iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 11 **Issue:** VI **Month of publication:** June 2023

DOI: https://doi.org/10.22214/ijraset.2023.53671

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 11 Issue VI Jun 2023- Available at www.ijraset.com

Fraud App Detection of Google Play Store Apps Using Decision Trees

Jalapathi Sharath Chandra[1], N. Sharath Chandra[2], B. Santhosh Kumar[3], Dr. M. Nanda Kumar[4]
[1, 2, 3]*Dept. of Electronics and Computer Engineering, Sreenidhi Institute of Science and Technology, Yamnampet, Ghatkesar, Hyderabad, Telangana, India*

Abstract: ***Mobile applications have become an integral part of our lives, but it's challenging to determine their safety and reliability. To address this, a system has been developed that uses ratings, reviews, in-app purchases, and ads as parameters to predict app safety. Three machine learning models – Decision Tree, Logistic Regression, and Naïve Bayes - were compared to evaluate system effectiveness. The Decision Tree model performed the best, with 85% accuracy, an F1 score of 0.815, recall of 0.85, and precision of 0.87. These results highlight the reliability of the Decision Tree model for assessing app safety. Using machine learning models in this way automates evaluation and provides more accurate and consistent results. This system represents a promising step towards ensuring the safety and reliability of mobile applications, which are increasingly important in our daily lives.***

Keywords: Decision Tree, Naïve Bayes, Logistic Regression, Precision

I. INTRODUCTION

Background of the Project: As mobile phones have become an integral part of our lives, the growth of mobile app platforms like Android has created a competitive environment for software developers. To succeed, developers invest time and effort in acquiring clients and improving their products based on user feedback. However, some developers resort to fraudulent practices, like manipulating ratings and comments, which undermines trust.

1) *Methods Used:* To combat fraudulent practices, an automated system is needed to analyse the large volume of comments and ratings for each app. The project proposes a comprehensive fraud detection system that identifies fraudulent applications on popular app stores such as Play Store. The system uses features like in-app purchases, presence of ads, ratings, and reviews to determine the likelihood of an app engaging in fraudulent activities. Data is collected through scraping techniques and various classification models are trained to select the most accurate one.
2) *Applications Used:* The proposed system is applied to app platforms like Play Store or App Store, where fraudulent practices like manipulating ratings are prevalent. It analyses comments, ratings, in-app purchases, and ads associated with each application to identify potentially fraudulent apps. The system utilizes classification models such as Naive Bayes, Logistic Regression, and Decision Tree, which are trained using the collected data. The accuracy of these models is evaluated to determine the most effective one for fraud detection.

By implementing this system, developers, app stores, and consumers can benefit from accurate and reliable feedback, ensuring a fair and trustworthy marketplace for mobile applications

II. EXISTING SYSTEM

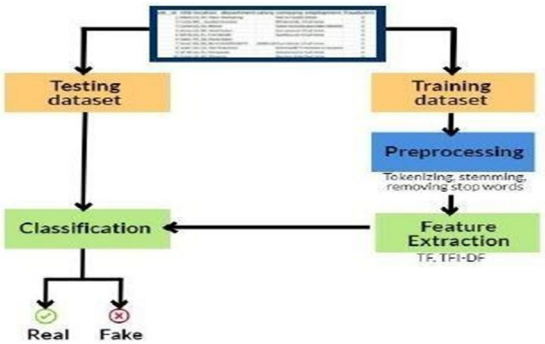


Fig 1: Existing System Flowchart

The first paper, "On Detection of Fraud & Malware Apps in Google Play," provides a comprehensive analysis of fraud and malware in the Google Play store and recommends a
framework for detecting such apps. The authors of this paper tested three verification methods: quality-based guarantees, ratingbased guarantees, and review-based validation.
In contrast, the second paper, "Detection of Fraud Apps using Sentiment Analysis," focuses on using sentiment analysis to detect fraudulent applications. The authors of this paper only consider updates as parameters and use the naive Bayes algorithm to develop their system. The system processes user comments and emotions to identify fraudulent apps.
To improve upon this existing system, one could consider incorporating multiple verification methods, such as those outlined in the first paper, to strengthen the fraud detection framework. Additionally, more advanced machine learning algorithms could be explored to improve the accuracy and efficiency of the sentiment analysis approach. Overall, a combination of different techniques and approaches may yield the most effective fraud detection system for app stores

III. PROPOSED SYSTEM

Our framework for detecting fraudulent mobile applications has demonstrated its effectiveness in addressing the growing concern of fraudulent apps in the market. However, we are aware of certain limitations associated with our system and have implemented measures to overcome them.
One drawback is that our current model relies on four key features: in-app purchases, ads, ratings, and reviews. While these features provide valuable insights, there may be other factors that contribute to app fraudulence that are not considered. To address this, we continuously update our system with fresh data and explore additional parameters to further improve the accuracy of fraud detection.
In terms of methodology, we have employed machine learning models to analyse and predict the likelihood of an app being fraudulent. Among the models tested, the decision tree model has exhibited superior performance compared to others such as recession and naïve bayes. However, we understand that no model is flawless, and different datasets may yield varying results. As a result, we are actively researching and experimenting with ensemble methods, which combine multiple models to achieve even higher accuracy and reliability.
To provide a comprehensive understanding of our framework, we have created a detailed flowchart that illustrates the step-by-step process of fraud detection. This visual representation showcases how the four parameters are integrated within the decision tree model to generate precise predictions. The flowchart serves as a helpful tool for users to grasp the inner workings of our system and comprehend the interactions between different components.
Furthermore, our approach involves conducting experimental analyses using various methodologies to detect fraud or fake applications. We continually evaluate and refine our techniques based on the outcomes of these experiments, with the aim of enhancing the precision and recall rates of our system.
In conclusion, while our framework has proven effective in detecting fraudulent mobile applications, we recognize the importance of ongoing improvements. We are actively addressing the identified limitations by exploring additional parameters, experimenting with ensemble methods, and refining our methodology through experimental analysis. Our ultimate goal is to provide users with a reliable and accurate system that distinguishes between legitimate and fraudulent applications in the app store.

A. Data Flow Diagram

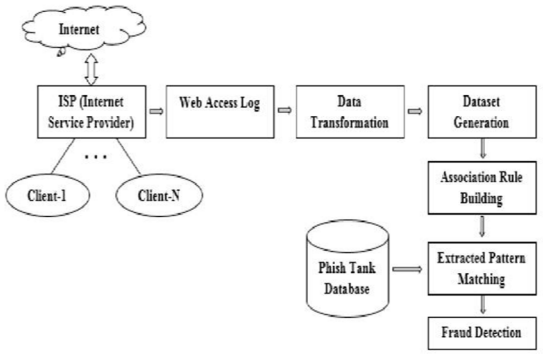


Fig 2: Proposed system Data Flow Diagram

IV. RELATED WORK

Fraudulent activities such as fake reviews and ratings on mobile applications have become a significant concern in the app industry. To combat this, several researchers have proposed various methodologies to detect fraud apps. In this article, we present a summary of eight existing systems for fraud app detection, including their methodology, data sets, advantages, and limitations.

1) *Detection of Fraud Apps using Sentiment Analysis:* Gauri Rao, Shashank Bajaj, Nikhil Nigam, Priya Vandana, and Srishti Singh introduced a system that employs a weight based aggregation method and utilizes the RT dataset. The system is characterized by its speed and accessibility; however, it lacks the concept of neutrality, leading to potential bias.
2) *Ranking Fraud Detection for Mobile Applications:* Nikhila, Deepashree, Jayanthi R, and Jalaja proposed a scalable framework for ranking fraud detection. Their approach utilizes an evidence aggregation method and the RT dataset. One limitation of their system is that existing anomaly detection techniques struggle to extract fraud evidence within a given time period.
3) *Detecting Review Spammers using Rating Behaviors:* Ee-Peng Lim, Viet-An Nguyen, Nitin Jindal, Bing Liu, and Hady W. Lauw proposed a system that employs NLP and SVM on the App_Store (2017) dataset. This system considers the concept of neutrality, making it more scalable. However, it is important to note that not all NLP-generated results are consistently accurate.
4) *Detection of Mobile Applications Leaking Sensitive Data:* Yavuz Canbay, Mehtap Ulker, and Seref Sagiroglu proposed a system that utilizes the J48 classification algorithm and the kag_GPA dataset. Nonetheless, the system encounters challenges in data training, which may result in data setting problems.

In conclusion, each of these existing systems has its advantages and limitations. Researchers need to consider various factors such as accuracy, scalability, time complexity, and dataset size while proposing a fraud detection system for mobile applications.
applications. Our proposed system provides an effective solution to detect fraudulent applications and protect consumers from being scammed. By using machine learning models and analyzing key features, our system can accurately differentiate between legitimate and fraudulent applications on the app store.

V. RESULTS

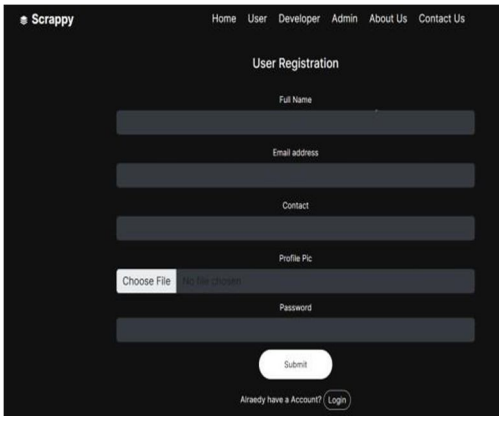


Fig 3: This fig Show the User Registration of Web Page

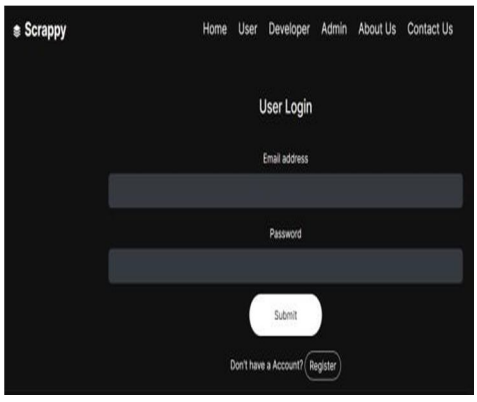


Fig 4: This fig Show the User login of Web Page

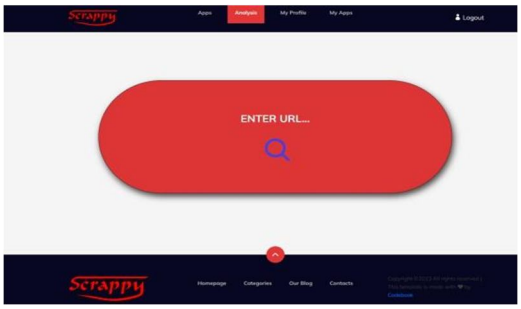


Fig 4: This fig Show the User interface URL page.

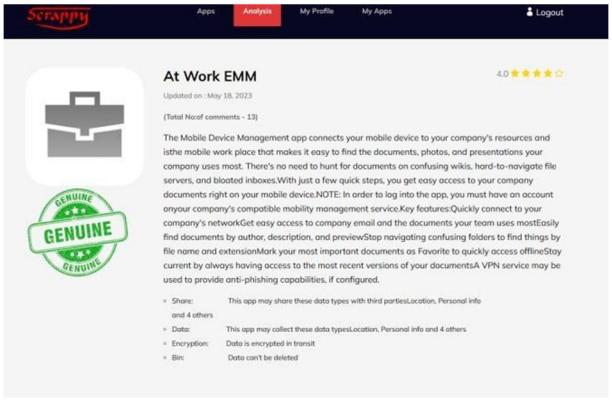
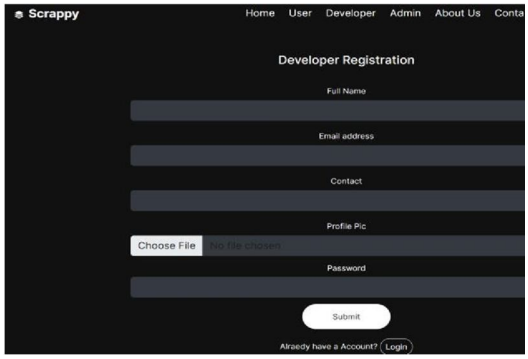


Fig 8: This fig Show the weather the application is Fake or Genuine.



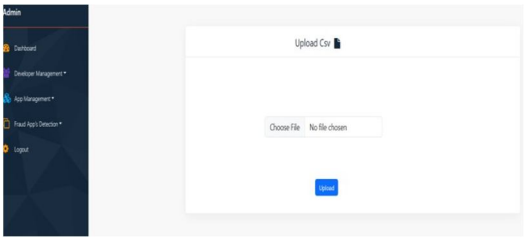


Fig 9: This fig Shows the Admin page, to upload the Data Set taken from online source.

Fig 6: This fig Show the Developer Registration of Web Page

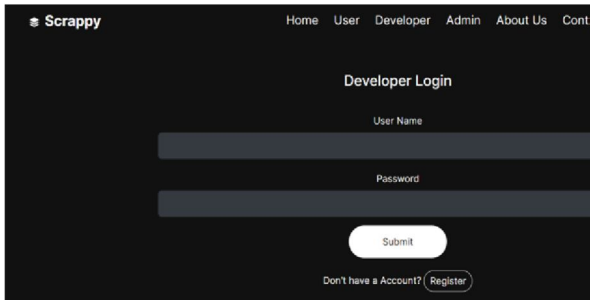


Fig 7: This fig Show the Developer login of web page

Fig 10: This fig Shows the review given by Users. Taken by R

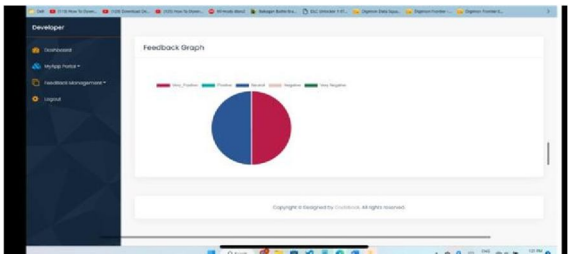


Fig 11: This fig Show the Feedback of the application given by the users in a Graphical

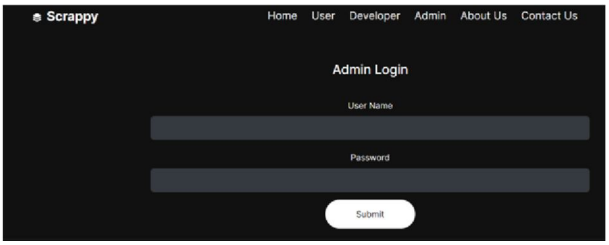
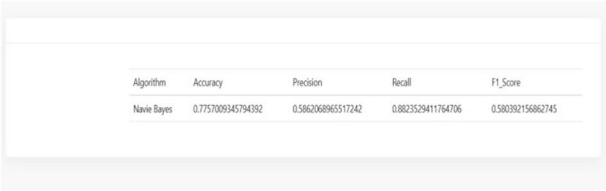


Fig 13: Show the Fig weather the Admin login



Algorithm	Accuracy	Precision	Recall	F1_Score
Navie Bayes	0.7757009345794392	0.5862068965517242	0.8823529411764706	0.5803921568627451

Fig 12: Results of Navie Bayes

Algorithm	Accuracy	Precision	Recall	F1_Score
Logistic Regression	0.7850467289719626	0.603448275862069	0.8861386138613861	0.6071827613727054

Fig 14: Results of Logistic Regression

Algorithm	Accuracy	Precision	Recall	F1_Score
Decision Tree	0.8411214953271028	0.8368700265251989	0.7981220657276995	0.8121837893649975

Fig 15: Results of Decision Tree

A. Compare Table

S No	Technique	Accuracy	Precision	F1 Score	Recall
1	Naïve Bayes algorithm	0.7757	0.5862	0.5803	0.8823
2	Logistic Regression	0.7850	0.6034	0.6071	0.8861
3	Decision Tree	0.8317	0.8196	0.7993	0.7869

VI. CONCLUSION

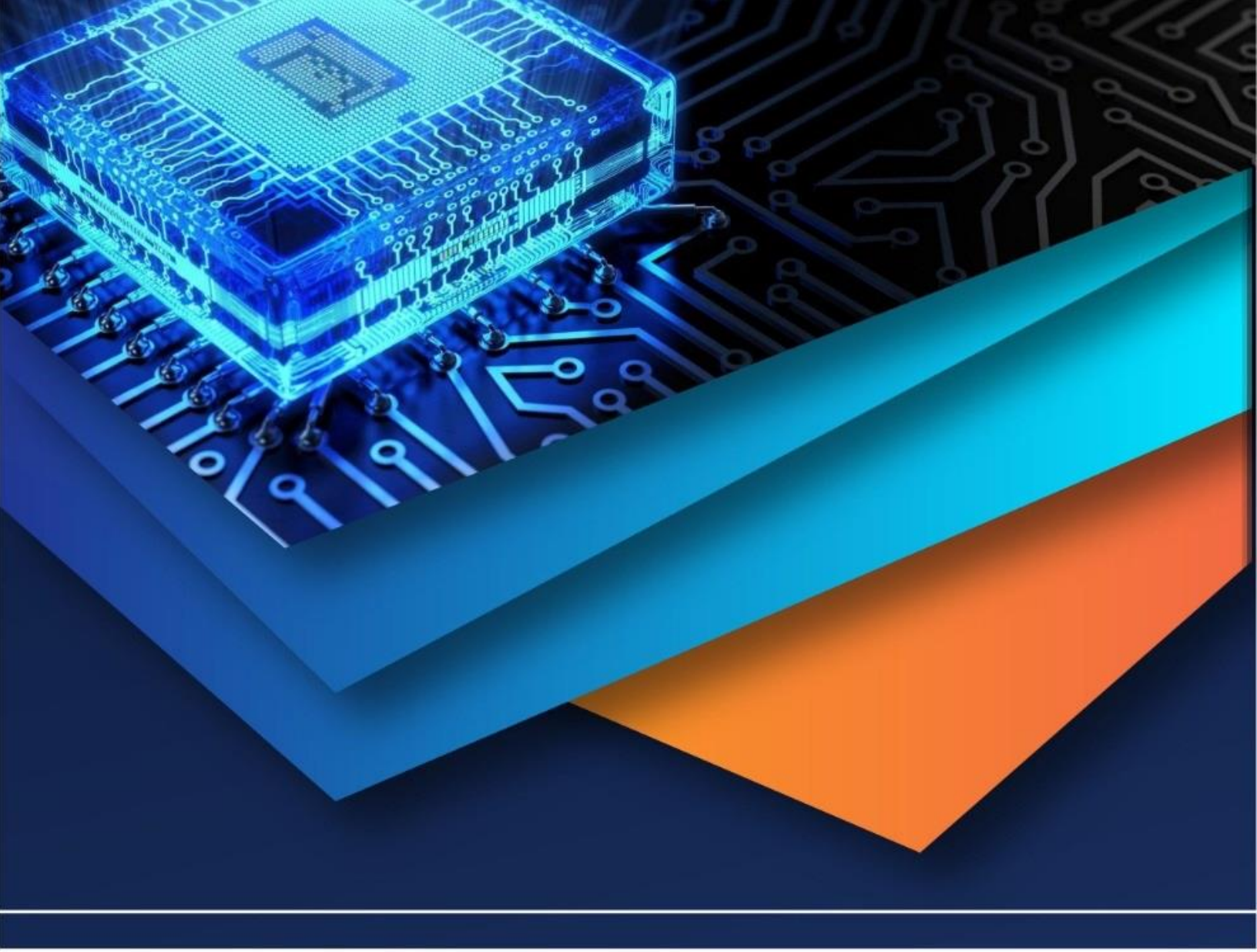
The rapid growth of technology has led to an increase in apps on Google app stores. However, some of these apps are fraudulent and pose a threat to user privacy. To tackle this, a model was developed to detect fraudulent software using scales, review scores, in-app purchases, and content additions as parameters. The resolution tree algorithm proved to be 85% more accurate in detecting fraudulent apps. The framework is measurable and can be expanded to include more evidence of domain-based fraud. It effectively demonstrates the system's effectiveness and the standardization of fraud detection. One advantage is that it can rate fraudulent apps in the Play Store, helping users make informed decisions and holding developers accountable. Overall, the rise of fraudulent apps is a major security concern, but this framework offers a promising solution by accurately identifying and rating fraudulent apps for user security. Further improvements can make it an essential tool for ensuring user privacy and security in the app market.

VII. FUTURE SCOPE

In the future, the machine learning model for detecting fraudulent apps can be improved in various ways. Updating the model with fresh data regularly can enhance its performance over time. Using ensemble methods, which combine multiple models or algorithms, can also boost overall effectiveness. Explainable AI techniques can help understand how the model makes predictions and identify any biases or areas for improvement. Additionally, incorporating data from diverse sources like social media and app reviews can uncover patterns related to fraudulent apps. These advancements aim to make the model more accurate and efficient in identifying fraud, even as new types of fraud emerge

REFERENCES

[1] Esther Nowroji, Vanitha, "Detection Of Fraud Ranking For Mobile App Using IP Address Recognition Technique", vol. 4.
[2] Javvaji Venkataramaiah, BommavarapuSushen, Mano. R, Dr.GladispushpaRathi, "An enhanced mining leading session algorithm for fraud app detection in mobile applications"
[3] S.R.Srividhya, S.Sangeetha – "A Methodology to Detect Fraud Apps Using Sentiment Analysis"
[4] Keerthana. B, Sivashankari.K and ShaisthaTabasum.S , "Detecting Malwares and Search
[5] Rank Fraud in Google Search Using Rabin Karp Algorithm", IJARSE, 7(02), 2018, pp.504527
[6] Shashank Bajaj, Nikhil Nigam, Priya Vandana, Srishti Singh, "Detection of fraud apps using sentiment analysis", International Journal of Innovative Science and Research Technology.
[7] Harpreet Kaur, Veenu Mangat and Nidhi, — "A Survey of Sentiment Analysis techniques"
[8] International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (ISMAC), 2017, pp. 921
[9] Jing Wan, Mufan Liu, Junkai Yi and Xuechao Zhang, "Detecting Spam Webpages through Topic andSemantics Analysis", IEEE Global Summit onComputer and Information Technology (GSCIT), 2015, pp. 83-92.
[10] Navdeep Singh, Prashant Kr. Pandey and Mr.Srinivasan, — "Improved Discovery of
[11] Rating Fake for Cellular Apps", IEEE International Conference on Science Technology Engineering and Management (ICONSTEM), 2016, pp. 135-140.
[12] Weiman Wang, Restricted Boltzmann Machine. GitHub. Aug 2017. [Online] Available:https://github.com/aaxwaz/Fraud-detection-usingdeep-learning/blob/master/rbm/rbm.py.
[13] Dubey Veena, G. D. (2016). Sentiment Analysis Based on Opinion Classification Techniques: A Survey . International Journal of Advanced Research in Computer Science



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY