Appendix A – Benford's Law (BL)

First significant leading digit (FSLD)

Let D be a positive real number on a probability space (Ω, F, ℙ). The logarithmic density function for the first leading digit is expressed as follows:

$$\mathbb{P}(\boldsymbol{D} = \boldsymbol{d}) = \boldsymbol{log_{10}}\left(\boldsymbol{1} + \frac{\boldsymbol{1}}{\boldsymbol{d}}\right) \quad (1)$$

where d ∈ {1, 2, …, 9}.

We briefly discuss the mathematical proof of BL theory using the fundamental theorem of equivalence [1]. Any positive real number $x \in \mathbb{R}^{+}$ can be written as scientific notation $x = S(x) * 10^{k}$. If we are interested in the leading digit of x, the value of the integer k is irrelevant. Therefore, instead of studying the raw value of x, we can apply the transformation: $x \rightarrow \log_{10} x \ mod\ 1$. Two positive real numbers i and j have the same first leading digits if and only if their significands $S(i)$ and $S(j)$ have the same first leading digits [1]. Thus, a set of real numbers $\{x_1, x_2, \ldots\}$ have subsets of leading digits d written as $\{x_n : S(x_n) \ \in [d, d+1)\}$, taking logarithms on this gives $\{x_n : \log_{10} S(x_n) \ \in [\log_{10} d, \log_{10}(d+1))\}$. Specifically, the probability that a value is in the interval $[\log_{10} d, \log_{10}(d+1))$ is just the length of this interval, which is $\log_{10}(d+1) - \log_{10} d$, and this is just BL in Equation (1) [1].

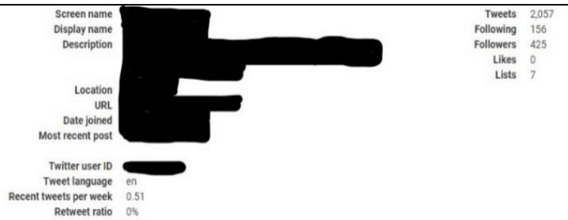
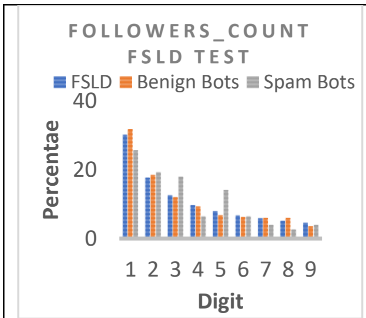
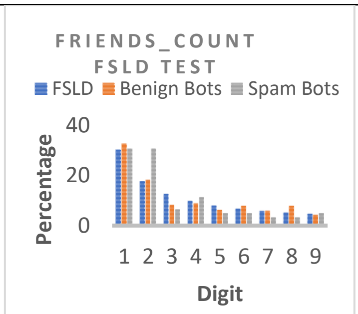
Benign bot example features	**Malicious bot example features**
	

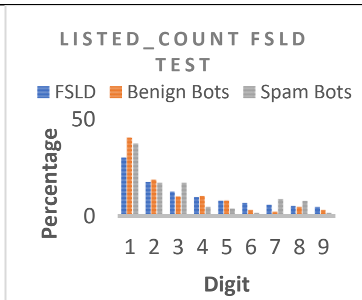
Table 16. Examples of Twitter benign and malicious bot features

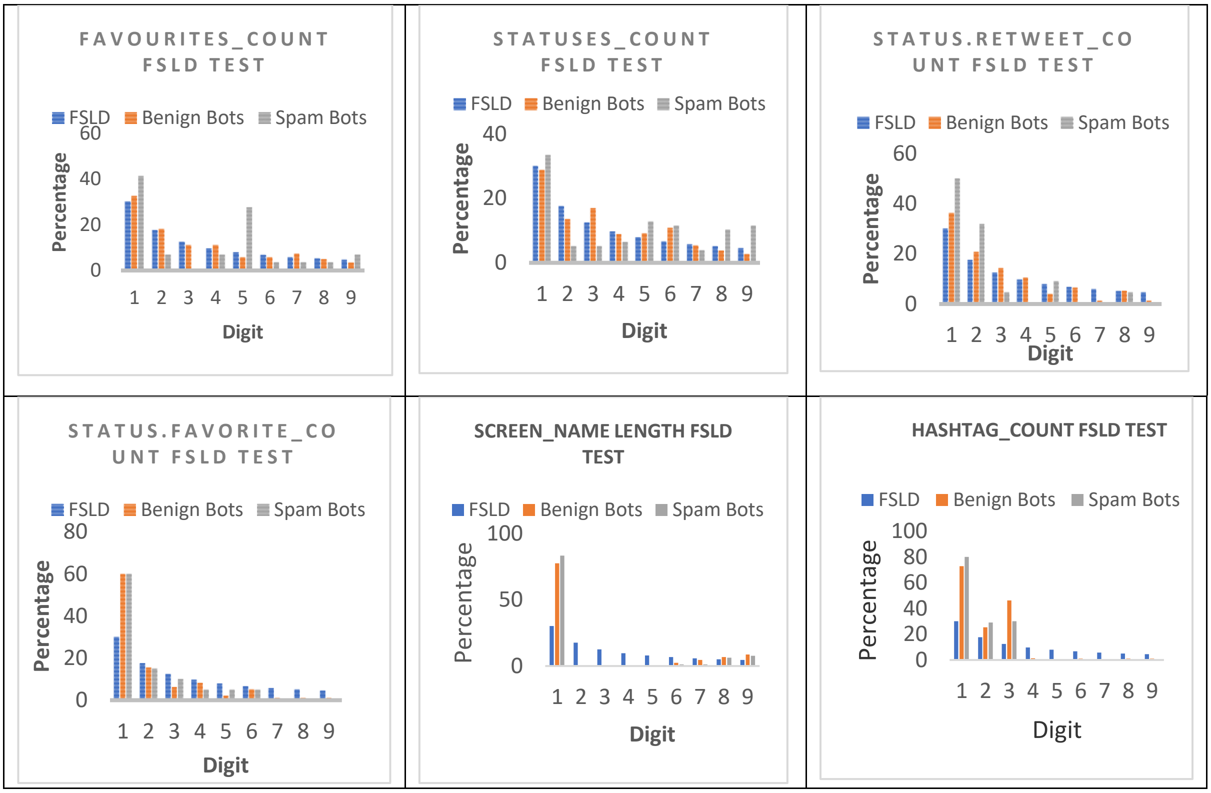
Table 16 depicts examples of benign and malicious bot features. The images are blurred to protect users' identities. Malicious bots appear to be tweeting spam bots as they have a high number of tweets and retweet ratio.

Appendix B – A sample of BL FSLD test









Appendix C – Semi-supervised Gaussian mixture model (GMM)

1. Given a set of features $\{x_j^1, x_j^2, \dots, x_j^n\}$ indicative of anomalous behavior,
   Apply the transformation $x \rightarrow \log(x)$
2. Fit the parameters of the model
   $\mu = \frac{1}{n}\sum_{i=1}^{n} x_j^i$ and $\Sigma = \frac{1}{n}\sum_{i=1}^{n}(x_j^i - \mu)\,(x_j^i - \mu)^T$
3. Given a test example, x_{test}, compute
   $$\mathbb{P}(x_{test}) = \frac{1}{(2\pi)^{\frac{n}{2}}\,|\Sigma|^{\frac{1}{2}}}\exp(-\frac{1}{2}(x^i - \mu)^T\Sigma^{-1}(x^i - \mu))$$
4. Prediction
   $$Predict = \begin{cases} malicious\ bot \text{ if } \mathbb{P}(x_{test}) < \mathcal{E} \\ benign\ bot \text{ otherwise} \end{cases}$$

where n is the number of samples and $\sum$ is the n-dimensional covariance matrix, the threshold $\mathcal{E}$ is determined automatically on the CV dataset. The main advantage of using GMM is that it captures correlations of input features naturally and hence can detect different "types" of anomalies. For example, we did not need to define a feature such as follower–friend ratio, as the correlation term naturally accounted for this. The major disadvantage is calculating $\sum^{-\mathbf{1}}$, which might be computationally expensive [2]. The results for the binary classification problem using GMM are indicated below. The ScikitLearn code is found in [3]. The results in Table 8 were achieved on the basis of $\mathcal{E} = 2.94e^{-05}$

Appendix D – Semi-supervised support vector machine (S3VM)

1. Let (x_i, y_i) denote the sample datasets given a set of l labeled samples and u unlabeled samples. Further, let a hyperplane be denoted as $(w.x) + b = 0$, we optimized the following objective function over w, b, η, ξ, and z

2. $\min\limits_{w,b,\eta,\xi,z} [\|w\| + C(\sum_{i=1}^{l} \eta_i + \sum_{j=l+1}^{l+u} \min(\xi_j, z_j))]$

3. $$\begin{cases} \boldsymbol{y_i(w^T.x_i + b) + \eta_i \geq 1 \; and \; \eta_i \geq 0, \forall \; i = 1,..,l} \\ \boldsymbol{(w^T.x_j - b) + \xi_j \geq 1 \; and \; \xi_j \geq 0, \forall \; j = l+1,..,l+u} \\ \boldsymbol{-(w^T.x_j - b) + z_j \geq 1 \; and \; z_j \geq 0, \forall \; j = l+1,..,l+u} \end{cases}$$

where $C = 1.0\ (> 0)$ is a fixed misclassification penalty. The first term of the S3VM minimization problem is the standard SVM. The second term was divided into two parts: (i) we added l slack variables η_i to ensure a maximum margin for labeled samples and (ii) accounted for unlabeled samples that could be classified as either +1 (benign bot) or −1 (malicious bot) by adding slack variables ξ_j and z_j. The ScikitLearn code can be found in [4].

Appendix E – Semi-supervised label propagation

Given a dataset with N labeled and M unlabeled data points, i.e.,

$$\begin{cases} X = (X_L, X_U) \text{ where } X \,\mathcal{E}\, \mathbb{R} \\ Y = (Y_L, Y_U) \text{ where } Y \in \{-1, +1, 0\} \end{cases}$$

1. Using the k-nearest neighbors (kNN), compute the affinity matrix W.
2. Compute the degree matrix

$D = diag(\left|\sum_j W_{ij}\right| \; \forall \; i = 1,2,..,N+M)$

3. Let $Y^{(0)} = Y$
4. Define $Y_L = \{y_0, y_1, \dots, y_N\}$
5. Iterate until convergence is achieved

$$\begin{cases} Y^{(t+1)} = D^{-1} W Y^{(t)} \\ Y_L^{(t+1)} = Y_L \end{cases}$$

Proof of convergence and ScikitLearn code can be found in [4].

Appendix F – Semi-supervised label spreading

1. Using the kNN method, compute the affinity matrix W.
2. Compute the degree matrix
   $D = diag(\left|\sum_j W_{ij}\right| \; \forall \; i = 1,2,..,N+M)$
3. Compute the normalized graph Laplacian:
   $L = D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$
4. Choose $\alpha \,\mathcal{E}\, (0,1]$
5. Iterate until convergence is achieved
   $Y^{(t+1)} = \alpha \, L \, Y^{(t)} + (1-\alpha) Y^{(0)}$

Proof of convergence and ScikitLearn code can be found in [4].

Appendix G – Mann–Whitney U test

Mann–Whitney U test

Given two samples of sizes n_1 and n_2 with rankings R_1 and R_2, respectively, compute

$$U_1 = n_1 n_2 + \frac{n_1(n_1+1)}{2} - R_1$$
$$U_2 = n_1 n_2 + \frac{n_2(n_2+1)}{2} - R_2$$
$$U = \min(U_1, U_2)$$

H_0 = no difference between two distributions; H_1 = difference between two distributions; reject H_0 if $p < 0.05$.

Feature	**Yang et al.** [5] **vs. combined**	**Mazza et al.** [6] **vs. combined**	**Yang et al.** [7] **vs. combined**
Favorites_count	Cannot reject H_0	Cannot reject H_0	Cannot reject H_0
Lists_count	Cannot reject H_0	Cannot reject H_0	Cannot reject H_0
Statuses_count	Cannot reject H_0	Cannot reject H_0	Cannot reject H_0
Status.retweet_count	Cannot reject H_0	Cannot reject H_0	Cannot reject H_0
Friends_count	Cannot reject H_0	Cannot reject H_0	Cannot reject H_0
Followers_count	Cannot reject H_0	Cannot reject H_0	Cannot reject H_0

Table 17. Statistical test results for the Mann–Whitney U test

References

1. Miller SJ (2015) Benford's Law: Theory and Applications. Princet Univ Press
2. Ding N, Ma HX, Gao H, et al (2019) Real-time anomaly detection based on long short-Term memory and Gaussian Mixture Model. Comput Electr Eng 79:. https://doi.org/10.1016/j.compeleceng.2019.106458
3. Amr T (2019) Hands-On Machine Learning with scikit-learn and Scientific Python Toolkits. OReilly Media 384
4. Bonaccorso G (2020) Mastering Machine Learning Algorithms: Expert techniques for implementing popular machine learning algorithms, fine-tuning your models, and understanding how they work. OReilly Media
5. Yang KC, Varol O, Hui PM, Menczer F (2020) Scalable and generalizable social bot detection through data selection. AAAI 2020 - 34th AAAI Conf Artif Intell 1096–1103. https://doi.org/10.1609/aaai.v34i01.5460
6. Mazza M, Cresci S, Avvenuti M, et al (2019) RTbust: Exploiting temporal patterns for botnet detection on twitter. WebSci 2019 - Proc 11th ACM Conf Web Sci 183–192. https://doi.org/10.1145/3292522.3326015
7. Yang KC, Varol O, Davis CA, et al (2019) Arming the public with artificial intelligence to counter social bots. Hum Behav Emerg Technol 1:48–61. https://doi.org/10.1002/hbe2.115