A Machine Learning-Driven Analysis of Phantom E911 Calls

Yang Hu*
Batoul Taki*
yh756@scarletmail.rutgers.edu
batoul.taki@rutgers.edu
Rutgers University
Piscatway, New Jersey, USA

Waheed U. Bajwa†
Rutgers University
Piscatway, New Jersey, USA
waheed.bajwa@rutgers.edu

Manoop Talasila
AT&T Labs
Bedminster, New Jersey, USA
talasila@att.com

Mukesh Mantan
AT&T Labs
Bedminster, New Jersey, USA
mm9430@att.com

Syed Anwar Aftab
AT&T Labs
Bedminster, New Jersey, USA
anwar@att.com

ABSTRACT

Phantom Enhanced 911 (E911) calls are automatically generated 2 second calls, are a serious concern on cellular networks, and consume critical resources. As networks become increasingly complex, detecting and troubleshooting the causes of phantom E911 calls is becoming increasingly difficult. In this paper machine learning (ML) tools are used to analyze anonymized call detail record data collected by a major US telecom network service provider. The data is carefully pre-processed and encoded using an efficient encoding method. Classification algorithms K Nearest Neighbors (KNN) and Decision Trees (DTs) are then implemented to study correlations between device and network level features and a mobile device's ability to initiate phantom calls. Based on the results, this work also suggests certain policy changes for network operators that may decrease the high volume of phantom E911 calls or alleviate the pressure of phantom E911 calls on a cellular network.

CCS CONCEPTS

• **Computing methodologies** → **Machine learning**; **Machine learning approaches**; *Cross-validation.*

KEYWORDS

Data Encoding, Decision Tree, Diagnosis, Feature Engineering, K-Nearest Neighbors, Machine Learning, Transductive Learning

ACM Reference Format:
Yang Hu, Batoul Taki, Waheed U. Bajwa, Manoop Talasila, Mukesh Mantan, and Syed Anwar Aftab. 2022. A Machine Learning-Driven Analysis of Phantom E911 Calls. In *Proceedings of the 2022 ACM Workshop on Wireless Security and Machine Learning (WiseML '22), May 19, 2022, San Antonio, TX, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3522783.3529527

*Both authors contributed equally to this research.
†Corresponding author.

This work was supported by the US NSF Grant CCF-1907658 ARO Grant W911NF-21-1-0301.

WiseML '22, May 19, 2022, San Antonio, TX, USA

ACM ISBN 978-1-4503-9277-8/22/05…$15.00
https://doi.org/10.1145/3522783.3529527

1 INTRODUCTION

An average of 240 million 911 calls are made annually in the United States, 80% of which originate from wireless devices [11]. E911 calls, which provide dispatchers with the caller's location, are especially important for public safety. Emergency services rely on such calls being routed from cellular devices to first responders in a timely manner. In recent years, telecommunication networks have been receiving high volumes of "phantom" E911 calls. Phantom calls are typically automatically generated two-second calls, placed often unbeknownst to the user. Phantom E911 calls are problematic as they pose a drag on cellular networks and take away from critical resources needed to respond to genuine calls and true emergencies. At the moment, it is unclear whether phantom E911 calls are a product of device configuration issues or are related to issues in the device operating system. Despite their high volume, the huge difference in data sample size between good calls vs phantom calls makes it challenging for the machine learning model to learn and classify them effectively. Additionally, mobile networks often contain mobile devices from third-party vendors; this adds further complexity to the network and makes the task of phantom call detection even more challenging. To this end, it is crucial to study the phenomenon of phantom E911 calls to diagnose where the issues lie (whether they are the result of a few isolated device types or a majority of devices), and to propose possible actions that can be taken by the general cellular community to help mitigate the problem.

In order to fulfill the current customer demands fueled by massive subscriber growth, mobile network technologies are enhancing at a fast rate. Phantom E911 calls are one of the by-products of this ever changing technological landscape where the devices and their software configurations need to support connectivity and call protocols for various changes across technologies (3G/4G/5G etc..). Multiple existing works have discussed the impact of complexity on the mobile network as well as on its users, management and maintenance. Some work study security threats (or vulnerabilities) at the network level [1, 3, 18]. These works characterize attacks as actions that usually take advantage of mobile devices, or deny service to them. Some of these works study the possible

security threats of rogue femtocells within a network [2, 5], including the ability of mobile operators to "create" rogue mobile devices. Other literature identify Denial of Service (DoS) attacks that may target a network and a collection of mobile devices due to rogue or faulty base stations in Self Organizing Networks (SONs) [15]. Other works of interest employ ML-driven approaches to study or detect abnormalities/anomalies in the network. These abnormalities are non-threatening phenomenon (an example of which is the problem of phantom E911 calls under study), but may cause an unwanted drag on the network resources. Anomalies in the current literature include base stations that exhibit irregular behavior or cells experiencing performance degradation due to mobility (such as during handovers) [7, 16]. Other abnormalities include sleeping cells [4], cells with unusually low or even high user activity (occurring perhaps due to malfunctions or events in the urban environment) [12, 17]. Furthermore, other works detect "spectrum anomalies," characterized as abnormal spectrum usage [8]. Much of the existing literature employs unsupervised or semi-supervised machine learning algorithms to detect such abnormalities/anomalies of interest, using collected data such as call-records or spectrum data. While these works study various anomalies that arise within the network architecture and try to detect them, the phenomenon of phantom E911 calls, and ways to detect them, has yet to be studied in detail- despite it being a major issue of concern among network operators. In addition to this, the types of anomalies studied in the literature are mainly network level related anomalies (such as sleeping cells or spectrum anomalies) [4, 8, 12]. However, an early investigation of the anonymized call records data by a major US network service provider indicates that phantom calls are most likely a consequence device level related anomalies (i.e. a mobile device"going rogue") among a particular cluster of devices. Additionally, anomalies mostly come from a particular cell site(s) in the network, whereby network devices can be used to locate the anomalous devices. Therefore, in this paper, we hypothesize that the possible causes of phantom E911 calls can be of two types: 1) those related to devices and 2) those related to the network properties. In order to verify this hypothesis we opt to study device and network related features separately. To take the first step towards "taming the beast" of phantom E911 calls we use readily available machine learning tools to study anonymized data on call detail records containing various device IDs and network IDs, and explore device and network-related features that may increase the potential of a device to initiate phantom calls. The aim of this paper is therefore diagnostic in nature [10], in which we utilize machine learning-driven techniques to identify the possible triggers of phantom E911 calls. Our main contributions in this direction are as follows:

- Call record data is carefully pre-processed, separated into device and network level features, and encoded.
- A KNN algorithm is used to classify call record data, verifying that only a handful of device-level features have a strong correlation with a mobile's ability to initiate phantom E911 calls. The fitted model is generalizable and works well even on new testing data.
- A DT algorithm is used to classify call record data, verifying that only certain network level features are associated with E911 phantom calls.

Figure 1: High-Level System view of LTE network. Two-sided arrow indicates a connection between UE and the network, dotted lines are signals, solid lines are data traffic

- Based on our findings, we provide network operator policy suggestions to mitigate the problem of E911 phantom calls.

It's worth noting that the machine learning algorithms were chosen due to their ease of implementation, interpretability and accuracy. Additionally, these machine learning approaches do not require any prior assumptions on the model as other signal processing based approaches may. The remainder of the paper is organized as follows: Section 2 formulates the problem under study. Section 3 gives a formal description of the data used for E911 phantom call diagnosis, the pre-processing steps, the machine learning model and experiments. Section 4 provides an interpretation of the results, and Section 5 concludes the work.

2 PROBLEM FORMULATION

Throughout this paper, we also refer to "rogue devices" as mobile devices that have initiated phantom E911 calls. We focus on LTE networks, as the majority of the present network calls are through 4G LTE and the phenomenon of phantom E911 calls has been prominently observed in such networks. A 4G LTE network is comprised of four essential blocks; the User Equipment (UE), the Evolved UTRAN (E-UTRAN), the Evolved Packet Core (EPC) and the Packet Data Network (PDN). Figure 1 depicts the LTE architecture at the block level. The EUTRAN controls all radio communications between a UE (i.e. a mobile device) and the EPC. The EPC communicates with external networks, or, PDN, to provide services to the mobile user, such as the Internet or other operator services. The EPC contains the core entities responsible for the functionality of the network, including management of voice and data calls. Figure 2 shows a more detailed (though not comprehensive) view of the LTE architecture blocks. The E-UTRAN has only a single physical component, the eNodeB (eNB). An eNB (located at a given point, or, "cell site") is assigned a cluster of one or more network cells (a wide geographical area with devices that communicate with a cell site), and controls the mobile devices located within its cluster. For a mobile device in a particular cell, the controlling eNB is responsible for radio interface transmission and reception. In addition to this, the eNB is responsible for ensuring secure connectivity between the mobile device and the rest of the network. In the EPC, the Mobile Management Entity (MME) is central to all other entities within the EPC. Each MME manages several eNBs - there are typically two or more MMEs in the entire network. A single MME communicates with the mobile devices in its service area through their corresponding eNBs. One of the many responsibilities of the MME is the authentication of the mobile device upon its initial connection to the network. To avoid security threats that may arise when a mobile device makes itself known to the network, the MME assigns each device under its management a unique but

temporary identity, or Globally Unique Temporary ID (GUTI). The MME also supports control signaling for handovers of a mobile device between eNBs or other MMEs. The MME communicates with two other major components; the Home Subscriber Server (HSS) and the Serving Gateway (S-GW). The HSS contains the user profiles of the mobile devices in the network, and the S-GW—much like a router—forwards data between the network and any PDN Gateway (P-GW), which then connects to the PDN if necessary [14]. The main pipeline of initiating a call in the network architecture is highlighted in purple, Figure 2. For a phantom E911 call to be initiated (for a device to "go rogue"), some malfunction, unexpected issue or error must have occurred somewhere along the pipeline. As Figure 2 suggests, the factors contributing to a device "going rogue" may be device related, network related, or both.

The data under study was collected from oracle device data sources. The main features of interest are The International Mobile Equipment Identity (IMEI) and International Mobile Subscriber Identifier (IMSI). IMEI is a unique identification or serial number assigned to mobile devices. IMSI is a number of 14-15 digits which uniquely identifies a mobile subscriber by their SIM card. Each data sample is labelled as "phantom - 1" if it was collected from a rogue device, and "role model - 0" if it was collected from a normal device.

Throughout this paper we identify a mobile device that has a tendency to initiate phantom calls (i.e a rogue device) through its IMEI, and we refer to it as a "phantom IMEI". A non-phantom IMEI is referred to as a "role model IMEI". Thus, given a dataset of call records with device and network level features, where phantom and role model IMEIs are known, we wish to build two transductive models that can identify device and network level features strongly correlated with a mobile's ability to go rogue. We must note here that a challenging aspect of the data under study is its categorical nature, and we must apply pre-processing and encoding techniques before any transductive model can be implemented.

3 METHODOLOGY

In this section we introduce the data, discuss its pre-processing (data anonymization, cleaning on feature and sample levels), describe encoding of categorical data, and discuss the experimental set up.

3.1 Data Pre-Processing and Encoding

Pre-processing raw features after data acquisition is an essential step in machine learning. It ensures the quality and usefulness of the data and directly affects the performance of the model. The initial step of pre-processing involves data-cleaning at the feature and sample level. This involves carefully filling in missing values (or removing samples with multiple missing values), and discarding features that are deemed unhelpful. Discarding of features is done sparingly and we rely on the following criteria to systematically determine if a feature should be discarded: 1) A feature has null/missing values across all data samples. 2) A feature is redundant (has only one value across all call records, for both phantom and role model IMEIs). 3) A feature is deemed meaningless/desultory, i.e, it has no affect on the data sample label.

The original features consisted of 209 features that described different device and network specific information, such as time stamps, and various identifiers for devices, gateways, nodes and area codes. After anonymizing and cleaning the data based on the criteria described above Table 1 shows a list and description of the final set of features under study. The sensitive personal information was anonymized by removing it from the dataset to maintain the customer privacy. As we can see, the device-related features are "IMEI", "IMSI", and the network related features are "MMEGI", "MMEC", "MTMSI", "PLMN_ID", "eNBId", "CELL_ID".

Table 1: Description of Device and Network Level Features

Feature	Description	Type
IMSI	International Mobile Subscriber Identity: The unique identifier of a mobile device's SIM card. It is used to authenticate a user in the mobile network.	Device
IMEI	International Mobile Equipment Identity: A unique identifier of an individual mobile/UE.	Device
CELL_ID	Geographic area that communicates with an eNB, and identified by its "cell ID".	Network
PLMN_ID	Public Land Mobile Network Identifier: A unique identifier of a PLMN (a terrestrial wireless communication network).	Network
MMEGI	MME Group ID: Unique within a PLMN. When MME Pooling is utilised within an LTE network, the MMEGI uniquely identifies which group (pool) the MME is assigned to.	Network
MMEC	MME Code: Uniquely identifies an MME within a MME Group.	Network
MTMSI	MME Mobile Subscriber Identity: Assigned randomly but unique for every device within an MME.	Network
eNBId	Identification number of the eNodeB.	Network

The features under study are categorical data, thus the pre-processed data must also be encoded. Some of the most notable encodign techniques include One hot Encoding, Binary Encoding and BaseN Encoding [6]. The major limitations of One Hot or Binary encoding are computation complexity and the storage. We apply Base$N$-Encoder to encode the cleaned data with $N = 10$, which is essentially decimal encoding. Compared to other encoding methods, decimal encoding requires less computational resources, especially when the number of distinct categories for any feature is large, (which is indeed the case for the data under study).

3.2 Supervised ML - Experimental Setup

This work employs two supervised classification algorithms: K Nearest Neighbours (KNN) and Decision Trees (DT), [9, 13, 19, 20]. We now introduce two sets of experiments:

(1) KNN classification model on data with the device level features, only.
(2) DT classification model on data with the network level features, only.

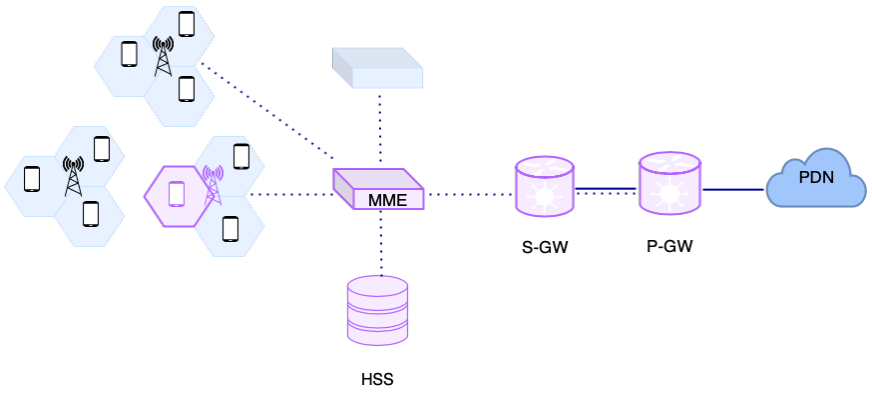

Figure 2: An in-depth view of LTE network architecture showing communication between the UE, E-UTRAN, EPC and PDN. The dotted lines indicate signals and the solid lines indicate data traffic.

We used a KNN model in order to achieve a higher level of accuracy in experiment setup 1. We thus use an auxiliary software to perform a feature importance analysis. Since we use DT in experiment setup 2 we are able to perform a feature analysis automatically. As in Figure 3, a main dataset for both experimental setupsconsists of network and device level features from (4G/LTE) network call records collected between the months of January and August of 2021. The data consisted of 7 distinct phantom IMEIs and 25000 distinct role model IMEIs. Each IMEI produces at least 1 - up to a few thousand- samples (i.e, we are not operating in the undersampling regime). The samples were labeled as "phantom -1 " if its corresponding IMEI was phantom and "role model - 0" (or "not phantom") if its corresponding IMEI was not phantom. The categorical data for device level features (IMEI and IMSI) from these samples were extracted and used in the first experimental setup, whilst the categorical data for the network level features were extracted and used in the second experimental setup. In addition to this, some supplementary device level data samples were also appended with the data for setup 1, comprised of 5890 distinct phantom IMEIs. We note here that many of the network level features shown in Table 1 are in fact elements composing the GUTI. As shown in Figure 4, the GUTI is comprised of the features "MMEGI", "MMEC", "MTMSI", "PLMN_ID". By considering the features individually, we can increase the richness and interpretability of the data under study. With this motivation in mind, feature engineering was then performed on the device level features. As shown in Figure 5, IMEI is also composed of several features, IMEI is structured using Type Allocation Code (TAC), the Serial Number, and a Check digit (which we are not interested in for the purposes of this paper). TAC is used to uniquely identify a mobile device in an LTE network and is composed of a Body Identifier and a Type Identifier, which indicate the device's brand owner and model type. The Serial Number is a unique number assigned to the device by the manufacturer. Similarly, IMSI can also be split into the Mobile Country Code (MCC), the Mobile Network Code (MNC), and the Mobile Subscriber Identification Number (MSIN), which are all identifiers stored on a SIM card in a device. Therefore - when studying device level features - rather than considering IMEI and IMSI alone, we opt to consider the 6 features that compose IMEI and IMSI (i.e. TAC

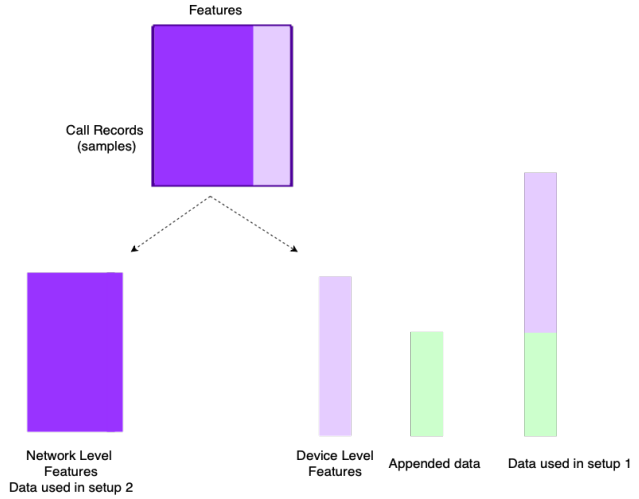

Figure 3: Data Partitioning for Experiments

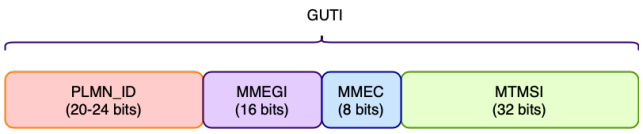

Figure 4: GUTI; no more than 80 bits and is PLMN_ID + MMEGI + MMEC + MTMSI.

Body, TAC Type, Serial Number, MCC, MNC, MSIN). In experiment setup 1, we hypothesize that the chosen features are indeed features of interest and are strongly correlated with a device's ability to initiate phantom E911 calls. Our goal is to verify this hypothesis. Appending supplementary device level data (as shown in Figure 3) was a balancing technique done in order to achieve comparability between the number of phantom and role model data samples respectively. For training the KNN model, we used a 67/33 split ratio. The distinct device features (i.e. the 6 components of IMEI and IMSI rather than IMEI and IMSI alone) were used, and data was sampled

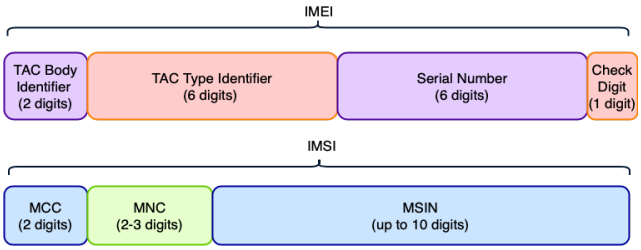

Figure 5: IMEI and IMSI; no more than 15 digits each. IMEI is TAC (body and type) + Serial Number + Code. IMSI is MCC + MNC + MSIN.

in a way such that there was no overlap in IMEI (phantom and role model) between the training and validation sets. The training stage included hyper parameter tuning in order to get the best number of neighbors K. In order to test the generalizability of the fitted model (and evaluate the model with changes in real networks), the model was also tested against a the selection of call records extracted from the year 2020, comprised of 19551 distinct phantom IMEIs and 10000 distinct role model IMEIs. In addition to fitting the KNN model, we are interested in understanding the contributions that the device level features have on the model. For this we used an inference tool, Lime, which helped in understanding the data features that impact the accuracy.

In experiment setup 2, we diagnose network related features that may be correlated to a device's ability to initiate phantom E911 calls, or may aid in further understanding some of the properties of phantom devices. We use a DT model and analyze feature importance. The data utilized in experiment setup 2 contained call records from 7 phantom IMEIs. Therefore, we dropped one phantom IMEI due to it's very low number of samples , and just performed 6-fold cross validation, where in each validation iteration (also called a trial) call records from an IMEI were used as test data. To ensure a balanced dataset in each iteration, we sample role model call records uniformly at random- of course, also ensuring no overlap in IMEIs between training and testing.

4 NUMERICAL RESULTS AND DISCUSSION

In both experimental setups, we consider two accuracy scores: Recall and Specificity. Denote "true/false positive" and "true/false negative" as TP/FP and TN/FN respectively. Recall is $(TP/(TP + TN))$ representing the model's ability to accurately label phantom call records. Specificity is $(TN/(TN+FP))$ representing the model's ability to accurately label role model call records. It is important to separate these two scores as our goal is to detect phantom call records specifically, and an average classification accuracy may not reflect the model's ability to do so.

4.1 Experimental Setup 1 (Device Level Features)

In experiment setup 1, training/validation of the KNN model had a 94% accuracy overall. As for testing, we stress that the testing dataset is never used in the training/validation process, thus becomes a perfect test data to evaluate the generalizability of the KNN model. The model was able to predict 80% of phantom call records, and 94% of normal records which indicates a high confidence in the model's predictive ability. The most impactful data features found using Lime are shown to be TAC Type and MNC. These results help customer care teams to only focus on a narrowed list of possible devices that have a potential to go rogue (in our experiments, 25% of devices were tagged as rogue compared to millions of active devices in the network), allowing proactive action on the identified devices, perhaps even before any phantom call is initiated.

4.2 Experimental Setup 2 (Network Level Features)

Table 2 shows the average test recall values for each validation trial (i.e, for each IMEI) over 14 Monte Carlos. We hypertune over the depth of the DT, with depth= 2 yielding best accuracy scores. Figure 6 shows the corresponding DT of the best cross-validation iteration (trial 2). The feature importance analysis of the DT shows that MTMSI is mostly correlated with a phantom call record, and occupies 98% – 100% of the decisive power. Table 3 shows the average recall and specificity scores across all validations, where $\sum$ denotes the sum of samples over all 6 trials. The average recall over all trials is 95.3%, which means we were able to correctly detect over 95% call records made by phantom devices. To validate the results of the DT, Table 2 also shows the average test recall values when using only the network feature MTMSI, and Table 4 shows the average recall and specificity over all trials, using only MTMSI. We see from the approximately equal accuracies in Table 3 and Table 4, that the results agree with the inference of the DT, whereby MTMSI is a feature most correlated with phantom devices and holds most of the decisive power in the phantom detection process. MTMSI is associated with MME group which in turn helps to recognize the geographic location involved in a phantom call. Identifying the MTMSI and MME associated to the phantom call can be very useful if a cluster of devices are involved in phantom calls. Such a cluster of MTMSIs can be easily identified through the DT itself. Based on these results, we conclude two aspects. First, the device specific machine learning model helps in the real networks to predict the devices that have the potential to go rogue (Namely IMEI and IMSI, but more specifically, TAC Body Type and MNC). Such predictions directly help network carrier's customer care teams to focus on helping customer's with such devices - if they are "bring your own device" (BYOD), then such devices can be suggested for discounted upgrades to solve the software issues related to features highly correlated with phantom devices. Applying these steps at an early stage saves network resources and reduces the volume of phantom E911 calls. Secondly, the network specific machine learning model helps in detecting the cluster of devices that are behaving in rogue (through their MTMSIs), so customer care teams can identify the behaviour and place appropriate measures to handle them automatically. This saves troubleshooting time for customer care personnel and network engineers, rather than having to manually figure out the pattern in such cluster of rogue devices.

5 CONCLUSION

In this paper we implemented two diagnostic models for device and network related features respectively, using supervised ML

Table 2: Recall accuracy for DT model over 6 Cross Validation iterations, for all network features and for MTMSI

Trial	Ph_IMEI	6 Features (Depth= 2)	Feature MTMSI (Depth= 2)
1	IMEI 1	0.962	0.979
2	IMEI 2	0.987	0.969
3	IMEI 3	0.978	0.986
4	IMEI 4	0.819	0.945
5	IMEI 5	0.438	0.393
6	IMEI 6	0.643	0.714

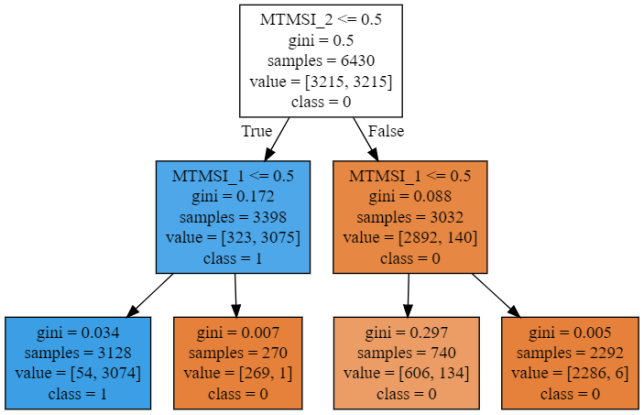

Figure 6: Decision Tree (depth=2) of Trial Based on the 6 Network Features

Table 3: Results of DT Model (6 Trials, 6 Network Features

Max_Depth	Criterion	Expression	Value
2	Average Recall	$\frac{\sum TP}{\sum TP+\sum FN}$	0.953
	Average Specificity	$\frac{\sum TN}{\sum TN+\sum FP}$	0.979

Table 4: Reults of DT Model Over 6 Trials for Feature 'MTMSI'

Max_Depth	Criterion	Expression	Value
2	Average Recall	$\frac{\sum TP}{\sum TP+\sum FN}$	0.960
	Average Specificity	$\frac{\sum TN}{\sum TN+\sum FP}$	0.981

algorithms: KNN and DT. Our results show that only a handful of features are strongly correlated with a device's ability to initiate phantom E911 calls. The device ML model has shown to be generalizable and works well even with unseen data. We are also able to conclude that the network feature "MTMSI" occupied around 99% importance in triggering the phantom E911 calls. Based on this, we give suggestions to network operators that may help reduce the large volume and provide early prevention of phantom E911 calls.

ACKNOWLEDGEMENTS

The authors would like to thank Anthony Caracciolo, Jia Wang and the entire development team for their collaboration and subject matter expertise towards initial data analytics. We thank Raj Savoor and Jennifer Yates for their invaluable support towards the innovative research collaboration.

REFERENCES

[1] Ramzi Bassil, Imad H Elhajj, Ali Chehab, and Ayman Kayssi. 2013. Effects of signaling attacks on LTE networks. In *2013 27th Int. Conf. on Advanced Information Networking and Applications Workshops*. IEEE, 499–504.

[2] Ravishankar Borgaonkar, Kevin Redon, and Jean-Pierre Seifert. 2011. Security analysis of a femtocell device. In *Proceedings of the 4th Int. Conf. on Security of Information and Networks*. 95–102.

[3] Jin Cao, Maode Ma, Hui Li, Yueyu Zhang, and Zhenxing Luo. 2013. A survey on security aspects for LTE and LTE-A networks. *IEEE communications surveys & tutorials* 16, 1 (2013), 283–302.

[4] Sergey Chernov, Michael Cochez, and Tapani Ristaniemi. 2015. Anomaly detection algorithms for the sleeping cell detection in LTE networks. In *2015 IEEE 81st Vehicular Technology Conference (VTC Spring)*. IEEE, 1–5.

[5] Nico Golde, Kevin Redon, and Ravishankar Borgaonkar. 2012. Weaponizing Femtocells: The Effect of Rogue Devices on Mobile Telecommunications.. In *NDSS*.

[6] John T Hancock and Taghi M Khoshgoftaar. 2020. Survey on categorical data for neural networks. *Journal of Big Data* 7, 1 (2020), 1–41.

[7] Bilal Hussain, Qinghe Du, and Pinyi Ren. 2018. Semi-supervised learning based big data-driven anomaly detection in mobile wireless networks. *China Communications* 15, 4 (2018), 41–57.

[8] Zhijing Li, Zhujun Xiao, Bolun Wang, Ben Y Zhao, and Haitao Zheng. 2019. Scaling deep learning models for spectrum anomaly detection. In *Proceedings of the Twentieth ACM International Symposium on Mobile Ad Hoc Networking and Computing*. 291–300.

[9] William McGinnis, Chapman Siu, Andre S, and Hanyu Huang. 2018. Category Encoders: a scikit-learn-contrib package of transformers for encoding categorical data. *The Journal of Open Source Software* 3 (01 2018), 501. https://doi.org/10.21105/joss.00501

[10] Robert Milne. 1987. Strategies for Diagnosis. *IEEE Transactions on Systems, Man, and Cybernetics* 17 (1987), 333–339.

[11] Yisroel Mirsky and Mordechai Guri. 2020. DDoS Attacks on 9-1-1 Emergency Services. *IEEE Trans. on Dependable and Secure Computing* 18, 6 (2020), 2767–2786.

[12] Jessica Moysen, Furqan Ahmed, Mario García-Lozano, and Jarno Niëmela. 2020. Unsupervised learning for detection of mobility related anomalies in commercial LTE networks. In *2020 European Conf. on Networks and Communications (EuCNC)*. IEEE, 111–115.

[13] Anthony J Myles, Robert N Feudale, Yang Liu, Nathaniel A Woody, and Steven D Brown. 2004. An introduction to decision tree modeling. *Journal of Chemometrics: A Journal of the Chemometrics Society* 18, 6 (2004), 275–285.

[14] Dhruv Sunil Shah. 2011. A Tutorial On LTE Evolved UTRAN (EUTRAN) and LTE Self Organizing Networks (SON). (2011).

[15] Altaf Shaik, Ravishankar Borgaonkar, Shinjo Park, and Jean-Pierre Seifert. 2018. On the impact of rogue base stations in 4g/lte self organizing networks. In *Proceedings of the 11th ACM Conference on Security & Privacy in Wireless and Mobile Networks*. 75–86.

[16] Bo Sun, Fei Yu, Kui Wu, Yang Xiao, and Victor CM Leung. 2006. Enhancing security using mobility-based anomaly detection in cellular mobile networks. *IEEE Transactions on Vehicular Technology* 55, 4 (2006), 1385–1396.

[17] Hoang Duy Trinh, Lorenza Giupponi, and Paolo Dini. 2019. Urban anomaly detection by processing mobile traffic traces with LSTM neural networks. In *2019 16th Annual IEEE Int. Conf. on Sensing, Communication, and Networking (SECON)*. IEEE, 1–8.

[18] Khyati Vachhani. 2018. Security threats against LTE networks: A survey. In *International Symposium on Security in Computing and Communication*. Springer, 242–256.

[19] Yiming Yang. 1999. An Evaluation of Statistical Approaches to Text Categorization. *Inf. Retr.* 1, 1–2 (may 1999), 69–90. https://doi.org/10.1023/A:1009982220290

[20] Yiming Yang and Xin Liu. 1999. A Re-Examination of Text Categorization Methods. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (Berkeley, California, USA) *(SIGIR '99)*. Association for Computing Machinery, New York, NY, USA, 42–49. https://doi.org/10.1145/312624.312647