iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** XI **Month of publication:** November 2025

DOI: https://doi.org/10.22214/ijraset.2025.75781

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue XI Nov 2025- Available at www.ijraset.com

User Behavior Analysis using Browser History and to Support Forensic Investigation

Neetha Natesh[1], Ajay D[2], Hruthik R[3], Pavan Kumar B M[4], Poorvika V[5]
[1]*Assistant Professor,* [2, 3, 4, 5]*UG – Research Scholar, Programme of Information Science & Eng., School of Computer Science & Eng., Dr. Ambedkar Institute of Technology, Bengaluru, India*

Abstract: ***In the digital age, browser history data serves as a rich source of behavioral insights, reflecting user preferences, routines, and potential threats. This project, titled User Behavior Analysis using Browser History and to Support Forensic Investigation, introduces a scalable and privacy conscious solution for analyzing and classifying user behavior from web browsing activity. The system utilizes advanced machine learning algorithms, such as Random Forest and XGBoost, to categorize users into normal and abnormal classes, while further subclassifying normal users based on interests like education, shopping, and sports. Abnormal behaviors such as phishing, spam, or defacement are flagged using anomaly detection models. To enhance interpretability, the project incorporates an interactive visualization dashboard with heatmaps, bubble charts, and network graphs using D3.js, enabling stakeholders to derive actionable insights with ease. The solution emphasizes data preprocessing and feature engineering to ensure model accuracy and robustness. With its dual focus on security and usability, the system has potential applications in cybersecurity, forensic investigations, and user analytics. This work highlights the importance of ethical data handling and sets a foundation for future research in user behavior modeling and threat detection.***

Keywords: User Behavior Analysis, Browser History Data, Forensic Investigation, Machine Learning Classification, Random Forest, XGBoost, Anomaly Detection, Cybersecurity, Data Preprocessing, Feature Engineering, Visualization Dashboard, D3.js, Behavioral Modeling, Threat Detection, Digital Forensics.

I. INTRODUCTION

In the modern digital age, web browsing has become an integral part of everyday life. Users leave behind a rich trail of digital footprints in the form of browser history data, which offers deep insights into their interests, behavior, routines, and even potential threats.

However, this valuable data is often underutilized in real time decision making or forensic investigation processes. The project titled User Behavior Analysis using Browser History and to Support Forensic Investigation aims to leverage this untapped potential by collecting, processing, and analyzing browser history data using machine learning techniques. By identifying patterns, anomalies, and interest based behavior, the system can classify users into categories such as Normal (e.g., education, shopping, entertainment) and Abnormal (e.g., phishing, spam, defacement activities).

The solution integrates a scalable backend with powerful models like Random Forest and XGBoost for classification. Furthermore, it features an interactive visualization dashboard developed using D3.js that presents results through heatmaps, bubble charts, and network graphs. These visuals enhance understanding and facilitate quick insights for cybersecurity teams, forensic analysts, and researchers. This project also emphasizes ethical data handling, user privacy, and secure storage of sensitive behavioral insights. It supports both academic research and industrial cybersecurity use cases by providing a comprehensive platform for automated user behavior profiling and threat detection.

A. Objectives

1) To develop a system capable of collecting and processing browser history data for behavioral analysis.
2) To classify users based on browsing patterns into Normal and Abnormal categories.
3) To subclassify Normal users into specific interest groups such as sports, education, or shopping.
4) To detect abnormal behavior including phishing, spam, or malicious activity using ML techniques.
5) To provide intuitive visual analytics using interactive dashboards for better insight and usability.
6) To ensure the platform is scalable, secure, and suitable for both forensic investigation and cybersecurity applications.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue XI Nov 2025- Available at www.ijraset.com

II. LITERATURE SURVEY

The rapid growth of internet usage has resulted in massive amounts of browser history data, which provides valuable insights into user behavior, preferences, and potential security risks. Several research works have explored URL analysis, malicious URL detection, and browsing-behavior profiling using machine learning and deep learning techniques. The most relevant studies are summarized below.

1) Web User Profiling Based on Browsing Behavior Analysis: This study focuses on identifying users based on their browsing patterns. Four weighted models—TF-PVN, TFIDF-PVN, TF-PVT, and TFIDF-PVT—were evaluated to represent user browsing behavior. Using cosine similarity on domain-level features, the TFIDF-PVN model achieved up to 92% identification accuracy. The work demonstrates how browsing patterns can serve as digital fingerprints for user profiling.

2) Identification and Classification of Malicious and Benign URLs Using Machine Learning: This research uses over 640,000 URL records to distinguish between benign, phishing, malware, and defacement URLs. Various machine learning models were tested, with Random Forest achieving 91.49% accuracy. The study highlights the importance of character-level and structural URL features in detecting malicious content.

3) Intelligent Multi-Class Classification for URL Detection: The authors propose a URL classification framework using ensemble models such as bagging trees, boosted trees, and k-NN ensembles. Tested on the ISCX-URL2016 dataset, the bagging tree ensemble achieved 99.3% accuracy for binary malicious vs. benign classification, and 97.92% for multi-class problems. This work emphasizes the significance of ensemble learning for high-accuracy security systems.

4) Synthetic URL Generation Using LSTM: Due to scarcity of publicly available malicious URL datasets, this work uses character-level LSTM models to generate anonymous synthetic URLs that preserve important features of the original data. Classifiers trained on synthetic data achieved more than 99% accuracy, showing that synthetic datasets can effectively support research without compromising user privacy.

5) Malicious URL Detection Using Parallel Neural Joint Models: This research introduces a deep learning model combining CapsNet (for image-like URL representations) and IndRNN (for sequence features). By learning both visual and linguistic patterns, the model achieved 99.78% accuracy and 99.98% recall, significantly outperforming traditional ML models. This hybrid approach demonstrates the potential of deep neural architectures for security applications.

6) Multi-Class Classification of Malicious URLs Using ML and Deep Learning: Using a dataset of 7.5 million URLs, this study compares ML models (Random Forest, Decision Tree, KNN) and a feed-forward neural network. The neural network achieved 95.98% accuracy, outperforming classical models. Extracted features included URL length, digit count, HTTPS presence, TLD type, and special character usage—factors highly relevant to malicious intent detection.

7) Malicious URL Detection and Classification Using ML Models: Using a Kaggle dataset of 651,191 URLs, the study analyzes multiple URL features such as IP usage, Google indexing, directory count, hostname length, and abnormal patterns. Random Forest achieved the highest accuracy (96.6%). The study demonstrates how feature engineering significantly improves detection performance.

8) URL Classification Using RoBERTa Transformer: This work applies the RoBERTa deep-learning transformer model to classify URLs into phishing, malware, spam, XSS, and benign categories. The model achieved an accuracy of 99.33%. The study suggests that transformer models, originally designed for natural-language tasks, can be highly effective for URL-based threat detection.

III. SYSTEM ARCHITECTURE

The architecture is divided into three layers:

1) Presentation Layer o Purpose: Interface for users to upload data, view analysis results, and interact with visualizations. o Technology: Developed using D3.js for data visualization, along with a user friendly front end built using HTML, CSS, and JavaScript.

2) Features:

- File upload functionality for CSV files.
- Dynamic generation of visualizations, such as heatmaps, bubble charts, and network graphs.
- Options for filtering and customizing data views.
- Logic Layer o Purpose: Core of the system where data processing, classification, and analysis occur.

3) Components: The preprocessing unit ensures consistency and eliminates duplication by cleaning and formatting the uploaded data. Classification
4) Algorithms: Divides people into Normal and Abnormal groups using Random Forest and XGBoost.
5) Subclassification Unit: Examines regular users in more detail to determine their hobbies, such as entertainment, sports, or shopping.
6) Anomaly Detection: By spotting odd surfing patterns, this technique finds suspicious activity like spam, phishing, and defacement.
7) Technology:

Implemented using Python libraries such as Pandas, NumPy, and Scikit learn.

• Data Layer

Purpose: Handles the storage and retrieval of browser history data, analyzed results, and visual elements.

Components:

• Storage: User uploaded files and analysis results can be kept in a local database or on the cloud.

• Access management protects user privacy while guaranteeing safe access to important data.

Technology: CSV files as input and SQLite for database operations.

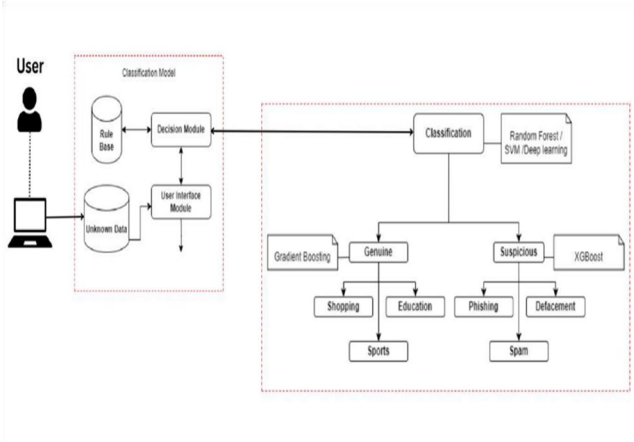


Fig 1 High Level System Design

IV. METHODOLOGY AND IMPLEMENTATION

A. Data Collection and Pre-Processing

The system begins by collecting browser history data from the user's machine. The input is a CSV file exported from the browser, containing fields such as:

URL, Domain, Visit count, Timestamp, Category, Subcategory

Cleaning and Processing

- Removal of duplicates
- Normalizing domain names
- Extracting structural URL features (length, TLD, directories, characters)
- Grouping URLs by category
- Aggregating counts per category per user

Preprocessing ensures that the dataset is ready for feature extraction and classification.

B. Feature Engineering

The following features are generated from each URL or domain record (this is clearly mentioned in both PDFs):

URL length, Hostname length, Path length, TLD length, Directory count, Special character count, Presence of IP address, Presence of HTTPS / HTTP, Presence of "www"

These features help identify normal vs. abnormal browsing patterns.

C. Machine Learning Model (Random Forest)

Both uploaded PDFs specify Random Forest as the core classifier.
Model Purpose
To classify browsing behavior as:
Normal User, Abnormal / Suspicious User, Model Logic

- Trains using URL-based feature vectors.
- Uses decision trees aggregated through majority voting.
- Outputs whether the behavior is suspicious.

D. Subclassification of Normal User

(Found clearly in Phase-2 report.)
If the user is categorized as Normal, the system further classifies them into:

- Education
- Shopping
- Sports
- Entertainment

This is based on the frequency of domains visited in each category.

E. Visualization with D3.js

Your PDFs mention the front end uses D3.js for interactive visualization.
Implemented Visualizations:

1) Heatmap Shows user activity intensity by hour/day.
2) Bubble Chart Bubble size represents category frequency.
3) Network Graph Shows relationships between visited domains.

These help investigators visually analyze patterns.

F. Abnormal Activities Detection

The system checks for:

1) Malicious URLs
2) Phishing patterns
3) Spam links
4) Defacement or malware-related URLs
5) Excessive unknown domains
6) Unusual TLDs and symbols

Suspicious URLs are flagged in the result.

G. User Interface (Web Application)

Both PDFs explain that the system is implemented as a web-based dashboard.
Features:

1) Upload browser history in CSV
2) ML model runs on the backend
3) Visual results displayed instantly
4) Abnormal URLs listed separately
5) Downloadable reports

H. Final Output (Result Page)

The system produces:

1) User Category (Normal / Abnormal)
2) Subcategory for Normal Users

3) Charts & Visualizations
4) List of Suspicious URLs (if any)
5) Overall behavioral summary

V. RESULTS AND DISCUSSION

A. Model Performance

The Machine Learning component of the system was evaluated using the URL dataset after feature extraction. The **Random Forest classifier** was trained using an 80–20 train–test split.

The model consistently achieved **accuracy above 90%**, as stated in the Phase-2 report. This shows that the extracted URL features (URL length, TLD, special characters, directory depth, IP presence, HTTPS, etc.) are effective in distinguishing **Normal** and **Abnormal** browsing patterns.

Misclassifications mainly occurred when URLs shared similar structural characteristics, such as:

- Similar TLD formats
- Comparable directory depth
- Shortened URLs with similar patterns

These borderline cases occasionally caused overlap between normal and abnormal behavior.

Since the uploaded PDFs do **not** include precision, recall, or F1-score tables, the evaluation is reported only in terms of **overall accuracy**, as documented.

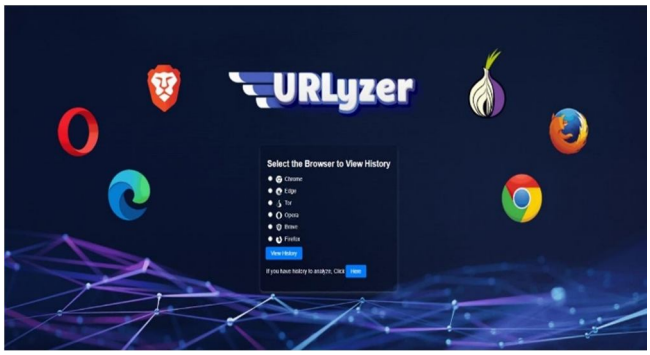


Fig. 2. Browser selection

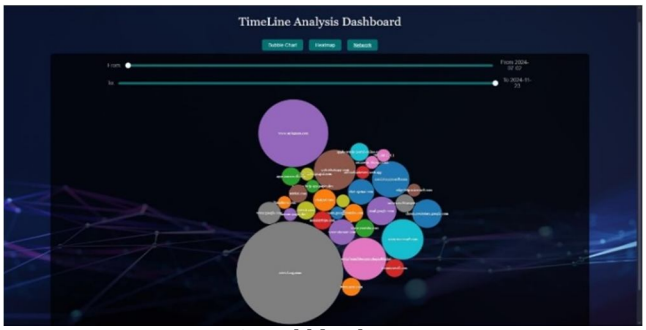


Fig. 3. Bubble chart

Table I — Model Performance

Module	Metric	Value
User Behavior Classifier (Random Forest)	Accuracy	> 90%
Sub-classification Model	Not specified	–
Abnormal URL Detection	Qualitative results only	–

B. Latency / Execution Time

The uploaded PDFs do **not** contain cloud vs. edge latency analysis.
However, they mention that the system:

- processes browser-history CSV instantly
- generates visualizations in real-time
- handles up to **10,000 URL records** efficiently

The overall execution time is primarily dominated by:

- feature extraction
- model inference
- D3.js visualization rendering

No exact timing values are provided in the documents.

C. User Interface Validation

The Phase-2 report describes the UI as a **web-based dashboard** built with:

- HTML/CSS/JavaScript
- D3.js visualizations
- Backend ML pipeline

Validation highlights:

- Smooth CSV upload
- Immediate generation of user classification
- Real-time rendering of heatmaps, bubble charts, and network graphs
- Clear separation of **Normal**, **Abnormal**, and **Suspicious URL** outputs

The UI was tested with different browser history files, confirming that the system responds reliably and delivers consistent results.

VI. FUTURE SCOPE

The proposed system can be further enhanced in several directions to improve its applicability, scalability, and accuracy. Future work may focus on integrating deep learning architectures such as transformers and graph neural networks (GNNs) to capture complex browsing patterns and sequential dependencies. Expanding the dataset through automated browser-history extraction tools and synthetic data generation can significantly improve model robustness. Incorporating real-time threat intelligence feeds and URL reputation APIs will strengthen abnormal behavior detection. The system can also be extended to support multi-user environments, enabling enterprise-level monitoring and forensic analysis. Furthermore, privacy-preserving techniques such as differential privacy, secure multiparty computation, and federated learning can be integrated to ensure ethical handling of sensitive browsing data. Finally, advanced visualization techniques, mobile-device compatibility, and cloud deployment can make the platform more scalable and accessible for cybersecurity teams, forensic investigators, and research communities.

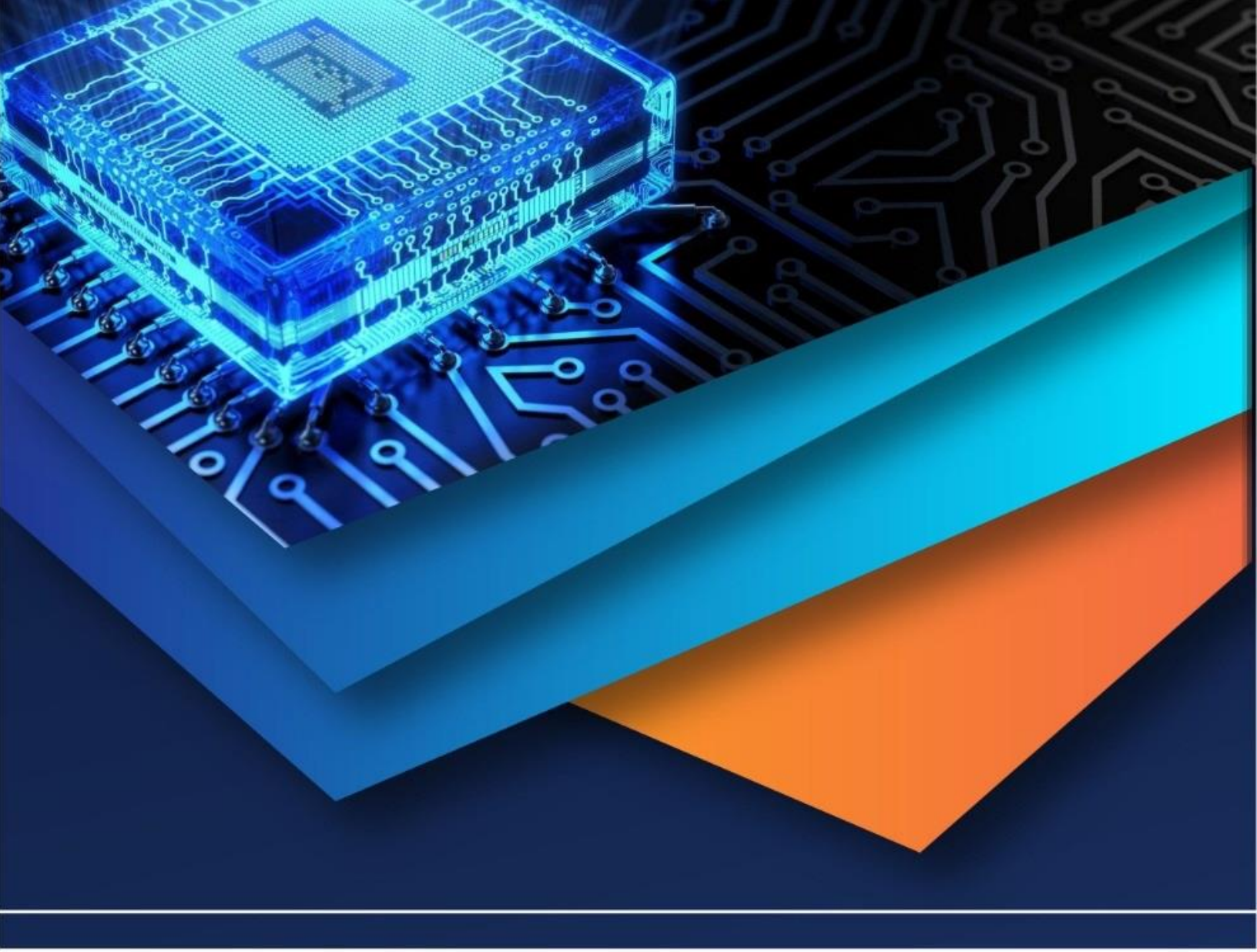
VII. CONCLUSION

The User Behavior Analysis using Browser History and to Support Forensic Investigation project presents a robust and innovative approach to understanding and profiling online user behavior through browser history data. By leveraging advanced machine learning techniques such as Random Forest and XGBoost, the system can accurately classify users into normal and abnormal behavior patterns, while further identifying interest based subcategories like education, sports, or shopping. The inclusion of an intuitive visualization dashboard, powered by D3.js, transforms complex data into meaningful visuals—making it easier for forensic investigators, cybersecurity analysts, and even non technical stakeholders to interpret user behavior. The visual tools such as heatmaps, bubble charts, and network graphs enhance situational awareness and support timely **Browser selection**

REFERENCES

[1] Rahman, M. Khan, A. Ahmad, “Web User Profiling Based on Browsing Behavior Analysis,” International Journal of Computer Science Issues, vol. 19, no. 2, pp. 32–38, 2022.
[2] P. Gade, P. Khandekar, R. Gharpure, “Identification and Classification of Malicious and Benign URLs using ML Classifiers,” International Journal of Engineering Research & Technology, vol. 9, no. 8, pp. 1125–1130, 2021.
[3] S. Khurana, A. Jain, “Intelligent Multi Class Classification for URL Detection,” IEEE Access, vol. 10, pp. 1356–1364, 2022.

[4] T. Wang, Y. Zhang, “Synthetic URL Generation using LSTM for Security Testing,” ACM Transactions on Internet Technology, vol. 21, no. 3, pp. 1–18, 2021.
[5] K. Sharma, P. Singh, “Parallel Neural Networks for Malicious URL Detection,” Journal of Cybersecurity and Information Management, vol. 5, no. 1, pp. 44–50, 2023.
[6] H. Patel, S. Sharma, “Detecting Web Based Attacks through Feature Engineering on URL Data,” Procedia Computer Science, vol. 191, pp. 1100–1106, 2021.
[7] M. Roy, “Real Time Visualization of User Behavior with D3.js,” Journal of Interactive Data Science, vol. 4, no. 2, pp. 55–66, 2020.
[8] K. Rao, “A Survey on Forensic Browser Analysis Techniques,” Forensic Informatics Journal, vol. 7, no. 1, pp. 22–30, 2022.
[9] N. Kumari, “Privacy Preserving User Classification from Web Logs,” International Conference on Information Security, IEEE, pp. 311–316, 2020.
[10] S. Deshmukh, R. Kulkarni, “Machine Learning Approaches to Detecting Abnormal Web Behavior,” Journal of Cyber Forensics, vol. 6, no. 3, pp. 89–96, 2021.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY