Clustering-based Outlier Detection in Social Media Activity for Identifying Fake Accounts

Muhammad Khubaib Shakeel (24L-8019)[1] and Asif Ali (24L-8024)[2]

I. Description of the Problem

As there is an extensive increase in the social media platforms and new users has been added almost everyday, fake and inauthentic accounts on these social media platforms has been increasing day by day. It poses significant threats to security of users and is also a threat for society. This threat also included the spreading of the misinformation and spam that causes serious problems to the Governments, Banks, Hospitals and many other parts of the society.

The main objective of this project is to create a **unsupervised learning framework** that would be scaleable and can easily detect the fake accounts in the vast sea of the social media platforms and social media users. It can be done by analyzing the behavioral outliers. Our goal is to analyze the fake-accounts using the multi-dimensional features or attributed like network level interactions, temporal patterens of the postings and user activity.

II. Related Work

In this research, our main goal is to find any anomaly on social media. For this purpose, we find two previous works that will help us to do out tasks.

- **Cao Xiao et al., "Detecting Clusters of Fake Accounts in Online Social Networks," NDSS 2015.** The authors proposed cluster-level classification to detect fake LinkedIn accounts using registration-based features, achieving an AUC of 0.98.[**?**]
- **Y. Li et al., "In a World That Counts: Clustering and Detecting Fake Social Engagement at Scale," WWW 2016.** This work used local spectral clustering on temporal engagement graphs to detect coordinated YouTube fake engagement. [**?**]

III. Project Contributions

A gap has been identified in these previous studies. The previous studies have been relying on supervised learning after clustering or they depends upon the features or attributes that has been derived from the single data view. Our project tackles these problems, by introducing the following:

- **Multi-view clustering Frameworks** In this step, features of behavior included activity and text, persistence features and network features has been joined or fuse with each other in order to get the Multi-view Clustering Frameworks.
- **Time-Aware persistence scoring** where features or attributes has been grouped together on the basis of age-normalized activity of accounts, like **activity_persistence_score**.
- **Scalable and Purely Unsupervised Working Environment** where a complete pipeline has been implemented with the help of the PySpark. In this we developed a pipeline that has used combined multiple unsupervised techniques (including the K-Means, GMM, DBSCAN,LOF) for final outlier scoring.

IV. Algorithms & Techniques

This project uses PySpark to create the pipelines for the processing of Big Data as e use in this project. PySpark provides us many Machine Learning and Data Mining Techniques so that we can handle big data processing. Following are the techniques and Algorithms that have been used in this project:

- **Big Data Processing** For the processing of the Big Data, its features preprocessing and aggregation, we use PySpark as it is helpful for scalable computation.
- **Clustering** Three clustering methods has been used in order to group the behaviors and helpful in order to generate the clusters-based features
  - **K-Means** use for partioning the users into 5 different clusters using the value of K=5. These are used in order to find the new features for example kmeans_distance means distance to the assigned centroid.
  - **Gaussian Mixture Models(GMM)** is used for data distribution for the model. it is done probabilistically in order to perform the soft-clustering. it K values is also set to 5.
  - **DBSCAN** It is used for the grouping on the basis of the density. It is also useful for identifying the sparse region which results in the approximation of the structural outliers.
- **Feature Engineering** in which features or attributes has been assembled together in a single vector and then the vector scaled with the help of the **MinMaxScaler**. Following are the features that has been used
  - **Behavioral/Text** It included the *tweet_count, vocab_richness, sentiment_polarity.*
  - **Temporal/Persistence** It included the *account_age_days, activity_persistence_score*
  - **Network(With the Help of Graph using *GraphX*)** For this purpose *degree_centrality, clustering_coeff, and pagerank_score* has been used. It has all been done with the help of the ***NetworkX***

library on the graph that gives the links of the followers/following.
  - **Cluster-Based** In this we calculate the kmeans_distance
- **Outlier Detection: Local Outlier Factor (LOF)** It is used for scoring the results of the clustering, or in short to find the outlier that are present in the clustering. It gives the score of the outlier on the basis of the local data density. If the score of LOF is lower, it means that there are the higher chances that anomaly are present. The final output included the LOF Score (*final_outlier_score*) and a binary prediction (*is_outlier*).

V. Dataset Details

- **Dataset Used**: For this purpose, ***TwiBot-20*** dataset has been used. it is an open-source dataset. the dataset has been taken from the kaggle. this dataset contains the Twitter datasets that contains the user profile, tweets and ground-truth labels.
- **Size**: Total of 11,826 user has been record in the evalution set.
- **Distribution:** The data shows the distribution in the following form
  - True Anomalies that consider as bots are total of 6589 bots
  - Inliners also considered as real or human accounts have 5237 users

VI. Results

The results have been shown in the form of the following Table 01. The results obtained after setting the factor ***contamination*** parameter to 0.5. The DBScan Cluster and Outlier score detection has been shown in figure 1 and figure 2 respectively.

VII. Problems face in Project

The results shows the problem significantly in the ROC-AUC Curve which is of 0.4459

- **Mismatch in Results**:Unsupervised outlier detection doesn't work properly as the bots make up the almost 56% of dataset TwiBot-20. Therefore the LOF-Algorithm doesn't work properly and incorrectly finds the sparse region instead of finding the dense region.
- **Separability Problem:** The AUC Score is low. It suggested that the features have not been able to create clusters although it uses multi-view structure. The anomaly and normal groups structually mixed with each other in feature space.
- **F1-Score didn't give accurate results:** F1-Score shows the reults of the 0.4910 is not correct as the contamination parameter set to 0.5 manually for LOF calculation. Therefore causes the model to set the 50% of users as robots. This causes high recall rates and disturbs the results of F1-Score.

VIII. Weakness and Limitations of the Work

- Local Outlier Factor (LOF) has been used on the dataset where a positive class is the present in abundance and hence it held as the major limitation in this case. Therefore it will become incabale of the solving the problem
- The features present in the network is dependent on the coverting the Spark Dataframe to PandasDataframe in order to use NetworkX to create the Network. Hence it limiting the scalibility of graph components. Therefore it causes the errors.

IX. Improvements

- More robust features should be incorporated and more complex temporal features should be used in order to detect the synchronization between accounts.
- Network features should be adjusted in such a way that the entire calculations would be perform in the spark instead of converting to Pandas DataFrame.

X. Computations

The computational experiment has been done in order to excute the multi-stage PySpark Data Pipeline effectively.

- TwiBot-20 has been used as the main dataset that contain the total records of the 11,826 users. The work done in Google Colab.
- Experiment has been performed in different stages:
  - **Data Preprocessing**: RAW .json files has been used in this experiment. these files first loaded, then merged and then shuffled. After that it was partioned in 10 different partition to perform the tasks. Tweets has been combined, then cleaned, then tokenized and then stop words removed from the dataset.
  - **Extracting Feature** All features were extracted and calculated. Behavioral and Temporal Features calculated the Tweet Counts, account age and the activity_persistence_score. The follower/following graph network has been constructed and featured has been calculated using the networkX
  - **Custering**: The 8-dimensional feature vectors has been assembled and scaled setting the K=5 in k-means. kmeans_distance has been calculated as an ensembled features. New 9-dimensions that has been develop during the k-means clustering has reassembled and and rescaled. With GMM(K=5) and DBSCAN (With epsilon=0.5, and Minimum Points set to 10 has been run on the feature vector that has been augmented.
  - **Outlier Detection and its evaluation** In this step, Local Outlier Factor has been used and run on the feature vectors that has been finalized. The contamination threshold of the LOF has been set to 0.5 in order to align it with outlier count and true bot ration. Hance the final score has been evaluted against the ground truth "label" using the ROC-AUC, F1-Score and Precision.

TABLE I: Evaluation Metric Results for the Clustering

Metric	Results	Interpretation
ROC-AUC	0.4450%	The model did not perform well compared to a random guess, which is 0.5. This shows that the detection has poor discriminatory ability.
F1-Score	0.4910%	It shows that precision and recall measure are balanced. .
Precision	0.5462%	It shows the true anomalies but it also contains many false positives
Predicted Anomalies	5913	Showes that almost 50% of abnormalities are the outliers.
True Positive	3069	It shows the number of results that are correctly identified.

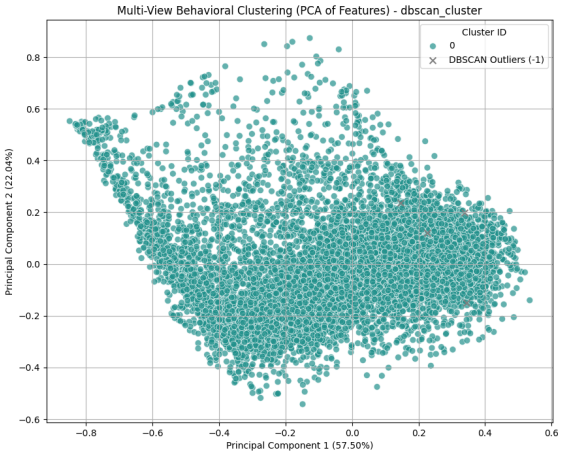

Fig. 1: A Multi-view behavioral clustering of our data.

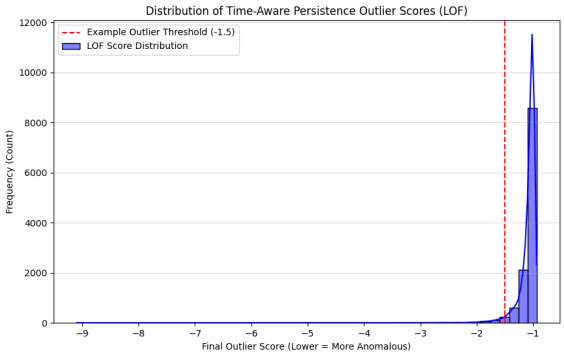

Fig. 2: Outlier score Distribution of Time Aware Persistence (LOF Score).

REFERENCES

[1] C. Xiao, D. M. Freeman, and T. Hwa, "Detecting Clusters of Fake Accounts in Online Social Networks," in *Proceedings of the 8th ACM Workshop on Artificial Intelligence and Security (AISec '15)*, Denver, Colorado, USA, 2015, pp. 91–101, doi: 10.1145/2808769.2808779.

[2] Y. Li, O. Martinez, X. Chen, Y. Li, and J. E. Hopcroft, "In a World That Counts: Clustering and Detecting Fake Social Engagement at Scale," in *Proceedings of the 25th International Conference on World Wide Web (WWW '16)*, Montréal, Québec, Canada, 2016, pp. 111–120, doi: 10.1145/2872427.2882972.