects to the Flask backend, which is locally running directly by using secure HTTP requests. In developer mode, the extension can be loaded by adding the option of Chrome which is the Load unpacked and choosing the project directory.

The project is not implemented on remote servers, however, the codebase is done in a way that best practices are considered to scale it and deploy it to remote servers in the future. The architecture also allows containerization with the help of Docker and version control with the help of GitHub so that it can be potentially integrated with CI/CD pipelines in the future production environment. Such an arrangement provides the system with the capability to be considered as modular, maintainable, and prepared to upgrade or even migrate to cloud environments in future in case of need.

Chapter 6

System Testing and Evaluation

This chapter will provide the description of various kinds of tests required to make the Echo Trace: OSINT-Based X Trend Analyzer Extension credible and of good quality.

6.1 Test Approach

A test approach refers to the implementation of the test strategy of a project and it is how testing would have been carried out. The OSINT X Trend Analyzer testing strategy was modeled to ensure that each and every module of the system is effectively checked and the non-functional and functional requirements are taken into consideration. To test the different layers of the system, including the next.js Chrome extension frontend, Python Flask backend, MongoDB database operations and Redis caching mechanisms, both manual testing and automated testing were performed.The manual testing was useful in testing the user interface, browser responsiveness, trend analysis workflow, sentiment visualization and influencer identification display. This helped in probing usability issues, working problems, and accuracy of analysis during live user systems. Auto-tests were written to the backend to verify the API functionality application which was written on Flask and database operations are managed with MongoDB Atlas. The tests ensured that the data retrieval, tweet preprocessing, sentiment classification, influencer ranking, and bot identification worked as expected in different input conditions.

Integration testing was done to ensure communication was done between the Chrome extension, MongoDB database, Redis cache, and NLP sentiment analysis modules as well. All in all, the test strategy gave us stability, performance, and accuracy in the codes, as well as, in the process of testing, it modeled the real user behaviors, to find out the edge cases as soon as possible during the development process.

6.1.1 Test Plan

A test plan is simply a summary of how an application will be tested, the resources that would be used, and the testing environment within which testing will occur and the time that will be spent in testing.

Features To be Tested

- Dashboard Navigation
- Hashtag Search and Analysis
- Sentiment Analysis Classification
- Influencer Identification
- Bot Activity Identification
- Engagement Metrics Visualization
- Time-Series Trend Graphs
- Advanced Filter Options
- API Response Time

6.1.2 Testing tools and Requirements

Following tools and environments are used for testing:

- Chrome DevTools – For UI testing and extension debugging.
- Postman – For API testing (Flask REST endpoints).
- MongoDB Compass – For tracking database queries, collections, and document updates.
- Redis CLI – For monitoring cache operations and hit/miss rates.
- Pytest – For backend unit and integration testing.
- Jest – For frontend component testing.
- Chrome Browser (Multiple Versions) – For cross-version compatibility testing.

6.1.3 Test Cases

The system testing is recommended whenever new deliverables and products can be used. A test case is a combination of an input, activity or incident with a reaction to verify a feature of a software application is functioning as anticipated. Test case may include such information as test case id, test case name, purpose, test data, steps and expected results. The possible test cases which may be used are:

This test case validates UI responsiveness and accessibility. It ensures users can seamlessly switch between different analytical modules of the extension.

This test case validates the core functionality of trend search and data retrieval from external Twitter data providers stored in MongoDB.

This test case ensures that the pretrained NLP model (BERT-based) accurately classifies tweet sentiment with acceptable accuracy rates.

This test case validates the multi-dimensional influencer scoring algorithm combining follower count, engagement rate, and verified status.

This test case validates heuristic-based bot detection and misinformation scoring algorithms for identifying suspicious content.

This test case ensures that D3.js and Chart.js visualizations render engagement metrics accurately and responsively.

This test case ensures temporal sentiment analysis is visualized effectively for tracking opinion shifts over time.

This test case validates drill-down functionality for exploring individual user profiles in detail.

This test case validates the effectiveness of Redis caching in improving system performance and user experience.

This test case validates that backend API performance meets responsiveness requirements for real-time analytics.

6.2 Summary

The testing and evaluation of OSINT-Based X Trend Analyzer Extension are justified by the fact that the system is fulfilling the requirements, both functional and non-functional. Key modules such as hashtag search, sentiment analysis, influencer identification, bot identification, geolocation analysis and data visualization were reliable in various test conditions.

Table 6.1: Dashboard Navigation

Test Case Name	Dashboard Navigation
Test Case ID	TC-01
Actor	Analyst/Researcher
Purpose	Ensure that users can navigate through different analytical modules seamlessly.
Setup	Logged-in user on main dashboard.
Instructions	– Access filter panel for hashtag search. – Navigate to sentiment analysis section. – View influencer ranking tables. – Check misinformation detection panels. – Access account settings.
Inputs	User navigation actions and clicks.
Expected Result	Smooth navigation without crashes or loading errors.
Observed Result	Navigation worked smoothly across all modules.
Verdict	Pass

Table 6.2: Hashtag Search and Analysis

Test Case Name	Hashtag Search and Analysis
Test Case ID	TC-02
Actor	Analyst/Researcher
Purpose	Ensure users can search trending hashtags and initiate analysis.
Setup	Logged-in user on dashboard.
Instructions	– Enter hashtag in search field (e.g., #ClimateChange). – Select date range (past 24 hours, 7 days, etc.). – Choose location filter if applicable. – Click "Analyze" button.
Inputs	Valid hashtag, date range, location.
Expected Result	System retrieves up to 800 tweets and displays analysis results.
Observed Result	Hashtag analysis initiated successfully with data retrieval from MongoDB.
Verdict	Pass

Table 6.3: Sentiment Analysis Classification

Test Case Name	Sentiment Analysis Classification
Test Case ID	TC-03
Actor	System / Analyst
Purpose	Ensure tweet sentiment is classified accurately as positive, negative, or neutral.
Setup	Tweets retrieved for analyzed hashtag.
Instructions	– System processes tweet text through NLP model. – Sentiment classification applied to each tweet. – Results aggregated and displayed.
Inputs	Tweet text content from retrieved dataset.
Expected Result	Each tweet labeled with sentiment and confidence score. Overall sentiment distribution displayed in pie chart.
Observed Result	Sentiment classification achieved approximately 82% accuracy against test dataset.
Verdict	Pass

Table 6.4: Influencer Detection and Ranking

Test Case Name	Influencer Detection and Ranking
Test Case ID	TC-04
Actor	System / Analyst
Purpose	Ensure influential Twitter users are identified and ranked based on multi-dimensional scoring.
Setup	Tweet dataset with user metadata retrieved.
Instructions	– System calculates engagement rates per user. – Follower count, retweet metrics analyzed. – Influence score computed using algorithm. – Top influencers ranked and displayed.
Inputs	User engagement data, follower counts, verified status.
Expected Result	Top 10 influencers displayed with scores in sortable table.
Observed Result	Influencer ranking displayed accurately with detailed metrics.
Verdict	Pass

Table 6.5: Bot identification

Test Case Name	Misinformation and Bot Detection
Test Case ID	TC-05
Actor	System / Analyst
Purpose	Ensure suspicious accounts and potential misinformation are flagged.
Setup	Tweet dataset analyzed for anomalies.
Instructions	– System analyzes posting patterns and account behavior. – Bot likelihood score calculated for each user. – Misinformation probability assigned to tweets. – Flagged content displayed in dedicated panel.
Inputs	Tweet metadata, account age, posting frequency, content patterns.
Expected Result	Bot likelihood distribution shown. Top suspicious tweets flagged with scores.
Observed Result	Bot detection achieved 72% precision and misinformation flags displayed appropriately.
Verdict	Pass

Table 6.6: Engagement Metrics Visualization

Test Case Name	Engagement Metrics Visualization
Test Case ID	TC-06
Actor	Analyst/Researcher
Purpose	Ensure engagement metrics are visualized clearly through interactive charts.
Setup	Analysis completed for a hashtag.
Instructions	– View total likes, retweets, replies displayed as cards. – Examine bar charts for engagement distribution. – Interact with charts using hover tooltips.
Inputs	Aggregated engagement data from analyzed tweets.
Expected Result	Clear visualization of metrics with interactive elements.
Observed Result	Charts rendered correctly using Chart.js with smooth interactions.
Verdict	Pass

Table 6.7: Time-Series Trend Graphs

Test Case Name	Time-Series Trend Graphs
Test Case ID	TC-11
Actor	Analyst/Researcher
Purpose	Validate sentiment evolution is displayed over time with line graphs.
Setup	User analyzed hashtag with date range spanning multiple days.
Instructions	– View time-series sentiment graph. – Observe positive, negative, neutral trends over time bins.
Inputs	Sentiment data aggregated per hour or day.
Expected Result	Line graph shows sentiment evolution with color-coded trends.
Observed Result	Time-series graph rendered correctly showing temporal patterns.
Verdict	Pass

Table 6.8: User Profile Details Display

Test Case Name	User Profile Details Display
Test Case ID	TC-07
Actor	Analyst/Researcher
Purpose	Ensure detailed Twitter user profiles are displayed when clicking influencers.
Setup	Influencer table populated with ranked users.
Instructions	– Click on an influencer username in table. – View modal with detailed profile information.
Inputs	User metadata from Twitter dataset.
Expected Result	Modal displays follower count, engagement stats, sample tweets, verified status.
Observed Result	Profile details displayed comprehensively in modal dialog.
Verdict	Pass

Table 6.9: Cache Performance and Hit Rates

Test Case Name	Cache Performance and Hit Rates
Test Case ID	TC-08
Actor	System
Purpose	Validate Redis caching improves response times for repeated queries.
Setup	Backend connected to Redis instance.
Instructions	– Submit analysis request for hashtag. – Submit identical request again immediately. – Measure response time difference.
Inputs	Identical hashtag queries.
Expected Result	Second request returns cached result in under 250ms. Cache hit rate exceeds 60% in production.
Observed Result	Cache hit reduced response time from 5 seconds to 200ms. Hit rates measured at 62%.
Verdict	Pass

Table 6.10: API Response Time

Test Case Name	API Response Time
Test Case ID	TC-09
Actor	System
Purpose	Ensure Flask API endpoints respond within acceptable time thresholds.
Setup	Backend server running, MongoDB and Redis accessible.
Instructions	– Submit analysis request via Postman. – Measure end-to-end response time. – Test with 800 tweet dataset (maximum daily limit).
Inputs	API request with hashtag, date range, location.
Expected Result	Non-cached analysis completes in 4-6 seconds for 800 tweets.
Observed Result	Average response time measured at 5.2 seconds for full analysis.
Verdict	Pass

In terms of security, the architecture adheres to widely accepted best practices for web and extension development, reducing exposure to common vulnerabilities such as injection attacks, and cross-site scripting (XSS). These considerations contribute to a more robust and reliable platform for handling publicly available social media data.

Overall, the assessment highlights the extension's potential as an effective social media intelligence tool. It demonstrates the ability to provide accessible and meaningful OSINT-based X trend analysis through a user-friendly, browser-based interface. By simplifying advanced analytical capabilities while maintaining reliability and accuracy, the system shows strong suitability for professional and academic use in areas such as journalism, digital marketing, and politics.

Chapter 7

Conclusions

The OSINT-Based X Trend Analyzer Extension is a powerful, easy-to-use system with a platform that allows one to analyze twitter data utilizing the Open-Source Intelligence (OSINT) method. It allows journalists, researchers, and analysts to access the discussions of the people, monitor the trending hashtags, evaluate the sentiment, and find the users with influence right in their browsers. The Chrome extension architecture is accessible and lightweight in nature, and does not need any outside deployment. The system is capable of converting raw X information into actionable insights by using Natural language processing (NLP), data visualization, and engagement analytics. The project is based on the next frontend and Python Flask backend, which makes it modular and scalable. MongoDB is the main database that gives storage and retrieval of the tweets, whereas Redis caching is optional and improves access to the data. With this hierarchical structure, the system is able to process and visualize more complex datasets efficiently and display trends, sentiments, influencers, and patterns of bot acitivites in a user-friendly interface. This renders the OSINT X Trend Analyzer a robust, privacy-conscious, and analytical software of for understanding the social dynamics on X.

7.1 Key Learnings

- **System Integration and Architecture:** Obtained practical experience in designing the modular system, which is a combination of a browser-based frontend and a Flask-based backend to perform data analytics.
- **Natural Language Processing and Data Analysis:** Learned how to process the text of tweets, apply text sentiment prediction with the help of models, and understand the patterns in public sentiment over time.

- **Database Management:** understood how to store, access and manipulate unstructured X datasets in MongoDB so as to enable efficient querying and scalability.
- **Data Visualization:** Implemented interactive dashboard with Chart.js/D3.js to display sentiment trends, engagement metrics, and rankings of the influencers in the most effective way.
- **Browser Extension Development:** Learned how to create and develop a Chrome extension with next, cope with permissions with manifest files, and establish a safe connection with back-end API.

7.2 Project Achievements

- **Hashtag Trend Analysis:** Created functionality to inspect hashtags, identify relevant tweets and summarize current trends of the population.
- **Sentiment Analysis Engine:** Built a sentiment analyzer which is an NLP-based sentiment analyzer and can classify the tweets as positive, negative, or neutral with contextual accuracy.
- **Influencer Identification:** Developed an influence-scoring system, which was founded on the engage- ment rate, the number of followers, and activity frequency, as a means of determining the key opinion leaders.
- **Bot Identification:** Implemented modules that indicate suspicious tweets and accounts that follow certain patterns of automated/coordinated behavior.
- **Interactive Visualization Dashboard:** Developed a very user-friendly, interactive dashboard displaying insights in the form of sentiment graphs, influencer table and flagged content panels in the Chrome extension.

7.3 Future Developments

OSINT-Based X Trend Analyzer Extension can be improved further in the future by:

- Deployment of backend to a cloud infrastructure that will be used to access remotely and support multi-users.
- Integration of live Twitter API streams for real-time trend tracking and sentiment monitoring.

- Implementation of multilingual sentiment analysis to accommodate various global customers.
- Integration of user-profile analysis module.
- Addition of AI models for implementing misinformation with bot activity as well.
- Development of a Firefox and Edge-compatible version of the extension for wider accessibility.

Appendix A

User Manual

A.1 Introduction

This user manual provides a step-by-step guide for using the **OSINT-Based X Trend Analyzer Extension**. The extension is designed to help users analyze real-time and historical trends on X by leveraging open-source intelligence techniques. It enables users to explore sentiment, engagement, influencers, bot activity, and geolocation patterns through an intuitive browser-based interface.

This manual is intended for a wide range of users including journalists, researchers, political analysts, marketers, cybersecurity professionals, and general users with no advanced technical background.

A.2 System Requirements

To use the OSINT-Based X Trend Analyzer Extension effectively, the following requirements must be met:

- Google Chrome browser (latest stable version recommended)
- Stable internet connection
- Desktop or laptop device (tablet support may be limited)
- No programming or data analysis knowledge required

A.3 Installation Guide

Follow the steps below to install the extension:

1. Open the Google Chrome browser.
2. Navigate to the Chrome Extensions page.
3. Enable **Developer Mode** (top-right corner).
4. Click on **Load unpacked**.
5. Select the folder containing the OSINT X Trend Analyzer Extension.
6. The extension icon will appear in the browser toolbar.

Once installed, the extension is ready for use.

A.4 Launching the Extension

1. Click on the extension icon in the Chrome toolbar.
2. The main dashboard interface will open in a popup window.
3. The user will be presented with input fields to begin analysis.

A.5 User Interface Overview

The extension interface is divided into the following main sections:

A.5.1 Header Section

- Displays the application title and logo.
- Shows system status and notifications.

A.5.2 Filter Panel

- Hashtag or keyword input field.
- Optional filters such as date range and location.
- Analyze button to start the analysis.

A.5.3 Main Analytics Panel

- Sentiment analysis charts.
- Engagement statistics.
- Influencer ranking tables.
- Bot activity indicators.
- Geolocation maps (if data is available).

A.6 Performing a Trend Analysis

To analyze a trend:

1. Enter a hashtag or keyword in the input field.
2. Select optional filters such as date range or location.
3. Click the **Analyze** button.
4. The system will fetch and process relevant X data.
5. Results will be displayed on the dashboard.

A.7 Understanding Sentiment Analysis

The sentiment analysis feature classifies tweets into three categories:

- **Positive**: Indicates favorable or supportive sentiment.
- **Negative**: Indicates criticism or negative opinion.
- **Neutral**: Indicates factual or emotionally neutral content.

Sentiment results are visualized using charts that show distribution and trends over time.

A.8 Influencer Identification

The influencer ranking feature identifies users who significantly impact discussions around a trend. Influencers are ranked based on:

- Number of followers
- Engagement metrics (likes, retweets, replies)
- Verified account status

Users can click on an influencer to view additional details.

A.9 Bot Activity Detection

The bot detection module analyzes tweet behavior patterns to flag potential automated or coordinated activity. Indicators include:

- Unusual posting frequency
- Repetitive content patterns
- Abnormal engagement behavior

Flagged tweets or users are visually marked to alert the user.

A.10 Engagement Metrics Visualization

The engagement section provides insights into how users interact with a trend:

- Total number of tweets analyzed
- Aggregate likes, retweets, and replies
- Time-based engagement graphs

These metrics help users understand the reach and popularity of a topic.

A.11 Geolocation Analysis

When location data is available, the extension displays geographic distribution of tweets. This feature helps identify:

- Regions with high activity
- Geographic spread of trends
- Regional sentiment variations

A.12 Historical Trend Analysis

Users can explore how a trend evolved over time by selecting a historical date range. This feature allows:

- Comparison of sentiment across time periods
- Identification of peak activity periods
- Analysis of long-term trend behavior

A.13 Error Handling and Notifications

The system provides clear feedback in case of issues:

- Informational messages during loading
- Error messages when data is unavailable
- Notifications when rate limits are reached

Users are guided with actionable messages to retry or adjust inputs.

A.14 Best Practices for Users

To obtain accurate insights:

- Use specific and relevant hashtags.
- Avoid overly broad keywords.
- Compare multiple time ranges for deeper analysis.
- Cross-check flagged bot activity before drawing conclusions.

A.15 Limitations

The extension operates within the following constraints:

- Dependent on availability of public X data.
- Geolocation analysis may be limited by missing metadata.
- Daily query limits may apply.

A.16 Conclusion

The OSINT-Based X Trend Analyzer Extension provides an accessible and powerful platform for social media intelligence analysis. By combining real-time analytics, sentiment evaluation, influencer detection, and visualization within a browser extension, it enables users to make informed, data-driven decisions efficiently.

References

[1] Amy Kim, Holly Hansen, James Murphy, Andrea Richards, Jennifer Duke, and Jason Allen. Methodological considerations in analyzing twitter data. *Journal of the National Cancer Institute Monographs*, (47):140–146, 2013. Cited on pp. 6 and 7.

[2] Federico Arolfo, Kevin Rodriguez, and Alejandro Vaisman. Analyzing the quality of twitter data streams. *Information Systems Frontiers*, 2020. Cited on pp. 6 and 7.

[3] Shashank Jain, Vishal Sharma, and Raghav Kaushal. Towards automated real-time detection of misinformation on twitter. In *International Conference on Advances in Computing, Communications and Informatics*, pages 2015–2020, 2016. Cited on pp. 6, 7, 8, and 13.

[4] Javier Pastor-Galindo, Pantaleo Nespoli, Guillermo Marmol, and Gregorio Perez. The not yet exploited goldmine of osint. *IEEE Access*, 8:10282–10304, 2020. Cited on pp. 6 and 7.

[5] David Zimbra, Ahmed Abbasi, Daniel Zeng, and Hsinchun Chen. The state-of-the-art in twitter sentiment analysis. *ACM Transactions on Management Information Systems*, 9:1–29, 2018. Cited on pp. 7 and 39.

[6] David Omand and Mark Phythian. The technoethics of contemporary intelligence practice. In *National Security Intelligence and Ethics*. Routledge, 2021. Cited on p. 8.

[7] Isabel Anger and Christian Kittl. Measuring influence on twitter. In *Proceedings of the International Conference on Social Informatics*, 2011. Cited on pp. 13 and 32.

[8] Donald Norman. *User Centered System Design*. CRC Press, 1986. Cited on p. 26.

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186, 2019. Cited on p. 39.