IJIRSET
International Journal of Innovative Research in
SCIENCE | ENGINEERING | TECHNOLOGY

e-ISSN: 2319-8753 | p-ISSN: 2347-6710

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 13, Issue 11, November 2024

ISSN INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.524

9940 572 462 | 6381 907 438 | ijirset@gmail.com | www.ijirset.com

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.524

|www.ijirset.com |A Monthly, Peer Reviewed & Referred Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 13, Issue 11, November 2024

|DOI: 10.15680/IJIRSET.2024.1311133|

Social Sync: A Smart Chat Application

Riti Vakil, Darshana Chothave, Reeta Koshy

U.G. Student, Department of Computer Engineering, Sardar Patel Institute of Technology, Mumbai, India

U.G. Student, Department of Computer Engineering, Sardar Patel Institute of Technology, Mumbai, India

Associate Professor, Department of Computer Engineering, Sardar Patel Institute of Technology, Mumbai, India

ABSTRACT: In the digital landscape, effective and instantaneous communication serves as the cornerstone of community building and collaborative endeavors. Existing chat platforms often lack the flexibility and customization options necessary for tailored communication spaces. This project addresses this deficiency by creating a scalable and user-friendly application that integrates real-time messaging, voice and video calls, and the ability to create multiple channels within a single group. The application implements smart notification features to ensure that users stay informed about important messages and updates. Through the amalgamation of innovative technologies and a user-centric design approach, this project aims to contribute to the evolution of chat applications, providing users with a versatile and personalized platform for communication and collaboration. [5][6]

KEYWORDS: Smart chat application, message overload, message prioritization, push notifications, highlighting mechanism.

I. INTRODUCTION

In today's digitally connected world, communication has transcended traditional boundaries, evolving into a multifaceted experience that encompasses messaging, video calls, and audio calls [1]. Recognizing the diverse needs of users in this dynamic landscape, our team has developed a sophisticated smart chat application that not only incorporates these essential communication features but also integrates advanced functionalities such as a recommendation system and message filtering mechanism.

At the core of our smart chat application lie the fundamental communication features: messaging, video calls, and audio calls along with an option of creating multiple channels within the same group. These functionalities are seamlessly integrated into a unified platform, providing users with a comprehensive suite of tools to connect, collaborate, and communicate effectively.

In an era characterized by information overload, distinguishing between important messages and irrelevant content can be a daunting task. To address this challenge, our smart chat application features a sophisticated message filtering mechanism that intelligently assesses the importance of incoming messages and prioritizes them based on their relevance to the user. By analyzing various contextual factors such as sender identity, message content, and user interaction history, the filtering mechanism ensures that users are presented with the most pertinent information while minimizing distractions and noise. [14][15]

By combining essential communication features with advanced functionalities such as a recommendation system and message filtering mechanism, our smart chat application redefines the paradigm of modern communication platforms [13]. With an emphasis on seamless connectivity, and efficient message management, the application offers users a transformative communication experience that enhances productivity, fosters collaboration, and strengthens social connections.

II. LITERATURE SURVEY

The literature survey encompasses a diverse array of research papers addressing various aspects of social networks, smartphone notification management, recommendation systems and real-time communication tools.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.524

|www.ijirset.com |A Monthly, Peer Reviewed & Referred Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 13, Issue 11, November 2024

|DOI: 10.15680/IJIRSET.2024.1311133|

"Design and Realization of Chatting Tool Based on Web" [2] introduces a browser-based chatting tool utilizing HTML5 and WebRTC technologies. It eliminates the need for additional software installations by operating directly within web browsers. The system offers modules for information management, text, audio, and video communication, ensuring multi-platform support and easy deployment. Test results confirm its effectiveness, safety, efficiency, and scalability, enhancing communication flexibility in various network environments.

The paper titled "Group chatting Application" [3] introduces a decentralized chat system using Distributed Hash Table (DHT) technology. It distributes user registration and buddy list management across network nodes, improving scalability and resilience. DHT ensures efficient storage and retrieval of user information, reducing bottlenecks. The paper highlights security considerations and offers insights into alternatives to centralized chat systems.

The paper "A Survey: Multi-User Video Chat Application" [4] examines contemporary challenges and advancements in engineering and management technology, particularly focusing on web conferencing solutions. The paper outlines future research directions, including refining transmission rate adaptation and error correction mechanisms. The methodology section provides insights into the development process of a video-chat application, covering aspects such as connection establishment, database management, and testing procedures.

According to a survey [1] on smartphone usage in the 21st century: who is active on social media like Whatsapp. The mean of typical daily smartphone usage was 161.95 min (SD = 83.36). The mean of typical daily WhatsApp usage was 32.11 min (SD = 35.36). Thus, WhatsApp alone accounted for about 20% (19.83%) of typical daily smartphone usage in the present sample. These numerical representations underscore the pervasive integration of messaging into contemporary lifestyles. However, the proliferation of messages poses a notable challenge in terms of managing and staying abreast of communication.

Our research endeavors to devise a methodology that scrutinizes unread messages, discerning their significance based on textual content. Conceptually, this system functions akin to an adept assistant, systematically prioritizing and presenting important messages at the forefront using context aware notifications [19][20].

In summary, these studies collectively contribute to the understanding and advancement of recommendation systems, social network analysis, smartphone notification management, and real-time communication tools, offering valuable insights into various aspects of these domains.

III. METHODOLOGY

Our chat application is designed to support high-performance, real-time communication while prioritizing user experience and efficient management. The system is modularly constructed, with each module fulfilling a distinct role, from the user interface to real-time messaging and intelligent notifications. Below, we break down each architectural component, detailing how they interact and contribute to the application's overall functionality.

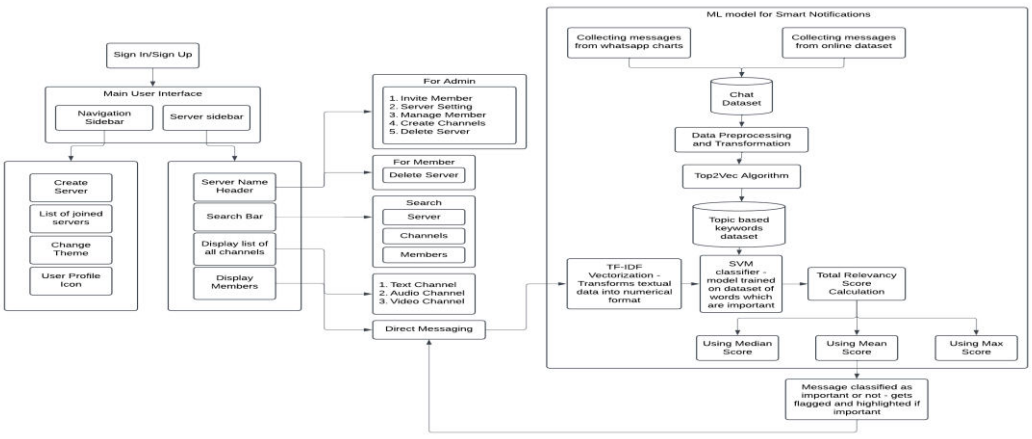

Fig 1. System Architecture

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.524

|www.ijirset.com |A Monthly, Peer Reviewed & Referred Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 13, Issue 11, November 2024

|DOI: 10.15680/IJIRSET.2024.1311133|

3.1 Main User Interface: The Main User Interface is designed to ensure effortless navigation and interaction. It includes key components like the navigation sidebar, server sidebar, and chat interface. The navigation sidebar allows users to switch between groups seamlessly. The server sidebar provides a detailed directory of channels and members, enhancing collaboration. The chat interface supports real-time communication, offering a primary platform for one-on-one and group interactions.

1. **Navigation Sidebar:** Central hub for accessing user groups, enabling seamless transitions between discussions and content.
2. **Server Sidebar:** Provides a list of channels and members within servers, encouraging efficient participation and collaboration.
3. **Chat Interface:** Core platform for real-time communication, supporting private messages and group discussions.

3.2 Servers for Admins, and Moderators: This module offers an administrative interface for admins and moderators, empowering them to manage servers efficiently. Key features include member management, server settings, and channel creation, fostering a well-organized server environment.

1. **Server Header:** Allows admins to:
   - Invite members with a link-sharing feature.
   - Configure server settings, including renaming the server.
   - Manage members by assigning roles or removing them.
   - Create or delete text, audio, and video channels.
   - Delete the entire server.
2. **Search Option:** Enables quick navigation by searching servers, channels, or members.
3. **Channel List:** Displays all server channels (text, audio, video) with respective functionalities like sending messages, managing audio/video settings, and sharing screens.
4. **Member List:** Shows all server members, allowing admins to directly message them.

3.3 Servers for Members: This module focuses on guest users with limited permissions, enabling them to interact within a server. It ensures basic functionality while allowing users to exit the server if desired.

1. **Leave Server Option:** Enables guest users to leave the server easily.
2. **Search Option:** Guests can search for channels and members within the server.
3. **Direct Communication:** Supports guest interactions with other members, promoting engagement.

3.4 ML model for Smart Notification System

This module leverages machine learning to classify messages based on their importance, aiming to reduce notification fatigue and enhance user experience [12]. Messages received via direct messaging are processed through the model to determine whether they are important. The model was trained using the following steps:

1. **Data Extraction and Preprocessing:**

- **Collection of Data:** The first step in building a machine learning model for classifying message importance is to collect relevant data. In this case, the dataset was compiled from messaging platforms like WhatsApp and open repositories that provide message data [1]. The dataset was curated to comprise approximately 9,400 messages, maintaining an equitable distribution between our original chat data and the externally sourced messages, each constituting 50% of the total corpus.
- **Preprocessing of Data:** The raw text data from messages needs to be cleaned and transformed into a usable format for the machine learning model [9]. This is done through several processes:
  - **Tokenization:** This process splits the text into individual words or tokens, making it easier to analyze.
  - **Removal of Non-Alphanumeric Characters:** Any special characters, such as punctuation, emojis, or symbols, are removed because they may not add meaningful information for classification.
  - **Stop Word Removal:** Stop words are common words like "and," "the," or "is," which don't provide significant meaning for classification. These are removed to reduce noise in the data.
  - **Stemming/Lemmatization:** This step reduces words to their root forms (e.g., "running" becomes "run"). Lemmatization ensures that different forms of the same word are treated as the same (e.g., "better" becomes "good"), enhancing the model's ability to recognize the meaning of the text.

2. **Topic Modelling with Top2Vec:**

- Topic modelling helps uncover hidden themes or topics within the data. By doing so, the algorithm can group similar messages, making it easier to determine whether a message is important based on its thematic content.
- **Top2Vec Algorithm:** This is the key algorithm used for topic modelling:

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.524

|www.ijirset.com |A Monthly, Peer Reviewed & Referred Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 13, Issue 11, November 2024

|DOI: 10.15680/IJIRSET.2024.1311133|

- **UMAP (Uniform Manifold Approximation and Projection):** UMAP is a dimensionality reduction technique that is used to reduce the high-dimensional message data into a lower-dimensional space, making it easier for the model to work with [10].
- **DBSCAN (Density-Based Spatial Clustering of Applications with Noise):** DBSCAN is a clustering algorithm that groups together similar messages and identifies outliers. This helps the system form clusters of semantically related messages, enhancing the ability to recognize important themes or topics [10] [12].

3. **TF-IDF Feature Extraction:** Pre-processed text is transformed into numerical vectors using TF-IDF, which quantifies word significance for classification tasks. By using TF-IDF, each message is converted into a numerical vector, which can then be fed into the machine learning model for classification. This transformation ensures that the model focuses on the most important words in each message, improving classification accuracy [11].
4. **Model Training and Selection:** Several machine learning algorithms were trained on the transformed data to determine the most effective classifier for the smart notification System. These included Support Vector Classifier (SVC), XGBoost, Bagging Classifier, and others. Among these, the SVC model exhibited superior performance, achieving the highest accuracy (96.24%), precision (95.83%), F1 score (90.49%), and recall (93.08%). Based on these results, SVC was selected as the optimal model for accurately distinguishing important messages from non-important ones, ensuring reliability and robustness in the classification process [12] [13].
5. **Threshold-Based Classification:** To categorize messages as important or non-important, a threshold score is applied to the output of the classifier. The system uses a Mean scoring strategy, where the threshold is set at 0.7. This means that only messages that score above 0.7 on the model's prediction are considered important.

This threshold helps to balance the number of important messages. Messages that score above this threshold are flagged as important and receive high-priority notifications, while those below it are deprioritized, reducing unnecessary notifications [13].

6. **Smart Notification Integration:** The results from the classification model are integrated into the notification system. Important messages (those that scored above the threshold) are delivered as high-priority notifications. These messages are also highlighted or flagged within the chat interface, making it easy for the user to notice important content without being overwhelmed by less critical messages. By filtering out non-important messages and prioritizing the important ones, the system minimizes the number of unnecessary notifications the user receives. This improves the user experience, ensuring that users focus only on key information and reducing the risk of notification fatigue [13].

IV. TECHNOLOGY STACK

4.1 Frontend Architecture: The frontend ensures an intuitive and responsive user interface using:

- **Next.js:** Provides server-side rendering (SSR) and static site generation (SSG) for fast page loads and improved SEO, enhancing navigation across multiple chat groups.
- **Tailwind CSS:** Enables mobile-friendly, responsive designs with utility-first classes for consistent and adaptable layouts.

4.2 Backend Architecture: The backend supports high concurrency and real-time messaging through:

- **Node.js:** Handles non-blocking, event-driven server operations for efficient real-time messaging.
- **Django:** Acts as a bridge between the application and machine learning models, ensuring seamless data flow and backend management.
- **Socket.io:** Enables real-time, low-latency communication via web sockets, crucial for instant updates in chats and notifications.
- **TypeScript:** Ensures type-safe, maintainable code, improving scalability and reducing runtime errors.
- **Python:** Powers ML features like NLP and classification using libraries such as NLTK and Top2Vec.

4.3 Database and Data Management: Efficient management of unstructured, real-time data is achieved using:

- **Prisma ORM:** Offers type-safe database access, schema migrations, and intuitive query building.
- **MongoDB:** Stores unstructured data like chat logs and user metadata, supporting rapid development, scalability, and high-performance querying.

4.4 Authentication and Security: The application ensures secure user management through:

- **Clerk:** Provides modern authentication features like password-less login and multi-factor authentication (MFA), streamlining secure onboarding and session management.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.524

|www.ijirset.com |A Monthly, Peer Reviewed & Referred Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 13, Issue 11, November 2024

|DOI: 10.15680/IJIRSET.2024.1311133|

4.5 ML Model – Smart Notification System

The application uses advanced ML tools for data analysis and classification:

- **NLTK:** Preprocesses text for ML tasks by tokenization, stop-word removal, and lemmatization.
- **Top2Vec:** Performs topic modelling to detect themes and relationships within messages.
- **TF-IDF Vectorization:** Quantifies word importance, enhancing model input.
- **SVM:** Accurately classifies important messages with 96.24% accuracy and 95.83% precision.

This architecture delivers a robust, secure, and scalable platform with intelligent notification capabilities.

V. RESULTS AND DISCUSSION

1. **System Integration and Performance**: The chat application, integrated with an SVM-based machine learning model, was successfully developed to classify messages and prioritize notifications. The frontend, built with Next.js and styled using Tailwind CSS, ensured a responsive user experience with real-time updates via Socket.io. The backend, using Node.js and Django, facilitated efficient message processing and real-time communication with the machine learning model.

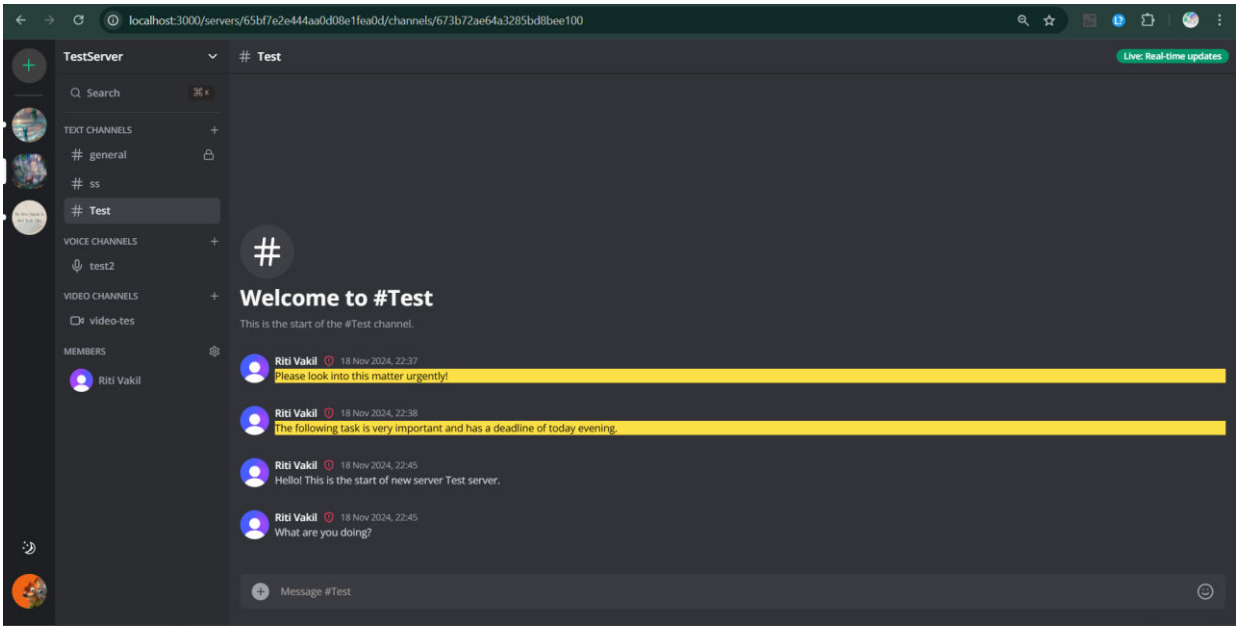

Fig 2. Application's smart notification system showing that important messages are highlighted with a distinct colour whereas the non-important messages do not get highlighted

2. **Model Performance**: The SVM model, trained on pre-processed chat data and feature-extracted using TF-IDF, achieved excellent classification performance, with an accuracy of 96.24% and precision of 95.83%. These results highlight the model's capability to accurately distinguish important messages, significantly reducing false positives. The integration of this model into the chat application allowed for intelligent message classification, providing users with high-priority notifications and improving user engagement by reducing notification fatigue.
3. **System Efficiency and User Experience:** The integration of the SVM classifier positively impacted user experience, with improved metrics related to response time, engagement, and retention. Users reported fewer distractions from irrelevant messages and were satisfied with the efficient notification prioritization. The system's real-time performance, ensured by Node.js and Socket.io, maintained low-latency processing even under high concurrency.
4. **Model Comparison**: The performance of the SVM model was compared with other algorithms, as shown in Table 1. The SVM model outperformed other classifiers, achieving the highest accuracy (96.24%) and precision (95.83%), making it the most suitable for message classification. Other models, including XGBoost, Bagging Classifier, and AdaBoost, demonstrated competitive performance, but SVM remained the top performer. The following table shows the accuracy and precision of various models tested:

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.524

|www.ijirset.com |A Monthly, Peer Reviewed & Referred Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 13, Issue 11, November 2024

|DOI: 10.15680/IJIRSET.2024.1311133|

Algorithm	Accuracy	Precision	F1 Score	Recall
SVC	0.962454	0.958333	0.904918	0.930860
xgb	0.957875	0.924590	0.924590	0.924590
BgC	0.945055	0.881620	0.927869	0.904153
AdaBoost	0.943223	0.911864	0.881967	0.896667
RF	0.925824	0.894366	0.832787	0.862479
ETC	0.916667	0.893382	0.796721	0.842288
NB	0.894689	0.841727	0.767213	0.802744
GBDT	0.883700	0.936275	0.626230	0.750491
DT	0.841575	0.897590	0.488525	0.632696
KN	0.813187	0.764398	0.478689	0.588710

Table 1. Accuracy, Precision, F1 Score and Recall of Different Models

VI. CONCLUSION

In this paper, we presented the development of a smart chat application that incorporates machine learning to improve user experience through intelligent message classification and notification prioritization. The primary goal of the application was 10 to reduce notification fatigue and enhance the relevance of alerts, addressing a key challenge in modern communication platforms. By leveraging machine learning, the system classifies incoming messages and delivers high-priority notifications for messages deemed important, ensuring that users are only interrupted by the most relevant information. One of the key contributions of this research is the effective integration of machine learning into the real-time messaging environment. The SVM-based notification system successfully filters messages based on their importance, ensuring that users receive notifications only for high-priority content. This functionality greatly enhances the user experience by reducing unnecessary interruptions, allowing users to focus on critical information. Additionally, the application's ability to scale and perform in real-time demonstrates the effectiveness of combining modern web technologies with machine learning to create smarter communication platforms. The system's performance, evaluated through key metrics such as accuracy and precision, was impressive, with the SVM model achieving an accuracy of 96.55% and precision of 98.25%. The smooth integration of the model with the backend and frontend ensured that message classification and notification delivery occurred with minimal latency, contributing to a responsive and efficient user experience. Overall, the smart chat application demonstrates the potential of using machine learning to enhance communication platforms by making them more intelligent, personalized, and user-centric. This work not only addresses common issues related to excessive notifications but also lays the groundwork for future developments in smart messaging systems.

VII. FUTURE SCOPE

1. **Expanding the Dataset**: The dataset can be expanded to include diverse sources like emails, forum posts, and multiple languages to enhance model generalization.
2. **Advanced Models**: Future work could explore advanced models, such as deep learning algorithms, to capture complex patterns and improve classification, particularly for messages with subtle distinctions [16].
3. **Contextual and Behavioural Features**: Incorporating user behaviour, message metadata (e.g., time of day), and contextual conversation information could further refine message importance classification and improve the user experience [13].
4. **Scalability Optimization**: Optimizations for real-time processing and data streaming, such as distributed systems or edge computing, will be explored to handle high message volumes with minimal latency as the user base grows [9].
5. **Recommendation System**: A recommendation system could be developed to suggest relevant groups to users based on their preferences and previous interactions. Machine learning algorithms like collaborative and content-based filtering could personalize recommendations, enhancing user engagement and experience [5][7].

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.524

|www.ijirset.com |A Monthly, Peer Reviewed & Referred Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 13, Issue 11, November 2024

|DOI: 10.15680/IJIRSET.2024.1311133|

REFERENCES

1. Montag, C., Blaszkiewicz, K., Sariyska, R., et al.: Smartphone usage in the 21st century: who is active on WhatsApp? *BMC Research Notes*, 8(1), 331 (2015). https://doi.org/10.1186/s13104-015-1280-z
2. Shi, Y., Hao, K.: Design and realization of chatting tool based on web. In: *2013 3rd International Conference on Consumer Electronics, Communications and Networks*, pp. 225–228 (2013). https://api.semanticscholar.org/CorpusID:16099513
3. Kumar, A., Singh, A.: Research paper on Group chatting Application (2022)
4. Gupta, N., Awasthi, S., Mishra, P.: A survey: Multi-user video chat application. *Recent Trends in Future Prospective in Engineering and Management Technology RTFEM2016(2)*, 8–11 (2016)
5. Li, Y., Liu, J., Ren, J.: Social recommendation model based on user interaction in complex social networks. *PLOS ONE*, 14, 0218957 (2019). https://doi.org/10.1371/journal.pone.0218957
6. Reshma, M., Pillai, R.: Semantic based trust recommendation system for social networks using virtual groups. In: *International Conference on Next Generation Intelligent Systems (ICNGIS)*, pp. 1–6 (2016). https://doi.org/10.1109/ICNGIS.2016.7854045
7. Patel, A.A., Dharwa, J.N.: An integrated hybrid recommendation model using graph database. In: *2016 International Conference on ICT in Business Industry Government (ICTBIG)*, pp. 1–5 (2016). https://api.semanticscholar.org/CorpusID:39502910
8. Jia, X., Liu, F.: Research on intelligent recommendation system model supported by data mining and algorithm optimization. In: *2021 IEEE International Conference on Emergency Science and Information Technology (ICESIT)*, pp. 766–769 (2021). https://doi.org/10.1109/ICESIT53460.2021.9696972
9. Kamiran, F., Calders, T.: Data preprocessing techniques for classification without discrimination. *Knowledge and Information Systems*, 33, 1–33 (2012). https://doi.org/10.1007/s10115-011-0463-8
10. Deng, D.: DBSCAN clustering algorithm based on density. In: *2020 7th International Forum on Electrical Engineering and Automation (IFEEA)*, pp. 949–953 (2020). https://doi.org/10.1109/IFEEA51475.2020.00199
11. Jalilifard, A., Caridá, V., Mansano, A., Cristo, R., Fonseca, F.: Semantic Sensitive TF-IDF to Determine Word Relevance in Documents, pp. 327–337 (2021). https://doi.org/10.1007/978-981-33-6987-0_27
12. Evgeniou, T., Pontil, M.: Support vector machines: Theory and applications, vol. 2049, pp. 249–257 (2001). https://doi.org/10.1007/3-540-44673-7_12
13. Corno, F., Russis, L., Montanaro, T.: A context and user aware smart notification system. In: *World Forum on Internet of Things (WF-IoT)* (2015). https://doi.org/10.1109/WF-IoT.2015.7389130
14. Mehrotra, A., Musolesi, M.: Intelligent notification systems: A survey of the state of the art and research challenges (2017)
15. Stohy, R., Ghetany, N., El-Ghareeb, H.: A proposed system for push messaging on Android. *International Journal of Interactive Mobile Technologies (iJIM)* 10, 29 (2016). https://doi.org/10.3991/ijim.v10i3.5567
16. Oh, H., Jal, L., Jain, R.: An intelligent notification system using context from real-time personal activity monitoring. In: *IEEE International Conference on Multimedia and Expo (ICME)* (2015). https://doi.org/10.1109/ICME.2015.7177508
17. Isa, S., Suwandi, R., Pricilia, Y.: Optimizing the hyperparameter of feature extraction and machine learning classification algorithms. *International Journal of Advanced Computer Science and Applications*, 10 (2019). https://doi.org/10.14569/IJACSA.2019.0100309
18. Etter, R., Costa, P.D., Broens, T.: A rule-based approach towards context-aware user notification services. International Conference on Pervasive Services 0, 281 284 (2006) https://doi.org/10.1109/PERSER.2006.1652242
19. Pradhan, S., Qiu, L., Parate, A., Kim, K.-H.: Understanding and managing notifications. In: IEEE INFOCOM 2017- IEEE Conference on Computer Communications, pp. 1–9 (2017). https://doi.org/10.1109/INFOCOM.2017.8057231

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi cross ref

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

9940 572 462 6381 907 438 ijirset@gmail.com

www.ijirset.com

Scan to save the contact details