DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
PUDUCHERRY TECHNOLOGICAL UNIVERSITY
PUDUCHERRY – 605 014.

BONAFIDE CERTIFICATE

This is to certify that the Mini Project Work titled **"User Activity Analysis to Detect Borderline Bots in Twitter"** is a bonafide work done by **Praveen M (20CS338)** and **Hari Prasath R (20CS330)** in partial fulfillment for the award of the degree of **Master of Technology** in **Computer Science and Engineering (Information Security)** of the **Pondicherry University** and that this work has not been submitted for the award of any other degree of this/any other institution.

Akila.V

Digitally signed by Akila.V
DN: cn=Akila.V, o=Pondicherry
Enginerring College, ou=CSE
department,
email=akila@pec.edu, c=IN
Date: 2021.10.21 09:44:24
+05'30'

Project Guide	**Head of the Department**
(Dr.V.Akila)	**(Dr. R. MANOHARAN)**

Submitted for the University Examination held on ___________.

Internal Examiner	**External Examiner**

ACKNOWLEDGEMENT

I am deeply indebted to **Dr. V. Akila**, **Assistant Professor** Department of Computer Science and Engineering, Pondicherry Engineering College, Pondicherry, for her valuable guidance throughout the two phases of this project.

I also express my heart-felt gratitude to **Dr. R. Manoharan**, Professor & Head, Department of Computer Science and Engineering, for giving constant motivation in succeeding my goal.

With profoundness I would like to express my sincere thanks to **Dr. K. Vivekanandan**, the Principal, Pondicherry Engineering College, for his kindness in extending the infrastructural facilities to carry out my project work successfully.

I would be failing in my duty if I do not acknowledge the efforts of the Project Review Panel members consisting of **Dr.R.Manoharan, Dr.P.Salini, Dr.Lakshmana Pandian, Dr.M.Sugumaran, Dr.K.Sathiyamurthy** for shaping our ideas and constructive criticisms during project review.

I also express my thanks to all the **Faculty** and **Technical Staff** members of the CSE department for their timely help and the **Central Library** for facilitating useful reference materials.

I would be failing in my duty if I don’t acknowledge the immense help extended by my friends, who have always been with me in all my trails and tribulations and encouraging me to complete.

Hari Prasath R
Praveen M

ABSTRACT

An Online Social Networks (OSN) like Twitter became a predominant platform for social expression and public relations. Twitter had 330 million monthly active users by the year 2019. With the gain in popularity, the ratio of virulent and automated accounts has also increased. It is estimated that 48 million of its functioning accounts are bots. Precisely, Twitter bots or Sybil accounts are kinds of automated web robot software that regulate activities like the tweet, retweet, like or follow via Twitter API. These bots misguide and delude genuine users by spreading spurious content. Hence, uncovering malicious bots from authentic users is obligatory to ensure a safe environment on Twitter. Then, we perform a set of simulations to benchmark different behavioral models and to identify the models that better resemble human behaviors in Twitter. In this project, We propose an effective approach to model the features of behavior analysis during bot detection. The behavior of a feature is modeled as a DNA sequence. This sequence is split to smaller fragments to represent a feature. The set of all fragments of all users is the data set. Whereas an efficient Bot detection model to detect borderline bots and to reduce False Positives based on Information Gain.The Borderline bots which follows unique patterns using Information gain approaches will yield better performance.

LIST OF FIGURES

TABLE OF CONTENT

CHAPTER I

INTRODUCTION

The Bot is a software application that runs automated tasks over the Internet. Typically, bots perform tasks that are simple and repetitive much faster than a person could. Bots are created for many different reasons such as conducting a distributed denial of service(DDOS), spreading spam, conducting click-fraud scams, stealing personal user information(e.g. credit card numbers, social security numbers), or taking advantage of the powerful computational resources offered by the bots to carry some distributed computing tasks. Bots interact over legitimate communication channels. Twitter bots are automated Twitter accounts controlled by bot software. While their purpose is to tweet, retweet, liking tweets and following other users for a specific goals on a large scale.

1.1 OVERVIEW

We propose a simple and effective approach to model the features of behavior analysis during bot detection. The behavior of a feature is modeled as a DNA sequence. This sequence is split to smaller fragments to represent a feature. The set of all fragments of all users is the data set. An Information Gain based method is used to identify the patterns that model the featured bots to contribute the false positives.

1.2 OBJECTIVE OF THE STUDY

- To detect the Borderline bots in Twitter through user Behavior Analysis.
- To identify what features contribute in detection of borderline bots.
- To design an efficient Bot detection model to detect borderline bots and to reduce False Positives based on Information Gain (Entropy).

1.3 NEED FOR THE STUDY

Bots evolve over a period of time. Existing Techniques detects Bots of same pattern/ similar kinds of bots, Thus Borderline bots are not detected. Most of the Existing Techniques use user, tweet or graphical features whereas, the use of user behaviours are still open field to explore. Uncovering borderline bots that follow unique patterns using Information gain (Entropy) approaches will yield better performance.

CHAPTER II

LITERATURE REVIEW

[1] Emergent properties, models, and laws of behavioral similarities within groups of twitter users

DNA-inspired online behavioral modeling techniques have been proposed and successfully applied to a broad range of tasks. Their investigation of fundamental laws that drives the occurrence of behavioral similarities among Twitter users, employing a DNA-inspired technique. Their findings are multifold. First, they demonstrate that, despite apparently featuring little to no similarities, the online behaviors of Twitter users are far from being uniformly random. Secondly, they benchmark different behavioral models through a number of simulations. They characterize the main properties of such models and they identify those models that better resemble human behaviors in Twitter. Then, they demonstrate that the number and the extent of behavioral similarities within a group of Twitter users obey a log-normal law, and they leverage this characterization to propose a novel bot detection system. In a nutshell, the results shed light on the fundamental properties that drive the online behaviors of groups of Twitter users, through the lenses of DNA-inspired behavioral modeling techniques. The study is based on a wealth of data gathered over several months that, for the sake of reproducibility, are publicly available for research purposes.

[2]Bot prediction on social networks of Twitter in altmetrics using deep graph convolutional networks

In the context of smart cities, it is crucial to filter out falsified information spread on social media channels through paid campaigns or bot-user accounts that significantly influence communication networks across the social communities and may affect smart decision-making by the citizens. They were focusing on two major aspects of the Twitter social network associated with altmetrics: (a) to analyze the properties of bots on Twitter networks and (b) to distinguish between bots and human accounts. Firstly, they were employed state-of-the-art social network analysis techniques that exploit Twitter's social network properties in novel altmetrics data. They found that 87% of tweets are affected by bots that are involved in the network's dominant communities. We also found that, to some extent, community size and the degree of distribution in Twitter's altmetrics network follow a power-law distribution. Applied a deep learning model, graph convolutional networks, to distinguish between organic (human) and bot Twitter accounts. The deployed model achieved the promising results, providing up to 71% classification accuracy over 200 epochs. Overall, the study concludes that bot presence in altmetrics-associated social media platforms can artificially inflate the number of social usage counts. As a result, special attention is required to eliminate such discrepancies when using altmetrics data for smart decision-making, such as research assessment either independently or complementary along with traditional bibliometric indices.

[3]A pattern-based approach for an early detection of popular Twitter accounts

Social networks (SN) are omnipresent in lives today. Not all users have the same behaviour on these networks. If some have a low activity, rarely posting messages and following few users, some others at the other extreme have a significant activity, with many followers and regularly posts. The important role of these popular SN users makes them the target of many applications for example for content monitoring or advertising. It is therefore relevant to be able to predict as soon as possible which SN users will become popular. Existed technique for early detection of such users based on the identification of characteristic patterns. We present an index, which allows a scaling up of our approach to large social networks. Where also describe our rest experiments that confirm the validity of our approach.

[4]Online Social Networks and Media_ On the capability of evolved spambots to evade detection via genetic engineering

Since decades, genetic algorithms have been used as an effective heuristic to solve optimization problems. However, in order to be applied, genetic algorithms may require a string-based genetic encoding of in- formation, which severely limited their applicability when dealing with online accounts. Remarkably, a behavioral modeling technique inspired by biological DNA has been recently proposed –and success- fully applied –for monitoring and detecting spambots in Online Social Networks. In this so-called *digital DNA* representation, the behavioral lifetime of an account is encoded as a sequence of characters, namely a digital DNA sequence. The authors proposed to create synthetic digital DNA sequences that resemble the characteristics of the digital DNA sequences of real accounts. The combination of (i) the capability to model the accounts' behaviors as digital DNA sequences, (ii) the possibility to create synthetic digital DNA sequences, and (iii) the evolutionary simulations allowed by genetic algorithms, open up the unprecedented opportunity to study –and even *anticipate* –the evolutionary patterns of modern social spambots. In this paper, we experiment with a novel ad-hoc genetic algorithm that allows to obtain behaviorally evolved spambots. By varying the different parameters of the genetic algorithm, we are able to evaluate the capability of the evolved spambots to escape a state-of-art behavior-based detection technique. Notably, despite such detection technique achieved excellent performances in the recent past, a number of our spambot evolutions manage to escape detection. An analysis, if carried out at large-scale, would allow to proactively identify possible spambot evolutions capable of evading current detection techniques.

[5] Twitter User Profiling: Bot and Gender Identification

Social bots are automated programs that generate a significant amount of social media content. This content can be harmful, as it may target a certain audience to influence opinions, often politically motivated, or to promote individuals to appear more popular than they really are. Then proposed a set of feature extraction and transformation methods in conjunction with ensemble classifiers for the PAN 2019 Author Profiling task. For the Bot identification subtask we used user behavior fingerprint and statistical diversity measures, while for the gender identification subtask we used a set of text statistics, as well as syntactic information and raw words.

CHAPTER III

EXISTING WORK

3.1 DESCRIPTION OF EXISTING TECHNIQUE

3.1.1 DNA MODELING

We define a digital DNA sequence as a row-vector of characters (i.e., a string),

$$s = (b1, b2, \dots, bn)\ b_i\ \varepsilon\ \mathbb{B} \vee i = 1, \dots, n$$

Characters bi in s are drawn from a finite set, called alphabet,

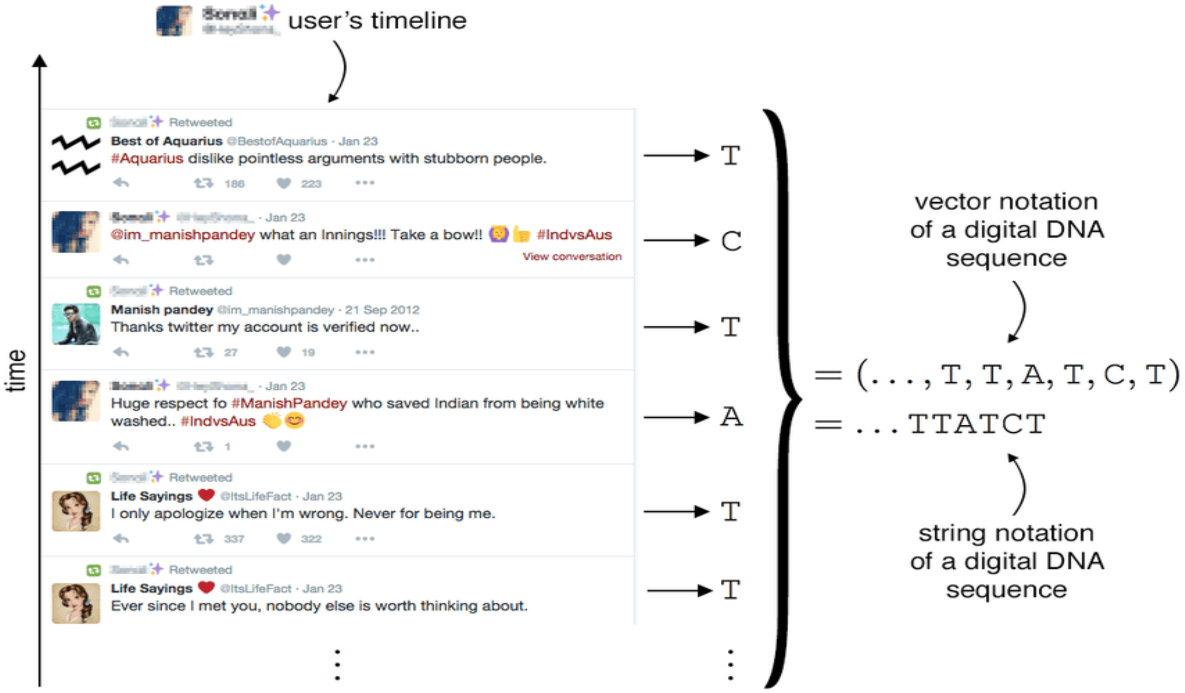
$$\mathbb{B} = \{B1,B2, \dots ,B_N\}\ B_i \neq B_j \vee i, j = 1, \dots ,N\ i \neq j$$

$$\mathbb{B}^3_{type} = \left\{\begin{array}{l} \mathtt{A} \Leftarrow \text{tweet}, \\ \mathtt{C} \Leftarrow \text{reply}, \\ \mathtt{T} \Leftarrow \text{retweet} \end{array}\right\} = \{\mathtt{A}, \mathtt{C}, \mathtt{T}\}$$

Fig 3.1 $\mathbb{B}^3_{type}$ alphabets

The B_i characters are also called the (DNA) bases of the alphabet $\mathbb{B}$. By representation of user's behavior with a digital DNA sequence by encoding each action of the user with a base. Then, by scanning user's actions in chronological order and by assigning the appropriate base to each action, we obtain the sequence of characters that makes up the digital DNA sequence of the user. Fig 3.2 shows the process of extracting the digital DNA sequence of a Twitter user, by scanning its timeline according to the alphabet $\mathbb{B}^3_{type}$ defined in Fig 3.1.

Fig 3.1 gives the definitions of all the alphabets used in this work. These alphabets represent possible encodings for OSNs actions, and have been already adopted in previous studies based on digital DNA . The intuition behind the alphabet notation is as follows. $\mathbb{B}$ stands for the set of bases bi, that is, the characters with which the DNA string can be composed. The superscript indicates the number of characters and the subscript what we intend to capture in the tweet. In summary, the $\mathbb{B}^3{}_{\text{type}}$ alphabet encodes user behaviors according to the type of tweets produced, either tweets, retweets, or replies. Alphabets $\mathbb{B}^3$ content and $\mathbb{B}^6$ content provide a way to model Twitter actions, with different granularities, by looking at the content of tweets rather than the type. In order to easily classify a tweet based on its content we exploited Twitter's notion of entities.


[redacted] user's timeline
time
T
C
T
A
T
T
vector notation of a digital DNA sequence
string notation of a digital DNA sequence

[redacted] Retweeted
Best of Aquarius @BestofAquarius · Jan 23
#Aquarius dislike pointless arguments with stubborn people.
186 223

[redacted] · Jan 23
@im_manishpandey what an Innings!!! Take a bow!! 👸👍 #IndvsAus
View conversation

[redacted] Retweeted
Manish pandey @im_manishpandey · 21 Sep 2012
Thanks twitter my account is verified now..
27 19

[redacted] · Jan 23
Huge respect fo #ManishPandey who saved Indian from being white washed.. #IndvsAus 👏😊
1

[redacted] Retweeted
Life Sayings ❤ @ItsLifeFact · Jan 23
I only apologize when I'm wrong. Never for being me.
337 322

[redacted] Retweeted
Life Sayings ❤ @ItsLifeFact · Jan 23
Ever since I met you, nobody else is worth thinking about.

$$= (\ldots, \mathtt{T}, \mathtt{T}, \mathtt{A}, \mathtt{T}, \mathtt{C}, \mathtt{T})$$
$$= \ldots\mathtt{TTATCT}$$

Fig 3.2 Except of a digital DNA extraction process for a Twitter user with the $\mathbb{B}^3{}_{\text{type}}$ alphabet

Another possible way of modeling the content of tweets could have involved the detection of the topic of a tweet . Then, it would have been possible to define an alphabet so as to have a different base for each of the main topics, such as politics, sports, technology, music, etc. Anyway, for the sake of simplicity, in our work we only exploited Twitter entities in order to obtain DNA sequences based on the content of tweets. Instead, we defined the alphabets $\mathrm{B}^3{}_{\text{interaction}}$ and $\mathrm{B}^6{}_{\text{interaction}}$ to capture in our model the interaction patterns of Twitter users, when considering the digital DNA sequences of a group of users. The idea is to employ different bases with respect to the popularity level of the peers with whom a given user interacts. Specifically, we exploited retweets and replies between users as a form of interaction and an account's followers count as a measure of popularity for that account. For this purpose, we computed the distribution of followers counts among a random sample of 400,000 Twitter users: such distribution is skewed towards low followers count values for ordinary users, since only few outliers count up to millions of followers, representing the Twitter celebrities. Then, in the definition of the alphabet $B^3_{\text{interaction}}$, we defined a base to represent interactions with celebrity users, another (different) base to represent interactions with ordinary users, and one last base to represent tweets that are not interactions (i.e., that are not retweets nor replies). In the definition of the $B^6_{\text{interaction}}$ alphabet we expanded this model by including more bases for the different levels of popularity determined by the percentiles of the distribution of followers count values. For instance, this approach could be used in order to understand whether users having a given level of popularity in a social network typically interact with other users on the same level, thus following an associative law.

Finally, the alphabet $B^3_{\text{account-age}}$ shows another possible way of modeling user interactions. In contrast with the two $B_{\text{interaction}}$ alphabets, this time the different bases represent the age of the accounts (i.e., time since the account's creation) with whom users interact, rather than their popularity. We used fixed thresholds to classify accounts as being new users in the Twitter platform (account age < 6 months), middle age users (6 months _ account age < 3 years) or longtime users (accounts age _ 3 years). We defined the alphabet as having one base for each of these 3 classes of accounts. Notably, to mark a difference with $B_{\text{ interaction}}$ alphabets, in $B^3_{\text{account-age}}$ we did not include a base for tweets that do not represent interactions. As a consequence, such tweets are not represented in digital DNA sequences obtained with $B^3_{account\text{-}age}$. As introduced before, in the above notations, alphabets are characterized by a subscript (e.g.,type) that identifies the kind of information captured by the bases, and by a superscript (e.g., 3) that denotes the number N of bases in the alphabet. These two indices are typically enough to unequally identify an alphabet. As demonstrated by the $B^3_{content}$ and $B^6{}_{content}$ alphabets, the superscript is useful to distinguish alphabets modeling the same facet with a different number of bases. Closing, it is worth noting like, although we limit our analyses to the digital DNA sequences obtained from the B^6 alphabets defined in Fig 3.1, the DNA modeling technique is suitable to encode a broad range of different behaviors and interactions, thus representing a flexible framework for online behavioral analyses. For instance, even if the account actions are ordered chronologically, our alphabets lack a temporal notion, which can be represented by the moment, or by the frequency, with which a tweet, a retweet, a mention are posted. The flexibility of the modeling technique makes easily possible to extend the alphabets with those kinds of temporal information.

3.1.2 LONGEST COMMON SUBSTRING

The Process is defined as the longest substring that is common to all the given sequences, provided that the elements of the substring are not required to occupy consecutive positions within the original sequences. It differs from the longest common substring problem: unlike substrings, substrings are not required to occupy consecutive positions within the original sequences.

The problem is a classic computer science problem, the basis of data comparison programs such as the diff utility, and has applications in computational linguistics and bioinformatics. It is also widely used by revision control systems such as get for reconciling multiple changes made to a revision-controlled collection of files.

3.3 EXISTING RESULTS

DNA LENGTH

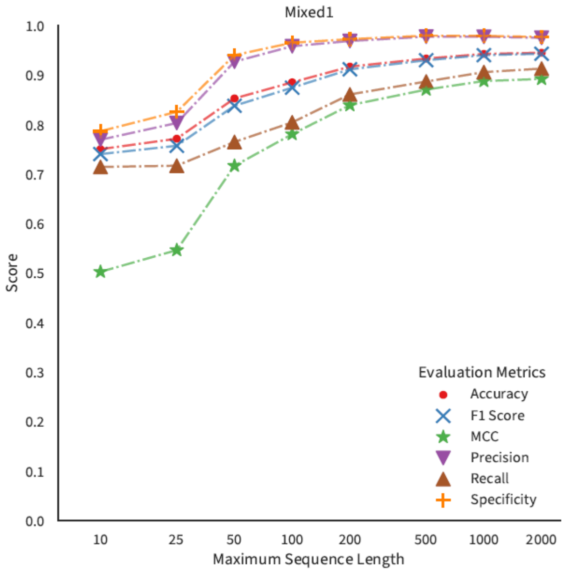


Fig 3.3 Mixed 1

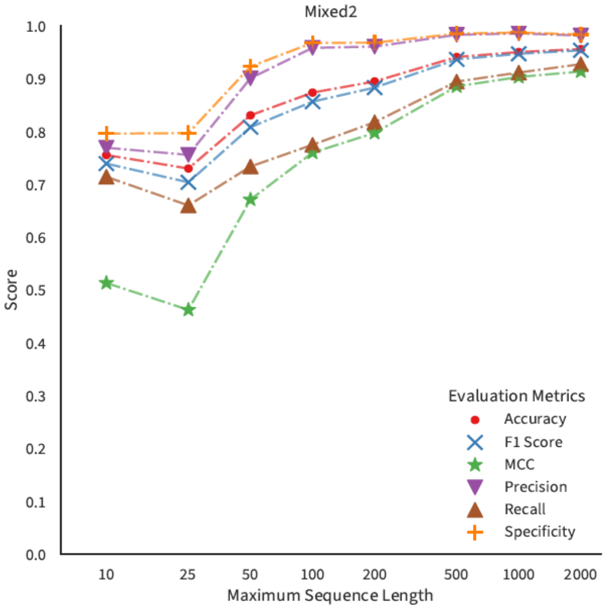


Fig 3.4 Mixed 2

DNA BOT PROPORTION

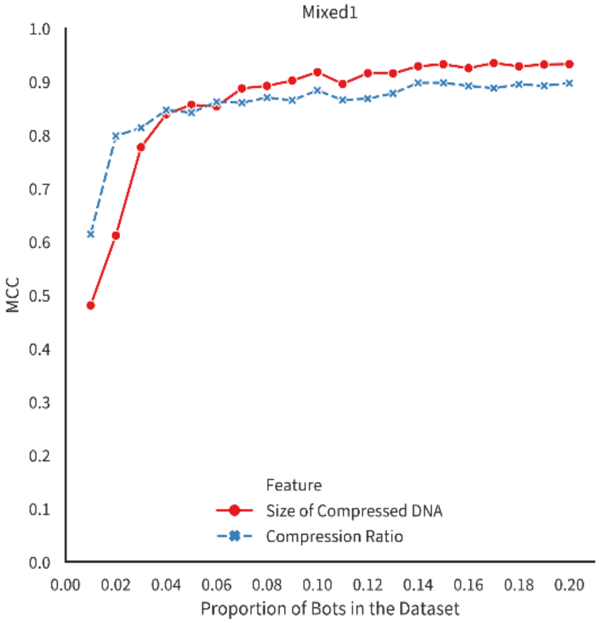


Fig 3.5 Bot Proportion 1

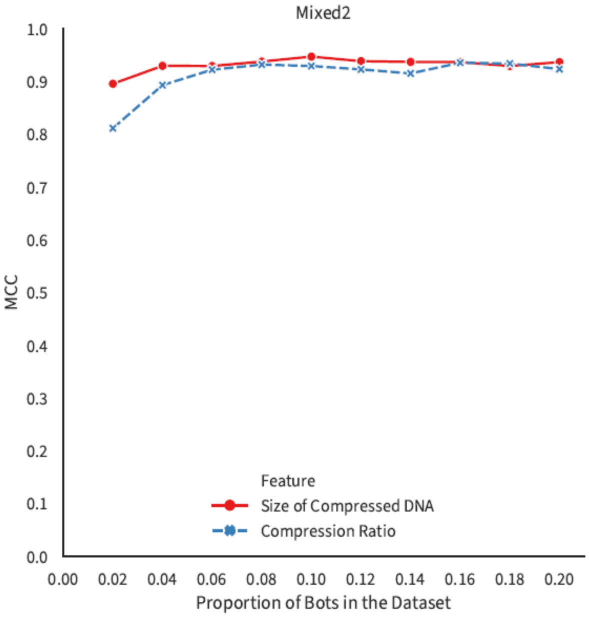


Fig 3.6 Bot Proportion 2

3.4 SUMMARY OF CONTRIBUTION

By analysing the collective behaviors of Twitter users. By leveraging a novel algorithm used to generate synthetic traces of human behavior, they demonstrated that such behaviors although extremely heterogeneous are not comparable to sequences of random actions. They showed evidence that user online activities lack a memory effect. Instead, they are characterized by a law of diminishing returns that follows a Weibull growth model. Most importantly, they also demonstrated that the emergence of behavioral similarities among a group of OSN users obey a log-normal law. Finally, exploited this log-normal characterization of human group behaviors to design a novel bot detection system that we successfully tested against four real-world datasets. Our results, other than being interesting on their own, also pave the way for future online behavioral research. Indeed, the diminishing returns property of our behavioral models suggests that null models relying either on block bootstraps or permutations fail in efficiently resembling real online behaviors. For this reason, in the future they could be replaced by other more powerful models for obtaining more accurate representations of human online behaviors. A promising line of direction could consist in letting online accounts evolve, as done in a recent work by means of a genetic algorithm, so that to evaluate the improvements one could achieve by adopting a mix of the two techniques, i.e., mutations and crossovers that are key elements of each genetic algorithm, and block resampling strategies, shown in this manuscript. Furthermore, regarding the choice of modeling online accounts through their digital DNA, the temporal dimension can convey important information about the nature of an account. For example, it might be supportive to consider not only the chronological order in which the account performs the actions, i.e., how the digital DNA model works now, but also to record each action with the timestamp at which it was performed. This would allow us to highlight, for example, groups of accounts that do certain actions in the same time frame. We will consider an extension of the DNA alphabets in future work.

CHAPTER IV

PROPOSED SYSTEM

4.1 SYSTEM ARCHITECTURAL DESIGN

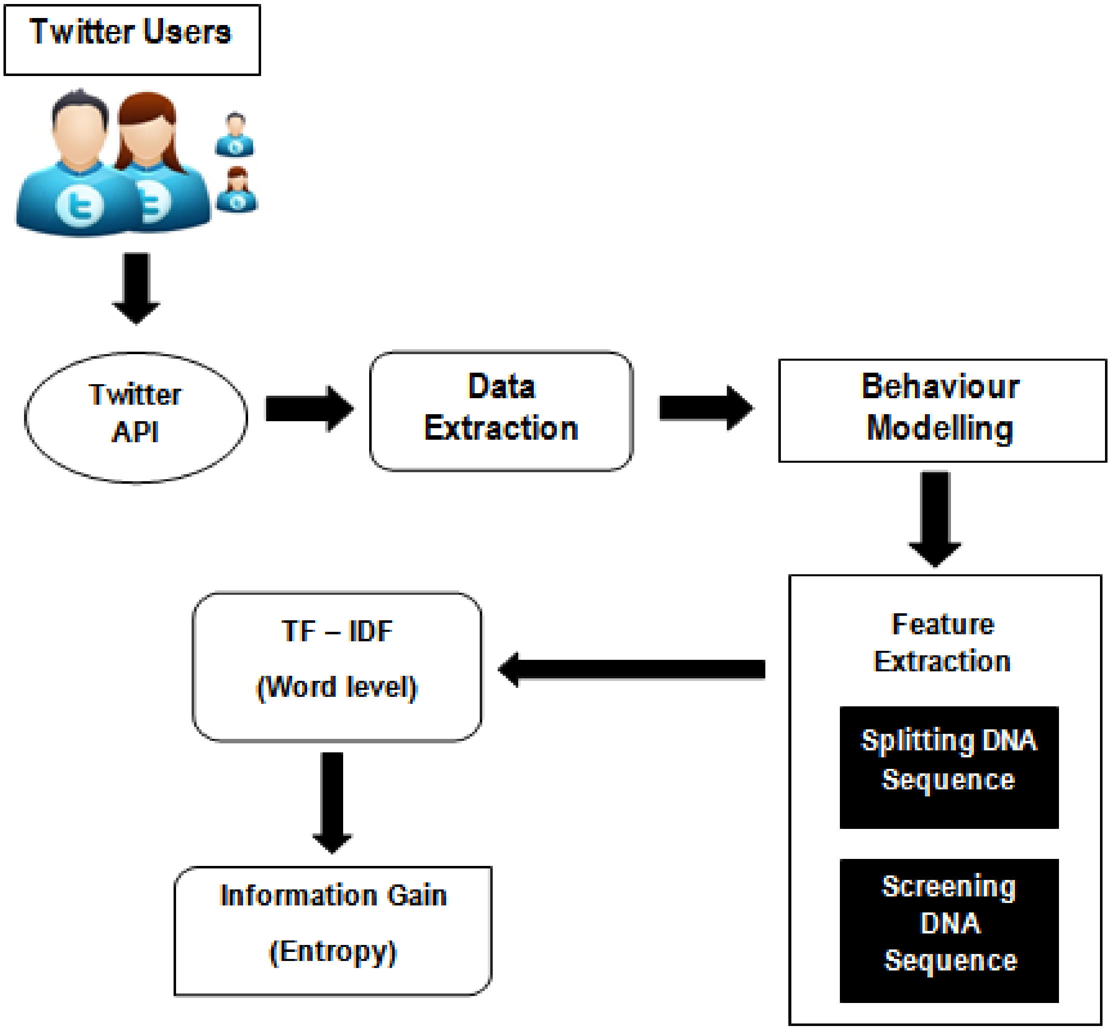


Fig 4.1 System Architecture

4.2 ARCHITECTURAL COMPONENT EXPLANATION

4.2.1 TWITTER API

The Twitter API enables programmatic access to Twitter in unique and advanced ways. Use it to analyze, learn from, and interact with Tweets, Direct Messages, users, and other key Twitter resources. Twitter for Websites. Twitter for Websites brings live conversation from Twitter to your website or app.

Twitter allows access to parts of our service via APIs to allow people to build software that integrates with Twitter, like a solution that helps a company respond to customer feedback on Twitter. Our API platform provides broad access to public Twitter data that users have chosen to share with the world.

APIs consist of three parts:

- User: the person who makes a request
- Client: the computer that sends the request to the server
- Server: the computer that responds to the request

Someone will build the server first, since it acquires and holds data. Once that server is running, programmers publish documentation, including the endpoints where specific data can be found. This documentation tells outside programmers the structure of the data on the server. An outside user can then query (or search) the data on the server, or build a program that runs searches on the database and transforms that information into a different, usable format.

4.2.2 DATA EXTRACTION

Data extraction involves pulling data from different sources and converting it into a useful format for further processing or analysis. In this tutorial, we will use python libraries such as pandas, json, and requests to read data from different sources and load them into a Jupyter notebook as a pandas dataframe.

Data extraction involves pulling data from different sources and converting it into a useful format for further processing or analysis. It is the first step of the Extract-Transform-Load pipeline (ETL) in the data engineering process. As a data scientist, you might need to combine data that is available in multiple file formats such as JSON, XML, CSV and SQL.We will use python libraries such as pandas, json, and requests to read data from different sources and load them into a Jupyter notebook as a pandas dataframe.

4.2.3 BEHAVIOUR ANALYSIS

The purpose is to activate latent users posts by modeling user behaviors by a transition of clusters that represent particular posting activities. Twitter has rapidly spread and become an easy and convenient that enables users to exchange instant text messages called tweets. There are so many latent users whose posting activities have decreased.

In this, two kinds of time-series analysis methods are proposed to clarify the lifecycles of Twitter users. In the first one, all users belong to a cluster consisting of several features at individual time slots and move among the clusters in a time series. In the second one, the posting activities of Twitter users are analyzed by the amount of tweets that vary with time.

This sophisticated evaluation using a large actual tweet-set demonstrated the proposed methods effectiveness. The authors found a big difference in the state transition diagrams between long- and short-term users. Analysis of short-term users introduces effective knowledge for encouraging continued Twitter use.

An the efficient user behavior model, which describes transitions of posting activities, is proposed. Two kinds of time longitudinal analysis method are evaluated using a large amount of actual tweets.

4.2.4 FEATURE EXTRACTION

4.2.4.1 SPLITTING DNA SEQUENCES: On Twitter could include a different base for each user-to-user interaction type: Reply(C), URL (D), Retweet (T), Tweet (A) and so on. Then, interactions can be encoded as strings formed by such characters according to the sequence of user-performed actions. Just like its biological predecessor, digital DNA is a compact representation of information for example, a Twitter user's timeline could be encoded as a single string of 3,200 characters (one character per tweet). Firstly, all tweets are sorted in chronological order. Secondly, the behavior of an account replacing each tweet, retweet & reply of this account is modelled by the characters A, D, T & C respectively, thus creating a DNA sequence.

4.2.4.2 SCREENING DNA SEQUENCE : The DNA Sequencing technique only exploits Twitter timeline data to perform bot detection. Furthermore, it does not require a training phase and can be employed pretty much like a clustering algorithm, in an unsupervised fashion. We also envision the possibility to exploit results of our DNA-inspired technique as a feature in a more complex detection system. Indeed, different types of DNA (such as the tweet type DNA and the tweet content DNA) can be exploited to model user behaviors along different directions. Then, results of these models could be used simultaneously in an ensemble or voting system. To this regard, the already interesting results achieved by exploiting only one type of digital DNA, namely tweet type DNA, might represent promising ground for further experimentation and research.

4.2.5 TF-IDF

TF-IDF stands for "**Term Frequency — Inverse Document Frequency**". This is a technique to quantify words in a set of documents. We generally compute a score for each word to signify its importance in the document and corpus. This method is a widely used technique in Information Retrieval and Text Mining.

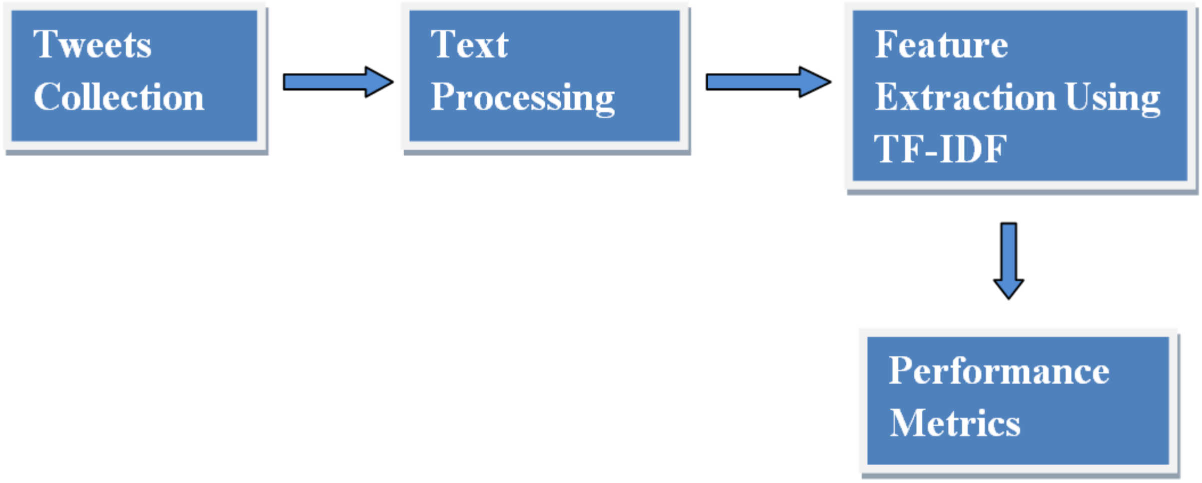


Fig 4.2 Experimental Design

TF-IDF is a popular approach used to weigh terms for NLP tasks because it assigns a value to a term according to its importance in a document scaled by its importance across all documents in your corpus, which mathematically eliminates naturally occurring words in the English language, and selects words that are more.

TERM FREQUENCY

Term frequency, tf(*t*,*d*), is the frequency of term *t*,

where $f_{t,d}$ is the *raw count* of a term in a document, i.e., the number of times that term *t* occurs in document *d*. There are various other ways to define term frequency

$$\mathrm{tf}(t,d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}}$$

- The raw count itself: tf(t,d) = $f_{t,d}$
- Boolean "frequencies": tf(t,d) = 1 if t occurs in d and 0 otherwise;
- Term frequency adjusted for document length:

$$\mathrm{tf}(t,d) = f_{t,d} \div (\text{number of words in d})$$

- Logarithmically scaled frequency: tf(t,d) = log (1 + $f_{t,d}$)
- Augmented frequency, to prevent a bias towards longer documents, e.g. raw frequency divided by the raw frequency of the most occurring term in the document

$$\mathrm{tf}(t,d) = 0.5 + 0.5 \cdot \frac{f_{t,d}}{\max\{f_{t',d} : t' \in d\}}$$

INVERSE DOCUMENT FREQUENCY

An Inverse document frequency is a measure of how much information the word provides, i.e., if it's common or rare across all documents. It is the Logarithmically scaled inverse fraction of the documents that contain the word (obtained by dividing the total number of documents by the number of documents containing the term, and then taking the logarithm of that quotient).

$$\mathrm{idf}(t,D) = \log \frac{N}{|\{d \in D : t \in d\}|}$$

with

- N: total number of documents in the corpus N= | D |

- $|\{d\ \varepsilon\ D: t\ \varepsilon\ D\}|$: number of documents where the term t appears (i.e., tf(t,d)≠0). If the term is not in the corpus, this will lead to a division-by-zero. It is therefore common to adjust the denominator to $1 + \ |\{d\ \varepsilon\ D: t\ \varepsilon\ D\}|$.

Then **tf–idf** is calculated as

$$tfidf(t,d,D) = tf(t,d).idf(t,D)$$

- A high weight in tf–idf is reached by a high term frequency (in the given document) and a low document frequency of the term in the whole collection of documents; the weights hence tend to filter out common terms. Since the ratio inside the idf's log function is always greater than or equal to 1, the value of idf (and tf–idf) is greater than or equal to 0. As a term appears in more documents, the ratio inside the logarithm approaches 1, bringing the idf and tf–idf closer to 0.

4.2.6 INFORMATION GAIN

Information gain is the reduction in entropy or surprise by transforming a dataset and is often used in training decision trees. Information gain is calculated by comparing the entropy of the dataset before and after a transformation.

We may considered to recall that information quantifies how surprising an event is in bits. Lower probability events have more information, higher probability events have less information. **Entropy** quantifies how much information there is in a random variable, or more specifically its probability distribution. A skewed distribution has a low entropy, whereas a distribution where events have equal probability has a larger entropy.

Information gain (T,x)=Entropy(T)-Average Entropy(T,x)

It measures how much "information" a feature gives us about the class. Entropy is the measures of impurity, disorder or uncertainty in a bunch of examples. Entropy controls how a Decision Tree decides to split the data. It actually effects how a Decision Tree draws its boundaries.

CHAPTER V

SIMULATION RESULTS

5.1 TOOL/PLATFORM

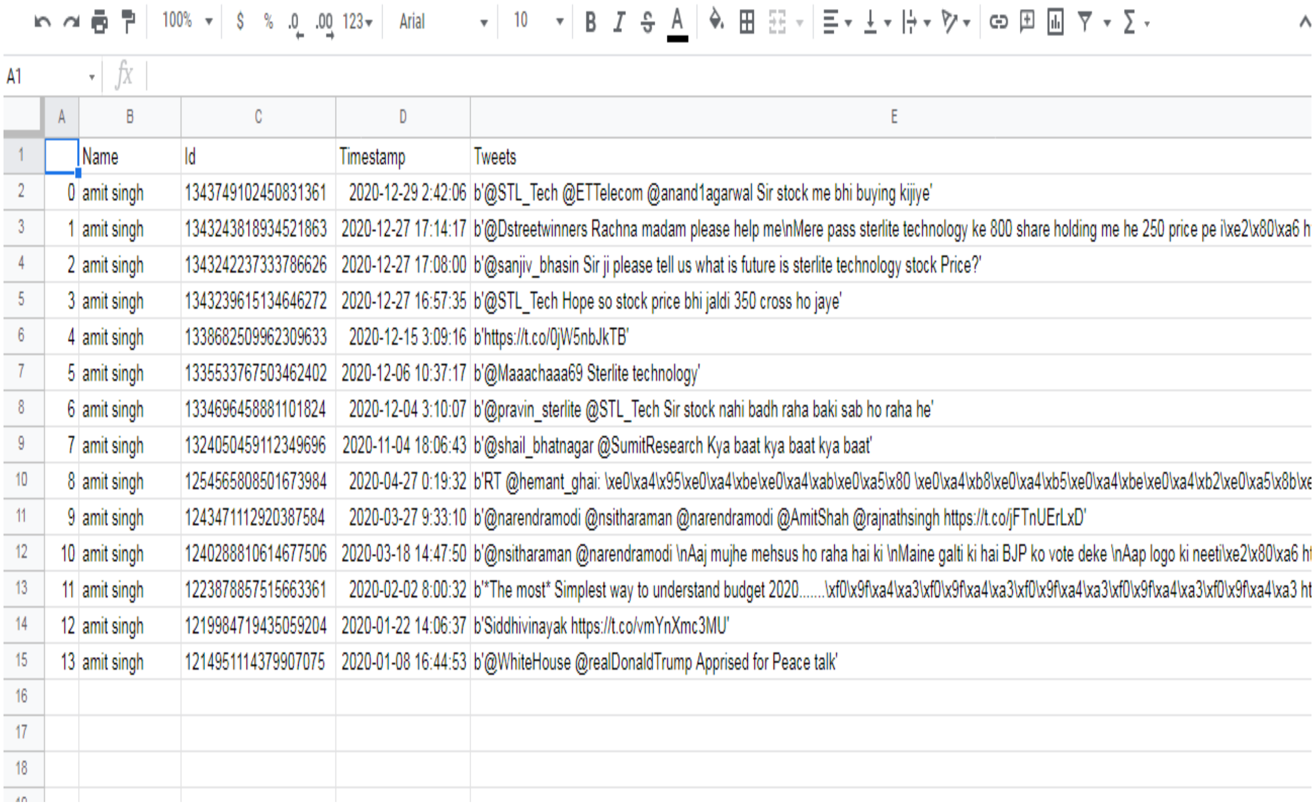
5.1.1 HARDWARE REQUIREMENTS

- **RAM: Minimum 4GB**
- **ROM: Minimum 20GB**

5.1.2 SOFTWARE REQUIREMENTS

- **Jupyter Notebook**
- **Python 3**

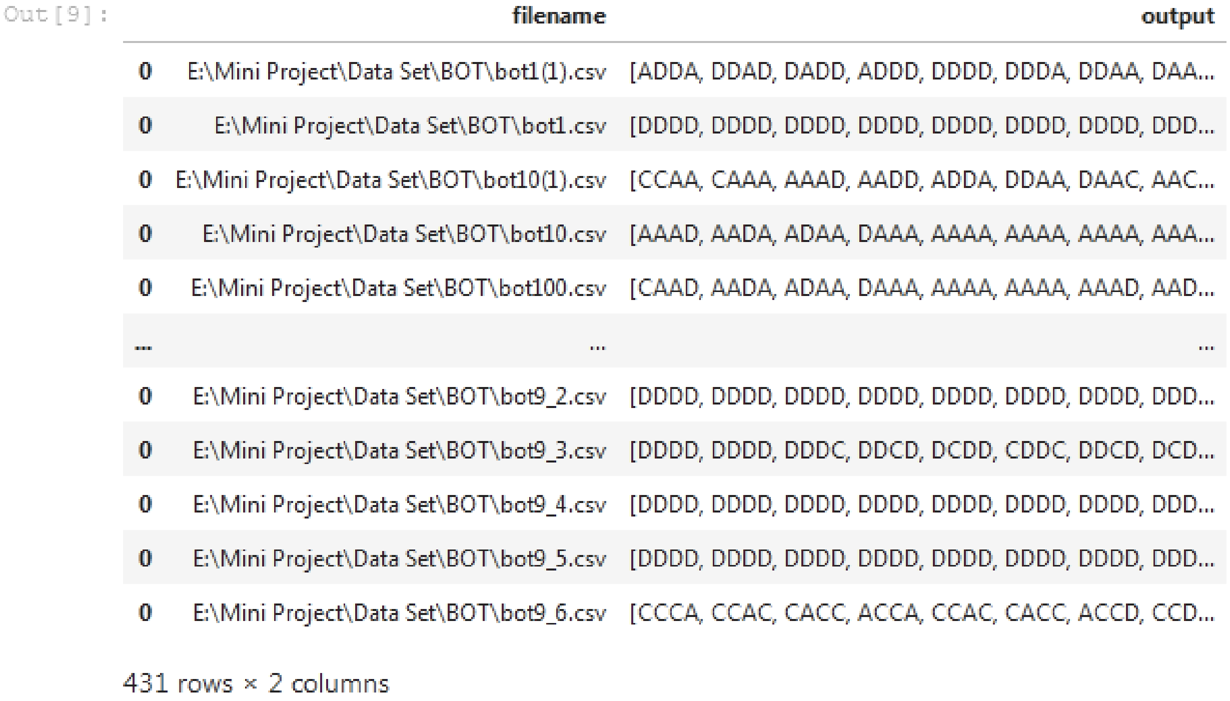
5.2 DATASET



	A	B	C	D	E
1		Name	Id	Timestamp	Tweets
2	0	amit singh	1343749102450831361	2020-12-29 2:42:06	b'@STL_Tech @ETTelecom @anand1agarwal Sir stock me bhi buying kijiye'
3	1	amit singh	1343243818934521863	2020-12-27 17:14:17	b'@Dstreetwinners Rachna madam please help me\nMere pass sterlite technology ke 800 share holding me he 250 price pe i\xe2\x80\xa6 h
4	2	amit singh	1343242237333786626	2020-12-27 17:08:00	b'@sanjiv_bhasin Sir ji please tell us what is future is sterlite technology stock Price?'
5	3	amit singh	1343239615134646272	2020-12-27 16:57:35	b'@STL_Tech Hope so stock price bhi jaldi 350 cross ho jaye'
6	4	amit singh	1338682509962309633	2020-12-15 3:09:16	b'https://t.co/0jW5nbJkTB'
7	5	amit singh	1335533767503462402	2020-12-06 10:37:17	b'@Maaachaaa69 Sterlite technology'
8	6	amit singh	1334696458881101824	2020-12-04 3:10:07	b'@pravin_sterlite @STL_Tech Sir stock nahi badh raha baki sab ho raha he'
9	7	amit singh	1324050459112349696	2020-11-04 18:06:43	b'@shail_bhatnagar @SumitResearch Kya baat kya baat kya baat'
10	8	amit singh	1254565808501673984	2020-04-27 0:19:32	b'RT @hemant_ghai: \xe0\xa4\x95\xe0\xa4\xbe\xe0\xa4\xab\xe0\xa5\x80 \xe0\xa4\xb8\xe0\xa4\xb5\xe0\xa4\xbe\xe0\xa4\xb2\xe0\xa5\x8b\xe
11	9	amit singh	1243471112920387584	2020-03-27 9:33:10	b'@narendramodi @nsitharaman @narendramodi @AmitShah @rajnathsingh https://t.co/jFTnUErLxD'
12	10	amit singh	1240288810614677506	2020-03-18 14:47:50	b'@nsitharaman @narendramodi \nAaj mujhe mehsus ho raha hai ki \nMaine galti ki hai BJP ko vote deke \nAap logo ki neeti\xe2\x80\xa6 ht
13	11	amit singh	1223878857515663361	2020-02-02 8:00:32	b'*The most* Simplest way to understand budget 2020.......\xf0\x9f\xa4\xa3\xf0\x9f\xa4\xa3\xf0\x9f\xa4\xa3\xf0\x9f\xa4\xa3\xf0\x9f\xa4\xa3 ht
14	12	amit singh	1219984719435059204	2020-01-22 14:06:37	b'Siddhivinayak https://t.co/vmYnXmc3MU'
15	13	amit singh	1214951114379907075	2020-01-08 16:44:53	b'@WhiteHouse @realDonaldTrump Apprised for Peace talk'
16					
17					
18					

Fig 5.1 Input Dataset

5.3 OUTPUT SCREENSHOT

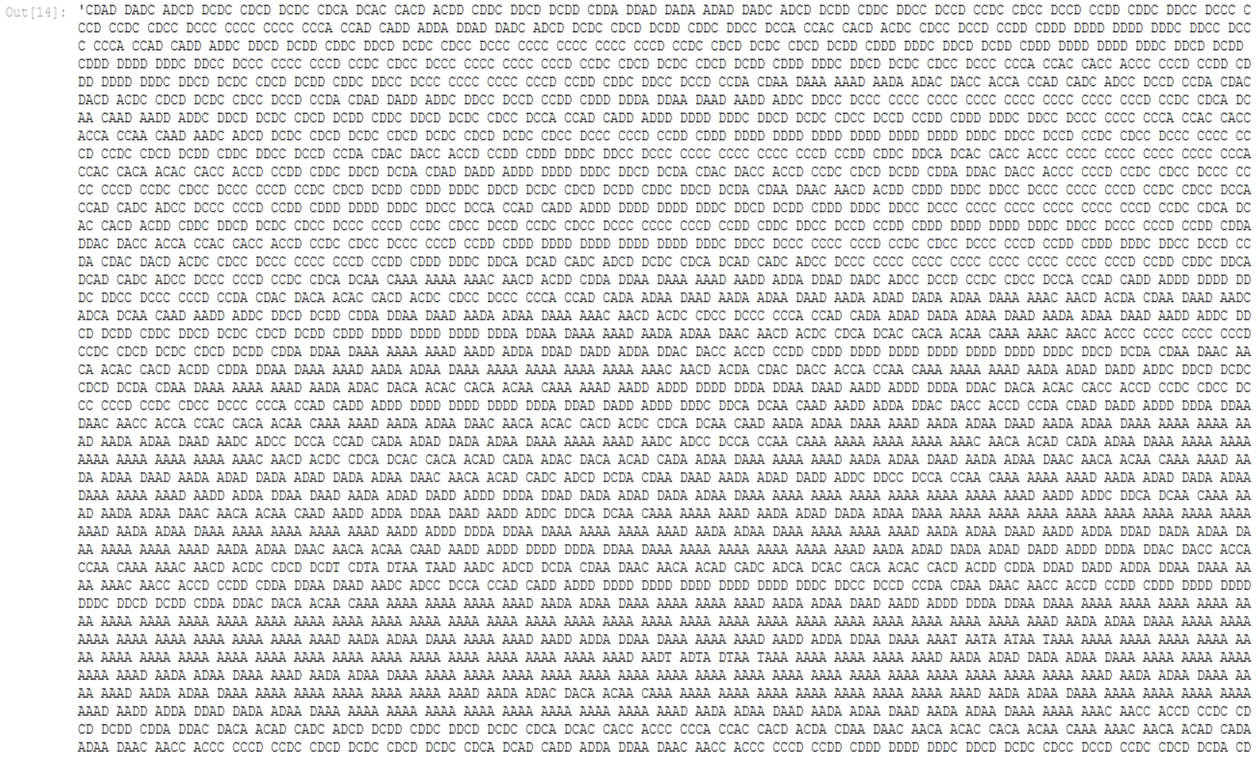


Out[9]:

	filename	output
0	E:\Mini Project\Data Set\BOT\bot1(1).csv	[ADDA, DDAD, DADD, ADDD, DDDD, DDDA, DDAA, DAA...
0	E:\Mini Project\Data Set\BOT\bot1.csv	[DDDD, DDDD, DDDD, DDDD, DDDD, DDDD, DDDD, DDD...
0	E:\Mini Project\Data Set\BOT\bot10(1).csv	[CCAA, CAAA, AAAD, AADD, ADDA, DDAA, DAAC, AAC...
0	E:\Mini Project\Data Set\BOT\bot10.csv	[AAAD, AADA, ADAA, DAAA, AAAA, AAAA, AAAA, AAA...
0	E:\Mini Project\Data Set\BOT\bot100.csv	[CAAD, AADA, ADAA, DAAA, AAAA, AAAA, AAAD, AAD...
...	...	...
0	E:\Mini Project\Data Set\BOT\bot9_2.csv	[DDDD, DDDD, DDDD, DDDD, DDDD, DDDD, DDDD, DDD...
0	E:\Mini Project\Data Set\BOT\bot9_3.csv	[DDDD, DDDD, DDDC, DDCD, DCDD, CDDC, DDCD, DCD...
0	E:\Mini Project\Data Set\BOT\bot9_4.csv	[DDDD, DDDD, DDDD, DDDD, DDDD, DDDD, DDDD, DDD...
0	E:\Mini Project\Data Set\BOT\bot9_5.csv	[DDDD, DDDD, DDDD, DDDD, DDDD, DDDD, DDDD, DDD...
0	E:\Mini Project\Data Set\BOT\bot9_6.csv	[CCCA, CCAC, CACC, ACCA, CCAC, CACC, ACCD, CCD...

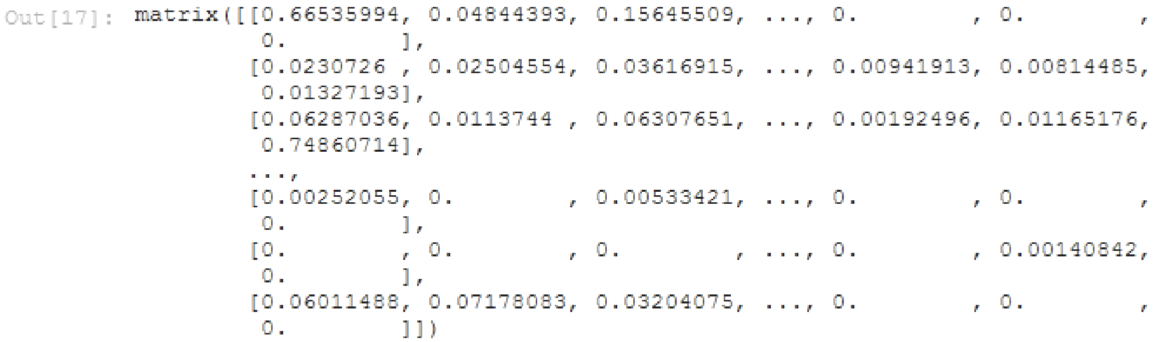
431 rows × 2 columns

Fig 5.2 Feature Splitting



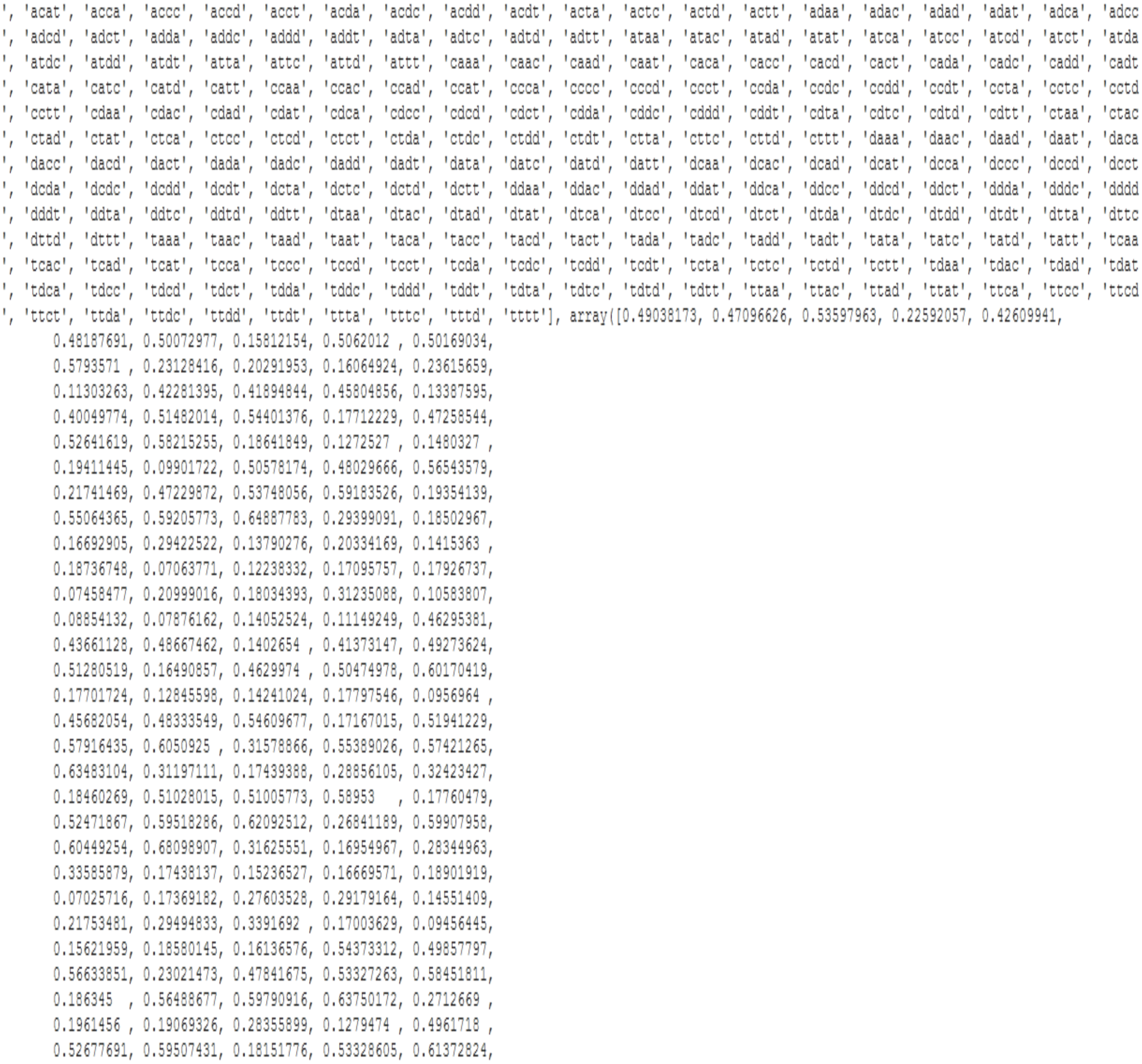
```
Out[14]: 'CDAD DADC ADCD DCDC CDCD DCDC CDCA DCAC CACD ACDD CDDC DDCD DCDD CDDA DDAD DADA ADAD DADC ADCD DCDD CDDC DDCC DCCD CCDC CDCC DCCD CCDD CDDC DDCC DCCC C
         CCD CCDC CDCC DCCC CCCC CCCC CCCA CCAD CADD ADDA DDAD DADC ADCD DCDC CDCD DCDD CDDC DDCC DCCA CCAC CACD ACDC CDCC DCCD CCDD CDDD DDDD DDDD DDDC DDCC DCC
         C CCCA CCAD CADD ADDC DDCD DCDD CDDC DDCD DCDC CDCC DCCC CCCC CCCC CCCC CCCD CCDC CDCD DCDC CDCD DCDD CDDD DDDC DDCD DCDD CDDD DDDD DDDD DDDC DDCD DCDD
         CDDD DDDD DDDC DDCC DCCC CCCC CCCD CCDC CDCC DCCC CCCC CCCC CCCD CCDC CDCD DCDC CDCD DCDD CDDD DDDC DDCD DCDC CDCC DCCC CCCA CCAC CACC ACCC CCCD CCDD CD
         DD DDDD DDDC DDCD DCDC CDCD DCDD CDDC DDCC DCCC CCCC CCCC CCCD CCDD CDDC DDCC DCCD CCDA CDAA DAAA AAAD AADA ADAC DACC ACCA CCAD CADC ADCC DCCD CCDA CDAC
         DACD ACDC CDCD DCDC CDCC DCCD CCDA CDAD DADD ADDC DDCC DCCD CCDD CDDD DDDA DDAA DAAD AADD ADDC DDCC DCCC CCCC CCCC CCCC CCCC CCCC CCCC CCCD CCDC CDCA DC
         AA CAAD AADD ADDC DDCD DCDC CDCD DCDD CDDC DDCD DCDC CDCC DCCA CCAD CADD ADDD DDDD DDDC DDCD DCDC CDCC DCCD CCDD CDDD DDDC DDCC DCCC CCCC CCCA CCAC CACC
         ACCA CCAA CAAD AADC ADCD DCDC CDCD DCDC CDCD DCDC CDCD DCDC CDCC DCCC CCCD CCDD CDDD DDDD DDDD DDDD DDDD DDDD DDDD DDDC DDCC DCCD CCDC CDCC DCCC CCCC CC
         CD CCDC CDCD DCDD CDDC DDCC DCCD CCDA CDAC DACC ACCD CCDD CDDD DDDC DDCC DCCC CCCC CCCC CCCC CCCD CCDD CDDC DDCA DCAC CACC ACCC CCCC CCCC CCCC CCCC CCCA
         CCAC CACA ACAC CACC ACCD CCDD CDDC DDCD DCDA CDAD DADD ADDD DDDD DDDC DDCD DCDA CDAC DACC ACCD CCDC CDCD DCDD CDDA DDAC DACC ACCC CCCD CCDC CDCC DCCC CC
         CC CCCD CCDC CDCC DCCC CCCD CCDC CDCD DCDD CDDD DDDC DDCD DCDC CDCD DCDD CDDC DDCD DCDA CDAA DAAC AACD ACDD CDDD DDDC DDCC DCCC CCCC CCCD CCDC CDCC DCCA
         CCAD CADC ADCC DCCC CCCD CCDD CDDD DDDD DDDC DDCC DCCA CCAD CADD ADDD DDDD DDDD DDDC DDCD DCDD CDDD DDDC DDCC DCCC CCCC CCCC CCCC CCCC CCCD CCDC CDCA DC
         AC CACD ACDD CDDC DDCD DCDC CDCC DCCC CCCD CCDC CDCC DCCD CCDC CDCC DCCC CCCC CCCD CCDD CDDC DDCC DCCD CCDD CDDD DDDD DDDD DDDC DDCC DCCC CCCD CCDD CDDA
         DDAC DACC ACCA CCAC CACC ACCD CCDC CDCC DCCC CCCD CCDD CDDD DDDD DDDD DDDD DDDD DDDC DDCC DCCC CCCC CCCD CCDC CDCC DCCC CCCD CCDD CDDD DDDC DDCC DCCD CC
         DA CDAC DACD ACDC CDCC DCCC CCCC CCCD CCDD CDDD DDDC DDCA DCAD CADC ADCD DCDC CDCA DCAD CADC ADCC DCCC CCCC CCCC CCCC CCCC CCCC CCCC CCCD CCDD CDDC DDCA
         DCAD CADC ADCC DCCC CCCD CCDC CDCA DCAA CAAA AAAA AAAC AACD ACDD CDDA DDAA DAAA AAAD AADD ADDA DDAD DADC ADCC DCCD CCDC CDCC DCCA CCAD CADD ADDD DDDD DD
         DC DDCC DCCC CCCD CCDA CDAC DACA ACAC CACD ACDC CDCC DCCC CCCA CCAD CADA ADAA DAAD AADA ADAA DAAD AADA ADAD DADA ADAA DAAA AAAC AACD ACDA CDAA DAAD AADC
         ADCA DCAA CAAD AADD ADDC DDCD DCDD CDDA DDAA DAAD AADA ADAA DAAA AAAC AACD ACDC CDCC DCCC CCCA CCAD CADA ADAD DADA ADAA DAAD AADA ADAA DAAD AADD ADDC DD
         CD DCDD CDDC DDCD DCDC CDCD DCDD CDDD DDDD DDDD DDDD DDDA DDAA DAAA AAAD AADA ADAA DAAC AACD ACDC CDCA DCAC CACA ACAA CAAA AAAC AACC ACCC CCCC CCCC CCCD
         CCDC CDCD DCDC CDCD DCDD CDDA DDAA DAAA AAAA AAAD AADD ADDA DDAD DADD ADDA DDAC DACC ACCD CCDD CDDD DDDD DDDD DDDD DDDD DDDD DDDC DDCD DCDA CDAA DAAC AA
         CA ACAC CACD ACDD CDDA DDAA DAAA AAAD AADA ADAA DAAA AAAA AAAA AAAA AAAA AAAC AACD ACDA CDAC DACC ACCA CCAA CAAA AAAA AAAD AADA ADAD DADD ADDC DDCD DCDC
         CDCD DCDA CDAA DAAA AAAA AAAD AADA ADAC DACA ACAC CACA ACAA CAAA AAAD AADD ADDD DDDD DDDA DDAA DAAD AADD ADDD DDDA DDAC DACA ACAC CACC ACCD CCDC CDCC DC
         CC CCCD CCDC CDCC DCCC CCCA CCAD CADD ADDD DDDD DDDD DDDD DDDA DDAD DADD ADDD DDDC DDCA DCAA CAAD AADD ADDA DDAC DACC ACCD CCDA CDAD DADD ADDD DDDA DDAA
         DAAC AACC ACCA CCAC CACA ACAA CAAA AAAD AADA ADAA DAAC AACA ACAC CACD ACDC CDCA DCAA CAAD AADA ADAA DAAA AAAD AADA ADAA DAAD AADA ADAA DAAA AAAA AAAA AA
         AD AADA ADAA DAAD AADC ADCC DCCA CCAD CADA ADAD DADA ADAA DAAA AAAA AAAD AADC ADCC DCCA CCAA CAAA AAAA AAAA AAAA AAAC AACA ACAD CADA ADAA DAAA AAAA AAAA
         AAAA AAAA AAAA AAAA AAAC AACD ACDC CDCA DCAC CACA ACAD CADA ADAC DACA ACAD CADA ADAA DAAA AAAA AAAD AADA ADAA DAAD AADA ADAA DAAC AACA ACAA CAAA AAAD AA
         DA ADAA DAAD AADA ADAD DADA ADAD DADA ADAA DAAC AACA ACAD CADC ADCD DCDA CDAA DAAD AADA ADAD DADD ADDC DDCC DCCA CCAA CAAA AAAA AAAD AADA ADAD DADA ADAA
         DAAA AAAA AAAD AADD ADDA DDAA DAAD AADA ADAD DADD ADDD DDDA DDAD DADA ADAD DADA ADAA DAAA AAAA AAAA AAAA AAAA AAAA AAAA AAAD AADD ADDC DDCA DCAA CAAA AA
         AD AADA ADAA DAAC AACA ACAA CAAD AADD ADDA DDAA DAAD AADD ADDC DDCA DCAA CAAA AAAA AAAD AADA ADAD DADA ADAA DAAA AAAA AAAA AAAA AAAA AAAA AAAA AAAA AAAA
         AAAD AADA ADAA DAAA AAAA AAAA AAAA AAAD AADD ADDD DDDA DDAA DAAA AAAA AAAA AAAD AADA ADAA DAAA AAAA AAAA AAAD AADA ADAA DAAD AADD ADDA DDAD DADA ADAA DA
         AA AAAA AAAA AAAD AADA ADAA DAAC AACA ACAA CAAD AADD ADDD DDDD DDDA DDAA DAAA AAAA AAAA AAAA AAAA AAAD AADA ADAD DADA ADAD DADD ADDD DDDA DDAC DACC ACCA
         CCAA CAAA AAAC AACD ACDC CDCD DCDT CDTA DTAA TAAD AADC ADCD DCDA CDAA DAAC AACA ACAD CADC ADCA DCAC CACA ACAC CACD ACDD CDDA DDAD DADD ADDA DDAA DAAA AA
         AA AAAC AACC ACCD CCDD CDDA DDAA DAAD AADC ADCC DCCA CCAD CADD ADDD DDDD DDDD DDDD DDDD DDDD DDDC DDCC DCCD CCDA CDAA DAAC AACC ACCD CCDD CDDD DDDD DDDD
         DDDC DDCD DCDD CDDA DDAC DACA ACAA CAAA AAAA AAAA AAAA AAAD AADA ADAA DAAA AAAA AAAA AAAD AADA ADAA DAAD AADD ADDD DDDA DDAA DAAA AAAA AAAA AAAA AAAA AA
         AA AAAA AAAA AAAA AAAA AAAA AAAA AAAA AAAA AAAA AAAA AAAA AAAA AAAA AAAA AAAA AAAA AAAA AAAA AAAA AAAA AAAA AAAA AAAA AAAA AAAD AADA ADAA DAAA AAAA AAAA
         AAAA AAAA AAAA AAAA AAAA AAAA AAAD AADA ADAA DAAA AAAA AAAD AADD ADDA DDAA DAAA AAAA AAAD AADD ADDA DDAA DAAA AAAT AATA ATAA TAAA AAAA AAAA AAAA AAAA AA
         AA AAAA AAAA AAAA AAAA AAAA AAAA AAAA AAAA AAAA AAAA AAAA AAAA AAAA AAAD AADT ADTA DTAA TAAA AAAA AAAA AAAA AAAD AADA ADAD DADA ADAA DAAA AAAA AAAA AAAA
         AAAA AAAD AADA ADAA DAAA AAAD AADA ADAA DAAA AAAA AAAA AAAA AAAA AAAA AAAA AAAA AAAA AAAA AAAA AAAA AAAA AAAA AAAA AAAA AAAA AAAA AAAD AADA ADAA DAAA AA
         AA AAAD AADA ADAA DAAA AAAA AAAA AAAA AAAA AAAA AAAD AADA ADAC DACA ACAA CAAA AAAA AAAA AAAA AAAA AAAA AAAA AAAA AAAD AADA ADAA DAAA AAAA AAAA AAAA AAAA
         AAAD AADD ADDA DDAD DADA ADAA DAAA AAAA AAAA AAAA AAAA AAAA AAAA AAAA AAAA AAAD AADA ADAA DAAD AADA ADAA DAAD AADA ADAA DAAA AAAA AAAC AACC ACCD CCDC CD
         CD DCDD CDDA DDAC DACA ACAD CADC ADCD DCDD CDDC DDCD DCDC CDCA DCAC CACC ACCC CCCA CCAC CACD ACDA CDAA DAAC AACA ACAC CACA ACAA CAAA AAAC AACA ACAD CADA
         ADAA DAAC AACC ACCC CCCD CCDC CDCD DCDC CDCD DCDC CDCA DCAD CADD ADDA DDAA DAAC AACC ACCC CCCD CCDD CDDD DDDD DDDC DDCD DCDC CDCC DCCD CCDC CDCD DCDA CD
```

Fig 5.3 Feature Screening



```
Out[17]: matrix([[0.66535994, 0.04844393, 0.15645509, ..., 0.        , 0.        ,
                  0.        ],
                 [0.0230726 , 0.02504554, 0.03616915, ..., 0.00941913, 0.00814485,
                  0.01327193],
                 [0.06287036, 0.0113744 , 0.06307651, ..., 0.00192496, 0.01165176,
                  0.74860714],
                 ...,
                 [0.00252055, 0.        , 0.00533421, ..., 0.        , 0.        ,
                  0.        ],
                 [0.        , 0.        , 0.        , ..., 0.        , 0.00140842,
                  0.        ],
                 [0.06011488, 0.07178083, 0.03204075, ..., 0.        , 0.        ,
                  0.        ]])
```

Fig 5.4 Matrix Form Feature Screening



```
', 'acat', 'acca', 'accc', 'accd', 'acct', 'acda', 'acdc', 'acdd', 'acdt', 'acta', 'actc', 'actd', 'actt', 'adaa', 'adac', 'adad', 'adat', 'adca', 'adcc
', 'adcd', 'adct', 'adda', 'addc', 'addd', 'addt', 'adta', 'adtc', 'adtd', 'adtt', 'ataa', 'atac', 'atad', 'atat', 'atca', 'atcc', 'atcd', 'atct', 'atda
', 'atdc', 'atdd', 'atdt', 'atta', 'attc', 'attd', 'attt', 'caaa', 'caac', 'caad', 'caat', 'caca', 'cacc', 'cacd', 'cact', 'cada', 'cadc', 'cadd', 'cadt
', 'cata', 'catc', 'catd', 'catt', 'ccaa', 'ccac', 'ccad', 'ccat', 'ccca', 'cccc', 'cccd', 'ccct', 'ccda', 'ccdc', 'ccdd', 'ccdt', 'ccta', 'cctc', 'cctd
', 'cctt', 'cdaa', 'cdac', 'cdad', 'cdat', 'cdca', 'cdcc', 'cdcd', 'cdct', 'cdda', 'cddc', 'cddd', 'cddt', 'cdta', 'cdtc', 'cdtd', 'cdtt', 'ctaa', 'ctac
', 'ctad', 'ctat', 'ctca', 'ctcc', 'ctcd', 'ctct', 'ctda', 'ctdc', 'ctdd', 'ctdt', 'ctta', 'cttc', 'cttd', 'cttt', 'daaa', 'daac', 'daad', 'daat', 'daca
', 'dacc', 'dacd', 'dact', 'dada', 'dadc', 'dadd', 'dadt', 'data', 'datc', 'datd', 'datt', 'dcaa', 'dcac', 'dcad', 'dcat', 'dcca', 'dccc', 'dccd', 'dcct
', 'dcda', 'dcdc', 'dcdd', 'dcdt', 'dcta', 'dctc', 'dctd', 'dctt', 'ddaa', 'ddac', 'ddad', 'ddat', 'ddca', 'ddcc', 'ddcd', 'ddct', 'ddda', 'dddc', 'dddd
', 'dddt', 'ddta', 'ddtc', 'ddtd', 'ddtt', 'dtaa', 'dtac', 'dtad', 'dtat', 'dtca', 'dtcc', 'dtcd', 'dtct', 'dtda', 'dtdc', 'dtdd', 'dtdt', 'dtta', 'dttc
', 'dttd', 'dttt', 'taaa', 'taac', 'taad', 'taat', 'taca', 'tacc', 'tacd', 'tact', 'tada', 'tadc', 'tadd', 'tadt', 'tata', 'tatc', 'tatd', 'tatt', 'tcaa
', 'tcac', 'tcad', 'tcat', 'tcca', 'tccc', 'tccd', 'tcct', 'tcda', 'tcdc', 'tcdd', 'tcdt', 'tcta', 'tctc', 'tctd', 'tctt', 'tdaa', 'tdac', 'tdad', 'tdat
', 'tdca', 'tdcc', 'tdcd', 'tdct', 'tdda', 'tddc', 'tddd', 'tddt', 'tdta', 'tdtc', 'tdtd', 'tdtt', 'ttaa', 'ttac', 'ttad', 'ttat', 'ttca', 'ttcc', 'ttcd
', 'ttct', 'ttda', 'ttdc', 'ttdd', 'ttdt', 'ttta', 'tttc', 'tttd', 'tttt'], array([0.49038173, 0.47096626, 0.53597963, 0.22592057, 0.42609941,
       0.48187691, 0.50072977, 0.15812154, 0.5062012 , 0.50169034,
       0.5793571 , 0.23128416, 0.20291953, 0.16064924, 0.23615659,
       0.11303263, 0.42281395, 0.41894844, 0.45804856, 0.13387595,
       0.40049774, 0.51482014, 0.54401376, 0.17712229, 0.47258544,
       0.52641619, 0.58215255, 0.18641849, 0.1272527 , 0.1480327 ,
       0.19411445, 0.09901722, 0.50578174, 0.48029666, 0.56543579,
       0.21741469, 0.47229872, 0.53748056, 0.59183526, 0.19354139,
       0.55064365, 0.59205773, 0.64887783, 0.29399091, 0.18502967,
       0.16692905, 0.29422522, 0.13790276, 0.20334169, 0.1415363 ,
       0.18736748, 0.07063771, 0.12238332, 0.17095757, 0.17926737,
       0.07458477, 0.20999016, 0.18034393, 0.31235088, 0.10583807,
       0.08854132, 0.07876162, 0.14052524, 0.11149249, 0.46295381,
       0.43661128, 0.48667462, 0.1402654 , 0.41373147, 0.49273624,
       0.51280519, 0.16490857, 0.4629974 , 0.50474978, 0.60170419,
       0.17701724, 0.12845598, 0.14241024, 0.17797546, 0.0956964 ,
       0.45682054, 0.48333549, 0.54609677, 0.17167015, 0.51941229,
       0.57916435, 0.6050925 , 0.31578866, 0.55389026, 0.57421265,
       0.63483104, 0.31197111, 0.17439388, 0.28856105, 0.32423427,
       0.18460269, 0.51028015, 0.51005773, 0.58953   , 0.17760479,
       0.52471867, 0.59518286, 0.62092512, 0.26841189, 0.59907958,
       0.60449254, 0.68098907, 0.31625551, 0.16954967, 0.28344963,
       0.33585879, 0.17438137, 0.15236527, 0.16669571, 0.18901919,
       0.07025716, 0.17369182, 0.27603528, 0.29179164, 0.14551409,
       0.21753481, 0.29494833, 0.3391692 , 0.17003629, 0.09456445,
       0.15621959, 0.18580145, 0.16136576, 0.54373312, 0.49857797,
       0.56633851, 0.23021473, 0.47841675, 0.53327263, 0.58451811,
       0.186345  , 0.56488677, 0.59790916, 0.63750172, 0.2712669 ,
       0.1961456 , 0.19069326, 0.28355899, 0.1279474 , 0.4961718 ,
       0.52677691, 0.59507431, 0.18151776, 0.53328605, 0.61372824,
```

Fig 5.5 Tf-Idf of Features

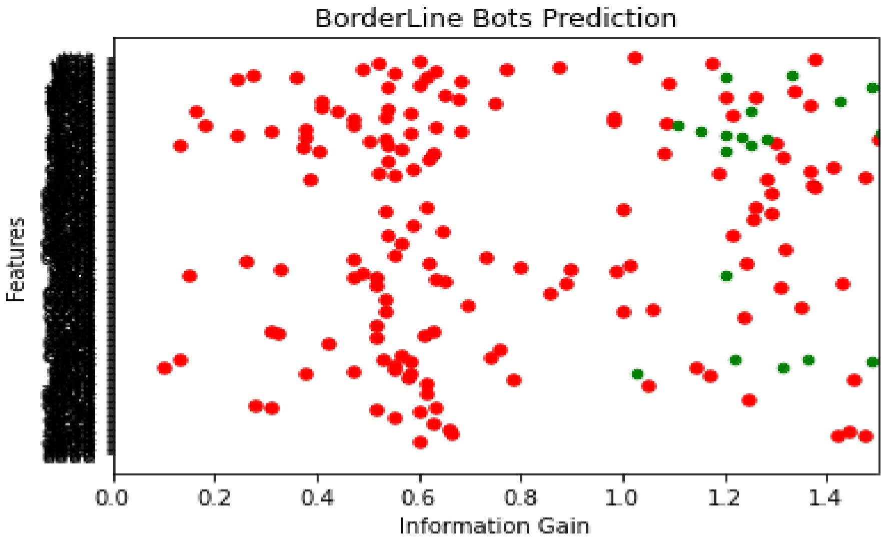


Fig 5.6 Information Gain of Bot Prediction

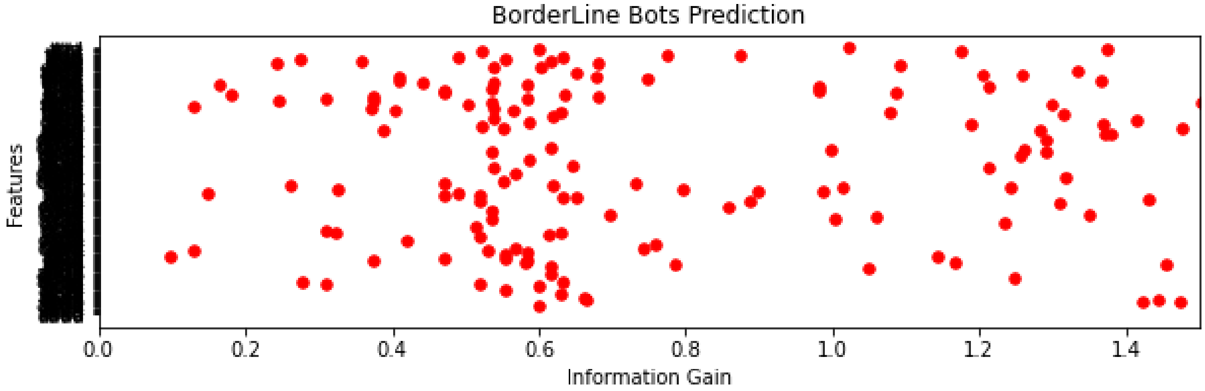


Fig 5.7 Information Gain of Borderline Bot

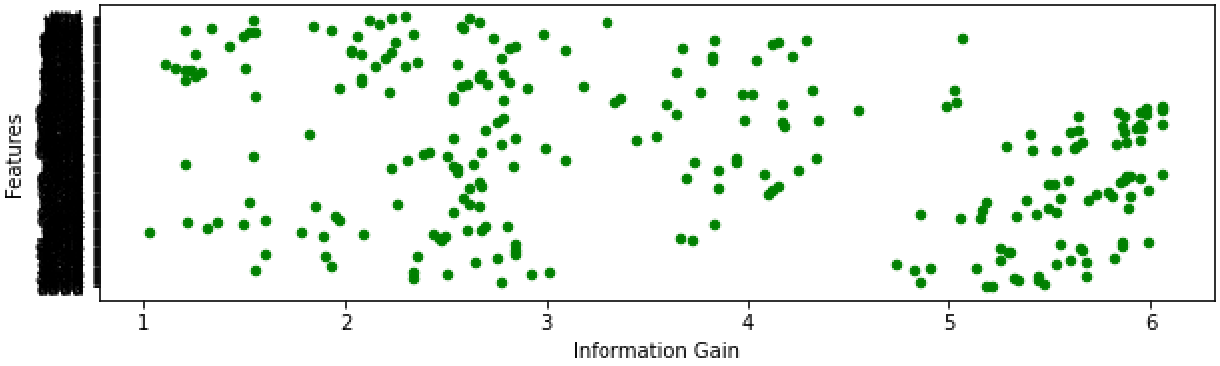


Fig 5.8 Information Gain of Human

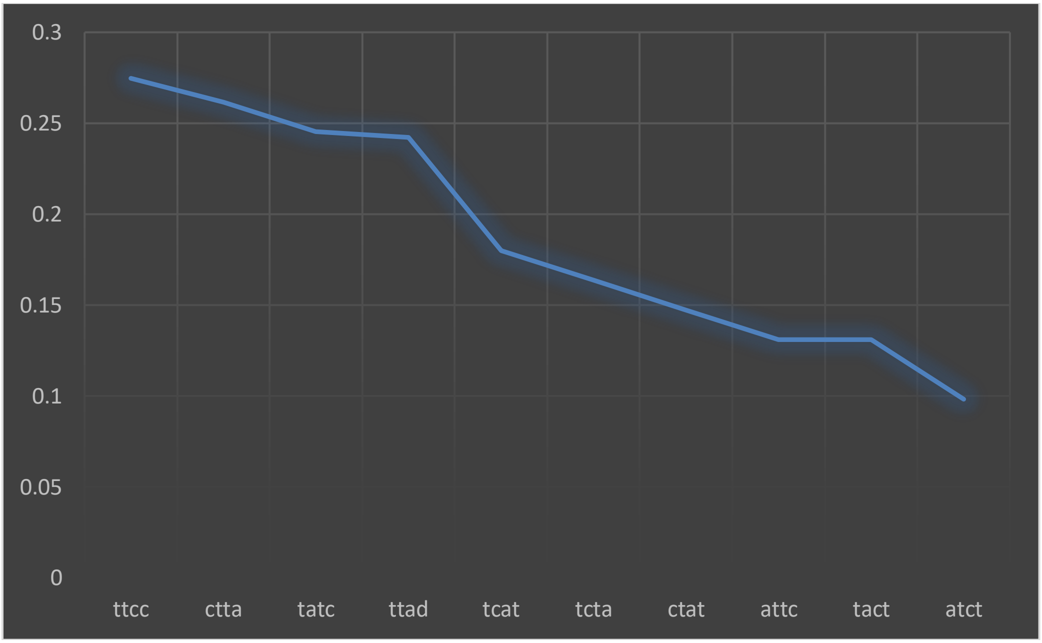


Fig 5.9 Anaysis of main features to detect bots

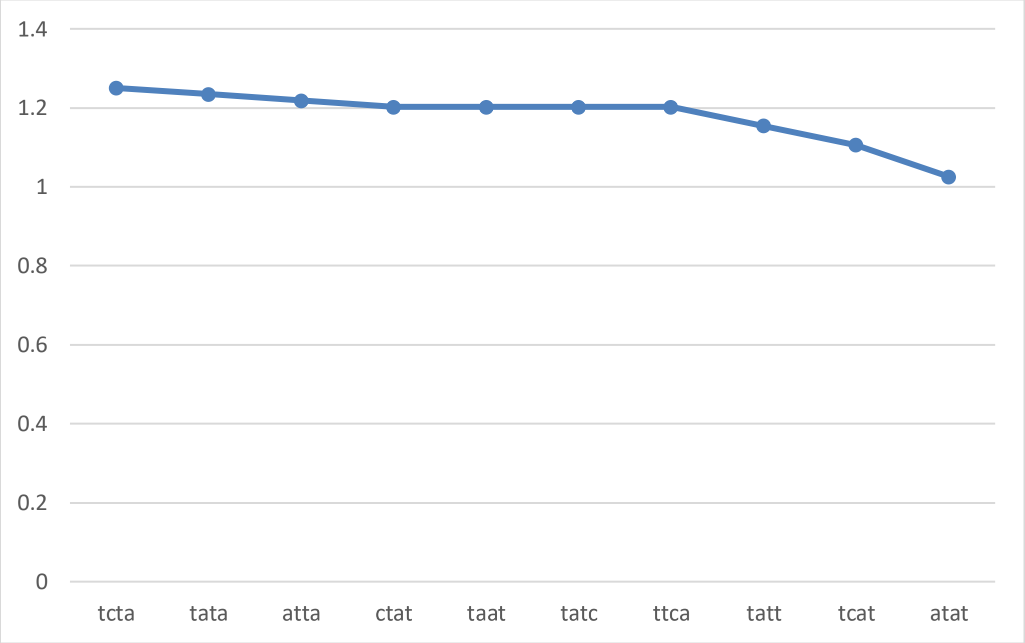


Fig 5.10 Analysis of main features to detect humans

CHAPTER VI

CONCLUSION AND FUTURE ENHANCEMENTS

In this work, we have considered only the user behaviors to detect Twitter Bots. The Research experiment includes a real time Twitter dataset collected through Twitter API. We model the user behaviors as DNA sequence and design Information gain (Entropy) based approach to analyze borderline bots from humans. The Essence of our proposed work identifies bots from humans through the entropy estimate on DNA sequences, which acts as the indicator for automation. Our modal achieves a significant results than existing approaches with F1 measure=0.9401 and accuracy=0.9490. Based on the interesting outcomes achieved using only profile’s timeline, the potency of information gain (entropy) based technique is foreseen. For future work, we plan to analyze entropy on various DNA modals of different lengths. Also, the performance of entropy on other attributes can be investigated.

REFERENCES

[1] S Cresci, R Di Pietro, M Petrocchi, A Spognardi, M Tesconi, “Emergent properties, models, and laws of behavioral similarities within groups of twitter users”, 2020.

[2] Aljohani NR, Fayoumi A, Hassan SU, “Bot prediction on social networks of Twitter in altmetrics using deep graph convolutional networks”, 2020.

[3] Jonathan Debure, Stephan Brunessaux, Camelia Constantin, Cédric du Mouza, “A pattern-based approach for an early detection of popular Twitter accounts”, 24th Symposium on International Database Engineering & Applications, 2020.

[4] Stefano Cresci , Marinella Petrocchi, Angelo Spognardi ,Stefano Tognazzi. “Online Social Networks and Media_ On the capability of evolved spambots to evade detection via genetic engineering”, 2019.

[5] Dijana Kosmajac, Vlado Keselj, “Twitter User Profiling: Bot and Gender Identification”, International Conference of the Cross-Language Evaluation Forum for European Languages, 2019.

[6] Feng Wei, Uyen Trang Nguyen, “Twitter Bot Detection Using Bidirectional Long Short-Term Memory Neural Networks and Word Embedding's”, IEEE International Conference on Trust, Privacy and Security, 2019.

[7] D Kosmajac, V Keselj, “Twitter bot detection using diversity measures”,International Conference of the Cross-Language, 2019.

[8] M Petrocchi, S Cresci, R Di Pietro, A Spognardi, M Tesconi, “ Exploiting digital DNA for the analysis of similarities in Twitter behaviors”, 2017.

[9] A Spognardi, Stefano Cresci, R Di Pietro, M Petrocchi, and M Tesconi. “DNA-inspired online behavioral modeling and its application to Spambot detection” IEEE INTELLIGENT SYSTEMS, arXiv:1602.00110v1 Jan 2016.

[10] X. Hu, J. Tang, and H. Liu, “Online social spammer detection,” in AAAI Conference on Artificial Intelligence, 2014, pp. 59–65.