IJNRD.ORG ISSN : 2456-4184

INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG

An International Open Access, Peer-reviewed, Refereed Journal

SPAM-SLAM: A MACHINE LEARNING APPROACH TO DETECT FAKE AND SPAM USERS IN SOCIALMEDIA PLATFORMS

Priyadharshini. S
Research Scholar and Assistant Professor
Department of Artificial Intelligence and Data Science
SNS College of Engineering
Coimbatore, Tamil Nadu, India
snpriyadharshini@gmail.com.

Revathi.K
Research Scholar and Assistant Professor
Department of Information Technology
SNS College of Engineering
Coimbatore, Tamil Nadu, India
revathiks09@gmail.com

Jeevaprasath. M
Bachelor of Technology
Department of Artificial Intelligence and Data Science
SNS College of Engineering
Coimbatore, Tamil Nadu, India
jeevaprasath2512@gmail.com

Preethi Anselin. J
Bachelor of Technology
Department of Artificial Intelligence and Data Science
SNS College of Engineering
Coimbatore, Tamil Nadu, India
preethiansi08@gmail.com

Tamilarasan. S
Bachelor of Technology
Department of Artificial Intelligence and Data Science
SNS College of Engineering
Coimbatore, Tamil Nadu, India
tamilarasan.officials@gmail.com

Yalini. S
Bachelor of Technology
Department of Artificial Intelligence and Data Science
SNS College of Engineering
Coimbatore, Tamil Nadu, India
yalinisai4@gmail.com

Abstract—Social networking sites engage millions of users around the world. The users' interactions with these social sites, such as Twitter and Facebook have a tremendous impact and occasionally undesirable repercussions for daily life. The prominent social networking sites have turned into a target platform for the spammers to disperse a huge amount of irrelevant and deleterious information. Twitter, for example, has become one of the most extravagantly used platforms of all times and therefore allows an unreasonable amount of spam. Fake users send undesired tweets to users to promote services or websites that not only affect legitimate users but also disrupt resource consumption. Moreover, the possibility of expanding invalid information to users through fake identities has increased that results in the unrolling of harmful content. Recently, the detection of spammers and identification of fake users on Twitter has become a common area of research in contemporary online social Networks (OSNs). In this paper, we perform a review of techniques used for detecting spammers on Twitter. Moreover, a taxonomy of the Twitter spam detection approaches is presented that classifies the techniques based on their ability to detect: (i) fake content, (ii) spam based on URL, (iii) spam in trending topics, and (iv) fake users. The presented techniques are also compared based on various features, such as user features, content features, graph features, structure features, and time features. We are hopeful that the presented study will be a useful resource for researchers to find the highlights of recent developments in Twitter spam detection on a single platform.

Keywords — Bus Social networking sites,Twitter spam detection,Fake users,Spam detection techniques,Online social networks (OSNs).

I. Introduction

In an era where online interactions dominate, ensuring the authenticity of users and shielding against spam has become paramount. Spam-slam is not just an app; it's a sophisticated solution meticulously crafted to redefine your digital experience.

Artificial Intelligence at Its Pinnacle: Spam-slam employs state-of-the-art AI algorithms, constantly learning and evolving to distinguish between genuine users and deceptive entities. This dynamic intelligence allows for real-time threat detection and adaptive responses.

Advanced Pattern Recognition: Our app goes beyond simple rule-based systems. Spam-slam utilizes advanced pattern recognition to analyze user behavior, profile data, and communication patterns, identifying anomalies indicative of fake accounts or spam activities.

Swift Action Against Deception: When Spam-slam detects a potential threat, it doesn't just alert; it acts. Through a seamless integration of machine learning and user behavior analysis, the app ensures swift and precise actions, preventing fraudulent activities before they can harm your online community.

Universal Compatibility: Whether you're managing a social platform, an e-commerce site, or a community forum, Spam-slam seamlessly integrates with diverse digital environments, offering a comprehensive solution tailored to your specific needs.

Zero-Tolerance for Spam: Bid farewell to unwanted solicitations and intrusive spam. Spam-slam's robust filters and personalized settings empower users to tailor their digital spaces, creating an environment free from the clutter of irrelevant content.

Privacy-First Approach: At Spam-slam, we understand the value of privacy. The app operates with a privacy-first mindset, ensuring that user data is handled responsibly and transparently. Our commitment is not just to security but also to preserving the integrity of personal information.

Step into a digital realm where authenticity reigns supreme. Spam-slam is not just an app; it's your partner in building a secure, genuine, and thriving online community. Embrace a new era of digital trust with Spam-slam – because your online experience deserves nothing less.

II. EXISTING SYSTEM

The existing system of Spam-slam revolves around a multifaceted approach to user verification and spam prevention.

1. User Profiling and Behaviour Analysis: Spam-slam extensively analyzes user profiles, including account creation details, activity history, and engagement patterns. By establishing a baseline for normal behaviour, the system can swiftly identify anomalies that might indicate a fake user or spam activity.

2. Machine Learning Algorithms: Leveraging machine learning, Spam-slam continuously refines its understanding of emerging threats. The algorithms adapt and learn from new patterns, staying ahead of evolving tactics used by malicious entities.

3. Real-time Monitoring and Alerts: The system operates in real-time, monitoring user activities as they unfold. Upon detecting suspicious behaviour or potential threats, Spam-slam triggers immediate alerts, allowing administrators or users to take prompt action.

4. Community Feedback Mechanism: Spam-slam incorporates a community feedback mechanism, empowering users to report suspicious accounts or content. This valuable input helps enhance the system's accuracy and responsiveness.

5. Content Filtering and Keyword Analysis: To combat spam, Spam-slam employs advanced content filtering techniques and keyword analysis. This ensures that unwanted or inappropriate content is identified and filtered out, contributing to a cleaner and more secure digital environment.

6. Integration with Third-Party Services*: The app seamlessly integrates with third-party services to cross-verify user information and enhance the accuracy of its assessments. This collaborative approach strengthens the overall effectiveness of the system.

7. Regular Updates and Maintenance: Spam-slam undergoes regular updates to stay ahead of emerging threats and maintain compatibility with evolving digital landscapes. Continuous maintenance ensures that the app remains a robust and reliable solution for user verification and spam prevention.

Spam-slam's existing system is a comprehensive ecosystem designed to tackle the complexities of online deception, providing users and administrators with the tools they need to create and maintain secure digital spaces.

2.1 Disadvantages

While Spam-slam boasts an advanced system for user verification and spam prevention, it's essential to acknowledge potential disadvantages and they are False Positives, Resource Intensive, Over-Reliance on AI, Privacy Concerns, Community Reporting Bias, Integration Challenges, Limited Compatibility, Continuous Maintenance Burden. Addressing these disadvantages involves a delicate balance between refining the AI algorithms, optimizing resource usage, maintaining privacy standards, and addressing user concerns to ensure Spam-slam remains an effective and user-friendly solution.

2.2 Objective and Scope

Objective:

The primary objective of Spam-slam is to establish a secure and authentic digital environment by employing advanced user verification and spam prevention mechanisms. The app aims to:

1. Mitigate Risks: Identify and neutralize fake user accounts and spam activities to protect users from potential online threats and deceptive practices.

2. Enhance User Trust: Foster a sense of trust and reliability within online communities, platforms, and networks by ensuring the authenticity of user interactions and content.

3. Optimize User Experience: Create a seamless and secure online experience by minimizing disruptions caused by fake profiles, intrusive spam, and malicious activities.

4. Adapt to Evolving Threats: Continuously evolve and adapt to emerging online threats through machine learning and real-time monitoring, staying ahead of deceptive tactics employed by malicious entities.

Scope:

Spam-slam's scope encompasses a wide range of features and functionalities to comprehensively address the challenges associated with fake user detection and spam prevention:

1. User Verification: Verify the authenticity of user profiles through in-depth analysis of account creation details, activity history, and behavioural patterns.

2. Real-time Monitoring: Monitor user activities in real-time to swiftly identify and respond to suspicious behaviour, minimizing the potential impact of deceptive entities.

3. AI-Powered Analysis: Utilize advanced artificial intelligence algorithms for continuous learning and adaptation, ensuring effective detection of evolving threats.

4. Content Filtering: Employ robust content filtering techniques, including keyword analysis, to identify and filter out spam, ensuring a cleaner and safer digital environment.

5. Community Feedback Integration: Allow users to contribute to the system's accuracy by reporting suspicious accounts or content, creating a collaborative approach to threat identification.

6. Third-Party Integration: Seamlessly integrate with third-party services to cross-verify user information, enhancing the accuracy and reliability of threat assessments.

7. Privacy-First Approach: Prioritize user privacy by handling data responsibly and transparently, maintaining a balance between security measures and user confidentiality.

8. Regular Updates and Maintenance: Ensure the app stays at the forefront of security measures by undergoing regular updates and maintenance to address emerging threats and maintain compatibility with evolving digital landscapes.

By defining clear objectives and a comprehensive scope, Spam-slam aims to provide users with a robust and reliable solution for creating secure, genuine, and trustworthy online communities.

III. Software Requirments

The software requirements for Spam-slam, the fake user and spam detection app, include:

1. Operating System: Compatible with major mobile and web platforms (iOS, Android, Windows, macOS). Web-based interface accessible through popular browsers (Chrome, Firefox, Safari).

2. Database Management System: Reliable database system to store user profiles, activity logs, and configuration data. Support for efficient data retrieval and management.

3. Programming Language: Utilize a versatile and efficient programming language (e.g., Python, Java, Swift) for app development.

4. Development Frameworks: Choose appropriate development frameworks (e.g., React Native, Flutter, Angular) for cross-platform compatibility and efficient coding.

5. Machine Learning Libraries: Integration of machine learning libraries (e.g., TensorFlow, PyTorch) for the implementation of advanced algorithms in user behaviour analysis.

6. AI Model Deployment: Mechanism for deploying and updating AI models seamlessly within the app to adapt to evolving threats.

7. Web Services: Integration with web services for third-party data verification and cross-referencing.

8. Authentication Mechanism: Robust authentication system to secure user accounts and access to sensitive data.

9. Real-time Monitoring Tools: Implement tools for real-time monitoring of user activities, ensuring prompt response to potential threats.

10. Content Filtering: Implement content filtering mechanisms using regular expressions and keyword analysis to identify and block spam.

11. User Reporting System: Develop a user-friendly reporting system allowing community members to flag suspicious content or accounts.

12. Privacy Compliance: Ensure compliance with privacy regulations and implement measures to safeguard user data.

13. Scalability Considerations: Design the app architecture to scale efficiently with increasing user bases and data volumes.

14. Security Measures: Incorporate encryption protocols, secure connections (HTTPS), and other security measures to protect sensitive data.

15. Cross-browser Compatibility: Ensure the web interface is compatible with popular browsers to provide a consistent user experience.

16. Regular Updates: Establish a mechanism for delivering regular updates to address security vulnerabilities, improve performance, and introduce new features.

17. Documentation: Comprehensive documentation for developers, administrators, and end-users to facilitate understanding and usage.

By meeting these software requirements, Spam-slam can deliver a robust and reliable solution for fake user and spam detection, ensuring a secure and trustworthy online environment.

System Requirements

1. RAM: 4GB or 8GB
2. Windows 10
3. Processor
4. Web Server
5. Internet Connectivity

Hardware and Software Setup

Hardware Setup:

1. Server Infrastructure: Multi-core processors with sufficient processing power to handle concurrent requests. Ample RAM to support the efficient execution of backend processes and algorithms.

2. Database Server: Powerful database server with fast storage to handle data storage and retrieval efficiently. RAID configurations or other redundancy mechanisms for data protection.

3. Networking Components: High-speed internet connection to facilitate real-time communication and updates. Firewalls and intrusion detection/prevention systems for network security. Load balancing equipment to distribute incoming traffic evenly.

4. Development and Testing Machines: Workstations for developers with sufficient resources for coding, testing, and debugging. Devices for quality assurance and testing purposes, including various mobile devices and browsers.

5. Cloud Services (Optional): Consider cloud-based solutions for scalability and flexibility, utilizing services like AWS, Google Cloud, or Azure.

Software Setup:

1. Operating Systems: Linux-based server operating system (e.g., Ubuntu Server) for backend infrastructure. Cross-platform compatibility for mobile apps (iOS, Android) and web-based interfaces.

2. Web Server: Configure a web server (e.g., Nginx, Apache) to serve the web-based interface securely. Implement SSL/TLS certificates for encrypted communication.

3. Database Management System: Choose a robust database system (e.g., PostgreSQL, MongoDB) for efficient data storage and retrieval. Optimize database configurations for performance.

4. Programming Languages and Frameworks: Use appropriate programming languages (e.g., Python, Java, Swift) for backend and frontend development. Employ frameworks like Django, Flask, or Express for backend development. Utilize React Native, Flutter, or similar frameworks for cross-platform mobile app development.

5. Machine Learning Libraries: Integrate machine learning libraries (e.g., TensorFlow, PyTorch) for implementing advanced algorithms for user behaviour analysis.

This comprehensive hardware and software setup ensures that Spam-slam operates efficiently, securely, and reliably, delivering a robust solution for fake user and spam detection.

3.4 System Integration and Testing

System Integration for Spam-slam:

1. Backend Integration: Integrate backend components, including the database, web server, and machine learning algorithms, ensuring seamless communication between each module.

2. Third-Party Services: Integrate with third-party services for user data verification and cross-referencing, establishing reliable connections and error handling.

3. Authentication Systems: Integrate secure authentication mechanisms for user registration, login, and access control.

4. Real-Time Monitoring: Implement real-time monitoring tools to capture and analyze user activities, enabling swift response to potential threats.

5. Content Filtering: Integrate content filtering mechanisms to identify and block spam, incorporating keyword analysis and pattern recognition.

6. Community Feedback System: Implement a user-friendly reporting system for community feedback, allowing users to report suspicious accounts or content.

7. Machine Learning Models: Integrate machine learning models for user behaviour analysis, ensuring they adapt and learn continuously from new patterns.

8. Web-Based Interface: Ensure seamless integration of the web-based interface with backend systems, providing a cohesive user experience.

9. Mobile App Integration: Integrate mobile app functionalities, including user verification features and real-time alerts, ensuring consistency across platforms.

Testing Strategies:

1. Unit Testing: Conduct unit tests for individual components, ensuring they function as intended. Verify the accuracy of machine learning models and algorithms through unit tests.

2. Integration Testing: Perform integration tests to validate the interactions between backend systems, third-party services, and the frontend interface.

3. End-to-End Testing: Conduct end-to-end tests to simulate user interactions across the entire system, checking for seamless integration and identifying any potential bottlenecks.

4. Security Testing: Perform security testing to identify and address vulnerabilities, including penetration testing, code reviews, and encryption audits.

5. Performance Testing: Assess the app's performance under various loads and conditions, ensuring optimal responsiveness and scalability.

6. User Experience Testing: Conduct usability tests to evaluate the overall user experience, including the reporting system, real-time monitoring, and user interactions.

7. Cross-Browser and Cross-Platform Testing: Ensure the web-based interface functions consistently across multiple browsers. Verify the mobile app's compatibility and performance on different devices and operating systems.

8. Regression Testing: Perform regression tests with each software update to ensure new features or bug fixes do not introduce unforeseen issues.

9. Community Feedback Validation: Validate the effectiveness of the community feedback system by simulating and responding to reported incidents.

10. Documentation Review: Review and update documentation to ensure accuracy and completeness, facilitating the on boarding process for users and administrators.

By implementing thorough system integration and testing strategies, Spam-slam can ensure a robust, secure, and user-friendly experience for detecting fake users and preventing spam in the digital environment.

3.5 Deployment and Maintenances

Deployment Process for Spam-slam:

1. Code Versioning: Utilize a version control system (e.g., Git) to manage code changes and ensure a stable release.

2. Continuous Integration/Continuous Deployment (CI/CD): Implement CI/CD pipelines to automate the build, testing, and deployment processes, ensuring a streamlined and error-free release cycle.

3. Environment Setup: Set up production, staging, and testing environments to facilitate a controlled deployment process.

4. Database Migration: Implement automated database migration scripts to update the database schema and ensure data consistency during deployment.

5. Security Measures: Apply security best practices, including encryption and secure configurations, to protect sensitive data during deployment.

Maintenance Practices for Spam-slam:

1. Regular Updates: Schedule and perform regular updates to address security vulnerabilities, introduce new features, and improve overall performance.

2. Data Backups: Regularly back up user data and configurations to prevent data loss in case of unforeseen issues.

3. Security Audits: Conduct periodic security audits to identify and address potential vulnerabilities, ensuring ongoing protection against emerging threats.

4. Scalability Planning: Continuously monitor user growth and plan for scalability enhancements to accommodate increasing loads on the system.

5. User Support and Training: Provide ongoing user support and training to address queries, resolve issues, and ensure users maximize the app's capabilities.

By following a well-structured deployment process and incorporating proactive maintenance practices, Spam-slam can consistently provide a secure, reliable, and evolving solution for fake user and spam detection in the digital space.

IV. IDEATE

5.1 Proposed System

Pam-slam aims to revolutionize user verification and spam prevention with an enhanced system that combines cutting-edge technologies and user-centric features.

1. Advanced User Profiling: Implement sophisticated user profiling mechanisms, incorporating behavioral analysis, activity history, and cross-referencing with third-party services for comprehensive user verification.

2. AI-Powered Threat Detection: Utilize advanced machine learning algorithms to enhance threat detection capabilities, enabling the system to adapt and evolve in real-time to counter emerging deceptive tactics.

3. Real-time Monitoring and Alerts: Enhance real-time monitoring features, providing administrators and users with immediate alerts when suspicious activities or potential threats are detected.

4. Community Feedback Empowerment: Strengthen the community feedback system by integrating user-friendly reporting tools, fostering a collaborative approach to identifying and addressing deceptive entities.

5. Multi-Layered Content Filtering: Introduce a multi-layered content filtering system that combines keyword analysis, pattern recognition, and advanced filtering rules to effectively identify and block spam.

6. Privacy-Centric Design: Implement privacy-centric design principles, ensuring secure handling of user data, transparent privacy policies, and granular user controls over their information.

7. Cross-Platform Compatibility: Optimize the app for cross-platform compatibility, providing a seamless experience on various devices and operating systems, including iOS, Android, and web browsers.

8. Enhanced User Authentication: Strengthen user authentication mechanisms with multi-factor authentication options, enhancing the overall security of user accounts.

9. Scalable Infrastructure: Design a scalable infrastructure that can efficiently handle growing user bases and increasing data volumes, ensuring the app's reliability and performance under varying loads.

10. User Education and On boarding: Develop user-friendly on boarding processes and educational resources to guide users in understanding the app's features, reporting mechanisms, and privacy controls.

This proposed system envisions Spam-slam as a dynamic and evolving solution that not only adapts to the changing landscape of online threats but also prioritizes user experience, privacy, and community collaboration.

5.2 Advantages

1. Highly Accurate User Verification: Leveraging advanced profiling and machine learning, Spam-slam ensures precise user verification, reducing the risk of fake accounts.

2. Effective Spam Prevention: The multi-layered content filtering system and real-time monitoring mechanisms empower Spam-slam to effectively identify and eliminate spam, creating a cleaner digital environment.

3. Real-Time Threat Detection: The app's use of AI algorithms enables real-time threat detection, allowing for immediate responses to emerging deceptive tactics and potential threats.

4. User-Friendly Reporting System: Spam-slam provides a user-friendly reporting system, encouraging community members to actively contribute to the identification and resolution of suspicious activities.

5. Adaptive Machine Learning: The incorporation of adaptive machine learning ensures that Spam-slam continuously evolves and learns from new patterns, enhancing its ability to detect evolving threats.

6. Cross-Platform Compatibility: With optimization for various platforms, including iOS, Android, and web browsers, Spam-slam offers a seamless and consistent user experience across different devices.

7. Privacy-Centric Design: Spam-slam prioritizes user privacy with a design that ensures secure handling of user data, transparent privacy policies, and granular user control over their information.

8. Scalable Infrastructure: The app is designed with scalability in mind, allowing it to efficiently handle growing user bases and increasing data volumes without compromising performance.

9. Enhanced User Authentication: Spam-slam strengthens user authentication mechanisms, providing an additional layer of security for user accounts.

10. Community Collaboration: The app fosters community collaboration through a robust reporting system, empowering users to actively contribute to the security of the digital space.

Spam-slam's advantages lie in its holistic approach to user verification, spam prevention, and community collaboration, creating a secure and user-friendly digital environment.

V. RESULT AND SCREESNSHOTS

5.1 Input

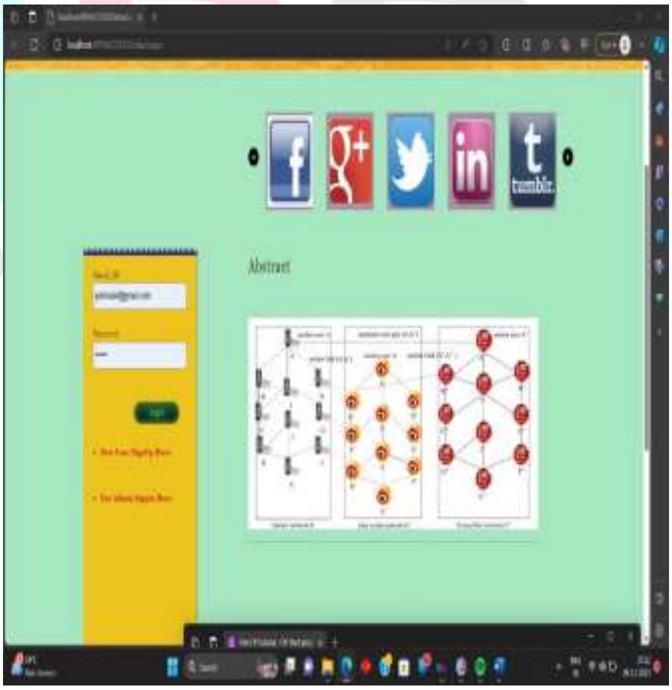

Fig 1: Input Page 1

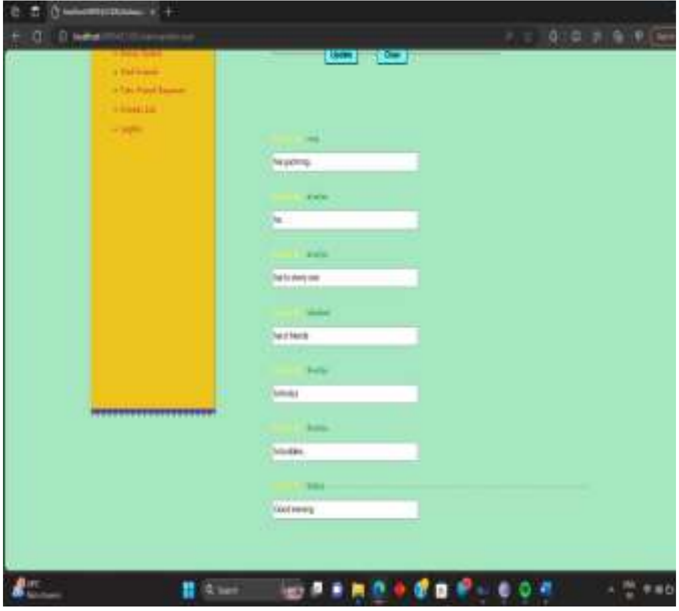

Fig 2: Input Page 2

5.2 Output

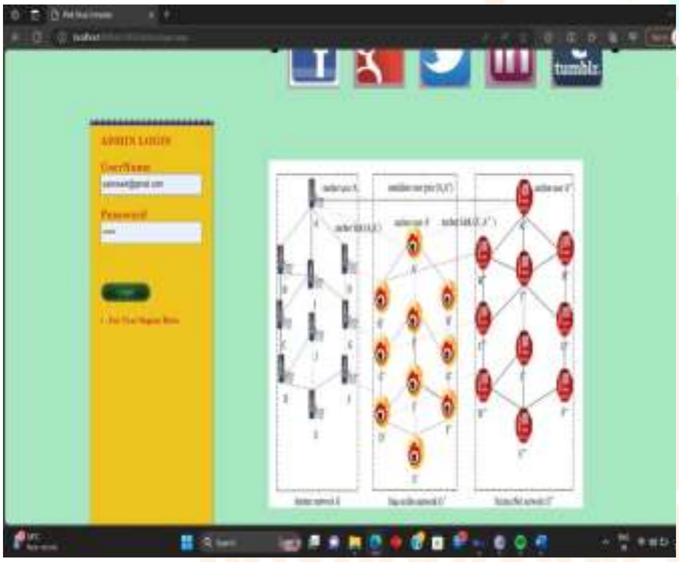

Fig 3: Output Page-1

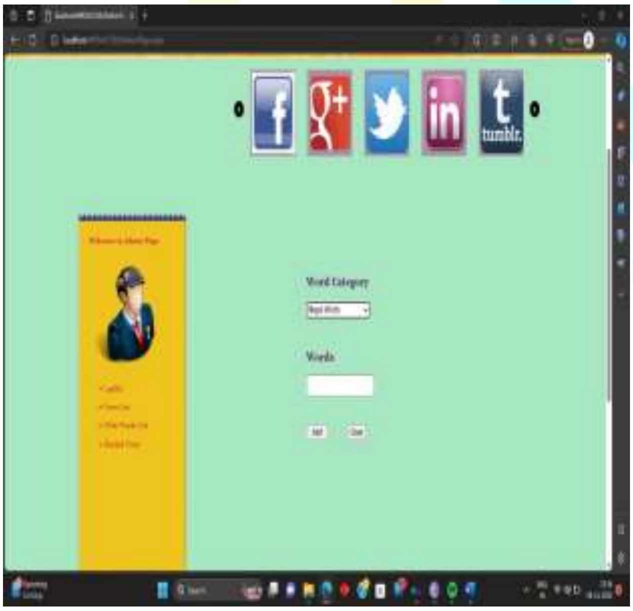

Fig 4: Output Page-2

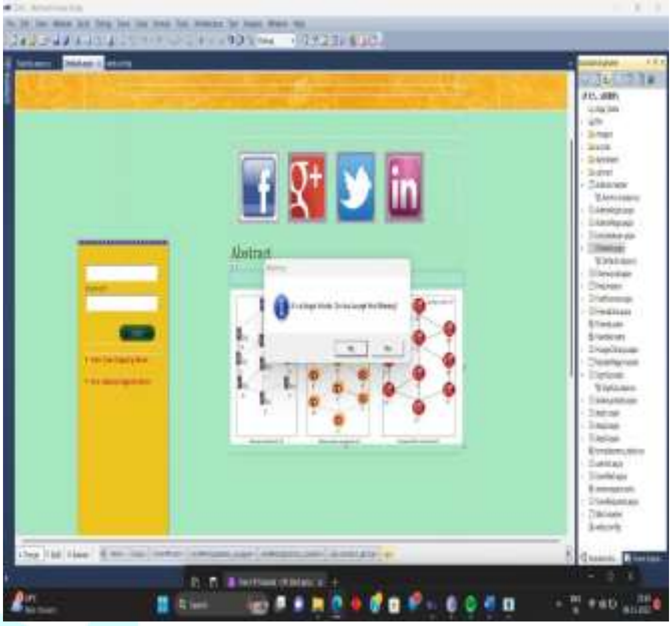

Fig 5: Output page-3

VI. Conclusion

This project SPAM – SLAM, with its focus on user authentication and content moderation, finds applications in a variety of online environments. It can be implemented in social media platforms to automatically detect and handle inappropriate or harmful content, creating a safer online community for users. Additionally, forums and community websites can utilize this system to ensure that discussions remain respectful and on-topic. Online marketplaces benefit from the ability to filter out misleading or malicious content in product reviews, enhancing consumer trust. Educational platforms can maintain a secure learning environment, and professional networking platforms can promote respectful interactions among users. In customer support forums, this project can help foster a positive and helpful atmosphere for users seeking assistance. Overall, the project's versatility in moderating user generated content and authenticating users makes it a valuable tool across various online platforms and communities.

VII. Future Scope

The SPAM – SLAM's future enhancements involves as content moderation and user authentication continue to be critical concerns in online platforms. As technology evolves, there is a growing need for more sophisticated and intelligent systems to detect and mitigate malicious content. Implementing advanced machine learning algorithms and natural language processing techniques could enhance the accuracy of content moderation. Additionally, integrating biometric authentication methods or advanced security measures can further bolster user authentication processes. As the digital landscape evolves, this project has the potential to adapt and incorporate cutting-edge technologies, ensuring its relevance and effectiveness in addressing emerging challenges in online content management and user security. Furthermore, it can be extended to support multiple languages and adapt to different cultural contexts, expanding its applicability on a global scale.

References

[1] Building APIs You Won't Hate" by Phil Sturgeon: This book teaches you how to design and build robust APIs that serve data to your mobile app. It covers topics like API design principles, authentication, data serialization, and best practices for building scalable and maintainable APIs.

[2] Internet of Things Programming with ESP32" by Adrian McEwen: This book focuses on programming the ESP32 microcontroller, a popular choice for building IoT devices like bus trackers. It covers connecting to various sensors, including GPS modules, and sending data over the internet using Wi-Fi or cellular networks.

[3] Android Programming with Kotlin for Beginners" by Stack Overflow: This book provides a comprehensive introduction to Android app development using Kotlin, the recommended language for modern Android apps. It covers building user interfaces, handling user interactions, and integrating with external services like GPS and map APIs

[4] iOS Programming with Swift" by Apple: This official guide from Apple teaches you how to develop iOS apps using Swift. It covers all the essential concepts, from user interface design to data persistence and network communication. You can choose this book if you're planning to build an app for iPhone or iPad users

[5] GPS for Beginners" by Brian Kerrigan: This book provides a good introduction to the fundamentals of GPS technology, including its working principles, signal acquisition, and accuracy limitations. It also covers basic concepts like geodetic data and coordinate systems.

[6] Xingshuai Huang, Benoit Boulet, and Di Wu, Ensemble Learning for Charging Load Forecasting of Electric Vehicle Charging Stations, IEEE Electric Power and Energy Conference, 2020.

[7] Haowei Wang, Xueliang Huang, et al, Electric vehicle charging load clustering and load forecasting based on long short term memory neural network, IEEE 5th International Electrical and Energy Conference, 2022.

[8] Jemima Graham and Fei Teng, Vehicle-to-grid plug-in forecasting for participation in ancillary services markets, IEEE Belgrade PowerTech, 2023.

[9] Hassan Khan, Muhammad Junaid Khan, and Amir Qayyum, Neural Network-based Load Forecasting Model for Efficient Charging of Electric Vehicles, 7th Asia Conference on Power and Electrical Engineering, 2022.

[10] Chenyang Liang and Wenyi Li, Electric Vehicle Charging Load Forecasting Model Considering Users’ Travel Rules, 3rd New Energy and Energy Storage System Control Summit Forum, 2023.

[11] Xingshuai Huang, Di Wu, and Benoit Boulet, MetaProbformer for Charging Load Probabilistic Forecasting of Electric Vehicle Charging Stations, IEEE Transactions on Intelligent Transportation Systems, Volume: 24, Issue: 10, 2023.

[12] Chenke He, Jizhong Zhu, et al, Optimal Planning of Electric Vehicle Battery Centralized Charging Station Based on EV Load Forecasting, 6th International Conference on Power and Renewable Energy (ICPRE), 2021.

[13] K.D.McBee, D.Bukofzer, et al, Forecasting Long-term Electric Vehicle Energy Demand in a Specific Geographic Region, IEEE Power & Energy Society General Meeting (PESGM), 2020.

[14] C.A.Anuradha, et al, Market Analysis and Forecasting using Trending Data Analytical Tool, International Conference on Sustainable Computing and Smart Systems, 2023, pp. 663–667.

[15] D.Praveen Kumar, et al, Data Mining Approaches in Healthcare Industry, Proceedings of the 5th International Conference on Inventive Research in Computing Applications, 2023, pp. 1019–1025