Gurukul International Multidisciplinary Research Journal (GIMRJ)*with* International Impact Factor 8.249
Peer Reviewed Journal
https://doi.org/10.69758/GIMRJ2406I8V12P033

e-ISSN No. 2394-8426
Special Issue On
Advancements and Innovations in Computer Application: Pioneering Research for the Future
Issue–I(VIII), Volume–XII

DEVELOPING A COMPREHENSIVE LIVE NEWS AGGREGATOR: INTEGRATING REAL-TIME DATA SOURCES FOR ENHANCED INFORMATION ACCESSIBILITY

Atharva Sastikar
PG scholar
Department of Computer Science
G. H. Raisoni University, Amravati, Maharashtra, India

Received on: *11 May ,2024* ***Revised on:*** *18 June ,2024* ***Published on:*** *29 June ,2024*

Abstract—News Aggregator is simply an online software which collects new stories and events around the world from various sources all in one place. News aggregator plays a very important role in reducing time consumption, as all of the news that would be explored through more than one website will be placed only in a single location. Also, summarizing this aggregated content absolutely will save reader's time. A proposed technique used called the TextRank algorithm that showed promising results for summarization. This paper presents the main goal of this project which is developing a news aggregator able to aggregate relevant articles of a certain input keyword or key-phrase. Summarizing the relevant articles after enhancing the text to give the reader understandable and efficient summary.

Index Term : News aggregator; text summarization; text enhancement; TextRank algorithm

I. INTRODUCTION

In the last few years, the world had incredible and huge growth in the rate of news that is published [1]. People live in a time full of information, data, and news [2]. So, nowadays news has an important part and position within the community. As people read the news daily to keep up with the most recent data and inputs. These data may be about technology, sports, weather, food, and celebrities or many other fields [3]. With the development of the Internet, and lot of websites that provide the same data and information, getting this has become simpler getting to it has become simpler [4]. So, users frequently discover it troublesome to decide which of these websites can provide the specified data within the most valuable and effective way [2]. The conventional commerce model of daily papers has been threatened by the internet to lessening their advertising income and by presenting new online media, such as web-only news, blogs and news aggregators [5]. The online social system is a valuable instrument for collecting, aggregating and expending the specific or common contents for different aims in a certain period of time. Daily papers are in competition with present day online media as shown in Fig. 1. Among online media sites, feeds aggregators show up to be more significant [5] [6]. An Outsell determination (2009) [5], 57% of feeds media clients go to computerized sites, and they are too more likely to turn to an aggregator 31 % than to a daily paper location 8% or other news sites 18% [5] Feeds aggregator combines news data, and regularly briefs it in a good format and design for the reader, from various sites, newspapers, and agencies [2]. News aggregators are frequently included in classifications such as "Websites Each Engineer Ought to Visit" [7]. Despite the pros of the presence of lots of information to the people through the internet, it will get us another problem which is information overload. There will be too much information that is in front of the user and might not be his interests [8]. This problem can be solved throughout the proposed system. As News Aggregator looks like a gateway that integrates different feeds websites, it organizes feeds by subject [9]. It could be a site that takes data and news from numerous sources and displays it in a single site [10]. Which simplifies readers' search and reading time for news by gathering content based on viewing history [11]. Using news aggregation is one of the best ways to stay on top of the news and topics you want. They offer convenience and time-saving features [12]. News Aggregator

system will have a major requirement which is Summarization. Summarizing articles from various sources talking about the same event then writing the content of this event on one summarized page with all perspectives [13]. Summarization is to create a shorter and smaller form of a text by protecting its meaning and the key substance of the initial content [14] [15]. So, summarization has a lot of pros like reducing the time of reading to the user and getting only useful and real news. Content summarization methods can be categorized into Extractive summaries and Abstractive summaries [16]. Extractive Summarization depends on extracting a few parts, such as phrases and sentences, from a piece of text and gather them together to form a summary. Therefore, identifying the right sentences for summarization is of the most extreme importance in an extractive method [17]. But Abstractive Summarization utilizes advanced NLP methods to generate an completely new summary. A few parts of this summary may not indeed appear within the original text [17]. In this paper, we follow extracting summarization technique which gives better output and right sentence for Summarization. The rest of the paper is structured as follows: Section II reviews related work on News aggregator based on summarization. The proposed system is presented in Section III. Section IV presents experimental results about the summarization and performance analysis of the system. Finally, Section V concludes the paper and discusses some future work.

II. RELATED WORK

The authors were focusing on gathering news using matrix-based analysis (MNA) with 5 main steps as follows: the first and second steps are data gathering and extracting the article from the websites and save it in the database. The third step is grouping where they categorize the articles. The last two steps are summarization and visualization that view the important article to the user. Before the grouping step, they added the matrix-based analysis where the matrix has entity as row and the column is the states about the entities. When starting analysis, the user defines what he's looking for where MNA prepare the default values for this purpose. After that, the initialization of the matrix extends a matrix over the two required chosen dimension and look in each cell for the cell documents. The summarization phase is done according to the following steps: topic summary, cell summary and summarizing both by using TF-IDF for each cell in the matrix. According to [11], the authors were aiming to accumulate the content from diverse websites such as articles fond moreover news headlines from blogs and websites. The belief that Rich Site Summary (RSS) gives us summarized and short data. Which is preferable for the news aggregator that they are still a successful solution for indexing articles. As reducing the time required for visiting some websites, subscribed users can quickly utilize Rich Site Summary feeds without wasting time going to numerous websites. Creating HTTP requests from the web-server is the primary step in the application and these requests are received from clients. At that point, they utilize Python to download Rich Site Summary feeds and extract articles from it according to the input. After periods of time, the web-server gives some requests to the subscribed users and in case there are any upgrades, it'll be stored and downloaded. Author in [19] was aiming to use Rich Site Summary integrated with HTML by using wrappers (programs) and parser in order to extract the information from a specific source, then adjust them according to news categories and personalized web views via a web-based interface. They explain how they do the content scanner by using HTML and Rich Site Summary. The first step is wrapping (HTML/Rich Site Summary wrapper) which involves identifying the URL address of the new items from the source with category per the news, and the address is stored in the database as for each category pair and also combined with the corresponding wrapper. The second step of wrapping is getting information from the new items, that will be used for getting and indexing the article, for each article they obtain the first sentence and pass it to the corresponding HTML page. According to [9], the authors were aiming to collect the news from multiple sites, newspapers, magazines, and television and merge them all in one summarized website. It progresses the goodness of results because the contents and data in it are brief and summarized. So, their work based on the Rich Site Summary fetcher for recovering Rich Site Summary reports from specific websites at a certain time. They also use web Crawling (Scraping) besides Rich Site Summary to get more accurate results. Web scraping may be a method utilized to collect huge amounts of information from websites. From all the above mentioned researches on the news aggregator, the quality of the aggregator system is still an open area to be introduced.

III. FUTURE SCOPE AND ENHANCEMENT

1) Customization & Personalization: Applying algorithms to comprehend user preferences and provide news

information that is specifically catered to them. This might entail looking at previous actions, direct user comments, and current contextual data.

2) AI-driven Curation: This method of selecting news stories ensures high relevancy, accuracy, and diversity of sources by combining artificial intelligence and natural language processing. AI is also useful for fact-checking and weeding out false information.
3) Multimedia Integration: Combining standard articles with different multimedia formats, like podcasts, interactive graphics, and videos, to accommodate a wide range of audience preferences.
4) Real-time Updates and Alerts: By giving consumers access to immediate updates and alerts on breaking news events, users can keep up with the most recent developments as they happen.
5) Community Engagement: Encouraging conversation and interaction among users by providing tools such as forums, comments, and user-generated content.
6) Cross-platform Integration: To offer a consistent user experience across several devices, cross-platform integration seamlessly integrates with a variety of platforms and devices, such as web browsers, mobile apps, smart speakers, and wearable technology.
7) Data Security and Privacy: Making user privacy and data security a top priority through the use of strong encryption protocols, anonymization strategies, and open data management procedures.
8) Monetization Strategies: Investigating cutting-edge monetization techniques that strike a balance between the requirement to generate income and the user experience, such as sponsored content, premium content offers, subscription models, and targeted advertising.
9) Integration with social networking platforms: enabling users to interact with their social networks, follow popular topics, and share news stories straight from the aggregator.
10) Emerging Technologies: Using cutting-edge tools like decentralized publishing platforms and blockchain for content verification to guarantee transparency.

IV. METHODOLOGY

1) News Aggregator:

a) Data Sources: Determine the news sources you want to aggregate. You can use APIs provided by news organizations or RSS feeds.
b) Fetching News: Implement a mechanism to fetch news articles from the chosen sources. Use libraries like requests or dedicated news API libraries to retrieve news data.
c) Data Storage: Decide how you want to store the fetched news articles. You can use a database like MySQL, PostgreSQL, or MongoDB to store the data.
d) Parsing and Processing: Extract relevant information from the news articles, such as headlines, summaries, publication date, and article URLs. Libraries like BeautifulSoup can help with parsing HTML content.
e) Categorization and Filtering: Implement logic to categorize news articles based on topics, keywords, or sources. Apply any filters you want to exclude certain articles.
f) Displaying News: Design a user interface to present the aggregated news to users. This can be a web application using a framework like Flask or Django.

2) Authentication:

a) User Registration: Implement a user registration mechanism where users can create accounts. Store user details, such as usernames and hashed passwords, in a database.
b) User Login: Create a login system where users can authenticate themselves using their credentials. Verify the provided username and password against the stored values.
c) Session Management: After successful login, manage user sessions to maintain authentication state. You can use cookies or tokens for session management.
d) Access Control: Determine which parts of the news aggregator require authentication. For example, you might restrict certain actions like commenting or favoriting articles to authenticated users only.

e) User Roles and Permissions: If you have different user roles (e.g., admin, regular user), implement role-based access control to define different levels of permissions.

3) **Security Considerations:**

a) Secure Password Storage: Hash and salt user passwords before storing them in the database. Use a library like bcrypt for secure password hashing.

b) Protection Against Cross-Site Scripting (XSS): Sanitize user input and ensure proper encoding when displaying user-generated content.

c) Protection Against Cross-Site Request Forgery (CSRF): Implement CSRF tokens to protect against CSRF attacks.

d) Secure Communication: Use HTTPS for secure communication between the user's browser and your web application. Obtain an SSL certificate for your domain.

e) Rate Limiting: Implement rate limiting mechanisms to prevent abuse and protect your application from excessive requests.

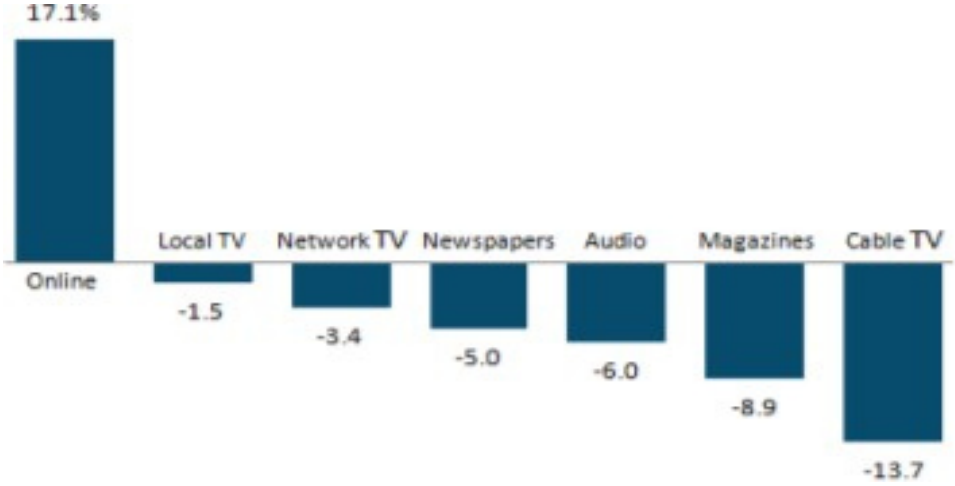

Figure 1: Fig. 1. Online feeds Develops Quickly contrast to Others

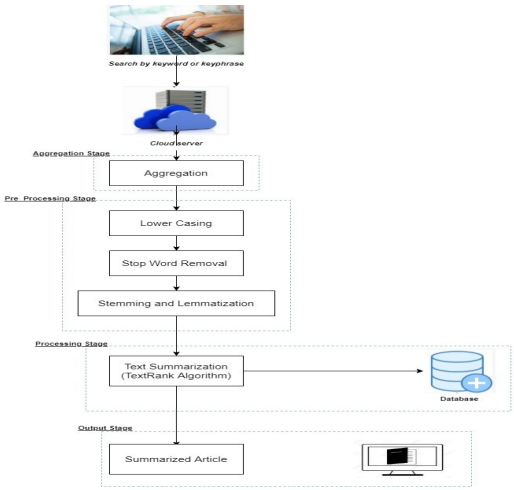

Figure 2. Proposed System Overview

V. RESULT AND DISCUSSION

Result:

Set up project dependencies: Create a new Python project and set up the necessary dependencies. You can use Flask or Django as a web framework, depending on your preference.
User authentication: Implement a user authentication system to handle user registration, login, and logout. You can use a library like Flask-Login or Django's built-in authentication system for this purpose.
News sources: Identify the news sources you want to aggregate. You can use various news APIs such as News API, GNews API, or RSS feeds from different news websites. Register for API keys or identify the RSS feeds you need to access.
Fetching news: Write code to fetch news articles from the selected news sources using the API keys or RSS feeds. Use appropriate libraries like requests or built-in modules like urllib to make HTTP requests and retrieve the news data.
Store news data: Decide how you want to store the news data. You can use a database like SQLite, MySQL, or PostgreSQL to store the news articles and associated information such as the source, timestamp, and category.
Aggregation and filtering: Develop a mechanism to aggregate the news articles from different sources and apply filters based on user preferences or categories. You can use data structures like lists or dictionaries to store and organize the news data.
User interface: Design and create the user interface for your news aggregator using HTML, CSS, and JavaScript. Integrate it with your Python backend using the chosen web framework (Flask or Django).
Authentication integration: Integrate the user authentication system with your user interface. Implement login and registration forms, and handle user sessions to allow authenticated users to access the news aggregator.
Display news articles: Create views or endpoints in your backend to serve the aggregated news data to the frontend. Customize the presentation of news articles based on your design requirements using HTML templates or a frontend framework like React or Vue.js. Real-time updates: To make your news aggregator live, you can consider implementing real-time updates using technologies like WebSockets or Server-Sent Events. This way, new articles can be pushed to the frontend as they are published by the news sources.
Testing and deployment: Test your application thoroughly to ensure it functions as expected. Deploy your application to a web server or a cloud platform such as Heroku or AWS to make it accessible to users.

VI. Discussion:

Data Sources: Identify the news sources you want to aggregate from. This can include RSS feeds, APIs, or web scraping from news websites. Choose reliable and credible sources to ensure the quality of the news content.
Fetching and Parsing News: Use libraries like feedparser or requests to fetch news data from RSS feeds or APIs. If scraping from websites, you can utilize tools like BeautifulSoup or Scrapy to extract relevant information from HTML pages.
Data Storage: Decide on the database to store the news data. You can use popular choices like SQLite, MySQL, or PostgreSQL. Design a schema to store the necessary details such as news headlines, descriptions, sources, publication dates, etc.
Authentication: Choose an authentication method, such as username/password or token-based authentication. Popular libraries like Flask-Login or Django's built-in authentication can be used for user management and session handling. Implement registration, login, and logout functionality to secure access to the news aggregator.
User Roles and Permissions: Define different user roles (e.g., admin, regular user) and associated permissions (e.g., editing, deleting news). Use authorization libraries like Flask-User or Django's permission system to manage user roles and access control.

User Interface: Design and develop a user-friendly interface to display the aggregated news. Utilize HTML, CSS, and front-end frameworks like Bootstrap or React for building responsive and visually appealing UI components.

Real-time Updates: Implement techniques like WebSockets or Server-Sent Events (SSE) to provide real-time updates for new articles or breaking news. You can use libraries like Flask-SocketIO or Django Channels to handle real time communication between the server and clients.

Error Handling and Logging: Implement proper error handling to gracefully handle exceptions and display meaningful error messages to users. Configure logging to record application activities and errors for debugging purposes.

Security: Implement measures like input validation, parameterized queries, and cross site scripting (XSS) prevention to protect against common security vulnerabilities. Use HTTPS and enforce secure communication between the server and clients to ensure data privacy.

Deployment and Scaling: Deploy your application on a production server using platforms like Heroku, AWS, or DigitalOcean. Consider scaling techniques such as load balancing and caching to handle increasing traffic and improve performance.

VI. KEY OBSERVATION

Real-time updates on a range of events can be obtained through live news aggregators, which present a thorough picture of current affairs. They frequently compile information from various sources, allowing users to remain up to date on current affairs, breaking news, and popular subjects. These resources can be quite helpful for keeping up with new developments, fashions, and global happenings.

A thorough understanding of current affairs can be obtained from live news aggregators, which can deliver real-time information on a variety of events. They frequently compile content from various sources, allowing visitors to remain up to date on breaking news, popular subjects, and world events. These resources can be extremely helpful for keeping up with growing narratives, new fashions, and important global events. They lower the danger of bias from single-source reporting by combining news from several sources to provide users a more comprehensive view of events. People are now able to create more educated opinions and make better decisions about the world around them because to the democratization of knowledge. Furthermore, real-time interaction with events is made possible by live news aggregators, which create a sense of community and connection among users who are actively monitoring and debating the most recent developments.

The role that live news aggregators play in improving accountability and transparency is another important feature. These platforms can draw attention to disparities or inconsistencies in reporting by compiling news from several sources, which can encourage more in-depth examination and research into particular topics. Furthermore, user-generated information and eyewitness reports are frequently accessible through live news aggregators, enhancing the news landscape with a variety of viewpoints and firsthand views. This openness can hold institutions responsible for their activities and promote a more active public conversation. Additionally, real-time trend tracking, pattern analysis, and public sentiment monitoring are made possible by live news aggregators, which provide scholars, journalists, and politicians with an invaluable tool for making better decisions and taking preventive measures in the face of new difficulties.

VII. OBERVATION DETAILS:

Real-time News Aggregation: With the increasing volume and speed of news production, there is a growing need for real-time news aggregation platforms. Python can be used to build robust systems that collect news from various sources, curate and filter them based on user preferences, and provide timely updates. Future developments may include improved natural language processing (NLP) techniques for better categorization and sentiment analysis of

news articles.

Personalized News Delivery: News aggregation platforms can leverage Python to provide personalized news recommendations to users based on their interests, browsing history, and social media activities. Future advancements may involve the integration of machine learning algorithms to understand user preferences and deliver tailored news content more accurately.

Data Visualization: Python offers powerful libraries like Matplotlib, Plotly, and Seaborn for data visualization. Future developments may focus on enhancing the visual representation of news data, such as interactive charts, maps, and graphs, to provide users with a more immersive and engaging news consumption experience.

Fact-Checking and Authenticity Verification: Fake news and misinformation are significant challenges in the digital age. Python can be used to develop algorithms for fact-checking news articles and verifying their authenticity. Future advancements may include the integration of advanced machine learning techniques, such as deep learning and natural language understanding, to detect and combat fake news more effectively.

Trust and Transparency: Ensuring trust and transparency in news aggregation platforms is crucial. Python can enable the implementation of blockchain technology to create immutable records of news sources, authors, and publication timestamps. This can help users verify the credibility and origin of news articles. Future developments may involve the integration of decentralized identity systems and smart contracts to enhance trust and transparency further.

Social Media Integration: Many news stories break first on social media platforms. Python can be used to integrate social media APIs and collect real-time updates from platforms like Twitter, Facebook, and Instagram. Future advancements may involve the development of sentiment analysis algorithms to gauge public reactions to news stories and provide more comprehensive coverage.

VIII. REFERENCES

[1] J. Gottfired and E. Shearer, "News use across social media platforms 2016," p. 19, 2016.

[2] C. S. Lee and L. Ma, "News sharing in social media: The effect of gratifications and prior experience," Computers in Human Behavior, vol. 28, no. 2, pp. 331–339, 2012.

[3] M. Brown, "Abandoning the news," Carnegiereporter Spring 2005, pp. 1–16, 2005.

[4] A. Jones, Losing the news: The future of the news that feeds democracy. New York: NY: Oxford University Press, 2008.

[5] T. Patterson, "Young people and news," Joan Shorenstein Center on the Press Politics and Public Policy Report Harvard University,2007.[Online].Available:http://www.hks.harvard.edu/presspol/research/carnegieknight/young_people_and_news_2007.pdf

[6] A. Casero-Ripollés, "Beyond newspapers: News consumption among young people in the digital era," Comunicar, vol. 20, no. 39, pp. 151–158, 2012.

[7] D. T. Mindich, "Tuned out: Why Americans under 40 don't follow the news," Journal of Communication, vol. 57, no. 1, pp. 175–176, 2007. [Online]. Available: http://onlinelibrary.wiley.com/doi/10.1111/j.1460-2466.2006.000339_2.x/abstract

[8] A. Mitchell and D. Page, "State of the News Media 2015," Pew Research Center, vol. 53, no. 9, pp. 1–97, 2015. [9] K. G. Barnhurst and E. Wartella, "Newspapers and citizenship: Young adults' subjective experience of newspapers," Critical Studies in Mass Communication, vol. 8, no. 2, pp. 195 209, 1991.

[9] K. Raeymaeckers, "Newspaper editors in search of young readers: content and layout strategies to win new readers," Journalism Studies, vol. 5, no. 2, pp. 221–232, 2004. [Online]. Available: http://www.tandfonline.com/doi/abs/10.
1080/1461670042000211195%5Cnhttp://www.tandfonline.com.proxy1.lib.umanitoba.
ca/action/doSearch?type=simple&filter=multiple&stemming=yes&searchText=page+
layout&publication=&searchType=journals&ajaxpagination=standard&a 97 BIBLIOGRAPHY

[10] I. Costera Meijer, "The paradox of popularity. How young people experience the News," Journalism Studies, vol. 8, no. 1, pp. 96–116, 2007.

[11] Usha Kosarkar, Gopal Sakarkar, Shilpa Gedam (2022), "An Analytical Perspective on Various Deep Learning Techniques for Deepfake Detection", 1st International Conference on Artificial Intelligence and Big Data Analytics (ICAIBDA), 10th & 11th June 2022, 2456-3463, Volume 7, PP. 25-30, https://doi.org/10.46335/IJIES.2022.7.8.5

[12] Usha Kosarkar, Gopal Sakarkar, Shilpa Gedam (2022), "Revealing and Classification of Deepfakes Videos Images using a Customize Convolution Neural Network Model", International Conference on Machine Learning and Data Engineering (ICMLDE), 7th & 8th September 2022, 2636-2652, Volume 218, PP. 2636-2652, https://doi.org/10.1016/j.procs.2023.01.237

[13] Usha Kosarkar, Gopal Sakarkar (2023), "Unmasking Deep Fakes: Advancements, Challenges, and Ethical Considerations", 4th International Conference on Electrical and Electronics Engineering (ICEEE),19th & 20th August 2023, 978-981-99-8661-3, Volume 1115, PP. 249-262, https://doi.org/10.1007/978-981-99-8661-3_19

[14] Usha Kosarkar, Gopal Sakarkar, Shilpa Gedam (2021), "Deepfakes, a threat to society", International Journal of Scientific Research in Science and Technology (IJSRST), 13th October 2021, 2395-602X, Volume 9, Issue 6, PP. 1132-1140, https://ijsrst.com/IJSRST219682

[15] Usha Kosarkar, Gopal Sakarkar (2024), "Design an efficient VARMA LSTM GRU model for identification of deep-fake images via dynamic window-based spatio-temporal analysis", International Journal of Multimedia Tools and Applications, 8th May 2024, https://doi.org/10.1007/s11042-024-19220-w

[16] Usha Kosarkar, Gopal Sakarkar, Shilpa Gedam (2022), "An Analytical Perspective on Various Deep Learning Techniques for Deepfake Detection", *1st International Conference on Artificial Intelligence and Big Data Analytics (ICAIBDA),* 10th & 11th June 2022, 2456-3463, Volume 7, PP. 25-30, https://doi.org/10.46335/IJIES.2022.7.8.5

[17] Usha Kosarkar, Gopal Sakarkar, Shilpa Gedam (2022), "Revealing and Classification of Deepfakes Videos Images using a Customize Convolution Neural Network Model", *International Conference on Machine Learning and Data Engineering (ICMLDE)*, 7th & 8th September 2022, 2636-2652, Volume 218, PP. 2636-2652, https://doi.org/10.1016/j.procs.2023.01.237

[18] Usha Kosarkar, Gopal Sakarkar (2023), "Unmasking Deep Fakes: Advancements, Challenges, and Ethical Considerations", *4th International Conference on Electrical and Electronics Engineering (ICEEE)*,19th & 20th August 2023, 978-981-99-8661-3, Volume 1115, PP. 249-262, https://doi.org/10.1007/978-981-99-8661-3_19

[19] Usha Kosarkar, Gopal Sakarkar, Shilpa Gedam (2021), "Deepfakes, a threat to society", *International Journal of Scientific Research in Science and Technology (IJSRST)*, 13th October 2021, 2395-602X, Volume 9, Issue 6, PP. 1132-1140, https://ijsrst.com/IJSRST219682

[20] Usha Kosarkar, Prachi Sasankar(2021), " A study for Face Recognition using techniques PCA and KNN", Journal of Computer Engineering (IOSR-JCE), 2278-0661,PP 2-5,

[21] Usha Kosarkar, Gopal Sakarkar (2024), "Design an efficient VARMA LSTM GRU model for identification of deep-fake images via dynamic window-based spatio-temporal analysis", Journal of Multimedia Tools and Applications, 1380-7501, https://doi.org/10.1007/s11042-024-19220-w

[22] Usha Kosarkar, Dipali Bhende, " Employing Artificial Intelligence Techniques in Mental Health Diagnostic Expert System", International Journal of Computer Engineering (IOSR-JCE),2278-0661, PP-40-45, https://www.iosrjournals.org/iosr-jce/papers/conf.15013/Volume%202/9.%2040-45.pdf?id=7557