JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

An Efficient Machine Learning Framework For Performance Improvement of Hijacked Social Media Accounts

[1]Mohit Kumar, [2]Prof. Rupali Chaure, [3]Dr. Ritu Shrivastava

[1]Research Scholar, [2]Assistant Professor, [3]Head and Professor
Department of Computer Science and Engineering
Sagar Institute of Research and Technology, Bhopal, India

Abstract : The twitter is very commonly online social media platform. It is widely used and acceptable by the government also to communicate the information. This online platform is less secure in terms of the user account. Malware can be disorienting, getting in the way of how you normally use your account and making you feel uneasy even if you aren't sure what's causing the problem. This paper presents an efficient machine learning framework for performance improvement of hijacked social media accounts. In this study, the machine learning classifiers are predicting the attack in twitter user accounts. The twitter dataset is taken as input data and applied into pre-processing method.

IndexTerms – Twitter, Attack, Online Social, Artificial intelligence, NLP, Machine Learning.

I. INTRODUCTION

The Social media are interactive technologies that facilitate the creation and sharing of information, ideas, interests, and other forms of expression through virtual communities and networks. While challenges to the definition of social media arise due to the variety of stand-alone and built-in social media services currently available, there are some common features:

- Social media are interactive Web 2.0 Internet-based applications.
- User-generated content—such as text posts or comments, digital photos or videos, and data generated through all online interactions is the lifeblood of social media.
- Users create service-specific profiles for the website or app that are designed and maintained by the social media organization.
- Social media helps the development of online social networks by connecting a user's profile with those of other individuals or groups.

The term social in regard to media suggests that platforms are user-centric and enable communal activity. As such, social media can be viewed as online facilitators or enhancers of human networks webs of individuals who enhance social connectivity

Users usually access social media services through web-based apps on desktops or download services that offer social media functionality to their mobile devices (e.g., smart phones and tablets). As users engage with these electronic services, they create highly interactive platforms which individuals, communities, and organizations can share, co-create, discuss, participate, and modify user-generated or self-curated content posted online. Additionally, social media are used to document memories, learn about and explore things, advertise one self, and form friendships along with the growth of ideas from the creation of blogs, podcasts, videos, and gaming sites. This changing relationship between humans and technology is the focus of the emerging field of technological self-studies. Some of the most popular social media websites, with more than 100 million registered users, include Facebook (and its associated Facebook Messenger), TikTok, WeChat, ShareChat, Instagram, QZone, Weibo, Twitter, Tumblr, Baidu Tieba, and LinkedIn. Depending on interpretation, other popular platforms that are sometimes referred to as social media services include YouTube, QQ, Quora, Telegram, WhatsApp, Signal, LINE, Snapchat, Pinterest, Viber, Reddit, Discord, VK, Microsoft Teams, and more. Wikis are examples of collaborative content creation.

Social media outlets differ from traditional media (e.g., print magazines and newspapers, TV, and radio broadcasting) in many ways, including quality,[11] reach, frequency, usability, relevancy, and permanence. Additionally, social media outlets operate in a dialogic

transmission system (i.e., many sources to many receivers) while traditional media outlets operate under a monologic transmission model (i.e., one source to many receivers). For instance, a newspaper is delivered to many subscribers, and a radio station broadcasts the same programs to an entire city.

This paper is organised into the V section. I section provides the overview & introduction of the twitter highjack prediction. The II section provides background. The III section provides the proposed methodology, IV section provides the simulation and results, and V section provides the conclusion of this paper.

II. LITERATURE SURVEY

S. Alterkav et al.,[1] presents a novel authorship verification framework for hijacked social media accounts, compromised by a human, is proposed. Significant textual features are derived from a Twitter-based dataset. They are composed of 16124 tweets with 280 characters crawled and manually annotated with the authorship information. XGBoost algorithm is then used to highlight the significance of each textual feature in the dataset. The reduced dataset is evaluated with many classifiers, and the achieved result of the F-score is 94.4%.

A. Alzubaidi et al.,[2] work centers around the ideas and dangers related with malware, and audits current methodologies and systems used to identify malware as for their technique, related datasets, and assessment measurements.

H. Kato et al.,[3] this, steady exhibition can be accomplished. Since our highlights are only eight-layered, the present plot takes less preparation time and is viable with other ML based plans. Moreover, our highlights can quantitatively offer clear data that assists human with understanding discovery results. Our plan is reasonable for useful use since every one of the prerequisites can be met.

C. Li et al et al.,[4] To stay aware of malware's advancement, the recognition models for the most part should be retrained intermittently (e.g., consistently) in light of the information gathered in nature. Be that as it may, this prompts harming assaults, explicitly indirect access assaults, which undermine the learning system and make avoidance burrows for controlled malware tests.

L. Gong et al.,[5] it is have put forth deliberate attempts in include designing, UI investigation, copying engineering, and run-time climate, hence keeping up with high identification exactness (97% accuracy and 97% review) and short per-application filter time (1.7 minutes) with just two ware servers, under a serious responsibility of 10K recently submitted applications each day.

I. Almomani et al.et al.,[6] Around the world, Ransomware assaults have prompted genuine misfortunes for people and partners. Be that as it may, the sensational increment of Ransomware families makes to the method involved with recognizing them more testing because of their consistently developed attributes.

F. Mercaldo et al.,[7] In this work, it is propose a strategy to derive versatile application perniciousness by distinguishing the having a place family, taking advantage of formal proportionality checking. it is acquaint a bunch of heuristics with lessen the quantity of portable application examinations and it is characterize a measurement mirroring the application vindictiveness. True tests on 35 Android malware families (going from 2010 to 2018) affirm the viability of the present technique in portable malware recognition and family distinguishing proof.

L. N. Vu et al.,[8] The curiosity of our review lies in the development of a nearness grid for every application. These lattices go about as "input pictures" to the Convolutional Brain Organization model, permitting it to figure out how to separate harmless and noxious applications, as well as malware families.

L. Gong et al et al.,[9] Regardless of being essential to the present portable biological system, application markets have in the interim turned into a characteristic, helpful malware conveyance channel as they in fact "loan believability" to noxious applications. In the beyond couple of years, AI (ML) strategies has been broadly investigated for computerized, hearty malware location, however till now it is have not seen a ML-based malware discovery arrangement applied at market scales.

W. Yuan et al.,[10] present locator principally involves a single shot calculation for model preparation. Consequently it very well may be completely or steadily prepared straightforwardly on cell phones. All things considered, our finder beats the shallow learning-based models, including support vector machine (SVM) and AdaBoost, and approaches the profound learning-based models multi-facet perceptron (MLP) and convolutional brain organization (CNN).

K. Liu et al.,[11] Android applications are growing quickly across the versatile biological system, yet Android malware is likewise arising in an interminable stream. Numerous scientists have concentrated on the issue of Android malware identification and have advanced speculations and techniques according to alternate points of view.

D. Li et al.,[12] This normally prompts another launch of antagonistic preparation, which is additionally outfitted to upgrading the troupe of profound brain organizations. it is assess protections utilizing Android malware indicators against 26 unique assaults upon two pragmatic datasets. Trial results show that the new ill-disposed preparing fundamentally upgrades the strength of profound brain networks against a wide scope of assaults, gathering techniques advance the heartiness when base classifiers are sufficiently vigorous, but troupe assaults can sidestep the improved malware locators successfully, even quite downsizing the VirusTotal administration.

Q. Han et al.,[13] As the most generally utilized operating system on the planet, Android is intensely focused on by vindictive programmers. However much work has been done on recognizing Android malware, programmers are turning out to be progressively proficient at avoiding ML classifiers.

J. Ribeiro et al.,[14] Past examination endeavors on fostering an Interruption Recognition and Avoidance Frameworks (IDPS) for Android cell phones depend for the most part on unified information assortment and handling on a cloud server. Be that as it may, this pattern is described by two significant impediments. In the first place, it requires a ceaseless association between observed gadgets and the server, which may be infeasible, because of versatile organization's blackout or halfway inclusion. Second, it expands the gamble of delicate data spillage and the infringement of client's protection.

X. Wang et al.,[15] In this work, to completely explore Android piece practices, it is first present a part include based system, CrowdNet, for distributed computing stages. CrowdNet incorporates a programmed information supplier that gathers impressions of piece highlights and an equal malware indicator that approves Android noxious practices. Then, at that point, it is compute and choose stowed away focuses by a heuristic methodology for 12,750 Android applications to decrease the quantity of emphasess and time intricacy.

III. PROPOSED METHODOLOGY

The proposed methodology is explained using following sub modules-

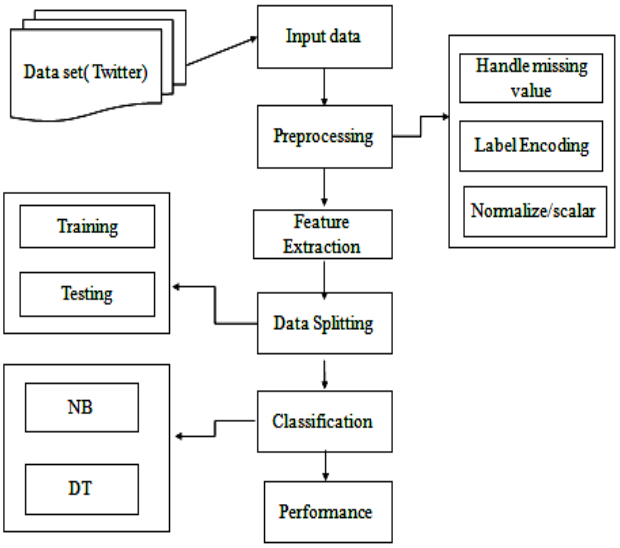

Figure 1: Flow Chart

1. Firstly, download the twitter dataset from kaggle website, which is a large dataset provider and machine learning repository Provider Company for research.
2. Now apply the preprocessing of the data, here handing the missing data, label encoding, and drop unwanted column.
3. Apply the model selection and feature extraction of the dataset
4. Now dataset is splitted into the training and the testing data. The 70-80% data is spllited into training and 20-30% data is splitted into the tested phase.
5. Apply the Naïve Bayes, and decision tree classification algorithm to predict the Hijacked twitter Social Media Accounts.
6. Now check and calculate the performance parameters.

The methodology step is discussed based on the following steps-

- Data Selection and Loading
- Data Preprocessing
- Feature Selection
- Feature Reduction
- Classification
- Result Generation

Data Selection and Loading

- The data selection is the process of selecting the data predicting twitter attack.
- The dataset which contains the information about android malware apps and permissions.

Data Preprocessing

- Data pre-processing is the process of removing the unwanted data from the dataset.
- It handles missing data removal and encoding categorical data.
- Encoding Categorical data: That categorical data is defined as variables with a finite set of label values.

Splitting Dataset

- Data splitting is the act of partitioning available data into. Two portions, usually for cross- valedictory purposes.
- One portion of the data is used to develop a predictive model. And the other to evaluate the model's performance.

Classification

Naïve Bayes Classifier Algorithm-

- Naïve Bayes algorithm is a supervised learning algorithm, which is based on Bayes theorem and used for solving classification problems.
- It is mainly used in text classification that includes a high-dimensional training dataset.
- Naïve Bayes Classifier is one of the simple and most effective Classification algorithms which help in building the fast machine learning models that can make quick predictions.

Decision Tree Classifier Algorithm

- Decision Tree is a supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome.
- Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches.
- The decisions or the test are performed on the basis of features of the given dataset.
- It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions.

Algorithm

Input: Twitter Dataset.

Take the initial data features like Id, Tweet, following, followers, actions, re-tweet location, type etc

Filtering the null value

Classify the dataset based on the selected features

Output: Optimal Precision, Recall, F-Measure, Accuracy and Error rate

Step: 1. Split train and test dataset Y_train, Y_test, X_train and X_test

2. Feature extractions, features = {} for attack count: features [attack count] = True

3. Model selection and split

Y train counts

Y test count

4. Apply the naïve bayes, and decision tree machine learning classifier.

5. Generate confusion matrix and show value of TP, FP, TN and FN

6. Calculate Accuracy, error rate, precision, recall and f-measure

7.Plot the ROC Curve

Evaluation

The confusion metrics used to evaluate a classification model are accuracy, precision, and recall.

- Precision = True Positive/(True Positive + False Positive)
- Recall = True Positive/(True Positive + False Negative)
- F1-Score = 2x (Precision x Recall)/(Precision + Recall)
- Accuracy = [TP +TN] / [TP+TN+FP+FN]
- Classification Error = 100- Accuracy

Result Generation

The final result is generated based on the overall classification and prediction. The performance of this proposed approach is evaluated using some measures like.

IV. SIMULATION AND RESULTS

The execution of the proposed calculation is done over python spyder 3.7. The sklearn, numpy, pandas, matplotlib, pyplot, seaborn, os library assists us with utilizing the capacities accessible in spyder climate for different strategies-

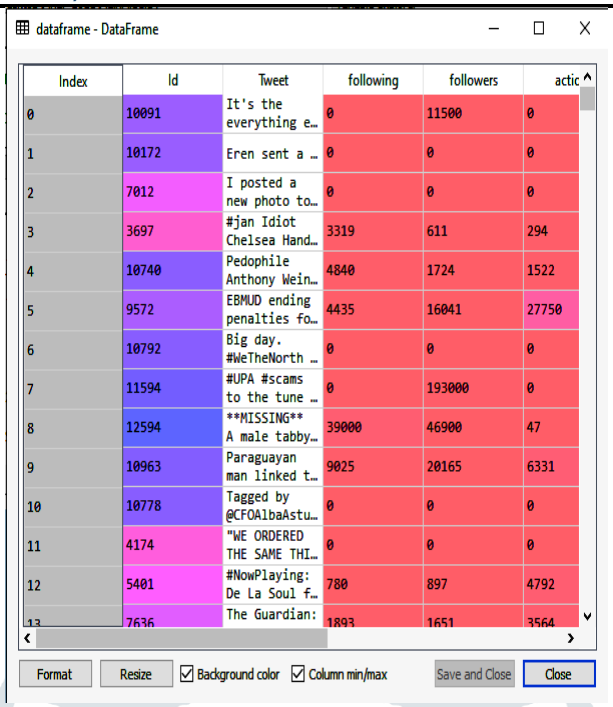

Figure 2: Dataset

Figure 2 is showing the dataset in the python environment. The dataset have various numbers of rows and column. The features name is mention in each column.

Figure 3: Y test

Figure 3 is showing the y test of the given dataset. The given dataset is divided into the 20-30% part into the train dataset. Total 2394 data is used for testing.

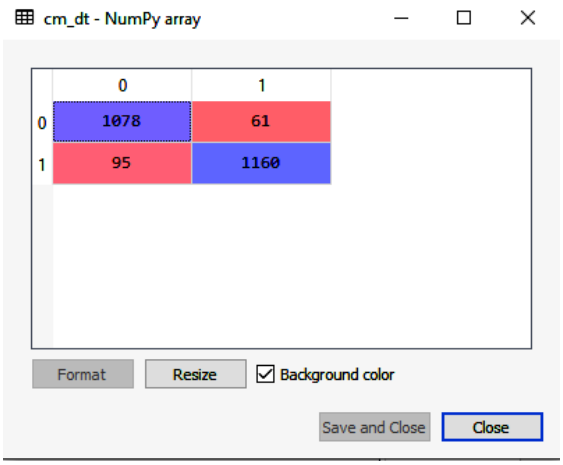

Figure 4: Confusion matrix of DT

Figure 4 is presenting confusion matrix, where true and false positive and negative values are shown.

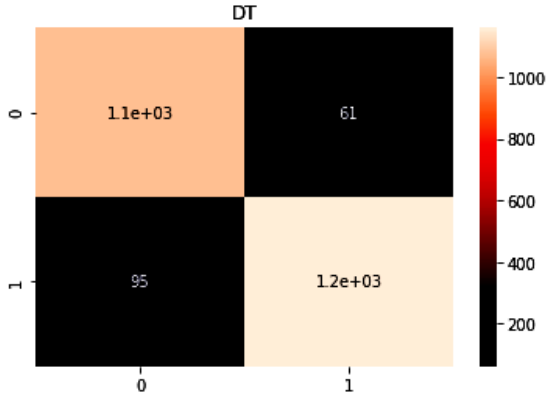

Figure 5: Confusion matrix heat map

Figure 5 is presenting confusion matrix heat map, Confusion matrix between true values and prediction values is formed by counting the number of matches within each interval.

Table 1: Result Comparison

Sr. No.	Parameters	Previous Work [1]	Proposed Work
1	Method	Logistic Regression	Decision Tree
2	Accuracy (%)	91.11	93.44
3	Classification Error (%)	8.89	6.56

V. CONCLUSION

This paper presents an efficient machine learning framework for performance improvement of hijacked social media accounts. The simulation is performed using python sypder 3.7 software. The simulation results show the improvement in the performance parameters. The overall accuracy achieved by the proposed work is 93.44 % while previous it is achieved 91.11 %. The classification error of proposed technique is 6.56% while 8.89% in existing work. Therefore it is clear from the simulation results; the proposed work is achieved significant better results than existing work.

REFERENCES

1. S. Alterkav, H. Erbay “Design and Analysis of a Novel Authorship Verification Framework for Hijacked Social Media Accounts Compromised by a Human” Security and Communication Networks, Volume 2021, Article 8869681.
2. A. Alzubaidi, "Recent Advances in Android Mobile Malware Detection: A Systematic Literature Review," in IEEE Access, vol. 9, pp. 146318-146349, 2021, doi: 10.1109/ACCESS.2021.3123187.
3. H. Kato, T. Sasaki and I. Sasase, "Android Malware Detection Based on Composition Ratio of Permission Pairs," in IEEE Access, vol. 9, pp. 130006-130019, 2021, doi: 10.1109/ACCESS.2021.3113711.
4. C. Li et al., "Backdoor Attack on Machine Learning Based Android Malware Detectors," in IEEE Transactions on Dependable and Secure Computing, doi: 10.1109/TDSC.2021.3094824.
5. L. Gong, Z. Li, H. Wang, H. Lin, X. Ma and Y. Liu, "Overlay-based Android Malware Detection at Market Scales: Systematically Adapting to the New Technological Landscape," in IEEE Transactions on Mobile Computing, doi: 10.1109/TMC.2021.3079433.
6. I. Almomani et al., "Android Ransomware Detection Based on a Hybrid Evolutionary Approach in the Context of Highly Imbalanced Data," in IEEE Access, vol. 9, pp. 57674-57691, 2021, doi: 10.1109/ACCESS.2021.3071450.
7. F. Mercaldo and A. Santone, "Formal Equivalence Checking for Mobile Malware Detection and Family Classification," in IEEE Transactions on Software Engineering, doi: 10.1109/TSE.2021.3067061.
8. L. N. Vu and S. Jung, "AdMat: A CNN-on-Matrix Approach to Android Malware Detection and Classification," in IEEE Access, vol. 9, pp. 39680-39694, 2021, doi: 10.1109/ACCESS.2021.3063748.
9. L. Gong et al., "Systematically Landing Machine Learning onto Market-Scale Mobile Malware Detection," in IEEE Transactions on Parallel and Distributed Systems, vol. 32, no. 7, pp. 1615-1628, 1 July 2021, doi: 10.1109/TPDS.2020.3046092.
10. W. Yuan, Y. Jiang, H. Li and M. Cai, "A Lightweight On-Device Detection Method for Android Malware," in IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 51, no. 9, pp. 5600-5611, Sept. 2021, doi: 10.1109/TSMC.2019.2958382.
11. K. Liu, S. Xu, G. Xu, M. Zhang, D. Sun and H. Liu, "A Review of Android Malware Detection Approaches Based on Machine Learning," in IEEE Access, vol. 8, pp. 124579-124607, 2020, doi: 10.1109/ACCESS.2020.3006143.
12. D. Li and Q. Li, "Adversarial Deep Ensemble: Evasion Attacks and Defenses for Malware Detection," in IEEE Transactions on Information Forensics and Security, vol. 15, pp. 3886-3900, 2020, doi: 10.1109/TIFS.2020.3003571.
13. Q. Han, V. S. Subrahmanian and Y. Xiong, "Android Malware Detection via (Somewhat) Robust Irreversible Feature Transformations," in IEEE Transactions on Information Forensics and Security, vol. 15, pp. 3511-3525, 2020, doi: 10.1109/TIFS.2020.2975932.
14. J. Ribeiro, F. B. Saghezchi, G. Mantas, J. Rodriguez and R. A. Abd-Alhameed, "HIDROID: Prototyping a Behavioral Host-Based Intrusion Detection and Prevention System for Android," in IEEE Access, vol. 8, pp. 23154-23168, 2020, doi: 10.1109/ACCESS.2020.2969626.
15. X. Wang, C. Li and D. Song, "CrowdNet: Identifying Large-Scale Malicious Attacks Over Android Kernel Structures," in IEEE Access, vol. 8, pp. 15823-15837, 2020, doi: 10.1109/ACCESS.2020.2965954.