2016 15th IEEE International Conference on Machine Learning and Applications

Uncovering the Landscape of Fraud and Spam in the Telephony Channel

Aude Marzuoli, Hassan A. Kingravi, David Dewey, Robert Pienta
Pindrop
Atlanta, GA, USA
Email: amarzuoli, hkingravi, ddewey, rpienta@pindrop.com

***Abstract*—Robocalling, voice phishing, and caller ID spoofing are common cybercrime techniques used to launch scam campaigns through the telephony channel, which unsuspecting users have long trusted. More reliable than online complaints, a telephony honeypot provides complete, accurate and timely information about unwanted phone calls across the United States. Our first goal is to provide a large-scale data-driven analysis of the telephony spam and fraud ecosystem. Our second goal is to uniquely identify bad actors potentially operating several phone numbers. We collected about 40,000 unsolicited calls. Our results show that only a few bad actors, robocallers or telemarketers, are responsible for the majority of the spam and scam calls, and that they can be uniquely identified based on audio features from their calls. This discovery has major implications for law enforcement and businesses that are presently engaged in combatting the rise of telephony fraud. In particular, since our system allows end-users to detect fraudulent behavior and tie it back to existing fraud and spam campaigns, it can be used as the first step towards designing and deploying intelligent defense strategies.**

I. Introduction

Telephony spam and fraud is a major concern because the telephony channel is less secure than most other communication channels. While email spam has led to a multi-billion dollar anti-spam industry [1], phone spam and fraud are less understood. Little is known about the telephony spam and fraud ecosystem, its tactics and value chain. Attacks on the telephony channel have recently increased, and this trend can be attributed to the availability of Voice over Internet Protocol (VoIP). Automated VoIP calls can be made at no or low cost at scale, from the United States or overseas. Caller ID spoofing occurs when a caller deliberately falsifies the information transmitted to a Caller ID display to disguise their identity. Spoofing and robodialing are easily accessible techniques. A robocall is defined as a phone call that uses a computerized autodialer to deliver a pre-recorded message. Cybercriminals are already exploiting the telephony channel to craft large-scale attacks such as voice phishing (vishing) [2]. An attacker can configure a VoIP software [3] to dial a group of phone numbers and play prerecorded outgoing calls, also allowing the recordings to be emailed upon completion. People have long trusted the telephony channel, making attacks relying on the telephone as a resource more successful [2]. The majority of the population can be more easily reached via phone than any other communication means. Telephony has become the weak link even for web security. For instance, cybercriminals regularly exploit social engineering over the phone to reset online banking credentials to steal money.

Millions of complaints from citizens have been sent to the Federal Trade Commission (FTC) about unwanted and fraudulent calls. Websites such as 800notes [4] receive thousands of online complaints about unwanted calls daily. Large-scale phone scams now regularly make news headlines and are widely reported to the FTC. The most famous ones include: scam 419, where the victims are convinced to send cash upfront by promising them a large amount of money that they would receive later if they cooperate [5]; the tech support scam, where consumers were tricked into paying for the removal of bogus viruses on their computers and giving the scammers remote access to their computers [6]; swindlers who overload emergency dispatch centers with automated calls [7].

Research regarding telecommunications fraud has long focused on exploiting very large data sets of all users, both genuine and fraudulent, of the telephony channel. Becker et al. [8] provided an overview of fraud detection at one of the top U.S. carriers. Weatherford [9] used neural networks to create long-term patterns of user behavior. Onderwater et al. [10] exploited outlier detection techniques on user profiles to identify fraudsters. Tseng et al. [11] studied unusual traffic patterns of millions of users, extracting features from patterns. The main drawback of these techniques is that they require access to millions of records of phone calls. Such records pose big imbalance problems since most phone calls placed by users are not fraudulent. Our proposed approach with a telephony honeypot is much simpler, avoids privacy concerns when providing all records of phone calls for millions of users, and has significantly less imbalance problems.

In previous research [12], the information obtained on a call to a telephony honeypot included the source phone number (which may not be a legitimate or a valid number because of spoofing), the destination phone number, and the time of call. However, it did not provide insight on the telephony fraud ecosystem. No one knows a priori whether there are thousands or just a few bad actors perpetrating attacks on the telephony channel. Contrary to other communication channels, very little information on a phone call is available besides the source phone number, the time of the call and sometimes its duration. Spammers and fraudsters using emails or social network platforms [13] usually leave a link to a website and semantic information is readily available in the corresponding

978-1-5090-6167-9/16 $31.00 © 2016 IEEE
DOI 10.1109/ICMLA.2016.195

email, tweet or SMS, including keywords. The sheer volume of unwanted calls combined with the fact that source phone numbers cannot be trusted (because of spoofing) makes the identification of telephony fraud challenging.

This paper establishes data-driven ground truth regarding the landscape of telephony spam and fraud, and gather threat intelligence in order to later design detection and defense mechanisms. The honeypot can be modeled as a bipartite graph, with source phone numbers as start nodes on one side, destination phone numbers as end nodes on the other side, and edges representing phone calls. We seek to uncover the hidden network between source phone numbers, reflecting the fact that one bad actor can be operating several source phone numbers. In this paper, we exploit the semantic information obtained from call recordings. Leveraging state-of-the-art audio transcription tools, in combination with natural language processing and clustering algorithms, clusters of calls playing identical recordings are extracted and automatically labeled. We extract the audio features [14] of the calls in each cluster and feed them to classifiers to create "phoneprints" of distinct telephony infrastructures. This technique overcomes issues with spoofed or restricted source phone numbers. The main contributions of this paper are outlined below:

- We provide a data-driven analysis of the telephony fraud and spam ecosystem.
- After collecting about 40,000 call recordings, we demonstrate that only a few bad actors are responsible for the majority of telephony spam and fraud, and that they can be uniquely identified by their audio signature.

The development of this system and our analysis can help businesses design adaptive defense strategies and provide law enforcement with threat intelligence that may allow for the disruption and potential prosecution of bad actors. This paper is structured to describe each core component of our processing pipeline. Section II introduces the telephony honeypot used, the call recordings collection and transcription. In Section III, call transcripts are used as input for a topic model and a similarity metric is computed to compare transcripts. In Section IV, spectral clustering is applied to transcript projections on the topic space. Combined with calling patterns, clusters of call transcripts provide insight on several spam and fraud campaigns, and the techniques used by fraudsters. In Section V, call audio features and clusters of call transcripts are used to train classifiers that can identify distinct bad actors. Section VI draws the conclusions of the paper and provides future research perspectives.

II. Data collection

In this section, we provide an introductory analysis of the traffic observed in the honeypot in 2015. We contrast the results with a set of online comments on unwanted calls collected in 2015. Finally, we present an overview of the data set collected from the honeypot, which is used in the remainder of the paper.

A. Overview of the Honeypot

The honeypot contains about 8,000,000 calls received in 2015 from about 880,000 distinct sources to about 80,000 distinct destinations. 39% of sources only called once, and 29% of sources only twice. Therefore obtaining historical information on a given source to apply machine learning techniques is very challenging. Some statistics are provided in Table I. During most of 2015 (except for a few weeks when we conducted experiments), calls were never answered. The calls received are entirely unsollicited, and our destination phone numbers never placed any calls. The distribution of calls per source phone number is shown in log-log scale in Figure 1. This plot, along with the fact that the median number of calls is lower than the average, suggests a power law or log-normal distribution, with a heavy tail.

B. Online comments

Earlier work by Gupta et al. [12] highlighted the limitations of online complaints data sets, and showed that a telephony honeypot provides three essential features needed to improve the quality of phone abuse intelligence. The first feature is completeness: quantitative and semantic information on a phone call are needed. The second feature is accuracy: a telephony abuse report should describe who made the call, the time at which the call was made, and information about the call suggesting it may be abusive. Accuracy of such a report means that the source and time are recorded correctly and its description is objective and supports why it is abusive. Due to the open nature of online forums, complaints on it are not limited to telephony fraud. This results in noisy data where some complaints do not pertain to telephony abuse. The noisy and often conflicting nature of user reports impacts the accuracy of such datasets. The third feature is timeliness: timeliness indicates how soon a complaint is filed after an abuse call is received. Generally, abuse calls and the phone numbers from where they come are reported several days after the time when the first call was received, and generally because people have been called multiple times. However, to understand the advantages and added value of a telephony honeypot, a set of online comments is examined.

In 2015, 660,145 online comments were scraped from the top five websites about phone numbers complaints, reporting 74,000 source phone numbers as unwanted callers. The number of comments per phone number is shown in Figure 1 in log-log scale, highlighting the fact that the distribution is skewed. Some statistics are provided in Table I.

TABLE I: Statistics regarding the number of calls to the honeypot and the number of online comments.

Data set	Total volume	Nb of sources	Average volume	Median volume	Max volume per source
Honeypot	8,000 k	880 k	9	2	21,329
Online Comments	660 k	74 k	9	2	2,156

The honeypot and the online comments are both observers of the telephony world, both reflecting a necessarily partial view of the world, so it is natural to wonder how the observations from these two sets are related. In fact, the honeypot and the online comments set have a significant overlap: about 16,000 sources that called the honeypot in 2015 were the subject of online comments. This represents 1.8% of sources that called the honeypot and 21% of sources reported in the online comments. In the honeypot, these sources placed 36% of all calls, with an average of 145 calls per source, and a median of 21 calls per source. In the comments set, these sources were responsible for 66% of all comments, with an average of 26 comments per source, and a median of 7 comments per source. The sources in the intersection of these two data sets are the tails of the calls per phone number distribution and of the number of comments per phone number distribution in Figure 1. Put simply, identifying the bad actors behind 1.8% of sources in the honeypot has the potential to address 66% of online complaints.

Figure 1 is a plot of the probability distributions of the honeypot dataset and of the online comments dataset, both are very similar and almost linear. Both distributions are fitted using scipy in Python with a maximum estimation. The lognormal fit is a better fit than the power law fit, and a much better fit than an exponential. These two distributions are heavy-tailed. They can be fitted with good accuracy as lognormal distributions, and modeled with generative techniques. We intend to tackle this aspect in future work to model the behavior of fraudulent callers.

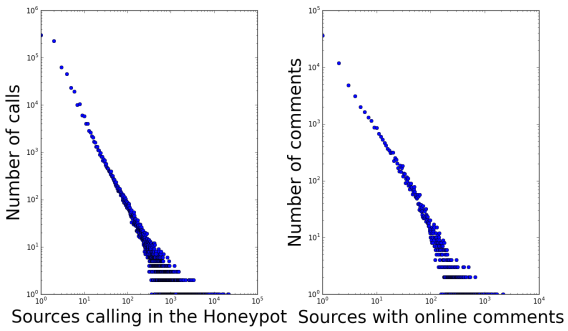


Fig. 1: Probability distributions of the calls per source phone number into the honeypot and the online comments per source phone number (log-log scale).

C. Overview of the data set collected from the Honeypot

TABLE II: Overview of the data sets collected with the honeypot.

Data set collected	Number of calls recorded	Number of distinct sources	Number of distinct destinations
Random recordings	8,871	5,179	4,867
Targeted recordings	30,910	1,338	8,838

The first experiment that we conducted consisted of randomly recording calls: over two weeks, every tenth call to destinations from one-party consent states was recorded. Moreover, we decided to target the tail of the distribution of callers, who we hypothesized corresponds to the most persistent robocallers or telemarketers. We conducted a second experiment over three weeks: if in the past week, a source number had called more than 10 times, it would be recorded if it called again. These two data sets are described in Table II. Throughout this paper, we heavily utilize both the random and targeted recordings for our analysis and compare the results obtained from both data sets.

Once a call is recorded, its audio is transcribed to a text file using Kaldi [15]. Kaldi is a free open-source speech recognition toolkit written in C++. Kaldi provides a speech recognition system based on finite-state transducers and it is intended for use by researchers. Transcripts are imperfect and contain spelling and grammatical mistakes, but identical recordings are transcribed into very similar transcripts.

The transcripts are then pre-processed before we apply language processing techniques. To filter out recordings that do not contain enough semantic information, stop words, such as prepositions or adverbs, are removed from transcripts. Then, unusual words that only appear once across all documents are removed. Transcripts containing fewer than 3 words (i.e. very little usable information) are discarded. The remaining words are lemmatized (i.e. the endings of the words are removed) and each transcript is stored as a bag-of-words.

After this initial filtering step, 3,574 calls (40%) from 1,899 sources (37%) remain for the random recordings, while 16,378 calls (52%) from 1,052 sources (79%) remain for the targeted recordings. This is the first indication that targeting heavy callers provides less noisy data. Robocallers are, by definition, playing a prerecorded message, and telemarketers read from a script, and they are the heaviest callers.

III. Detection of fraud and spam telephony campaigns

In this section, we use the transcripts from the honeypot recordings presented as input to a topic model, to perform a dimensionality reduction.

In machine learning and natural language processing, topic models are algorithms used to analyze large volumes of unlabeled text documents [16]. They uncover patterns and thematic structures in document collections. Commonly used topic models include Latent Semantic Indexing (LSI), Latent Dirichlet Allocation (LDA) and Non-negative Matrix Factorization (NMF). All three were tested on the transcripts corpus and LSI provided the most intuitive and consistent results. LSI [17] is a dimensionality reduction technique that projects documents and document queries into a space of smaller dimension, called the "topic space", than the original space of dictionary words they were expressed in. The intuition behind LSI is that there is a set of independent underlying variables which span the meanings behind the data. We outline the basics behind this algorithm for completeness.

Let the corpus be represented by a $d \times n$ weighted term-document matrix X, where d is the size of the dictionary (all distinct words observed in the corpus), and n is the number of transcripts. The columns of X are the transcripts in filtered bag-of-words form, and each term in the dictionary is represented by a row. A Term Frequency Inverse Document Frequency (TF-IDF) [18] local weighting function is applied to condition the data . TF-IDF determines how relevant a particular word is in a given document. Words that are common in a small group of documents tend to have higher TF-IDF values than common words across documents such as prepositions. Define X_{tfidf} as the weighted term-document matrix once the TF-IDF transformation has been applied: then LSI is simply the singular value decomposition (SVD) of X_{tfidf}, where only the largest k singular values are kept, and $k \ll min(d, n)$: this reduction preserves the most important semantic information in the text while reducing noise and other undesirable artifacts of the original space of X. The n columns of $S_{k\times k}V^T_{n\times k}$ are the new coordinates of each transcript after dimensionality reduction. This new coordinate system helps perform the dimensionality reduction for documents that are not in the original corpus, instead of increasing the size of the corpus and performing the full LSI again.

The results for each experiment are described in Table III. The number of topics was selected with the consideration that a smaller number of topics ensures a smoother projection space while eliminating a large share of the noise. The first topics, i.e. the topics corresponding to the largest singular values, reflect the most important scams in volume.

TABLE III: Topic Modeling Inputs.

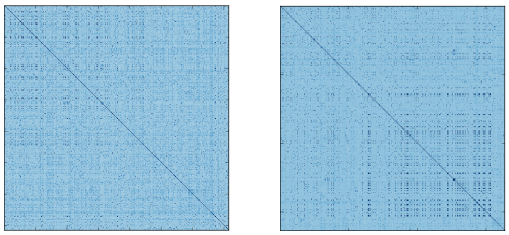
Experiment	Number of transcripts	Number of words	Number of nonzero entries
Random recordings	3,574	4,044	68,659
Targeted recordings	16,378	5,015	355,312

Once each transcript has been mapped to a lower dimensional projection, these projections can be compared in the space of topics. We use the cosine similarity measure between the projections of any pair of transcripts, which is normalized between 0 and 1, with 1 indicating identical projections.

From the pair-wise similarity scores computed across the whole corpus, an $n \times n$ similarity matrix is constructed. The similarity matrix obtained for both recordings experiments are pictured in Figure 2. The darker the blue dot between row i and column j, the more similar transcript i is to transcript j. The fact that the diagonal of the matrix is dark blue is expected, since each transcript is maximally similar to itself.

IV. Clustering of similar transcripts

In the previous section, we computed pair-wise similarity between the transcripts' projections in the topic space. In this section, we identify clusters of identical transcript projections, whose calls are playing the same recording . Initially, we tried clustering on transcripts directly, instead of clustering on projections in the topic space. However, because many transcripts from different spam and scam campaigns use identical words, it was not effective, hence the extra step taken to perform a dimensionality reduction before clustering.



(a) 3,040 random transcripts.

(b) 14,128 targeted transcripts.

Fig. 2: Similarity matrix. No pattern is easily distinguishable at this point, hence the need for clustering.

Clustering algorithms address the classical unsupervised learning problem of finding a partition for a given set of items. There are often many ways to partition the data. Spectral clustering [19] is a powerful non-parametric technique to uncover structure in data using the spectrum of a pairwise similarity matrix. The algorithm takes as inputs S, the similarity matrix, and k, the number of clusters wanted. Spectral clustering works well in practice because the graph Laplacian encodes geometric information relevant to cluster similariy, and its spectral decomposition induces a lower dimensional space upon which partitional clustering algorithms such as k-means can infer that potentially nonlinear structure.

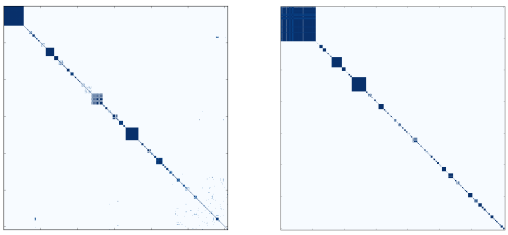
The results of the spectral clustering on the similarity matrices corresponding to the random and the targeted recordings are shown in Figure 3. The data suggests that a set of phone numbers are fraudulent when all their calls cluster, indicating that all the callers are either playing the same recording or they are humans reading from the same script. A good cluster is a cluster with very high average intra-cluster similarity, i.e. corresponding to a group of identical recordings. A dark square block in Figure 3 corresponds to a subset of identical transcripts (which are all perfectly similar to one another). Then the corresponding source phone numbers propagating the same scam or spam campaign behind the corresponding calls can be identified.

In both sets, only a few clusters contain the majority of spam and scam calls. The detailed results are presented in Table IV. Moreover, some of the clusters in both sets of recordings overlap and correspond to the same scam or spam calls.

TABLE IV: Proportion of transcripts in "good" clusters (clusters of sufficient size for a phoneprint and high average intra similarity).

Experiment	Nb of good clusters	Nb of transcripts clustered	Percentage of transcripts
Random	23	1,144	37%
Targeted	93	9,924	61%

One of the most prevalent scam campaigns is related to



(a) 3,040 random transcripts.

(b) 14,128 targeted transcripts.

Fig. 3: Similarity matrix after clustering. Each dark block corresponds to a cluster.

Google. We found more than 10 clusters of distinct recordings linked to Google, such as: *"Our records indicate that you have not optimize your google business listing of it is critical that you're listing is up to date and optimize in order to be found online press one to verify and optimize your free google listing press nine to be removed from this list."* If the destination answers, the scammer will then try to socially engineer the person answering and convince them to either pay for a bogus service or disclose personal information such as passwords [20]. The "Optimize your Google listing scam" contains 228 recordings from 3 sources to 197 destinations, where each source mimics the area code of the destinations it calls.

V. Classification of distinct bad actors

In the previous sections, traffic patterns and semantic information were extracted for a set of recordings. In this section, audio features from each recording are used to train classification models, in order to uniquely identify distinct telephony infrastructures.

A. Phoneprinting

For each recording, 150 distinct features [21], including packet loss, spectrum, and VoIP information are extracted for the purposes of training a classification model. Experiments were conducted with standard classification algorithms such as random forests, gradient boosting and kernel support vector machines (SVMs), and the latter was chosen due to its superior performance for this task (maximizing true detection and minimizing false acceptance rates). Recall that kernel SVMs [22] train nonlinear classifiers by utilizing kernel functions $k(x, y)$ that map the data x, in $\mathbf{R}^D$ using an implicit feature map $\psi : \mathbf{R}^D \to \mathcal{H}$ to a high-dimensional Hilbert space $\mathcal{H}$, and then learning a hyperplane in feature space to discriminate two classes by maximizing the margin of separation between the classes [23]. The kernel function completely determines the richness of the feature space, and even though many choices for the function exist, we chose the standard RBF kernel due to its expressive power . We call the final classification model using the aforementioned audio features a *phoneprint*. Note that the phoneprinting process involves cross-validation to infer the best parameters of the SVM model.

B. Cluster phoneprinting and bad actor identification

Using spectral clustering on the similarity matrix between transcripts, clusters of transcripts, and hence of audio recordings, are obtained. In this subsection, we report the performance of phoneprinting clusters of sufficient size that have a very high intra-cluster similarity. Phoneprinting clusters of recordings from different source phone numbers enables us to overcome the fact that most phone numbers only call once or twice in the honeypot. The experiments we conducted showed that phoneprinting clusters provides better results than phoneprinting a phone number alone, especially for robocallers and telemarketers. The intuition behind this is as follows: if several calls are placed from a given phone infrastructure, their audio features tend to be highly similar in the audio feature space. Bad actors tend to use several distinct phone numbers to perpetrate the same scam. Indeed, we saw in the honeypot the exact same recording being played by several phone numbers. Even if the caller is hiding behind several source phone numbers, by spoofing, if he or she calls from the same infrastructure, the audio features of the call recordings will be highly similar. Hence, the hypothesis is that clusters in the topic space will also tend to cluster in the audio space, except that the semantic information associated with the transcript space will allow for more accurate groupings.

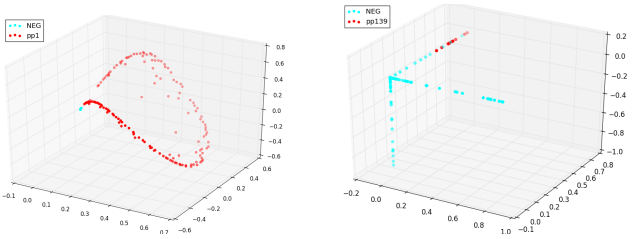
To prove this point, audio features are extracted from recordings associated to clusters with high similarity in the topic space, and are systematically phoneprinted. Across more than eighty phoneprints trained for the random and the targeted recordings data sets: the average training TPR obtained was 72%, and the average testing FPR was 0.11%. The maximum training TPR was 98%. Our results show that phoneprinting a cluster may perform very well even with less than two calls per source phone number on average. Another advantage of the methodology developed is that it enables catching bad actors hiding behind "restricted" or "anonymous" phone numbers, or spoofing phone numbers. Several clusters from the random recordings data set contained "anonymous" calls. when the caller calls again, from any phone number, he or she will be flagged as]the same bad actor by extracting the audio features and testing them through the set of existing phoneprints.

C. Kernel PCA and Visualization of Audio Features

We can demonstrate some of the insights of the last section by visualizing the audio features data in an approximation of the Hilbert space $\mathcal{H}$ that they are mapped to for classification. Recall that principal component analysis (PCA) can be used to create visualizations of data in high dimensions: given zero-mean data $\{x_i\}_{i=1}^N$, the covariance matrix $C = \frac{1}{N}\sum_{i=1}^N x_i x_i^T$ can be diagonalized into principal components, and the data can be projected to k dimensions capturing the largest possible variance. We can apply a similar procedure to compute an embedding of the audio feature data in the Hilbert space associated to the kernel $k(x, y)$ by using kernel PCA [24]: here, the $N \times N$ Gram matrix $K := k(x_i, x_j)$ between the data must be diagonalized to estimate the principal eigenfunctions, which represent the most important coordinates in $\mathcal{H}$. For a

given phoneprint, we used the RBF kernel with its chosen γ parameter to compute K, and then perform the embedding of the data that generated the phoneprint.

Using the clusters extracted above, and the performance of the associated phoneprints, examples of well-separated classes are depicted in Figure 4. For cluster 1, several thousands of data points in the negative class, in light blue, are grouped into a very localized region, while the positive class occupies a different region of the space. This shows that, in kernel space, both classes are easily separable, as the good phoneprint performance suggested. For cluster 139, the projections of the audio features (in kernel space) are not easily separable in the first three dimensions. The corresponding phoneprint performance for cluster 139 was much lower with a TPR of 46.67% against 98.01% for cluster 1 and 91.89% for cluster 110 respectively. For lower than usual phoneprint performance, the kernel PCA visualization helps us understand whether the positive class or the negative class or both contain wrong labels.



(a) Example of good separation between classes: cluster 1.

(b) Example of poor separation between classes: cluster 139.

Fig. 4: Visualization of the first three components of the Kernel PCA on audio features used for phoneprinting.

VI. Conclusion

In this paper, we provided a data-driven analysis of the telephony fraud and spam ecosystem based on recordings obtained from a telephony honeypot. We developed a technique to exploit heterogenous aspects of the telephony ecosystem (call recordings, audio signal, traffic pattern), by combining tools from supervised and unsupervised learning. We recorded about 40,000 calls to our Honeypot, both randomly and by targeting heavy callers. Our results show data-driven evidence that only a few bad actors, operating distinct telephony infrastructures, are responsible for the majority of telephony spam and fraud. Moreover, only 1.8% of sources are responsible for 66% of online complaints, and we now have techniques to detect and identify the bad actors behind them. Our system allows end-users to detect fraudulent behavior and tie it back to existing fraud and spam campaigns, it can be used as the first step towards designing and deploying intelligent defense strategies.

VII. Acknowledgments

The authors would like to acknowledge their colleagues Telvis Calhoun, Aaron Dallas and Dr. Mustaque Ahamad for their help.

References

[1] R. Dhamija, J. D. Tygar, and M. Hearst, "Why phishing works," in *Proceedings of the SIGCHI conference on Human Factors in computing systems*. ACM, 2006, pp. 581–590.

[2] G. Ollmann, "The vishing guide," *http://www. infosecwriters. com/text resources/pdf/IBM ISS vishing guide GOllmann. pdf, IBM, Tech. Rep*, 2007.

[3] *Asterisk*, (accessed December 1, 2015), http://www.asterisk.org/.

[4] *Directory of Unknown Callers*, (accessed December 1, 2015), http://www.800notes.com/.

[5] *419 Scam Directory*, (accessed December 1, 2015), http://www.419scam.org/.

[6] *FTC, Pennsylvania and Connecticut Sue Tech Support Scammers That Took More Than $17 Million From Consumers*, (accessed December 1, 2015), https://www.ftc.gov/news-events/press-releases/2015/11/.

[7] *Swindlers Use Telephones, With Internet Tactics*, (accessed December 1, 2015), http://www.nytimes.com/2014/01/20/technology/swindlers-use-telephones-with-internets-tactics.html.

[8] R. A. Becker, C. Volinsky, and A. R. Wilks, "Fraud detection in telecommunications: History and lessons learned," *Technometrics*, vol. 52, no. 1, 2010.

[9] M. Weatherford, "Mining for fraud," *Intelligent Systems, IEEE*, vol. 17, no. 4, pp. 4–6, 2002.

[10] M. Onderwater, "Detecting unusual user profiles with outlier detection techniques," 2010.

[11] V. S. Tseng, J.-C. Ying, C.-W. Huang, Y. Kao, and K.-T. Chen, "Fraudetector: A graph-mining-based framework for fraudulent phone call detection," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2015, pp. 2157–2166.

[12] P. Gupta, B. Srinivasan, V. Balasubramaniyan, and M. Ahamad, "Phoneypot: Data-driven understanding of telephony threats," 2015.

[13] K. Thomas, D. McCoy, C. Grier, A. Kolcz, and V. Paxson, "Trafficking fraudulent accounts: The role of the underground market in twitter spam and abuse." in *USENIX Security*. Citeseer, 2013, pp. 195–210.

[14] V. A. Balasubramaniyan, A. Poonawalla, M. Ahamad, M. T. Hunter, and P. Traynor, "Pindr0p: using single-ended audio features to determine call provenance," in *Proceedings of the 17th ACM conference on Computer and communications security*. ACM, 2010, pp. 109–120.

[15] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The kaldi speech recognition toolkit," in *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society, Dec. 2011, iEEE Catalog No.: CFP11SRW-USB.

[16] D. M. Blei and J. D. Lafferty, "Topic models," *Text mining: classification, clustering, and applications*, vol. 10, no. 71, p. 34, 2009.

[17] P. Wiemer-Hastings, K. Wiemer-Hastings, and A. Graesser, "Latent semantic analysis," in *Proceedings of the 16th international joint conference on Artificial intelligence*. Citeseer, 2004, pp. 1–14.

[18] J. Ramos, "Using tf-idf to determine word relevance in document queries," in *Proceedings of the first instructional conference on machine learning*, 2003.

[19] A. Y. Ng, M. I. Jordan, Y. Weiss *et al.*, "On spectral clustering: Analysis and an algorithm," *Advances in neural information processing systems*, vol. 2, pp. 849–856, 2002.

[20] *Google Scam*, (accessed January 31, 2016), https://support.google.com/faqs/answer/2952493?hl=en.

[21] V. Balasubramaniyan and M. Ahamad, "Patent US 20130109358 A1: Systems and methods for detecting call provenance from call audio," Patent US 20 130 109 358 A1, 06 29, 2011.

[22] J. Shawe-Taylor and N. Cristianini, *Kernel methods for pattern analysis*. Cambridge university press, 2004.

[23] I. Steinwart and A. Christmann, *Support vector machines*. Springer Science & Business Media, 2008.

[24] B. Schölkopf, A. Smola, and K.-R. Müller, "Kernel principal component analysis," pp. 583–588, 1997.