Data and Network Security

(Master Degree in Computer Science and Cybersecurity)

Lecture 12

Outline for today

- Recap last lecture
- Bot detection
- Detection of Spambot Groups

Software defined networking

SDN is an approach to networking that uses software controllers that can be driven by application programming interfaces (APIs) to communicate with the hardware infrastructure to direct network traffic.

Approach to networking that aims:

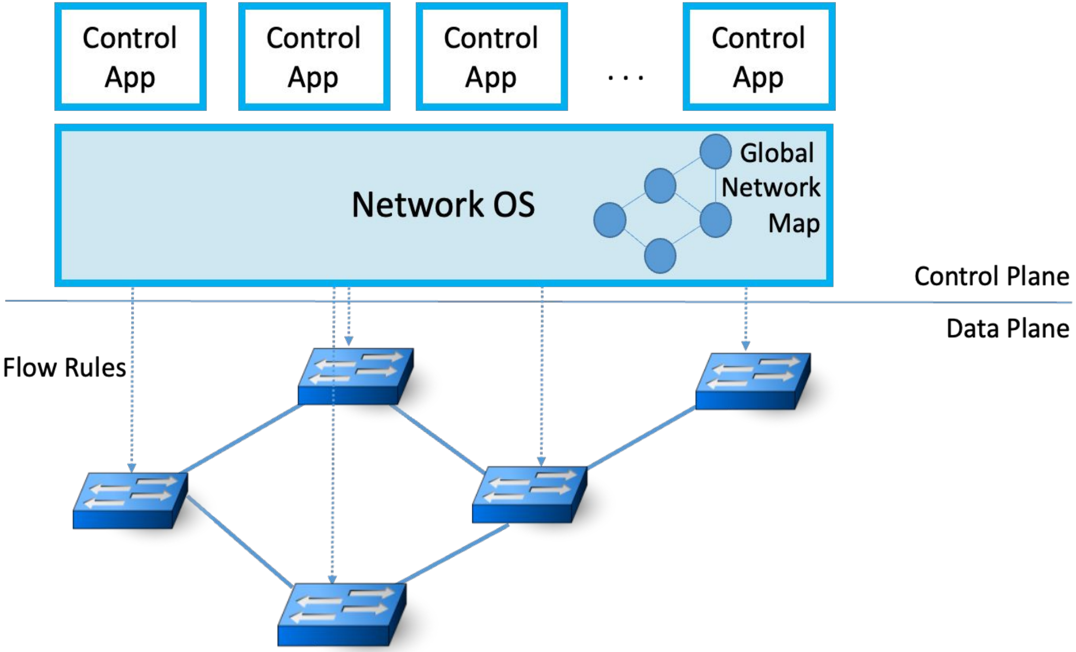
- to make networks more flexible, scalable, and programmable.
- Make the topology independent of physical one

Traditional networking vs. SDN

- In traditional networking, both the control plane (which determines how data packets are forwarded) and the data plane (which handles the actual forwarding of packets) are tightly integrated within network devices.

- SDN is the separation of control and data planes. This separation allows for programmability and automation of network configurations and policies. This enables network administrators to dynamically control and manage network traffic flows according to application requirements and business needs.

SDN



The control plane is centralized in an SDN controller, while the data plane remains distributed across network devices.

Issue

- Allowing for a flexible network management, this on-demand management of network flows also introduces some security threats.

- Extensive communication between the data and control plane can potentially result in a bottleneck for the whole system.

Control plane saturation attack

Installation of rules on the switches is driven by the traffic generated from network users.

- an attacker can exploit this behavior to attack the control plane, by flooding an OpenFlow switch with a large number of **unique** flows.
- for each network flow, the switch will forward a request to the controller, overwhelming it if the rate of new inbound network flows is high enough.

Control plane saturation - the attack

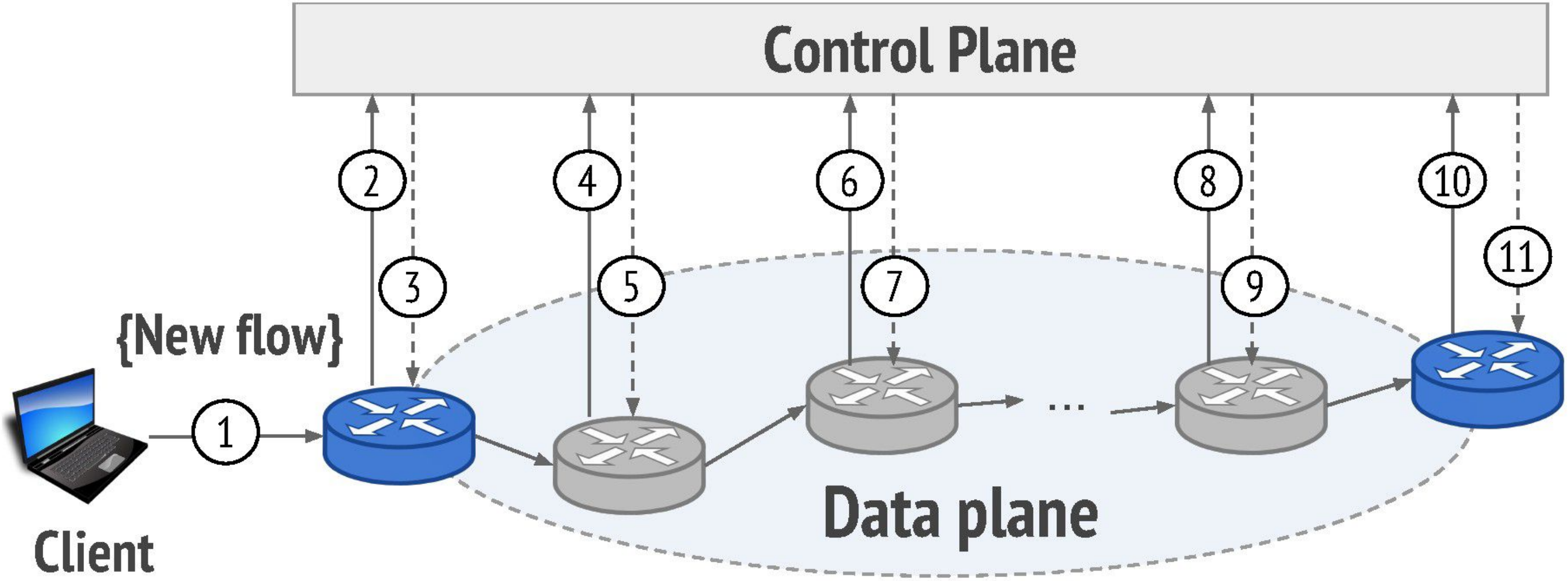
- High number of unique flows:
  - OpenFlow switch will contact the controller to ask for a new flow rule
  - The controller then processes the request, crafts a response containing a series of flow rules, and forwards it to the OpenFlow switch. (performed for each new inbound network flow)
  - Attack Rationale: generate new network flows quickly enough, such that the controller will not be able to keep up with the incoming requests and will be incapacitated from serving other legitimate connections.

Control plane saturation - long paths

Given a new flow -> the controller installs a flow rule only on the OpenFlow switch that performed the flow request.

Control plane saturation - long paths

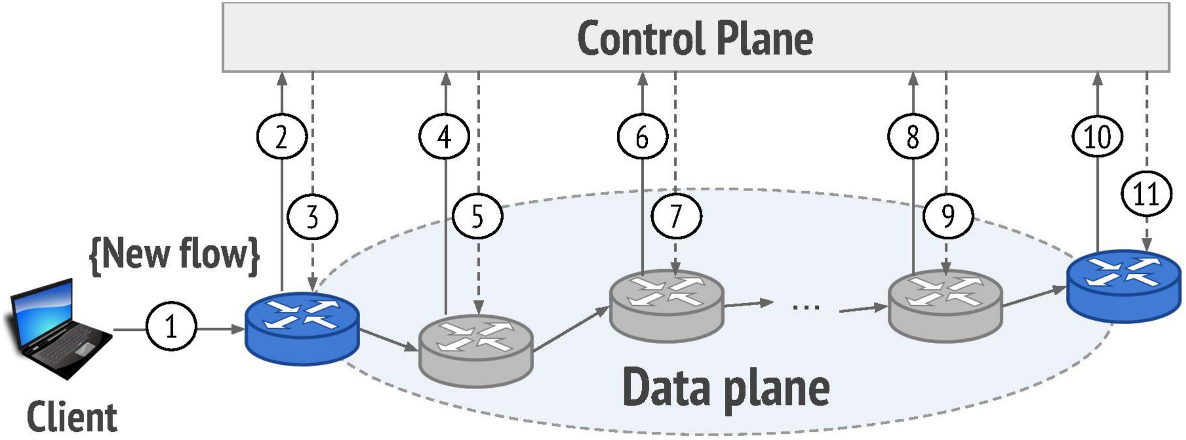
If the SDN subnetwork is big enough, the packet will be routed through other switches of the subnetwork, each of which will send a flow rule request to the controller



Control plane saturation - long paths

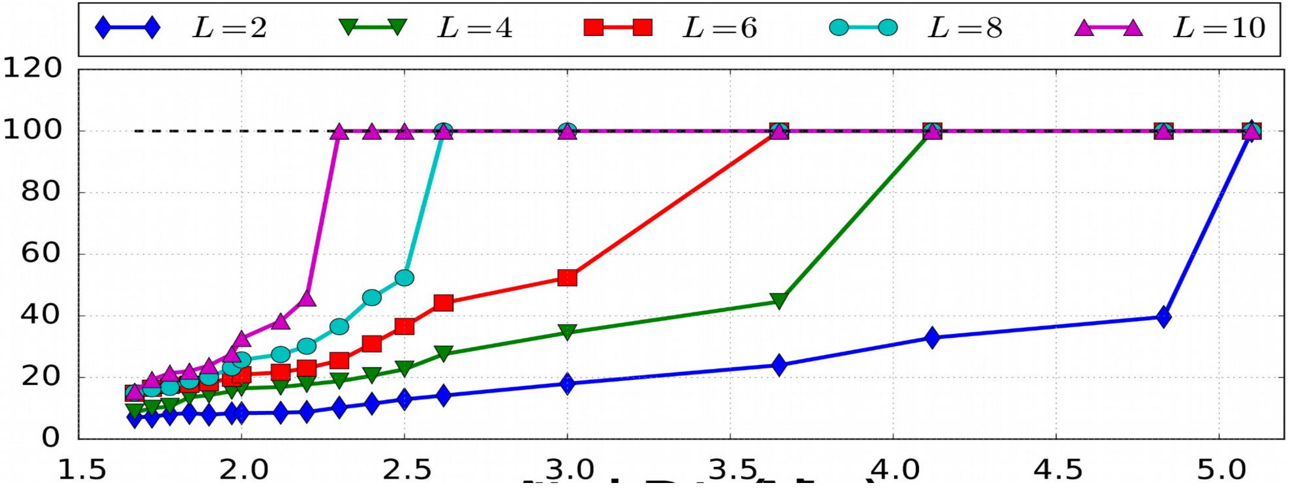
If the SDN subnetwork is big enough, the packet will be routed through other switches of the subnetwork, each of which will send a flow rule request to the controller.

Long paths – amplification of the effect on the control plane

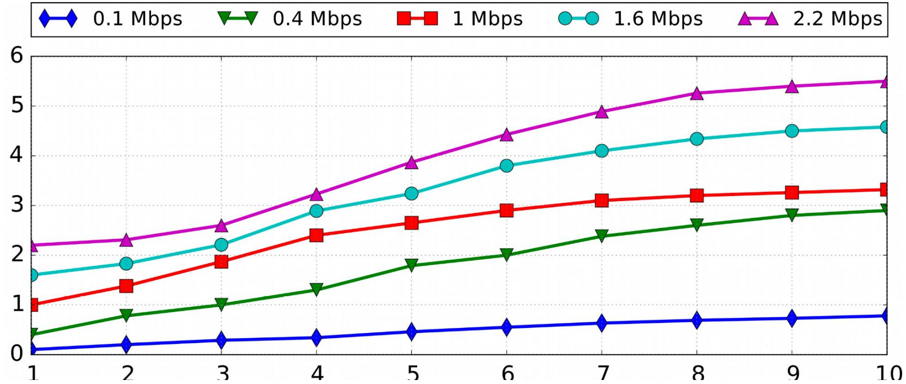


Impact on the client

- Leveraging a path of length L = 4, need around 4 Mbps attack rate to perform a saturation attack
- Using a path of length L = 2 an attacker will only increase the webpage retrieval time to 30 sec.
  - (100 second as controller incapacitated)



Impact on the controller



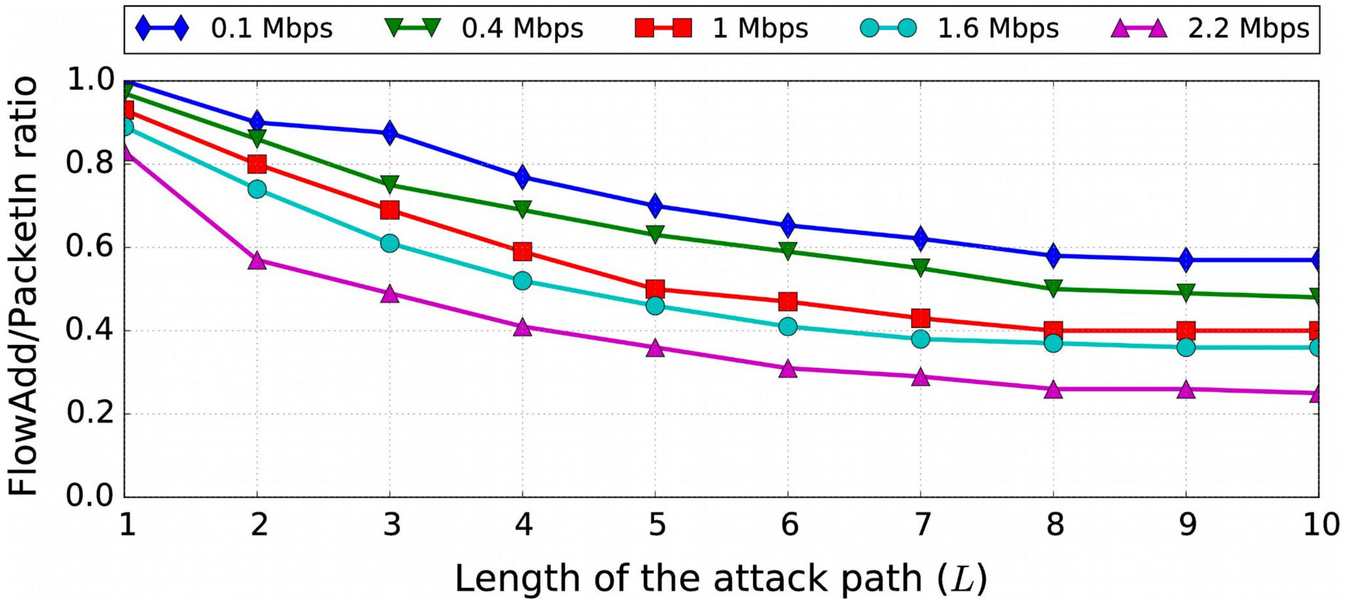
There is a loss of FlowAdd packets attributed to controller capacity saturation and to flow table saturation at each switch.

As the controller approaches the point of saturation, it is not able to keep up with the rate of incoming PacketIn requests anymore.

Large portion of the attack packets are not forwarded to the next hop in the attack path, thus causing a reduced amplification effect on the attack.

Impact on the controller - FlowAdd/PacketIn

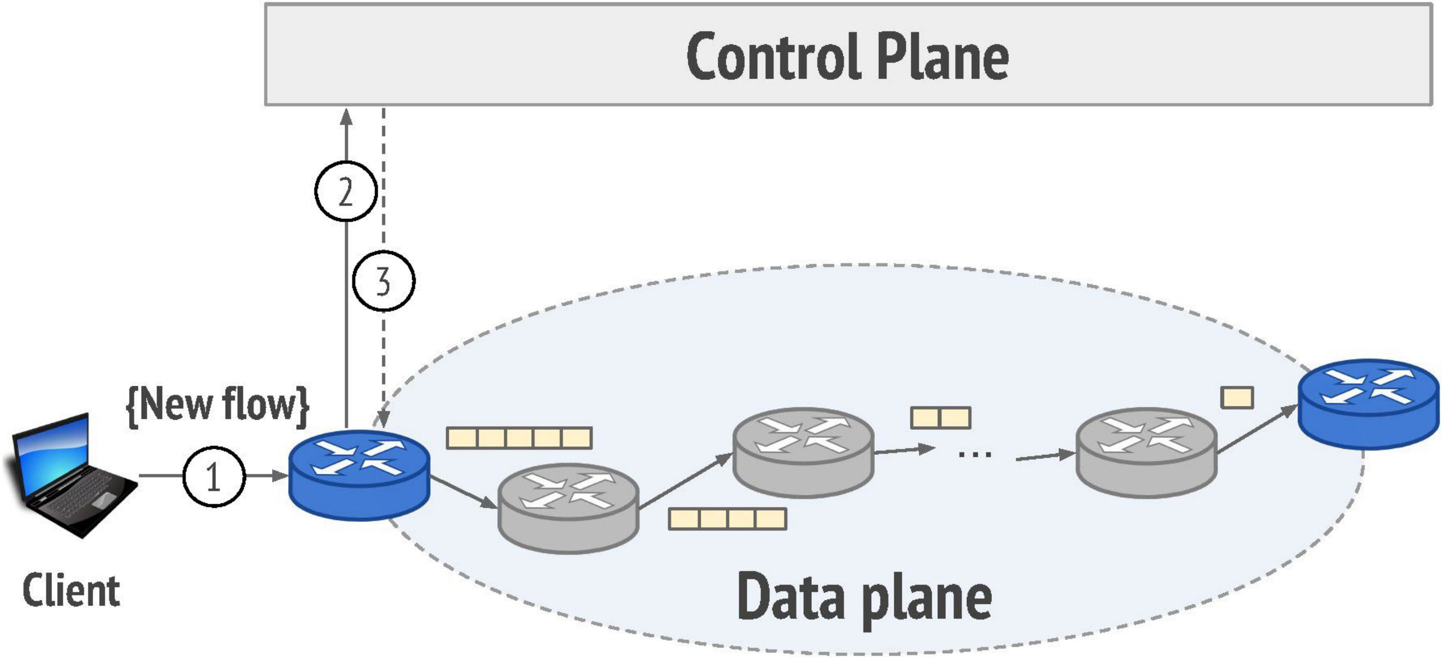
Large portion of the attack packets are not forwarded to the next hop in the attack path, thus causing a reduced amplification effect on the attack.



Possible mitigation strategies

Flow rule piggybacking

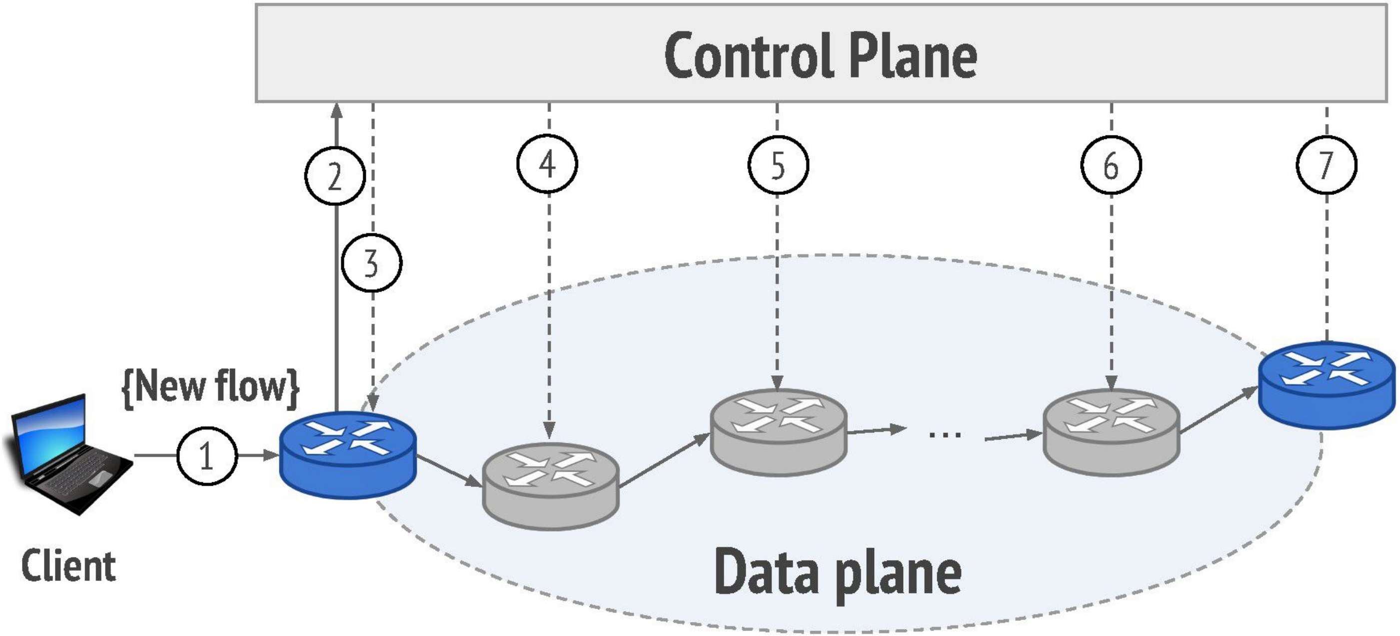
- control plane pushes all rules in one response
- switch install first rule and forwards remaining



Possible mitigation strategies

Flow rule push

- Controller pushes rules on all switches on path after the first request



Outline for today

- Recap last lecture
- Social spambots
- Detection of Spambot Groups

Social Bots - Spam bots

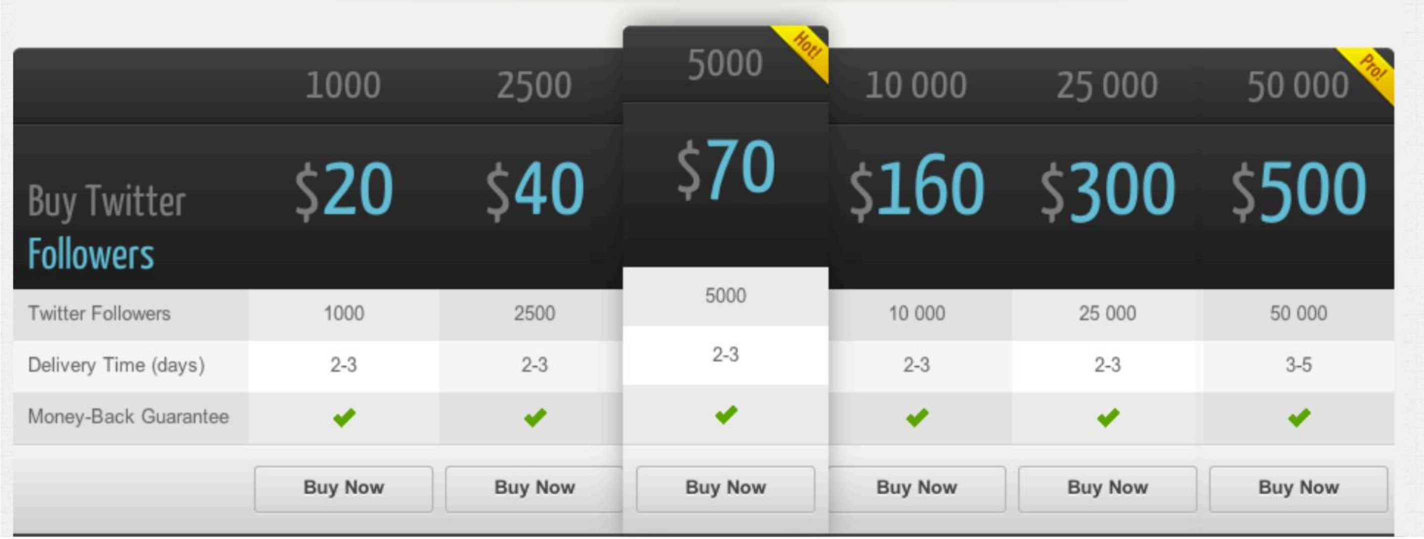
(Semi-)automated accounts with usually harmful intention designed to mimic human behavior on social media platforms.

Usual intentions:

- Misinformation spreading
- Stealing personal information/data
- Stock market manipulation
- Political influence

Social Bots - some history

- Started with fake followers



Buy Twitter Followers	1000	2500	5000 Hot!	10 000	25 000	50 000 Pro!
	$20	$40	$70	$160	$300	$500
Twitter Followers	1000	2500	5000	10 000	25 000	50 000
Delivery Time (days)	2-3	2-3	2-3	2-3	2-3	3-5
Money-Back Guarantee	✔	✔	✔	✔	✔	✔
	Buy Now	Buy Now	Buy Now	Buy Now	Buy Now	Buy Now

Social Bots - current spambots

- Nowadays we have social spambots
  - Aim: Indistinguishable from genuine accounts

They can be:

- follower bots,
- like bots,
- comment bots,
- content-sharing bots
- …

Social Bots - what they usually do/affect?

- Distortion of online discussion
- Manipulation of overall public opinion
- Lowering trust levels
- Misinformation amplification
- Possible economic implications
- Psychological implications

Social Bots - what they usually do/affect?

- Distortion of online discussion

By typically flooding platforms with repetitive or misleading content, spambots can push over the audiences certain viewpoints or narratives, artificially inflating their popularity or significance.

This distortion can make it difficult for users to discern genuine opinions and information from manipulated or fabricated content.

Social Bots - what they usually do/affect?

- Manipulation of overall public opinion

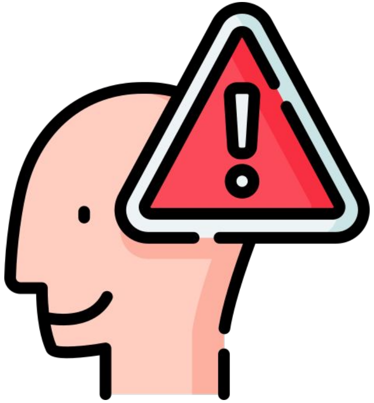
Engaging in coordinated campaigns to promote or discredit certain viewpoints, products, or political candidates this artificially generated content can alter public perceptions and attitudes, potentially impacting real-world outcomes such as elections or consumer behavior.

Social Bots - what they usually do/affect?

- Lowering trust levels

Aware users may become skeptical of the authenticity of social media engagement, leading to decreased trust in platforms and their content.

Fake accounts and automated interactions can tarnish the reputation of legitimate users and organizations, further eroding trust within online communities.



Social Bots - what they usually do/affect?

- Economic implications

Influencing consumer behavior and online market dynamics.
Spambots can inflate metrics such as:

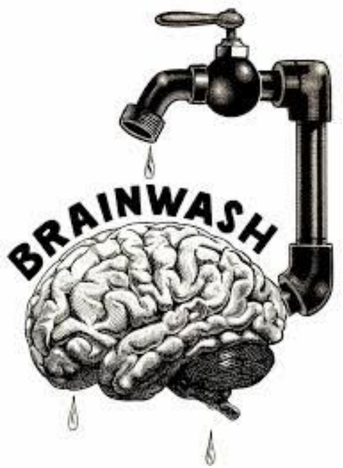
- likes, shares, and followers,

This can mislead advertisers/investors about the true reach and engagement of content of a “celebrity” which down the line with result in financial loss.

Social Bots - what they usually do/affect?

- Psychological implications

Exposure to manipulated or deceptive content may disrupt individuals' confidence/judgement in their ability to discern truth from false.



Outline for today

- Recap last lecture
- Social spambots
- Detection of Spambot Groups

Social Bots - How to detect them?

Social Bots - How to detect them?

Nowadays spambots have become indistinguishable from genuine accounts if analyzed one-by-one so:

How about analyzing the online behavior of large groups of users, with the goal of detecting possible spambots among them?

Behavioral analysis

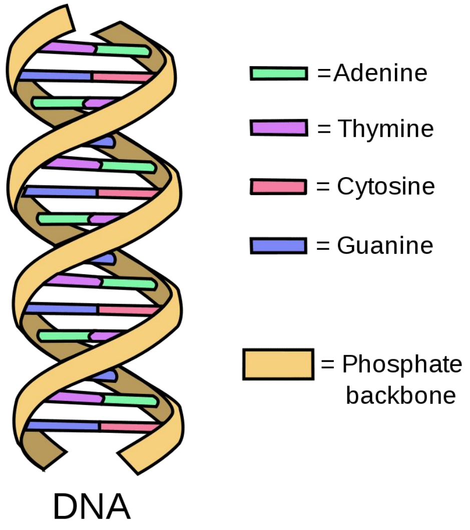
`Behaviour - sequence of actions performed by an account`

Behavioral analysis

Behaviour - sequence of actions performed by an account.

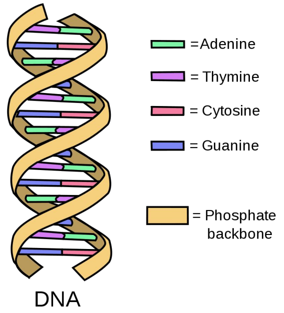
Inspiration from DNA:

- Each type of action is associated to a character (e.g., A, B, C)
- The online behaviour of an account is modeled as a sequence of characters according to the sequence of actions performed by that account



Behavioral analysis

Behaviour - sequence of actions performed by an account.

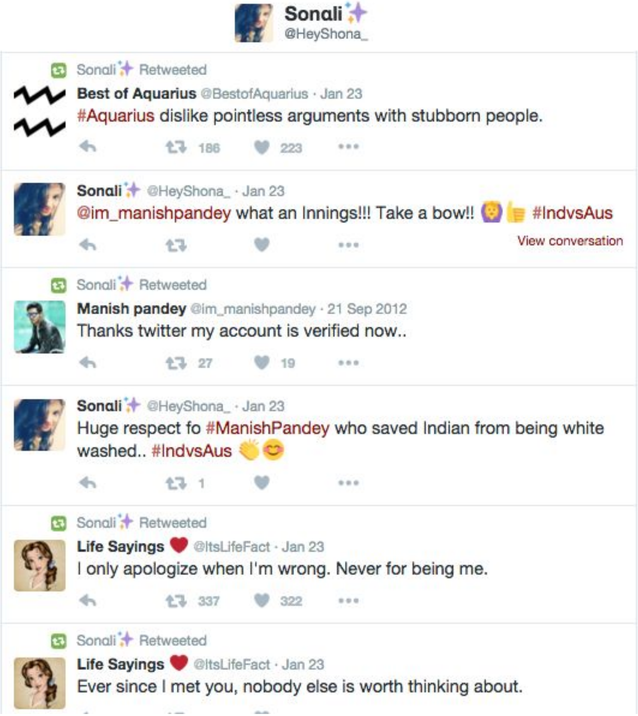


By drawing a parallel with biological DNA, we will see how to model users behaviors and interactions by means of strings of characters, representing the sequence of their actions.

Online actions-such as posting new content, replying to another user, following an account can be encoded with different characters, similarly to DNA sequences.

Digital DNA

Encoding
T - tweet
R - retweet
P - reply



R

P

R

T

R

R

RPRTRR

Digital DNA vs biological DNA

T - tweet
R - retweet
P - reply

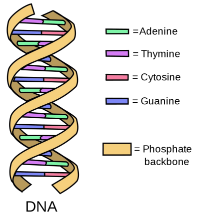
…RRTRPRTPRRPRTPRPTPRRTRPR
…RPRTPTTRPTRPTPRRRRTPPRPP
…TTTRRRPPTPRPTPRTRPTRRRTP
…PRTRPRTPPPPRTPRRPRTPPRRT
…TRTRPRTPRRPRTPRPTPTPPRTT
…TRPPRTPPTRPPTPRRTTTPPRPR

A - adenine
G - guanine
T - thymine
C - cytosine

…AGTCTCCATTTTCAGGTCGTA
…GTTTAAGATCGCCTCATCACC
…AGGCAATTCGCCTGAACTGG
…AGTCTCGATCCTTTCCTCGTT
…AAAATCGAACGCCTTGTCGG
…ATTCTCCATCGCCTAAACAAC

Spambot characterization

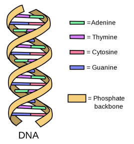
Automated accounts have similar DNA sequences.



The most well-known and widely adopted analysis techniques are **sequence alignment** and **repetition/pattern elicitation.** One of the main goals of these techniques is to find commonalities and repetitions among DNA sequences.

Via an analysis of common sub-sequences and substrings it is possible to predict specific characteristics of the individual and to uncover relationships between different individuals.

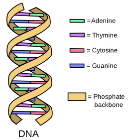
Spambot characterization



To create a digital DNA sequence all the possible actions are represented as a finite set of unique characters:

$$\mathbb{B} = \{B_1, B_2, \ldots, B_N\} \quad B_i \neq B_j \; \forall \; i, j = 1, \ldots, N \; \wedge \; i \neq j.$$

Spambot characterization



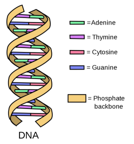
To create a digital DNA sequence all the possible actions are represented as a finite set of unique characters:

$$\mathbb{B} = \{B_1, B_2, \ldots, B_N\} \quad B_i \neq B_j \;\forall\; i, j = 1, \ldots, N \wedge i \neq j.$$

A digital DNA sequence is an ordered tuple, or row vector, of characters (i.e., a string) whose possible values are defined by the bases of its alphabet. A sequence s is defined as:

$$s = (b_1, b_2, \ldots, b_n) \quad b_i \in \mathbb{B} \;\forall\; i = 1, \ldots, n.$$

Spambot characterization

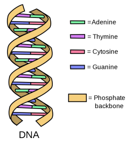


Different alphabets can model different granularities

$$\mathbb{B}^3_{content} = \left\{ \begin{array}{l} \mathtt{N} \longmapsfrom \text{tweet contains no entities (plain text)}, \\ \mathtt{E} \longmapsfrom \text{tweet contains entities of one type}, \\ \mathtt{X} \longmapsfrom \text{tweet contains entities of mixed types} \end{array} \right\}$$
$$= \{\mathtt{N}, \mathtt{E}, \mathtt{X}\}$$

$$\mathbb{B}^6_{content} = \left\{ \begin{array}{ll} \mathtt{N} & \longmapsfrom \text{tweet contains no entities (plain text)}, \\ \mathtt{U} & \longmapsfrom \text{tweet contains one or more URLs}, \\ \mathtt{H} & \longmapsfrom \text{tweet contains one or more hashtags}, \\ \mathtt{M} & \longmapsfrom \text{tweet contains one or more mentions}, \\ \mathtt{D} & \longmapsfrom \text{tweet contains one or more medias}, \\ \mathtt{X} & \longmapsfrom \text{tweet contains entities of mixed types} \end{array} \right\}$$
$$= \{\mathtt{N}, \mathtt{U}, \mathtt{H}, \mathtt{M}, \mathtt{D}, \mathtt{X}\}.$$

Longest common substring

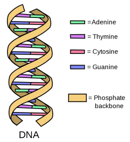


Observe the Longest substring between N sequences of digital DNA

…TRRRPRRTRRPRTPRPTPRRTRPR
…RPRTPTTRRRPRRTPRRRRTPPRP
…TTTRRRPRRRPRRTRTRPTRRRTP
…PRTRPRTPPPPRTPRRRRRPRRTR

Intuitively, users that share long behavioral patterns are much more likely to be similar than those that share little to no behavioral patterns.

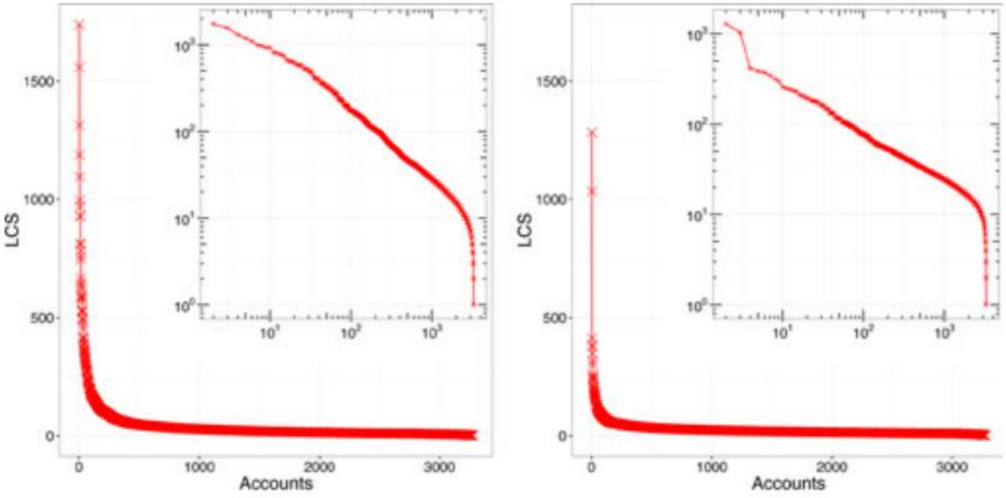
Longest common substring



Goal: find the LCS that is common to at least k of these strings:

$$A = \begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_M \end{pmatrix} = \begin{pmatrix} (b_{1,1}, b_{1,2}, \ldots, b_{1,n}) \\ (b_{2,1}, b_{2,2}, \ldots, b_{2,m}) \\ \vdots \\ (b_{M,1}, b_{M,2}, \ldots, b_{M,p}) \end{pmatrix}$$

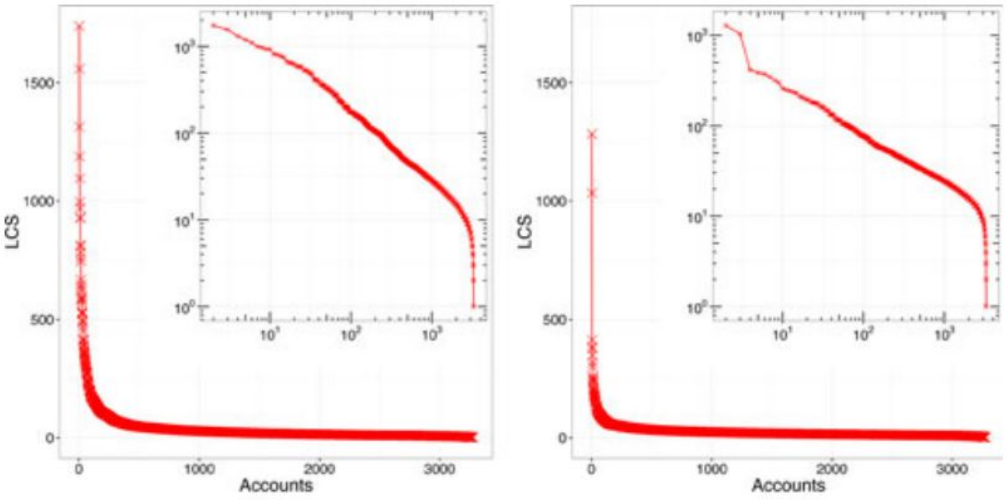
Longest common substring



(a) $\mathbb{B}^3_{type}$ alphabet. (b) $\mathbb{B}^3_{content}$ alphabet.

- x axis - the number of k accounts (corresponding to the k strings, used to compute LCS values)
- y axis - the length of the LCS common to at least k accounts.

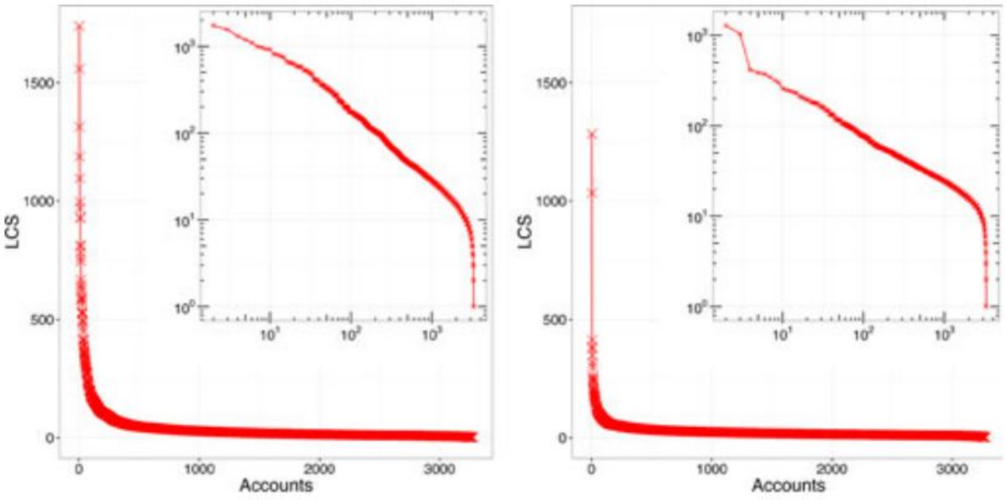
Longest common substring



(a) $\mathbb{B}^3_{type}$ alphabet.   (b) $\mathbb{B}^3_{content}$ alphabet.

Each point in an LCS curve corresponds to a subset of k accounts that share the longest substring (of length y) among all those shared between all the other possible subsets of k accounts.

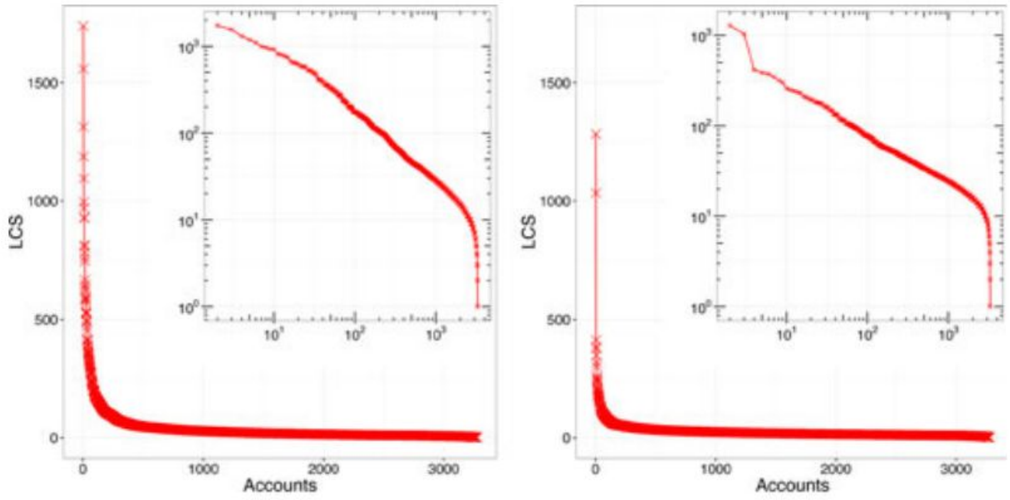
Longest common substring



(a) $\mathbb{B}^3_{type}$ alphabet. (b) $\mathbb{B}^3_{content}$ alphabet.

As the number k of accounts grows,
the length of the LCS common to all of them shortens.
LCS curves are monotonic non-increasing functions.

Longest common substring



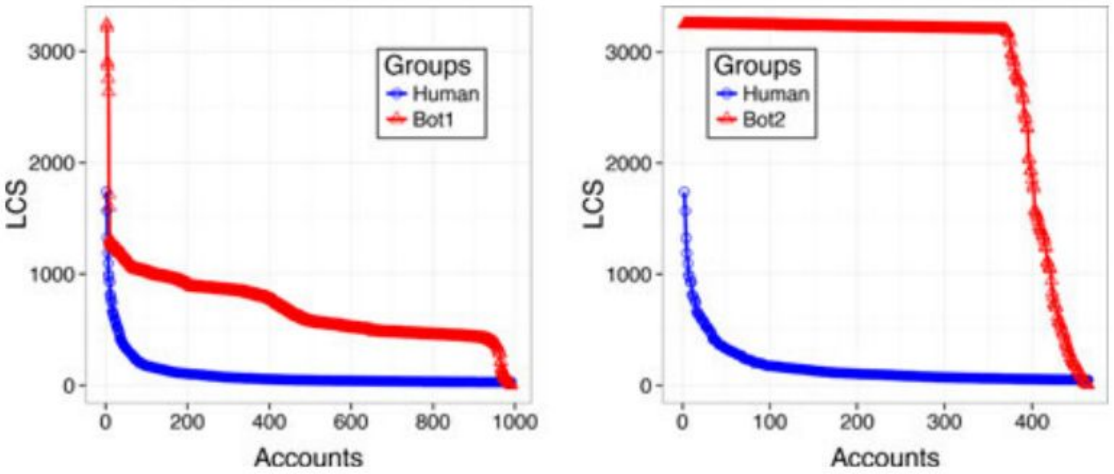
(a) $\mathbb{B}^3_{type}$ alphabet.

(b) $\mathbb{B}^3_{content}$ alphabet.

Thus, it is more likely to find a long LCS among a few accounts rather than among large groups.

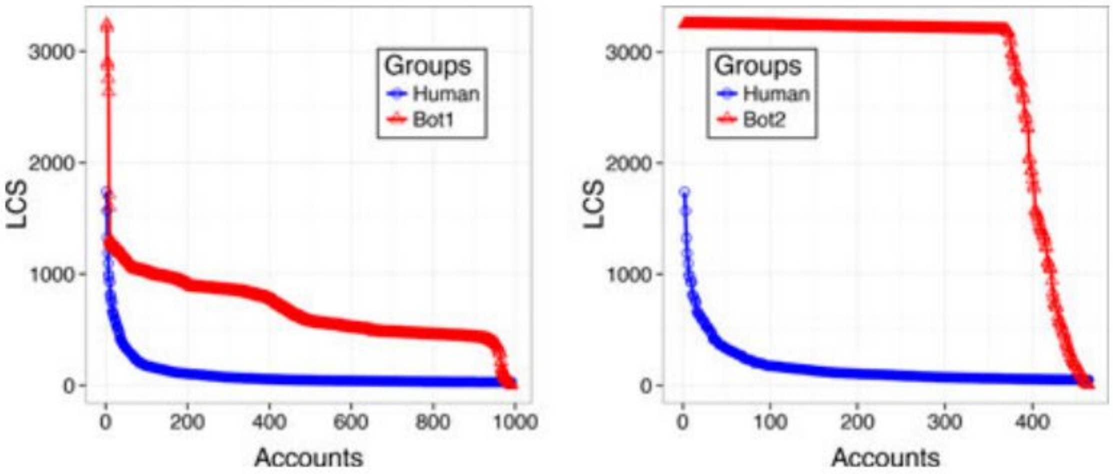
LCS curves are monotonic non-increasing functions.

Bots versus Humans



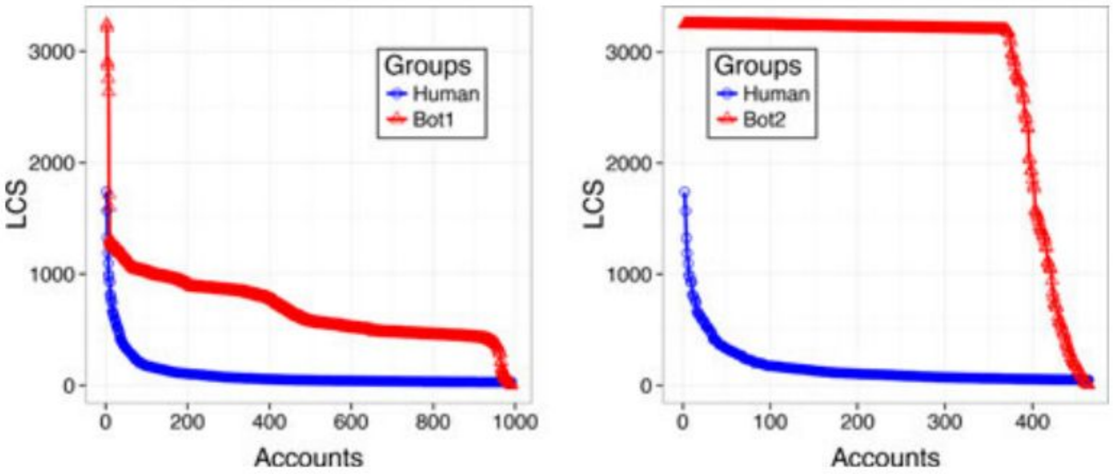
Observation 1: LCS of both groups of spambots are rather long even when the number of accounts grows.

Bots versus Humans



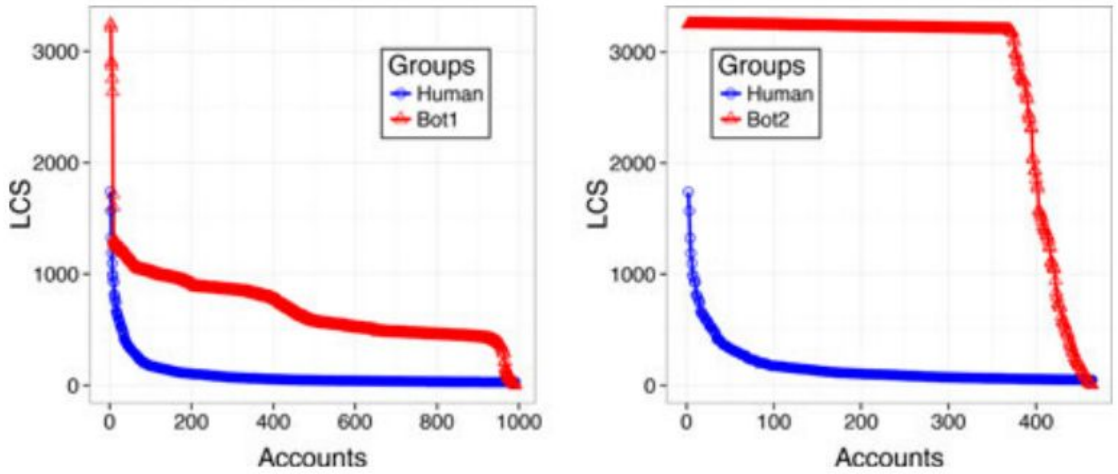
Observation 2: Sudden drop of the LCS length when the number of accounts gets close to the group size for spambots

Bots versus Humans



Observation 3: Genuine accounts show little to no similarity-as represented by LCS curves that exponentially decay, rapidly reaching the smallest values of LCS length.

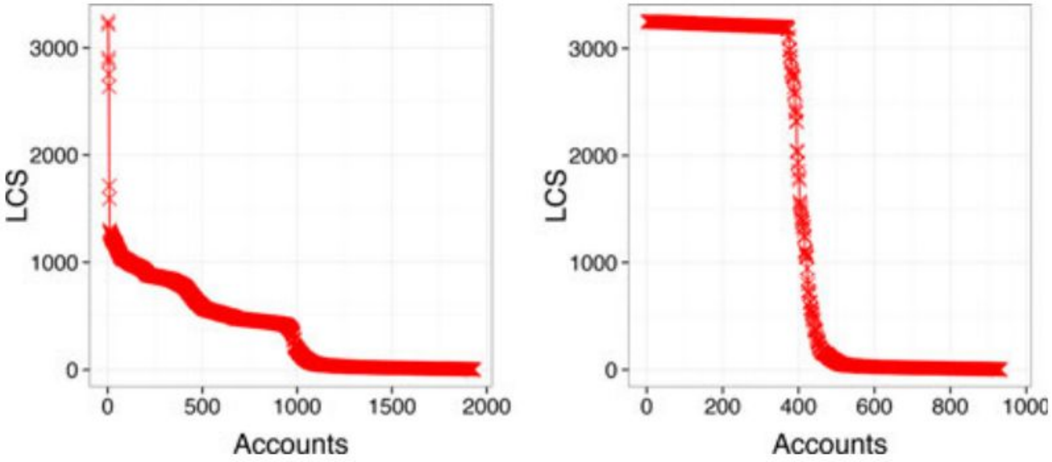
Bots versus Humans - Automation in distinguishing?



Consider high behavioral similarity as a proxy for automation and, thus, a high level of similarity among a large group of accounts might serve as an indicator for anomalous behaviors.

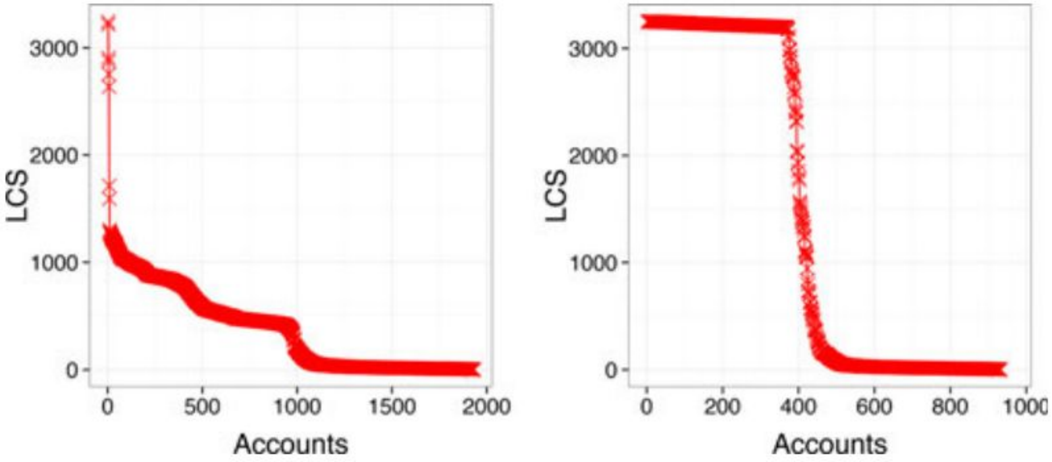
Heterogeneous Users - LCS

Mix various bot families and same quantity of normal users data.



Heterogeneous Users - LCS

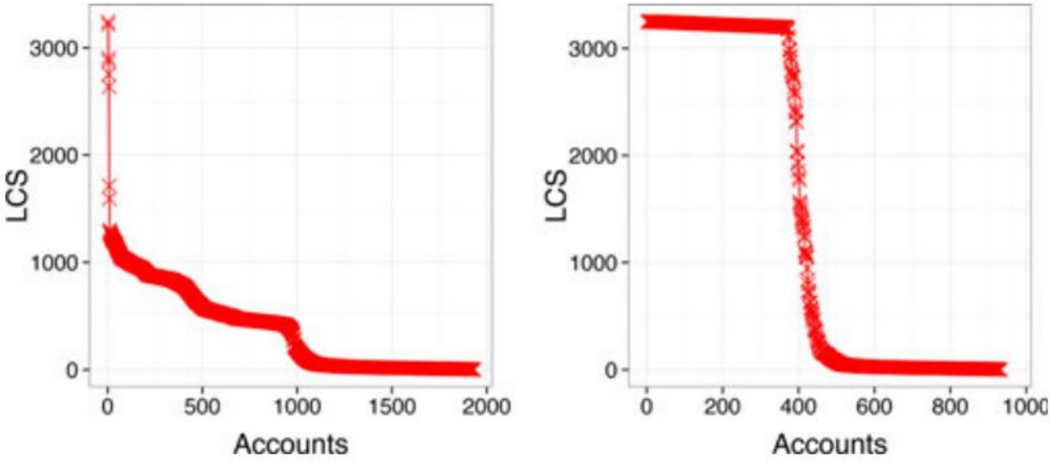
Mix various bot families and same quantity of normal users data.



LCS curves in both plots asymptotically reach their minimum value as the number of accounts grows.

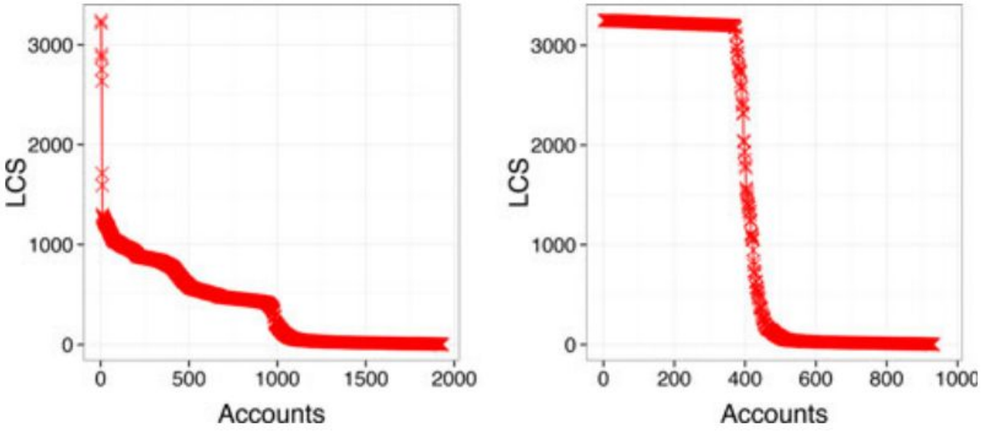
Heterogeneous Users - LCS

Mix various bot families and same quantity of normal users data.



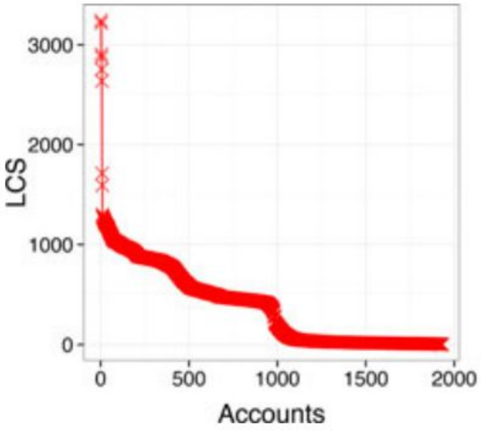
We have a different behaviour compared to the individual groups

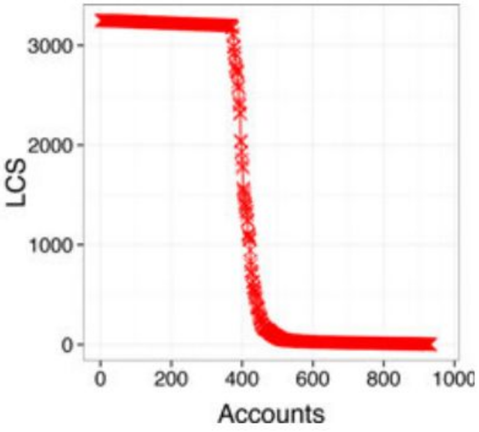
Heterogeneous Users - LCS



Observation: A trend seems to be dominant only until reaching a certain threshold. Then, a steep fall occurs and another possibly different-trend kicks in.

Heterogeneous Users - LCS

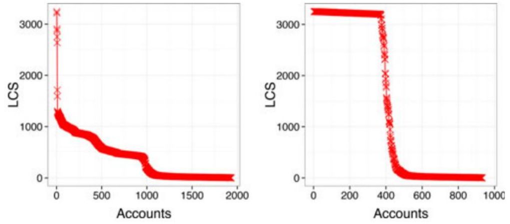




Observation: A trend seems to be dominant only until reaching a certain threshold. Then, a steep fall occurs and another possibly different-trend kicks in.

Observation: These portions of the LCS curves separated by the steep drops resemble LCS curves of the single groups of similar users

Heterogeneous Users - LCS



The steep drops of LCS curves separate areas where the length of the LCS remains practically unchanged, even for significantly different numbers of considered accounts.

These plateaux in LCS curves are strictly related to homogeneous groups of highly similar accounts. (multiple plateaux - multiple sub groups existing).

The steeper the drop in a LCS curve, the more different are the two subgroups of accounts split by that drop.

Slow and gradual decreases in LCS curves represent areas of uncertainty, where it might be difficult to make strong hypotheses about the characteristics of the underlying accounts.

Time for detection

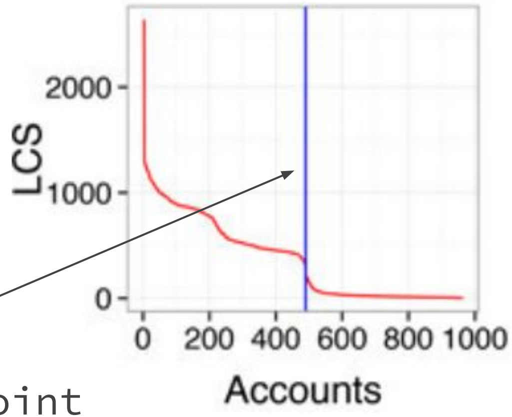
Supervised approach - case 1

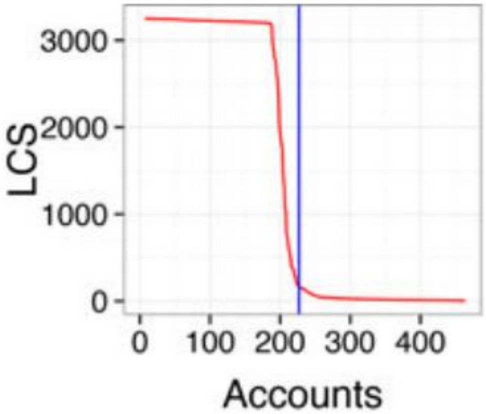
A good division of the original set of users into several subgroups is one where all the users belonging to a given class are assigned to the same subgroup.

LCS curve of a heterogeneous group of users can be used as a splitting point to obtain two subgroups of more homogeneous users.

Supervised approach - case 1 - evaluation

Using a labeled dataset check all possible splitting point in the LCS curve of the training-set users and find the one that yields the best possible subgroup division. (Every point generates a different “classifier”)



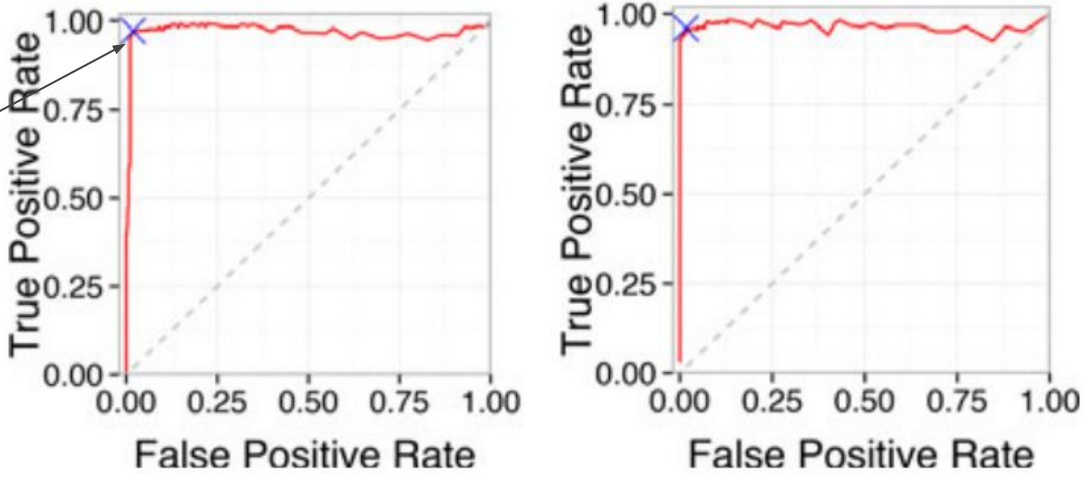


Best splitting point
Left - spambots
Right - genuine users

Supervised approach - case 1 - evaluation

Supervised - Cannot guarantee that the learned LCS value would still be effective when applied on a test-set different from the one used to derive the LCS value.

Best model



Unsupervised approach

Exploit the discrete derivative of a LCS curve to recognize the points corresponding to the steep drops.

Unsupervised approach

Exploit the discrete derivative of a LCS curve to recognize the points corresponding to the steep drops.

The steep drops of LCS curves appear as sharp peaks in the derivative plot

- represent suitable splitting points to isolate different subgroups among the whole set of users.

Unsupervised approach

Exploit the discrete derivative of a LCS curve to recognize the points corresponding to the steep drops.

The steep drops of LCS curves appear as sharp peaks in the derivative plot

- represent suitable splitting points to isolate different subgroups among the whole set of users.

Rank the suitable points according to their corresponding derivative value

- how steep is the corresponding drop

Unsupervised approach

Exploit the discrete derivative of a LCS curve to recognize the points corresponding to the steep drops.

The steep drops of LCS curves appear as sharp peaks in the derivative plot

- represent suitable splitting points to isolate different subgroups among the whole set of users.

Rank the suitable points according to their corresponding derivative value

- how steep is the corresponding drop

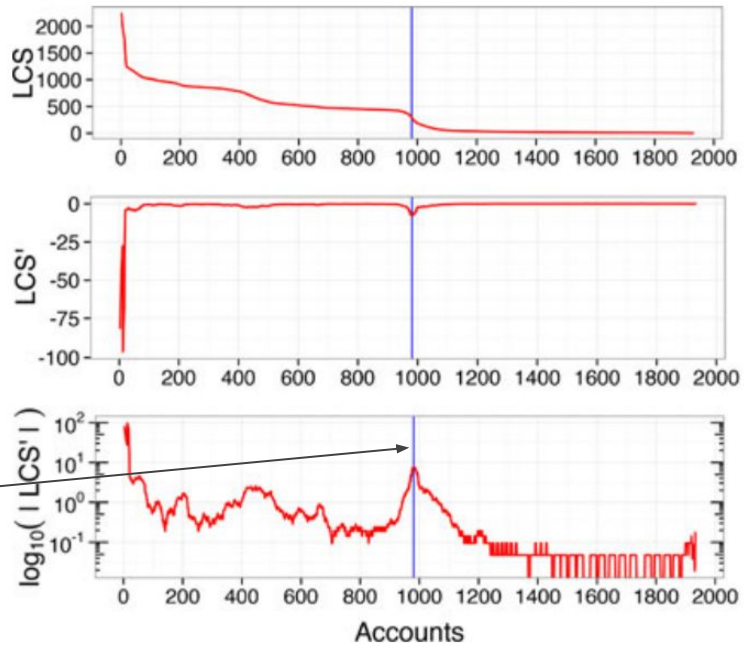
Repeatedly divide the whole set of users based on the ranked points, leading to a dendrogram structure.

- Useful when the LCS curve exhibits multiple plateaux and steep drops, (find best possible clusters)

Unsupervised approach

Exploit the discrete derivative of a LCS curve to recognize the points corresponding to the steep drops.

Most pronounced peak



Reading Material

1. Social Spambot detection: Link-1, Link-2, Link-3