e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:07/Issue:03/March-2025 Impact Factor- 8.187 www.irjmets.com

WEB DATA MINING TO DETECT ONLINE SPREAD OF TERRORISM

K. Jaya Krishna*1, K. Pavan Kalyan*2, Kakarla Vinay Kumar*3,
Mettu. Manoj*4, Vaibhavee Jain*5

*1,2,3,4Student, Computer Science And Engineering, Parul University, Vadodara, Gujarat, India.

*5Asst. Prof., Computer Science And Engineering, Parul University, Vadodara, Gujarat, India.

DOI: https://www.doi.org/10.56726/IRJMETS70070

ABSTRACT

Terrorism has taken deep roots in certain regions of the world, necessitating proactive measures to counter its spread. The internet serves as a key platform for the dissemination of terrorist propaganda, recruitment, and radicalization. This paper proposes a robust web data mining system to detect and tag online content related to terrorism for human review. Using advanced text mining techniques and machine learning algorithms, such as K-Nearest Neighbors (KNN) and Naive Bayes, the system efficiently identifies harmful content. This research highlights the potential of web data mining to support anti-terrorism and cybersecurity initiatives, providing actionable intelligence to law enforcement agencies and policymakers.

Keywords: Web Data Mining, Online Terrorism, Machine Learning, Knn Classifier, Counterterrorism.

I. INTRODUCTION

The spread of terrorism on the internet has become a significant global issue, with terrorist organizations exploiting online platforms to propagate propaganda, radicalize individuals, and recruit members. This study aims to develop a web data mining system capable of identifying and flagging extremist content. By utilizing text mining techniques, the proposed system analyzes unstructured data, extracts meaningful patterns, and enhances the detection of terrorism-related content. The system addresses existing gaps in accuracy, scalability, and efficiency, offering a valuable tool for combating online terrorism.

II. METHODOLOGY

1. Research and Analysis:

The project employs social network analysis techniques to study the behavior and composition of internet communities engaged in terrorist actions. This involves analyzing connections, interactions, and information flow within networks to identify important nodes, influential actors, and potential threats.

Key areas of focus include:

- Machine Learning Algorithms: Utilizing categorization, clustering, and anomaly detection techniques with algorithms like ensemble methods, support vector machines (SVM), decision trees, and deep learning models to detect terrorism-related content.
- Natural Language Processing (NLP): Adapting techniques such as named entity recognition, sentiment analysis, topic modeling, and semantic analysis to identify extremist language, detect propaganda, and uncover hidden meanings in communications.

2. Data Collection and Preprocessing:

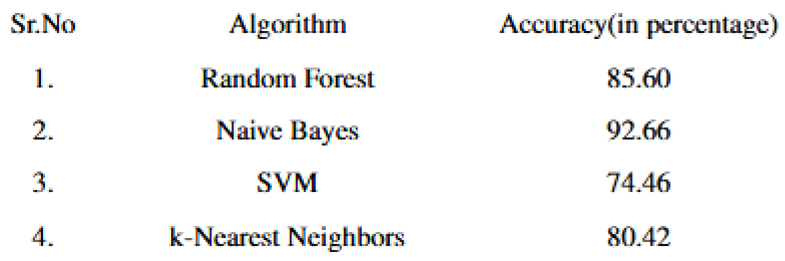
- Data is gathered from social media, forums, blogs, and other online sources using web scraping techniques.
- Preprocessing involves cleaning the data by removing noise, duplicates, and irrelevant metadata. Steps like normalization, tokenization, stop-word removal, and stemming ensure standardized data for analysis.

3. Model Training and Evaluation:

- A K-Nearest Neighbors (KNN) algorithm is implemented for classification.
- Training datasets are manually labeled as "terrorism-related" or "non-terrorism-related."
- Hyperparameter tuning (e.g., selecting optimal values for 'K') is performed using cross-validation to optimize the model's performance.
- Metrics like accuracy, precision, recall, and F1-score are used to evaluate the model's effectiveness.

Evaluation: This modeling approach highlights the integration of machine learning and NLP techniques in detecting and analyzing extremist activities online.

Table 1: Comparison of machine learning algorithms



Sr.No	Algorithm	Accuracy(in percentage)
1.	Random Forest	85.60
2.	Naive Bayes	92.66
3.	SVM	74.46
4.	k-Nearest Neighbors	80.42

Requirement Gathering

- Emphasizes user interviews, surveys, and consultations to shape the project scope and ensure alignment with stakeholder needs.

Design and Planning

- Focuses on designing architecture, wireframes, and UI/UX prototypes while also preparing timelines and resource allocation.

Development

- Covers implementation phases, including frontend, backend, and database development, ensuring all functionalities are integrated.

Quality Assurance and Testing

- **Unit Testing**: Ensures individual components perform correctly.
- **Integration Testing**: Validates seamless interaction between system components.
- **Functional Testing**: Confirms the system meets both technical and user-driven requirements.
- **System Testing**: Verifies overall system reliability and stability.
- **White Box Testing**: Examines internal logic and code quality.
- **Black Box Testing**: Focuses on the system's external behavior without considering its internal structure.

Deployment

- Finalizes production rollout, addressing server configurations, database setup, and code deployment.

Observation and Maintenance

- Continuous monitoring for performance, security updates, and user support, ensuring long-term reliability and alignment with emerging trends.

III. MODELING AND ANALYSIS

Architecture:

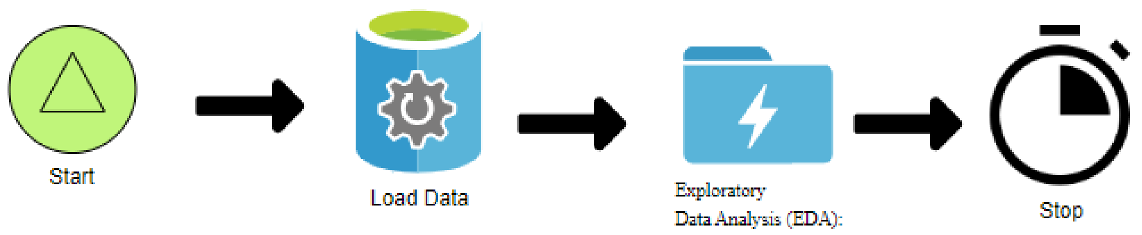


Figure 1: Architecture

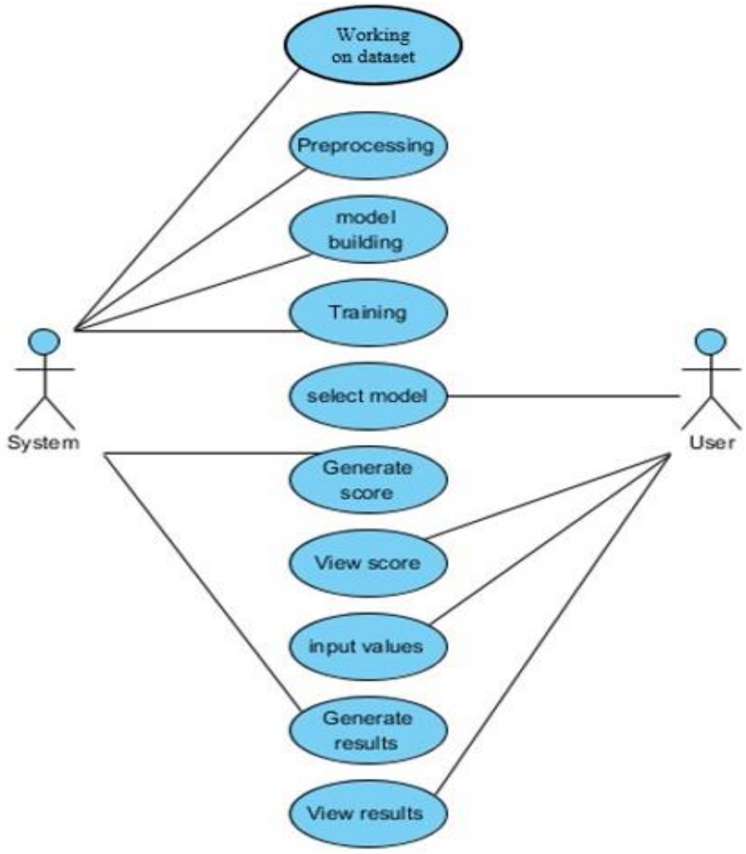


Figure 2: USE CASE DIAGRAM.

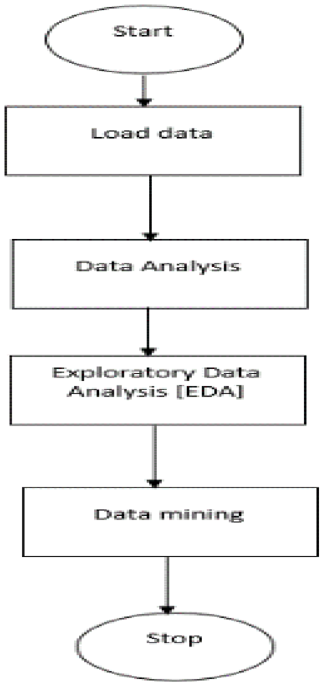


Figure 3: Block Diagram.

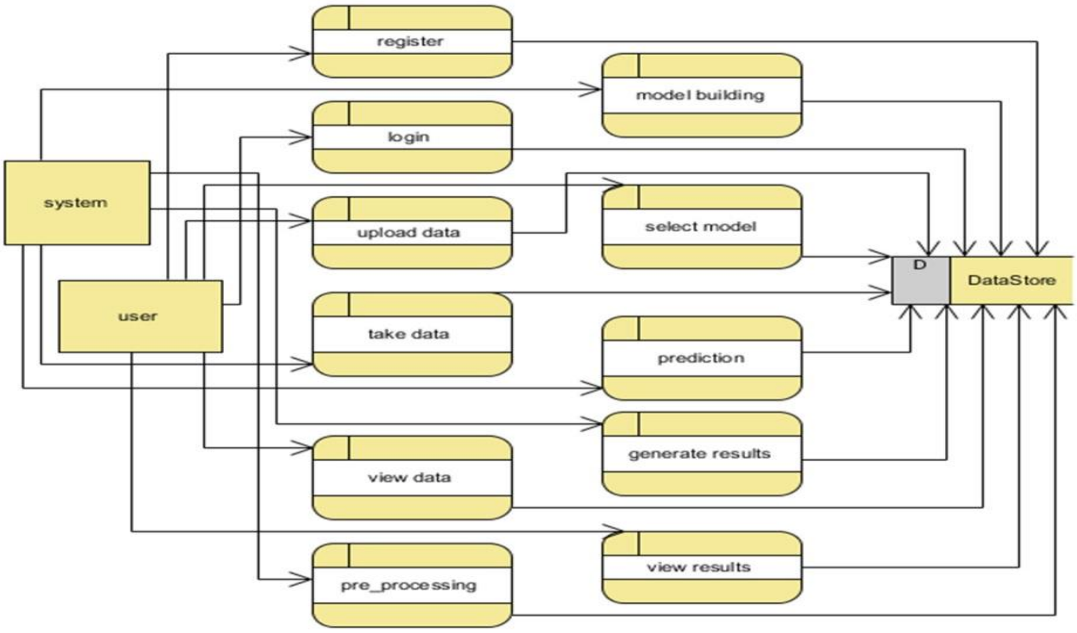


Figure 4: DFD DIAGRAM

IV. RESULTS AND DISCUSSION

System Requirements Specification

Functional and Non-Functional Requirements:

Requirements analysis is critical to the success of a system or software project, dividing requirements into functional and non-functional categories.

- Functional Requirements define the essential features that the system must provide, directly influencing the user experience. These include user authentication, system shutdown during cyber-attacks, and sending verification emails upon user registration.
- Non-Functional Requirements focus on the system's quality attributes, such as portability, security, reliability, performance, and scalability. Examples include ensuring email delivery within 12 hours, processing requests within 10 seconds, and maintaining a site load time of 3 seconds even with over 10,000 simultaneous users.

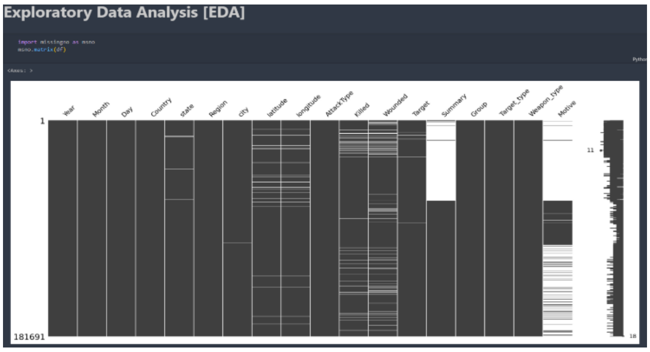
Hardware Requirements:

- Operating System: Windows 7 or later.
- RAM: 8 GB.
- Storage: More than 500 GB (HDD or SSD).
- Processor: Intel 3rd generation or newer, or Ryzen equivalent with 8 GB RAM.

Software Requirements:

- Software: Python 3.6 or higher.
- IDE: Jupyter Notebook or VS Code.

By addressing both functional and non-functional requirements and ensuring appropriate software and hardware configurations, a robust system can be developed, meeting user needs and maintaining high performance and reliability.

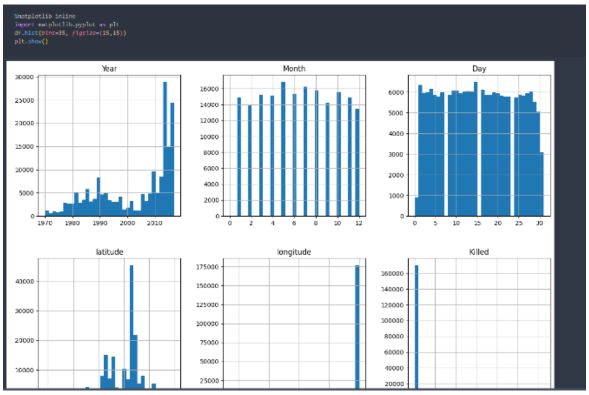


Exploratory Data Analysis [EDA]

```
import missingno as msno
msno.matrix(df)
```

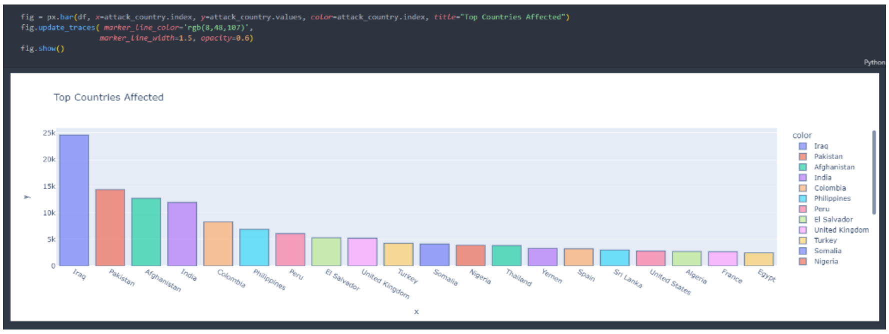
```
<Axes: >
```

Figure 5: Exploratory Data Analysis [EDA]



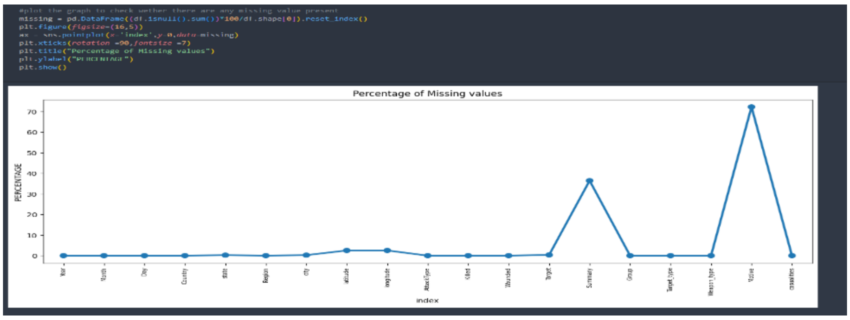
```
%matplotlib inline
import matplotlib.pyplot as plt
df.hist(bins=35, figsize=(15,15))
plt.show()
```

Figure 6: Hist Plot



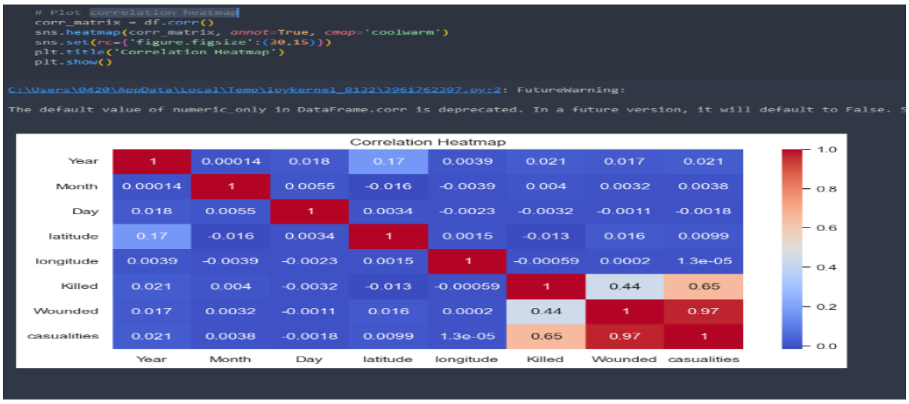
```
fig = px.bar(df, x=attack_country.index, y=attack_country.values, color=attack_country.index, title="Top Countries Affected")
fig.update_traces( marker_line_color='rgb(8,48,107)',
                  marker_line_width=1.5, opacity=0.6)
fig.show()
```

Figure 7: Bar Plot



```
#plot the graph to check wether there are any missing value present
missing = pd.DataFrame((df.isnull().sum())*100/df.shape[0]).reset_index()
plt.figure(figsize=(16,5))
ax = sns.pointplot(x='index',y=0,data=missing)
plt.xticks(rotation =90,fontsize =7)
plt.title("Percentage of Missing values")
plt.ylabel("PERCENTAGE")
plt.show()
```

Figure 8: Missing Value Plot



```
# Plot correlation heatmap
corr_matrix = df.corr()
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm')
sns.set(rc={'figure.figsize':(30,15)})
plt.title('Correlation Heatmap')
plt.show()
```

```
C:\Users\0420\AppData\Local\Temp\ipykernel_8132\3961762397.py:2: FutureWarning:

The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it will default to False.
```

Correlation Heatmap

	Year	Month	Day	latitude	longitude	Killed	Wounded	casualities
Year	1	0.00014	0.018	0.17	0.0039	0.021	0.017	0.021
Month	0.00014	1	0.0055	-0.016	-0.0039	0.004	0.0032	0.0038
Day	0.018	0.0055	1	0.0034	-0.0023	-0.0032	-0.0011	-0.0018
latitude	0.17	-0.016	0.0034	1	0.0015	-0.013	0.016	0.0099
longitude	0.0039	-0.0039	-0.0023	0.0015	1	-0.00059	0.0002	1.3e-05
Killed	0.021	0.004	-0.0032	-0.013	-0.00059	1	0.44	0.65
Wounded	0.017	0.0032	-0.0011	0.016	0.0002	0.44	1	0.97
casualities	0.021	0.0038	-0.0018	0.0099	1.3e-05	0.65	0.97	1

Figure 9: Correlation heat map

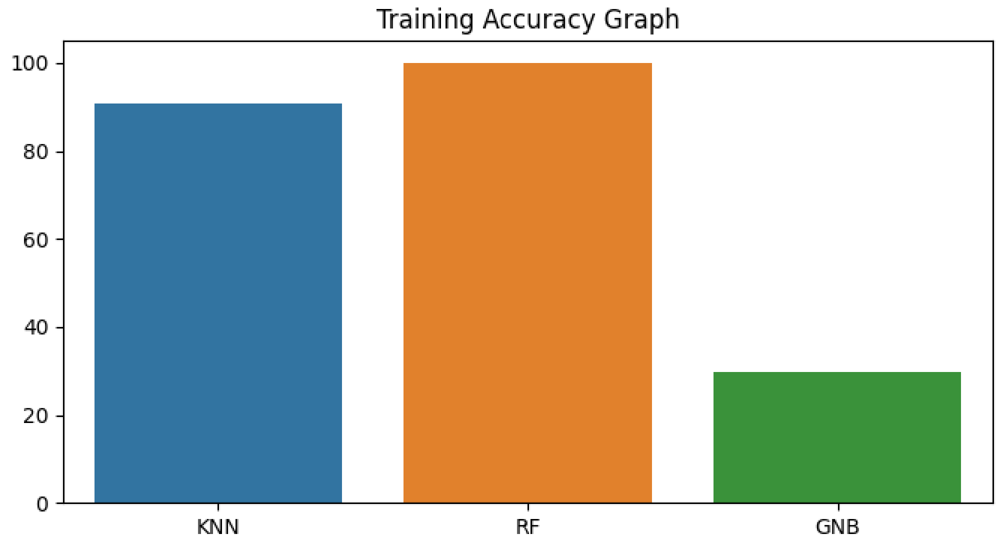


Figure 10: Training Accuracy Graph

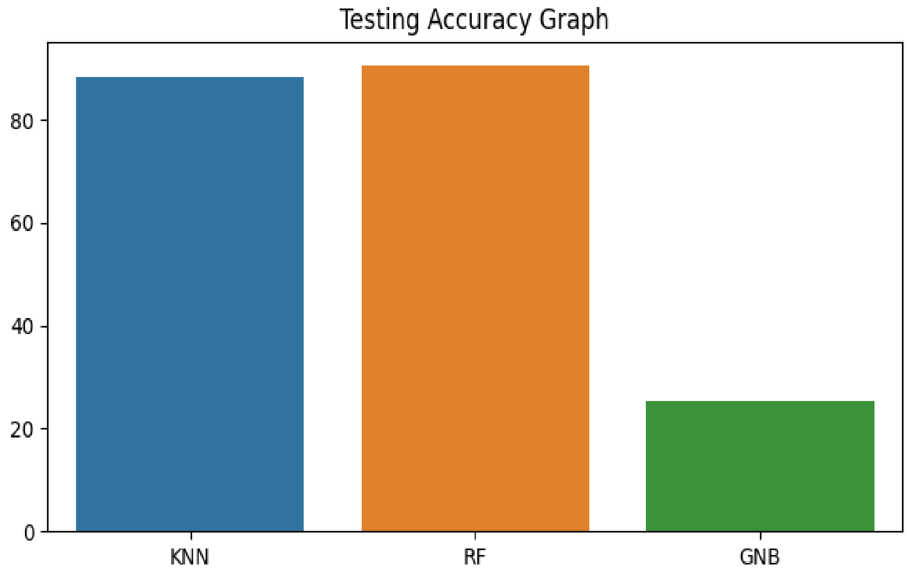


Figure 11: Testing Accuracy Graph

V. CONCLUSION

This research demonstrates the feasibility of using web data mining techniques to detect and flag online terrorism-related content. The proposed system offers a scalable and efficient solution for combating online extremism, supporting law enforcement and cybersecurity initiatives. Future work will focus on enhancing the system to classify content by context (e.g., propaganda or recruitment) and developing real-time monitoring capabilities to track extremist networks more effectively.

VI. FUTURE WORK

The current implementation of the system just checks whether the page has terrorism related content or not. Future enhancement to this project would be to identify with what respect is the content i.e. informative orspreading terrorism. There may be a module where the system itself tracks terrorists using codewords to communicate. The suggested system keeps track of vulnerable IP addresses and gives the offices that use it information. Later on, the system would monitor the specific vulnerable individual sending terrorism-related messages.

VII. REFERENCES

[1] L.K . Joshila Grace, V. Maheswari, Dhinaharan Nagamalai, Analysis of Web Logs and Web User.

[2] Guler, E. R., and Ozdemir, S. (2018). Applications of Stream Data Mining on the Internet of Things: A Survey. 2018 International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism (IBIGDELFT), 51–55.

[3] Baghel, S., and Yogesh. (2018). Detecting Future Terrorism Trend in India Using Clustering Analysis. 2018 7th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), 431–438.

[4] Ramesh Yevale, Mayuri Dhage, Tejali Nalawade,.Trupti Kaule, Unautho- rized Terror Attack Tracking Using Web Usage Mining, (IJCSIT) Inter- national Journal of Computer Science and Information Technologies,ISSN: 0975-9646, Vol. 5 (2) , 2014,1210-1212.

[5] Nisha Chaurasia1, Mradul Dhakar1, Akhilesh Tiwari2 and R. K. Gupta2, A Survey on Terrorist Network Mining: Current Trends and Opportunities, International Journal of Computer Science Engineering Survey (IJCSES) Vol.3, No.4, August 2012.

[6] Agrawal, R. and Srikant, R. (1994) Fast Algorithms for Mining Association Rules in Large Databases. Proceedings of the 20th International Conference on Very Large Data Bases, Santiago de Chile, 12-15 September 1994, 487-499.

[7] Sentiment Analysis in Social Media Texts: Alexandra Balahur.

[8] T.Anand,S.Padmapriya,E. Kirubakaran Terror Tracking Using Advanced Web Mining 2009 International Conference on Intelligent Agent and Multi-Agent Systems. Web mining techniques can be used for detecting and avoiding terror threats caused by terrorists all over the world.

[9] Nasraoui Olfa FH, Joshi A, Krishnapuram R. Mining web access logs using relational competitive fuzzy clustering. The International Fuzzy Systems Association World Congress. Taipei, Taiwan, 17–20 August 1999, 195–204.

[10] Krishnapuram R, Joshi A, Nasraoui O, Yi L. Low-complexity fuzzy relational clustering algorithms for web mining. IEEE Trans Fuzzy Syst 2001,)9): 595–607.

[11] Pal SK, Talwar V, Mitra P. Web mining in soft computing framework: relevance, state of the art and future directions. IEEE Trans Neural Netw 2002,)13): 1163–1177.

[12] Cooley R, Mobasher B, Srivastava J. Grouping web page references into transactions for mining world wide web browsing patterns. IEEE Knowledge and Data Engineering Exchange Workshop. Newport Beach, California, 4 November 1997, 2–9.

[13] A survey of fuzzy web mining† Chun-Wei Lin, Tzung-Pei Hong First published: 18 April 2013.

[14] Detecting and removing web application vulnerabilities with static analysis and data mining I Medeiros, N Neves, M Correia - IEEE Transactions on . . . , 2015 - ieeexplore.ieee.org.

[15] P. N. Tan, M. Steinbach, Vipin Kumar, Introduction to Data Mining, Pearson Education.