CAPP: A Context-Aware Proof of Presence for Crowdsensing Incentives

Nicholas Handaja*, Brent Lagesse†
Computing and Software Systems
University of Washington Bothell
Bothell, WA USA
Email: *nhandaja@uw.edu, †lagesse@uw.edu

***Abstract*—As crowdsensing applications become more prevalent, they are becoming a more frequent target of attacks. Attacks on incentive systems are difficult to prevent because the attacker is not necessarily submitting malicious data, in fact, the data is typically completely valid, so previous work in data trust and integrity will not prevent these attacks.**

We have taken a context-aware approach to preventing attackers from submitting useless information to exploit incentive systems. Our system requires participants to submit a context package along with the data they are collecting. This context package acts as a proof that they were where they said they were when they said they were and is evaluated using a machine learning model. The system is able to achieve an F1 score of 0.93 against an attacker that submits arbitrary context information. We have demonstrated the feasibility of this approach in such a way that it raises the cost to attackers to submit arbitrary data to a crowdsensing system to exploit its incentive mechanisms.

***Index Terms*—Context-Aware Security, Crowdsensing, Incentives**

I. Introduction

Crowdsensing is an emerging method for data collection that lowers the barrier of entry from traditional infrastructure-driven approaches. Whereas traditional sensing requires that the investigator or company to deploy and maintain devices in the environment, crowdsensing utilizes sensors that people carry with them, typically in the form of a mobile phone, to acquire experimental data. Researchers have already used crowdsensing for a variety of applications and scientific studies including tracking the spread of invasive species[1], assessing noise pollution[2], monitoring traffic conditions[3]. In all of these cases, there is a spatio-temporal component to the data that is critical for it to be useful. If a user lies about the time or location of the sensed information, that information loses its value to the crowdsensing system.

Since users that collect information for crowdsensing applications are volunteering their time and resources, it is necessary to incentivize the users to participate. There are a several ways that this is done. Most taxonomies describe incentives in some form similar to incentives based on the value of the application, incentives by gamification of the sensing task, and incentivization by payments[4], [5], [6]. In crowdsensing, the security concern that we address is methods by which we can prevent attackers from being rewarded for contributing information that does not help the system. The focus of this work differs from work that focuses on detecting bad data, because the contributed data could be indistinguishable from good data, but not add any additional value to the system.

GPS spoofing has frequently been discussed in terms of attacks on UAVs, Smart Grid, and military applications[7], [8], [9], but recently there has been an uptick in attacks that focus on economic incentives related to location. A recent news article [10] describes a case in 2018 where a user of AEON Kyushu app in Japan set up machines to check in to locations non-stop in exchange for micro-incentives. The user accrued nearly 2.7 million check-ins which totaled approximately 5,380,000 Yen ($53,500 USD) worth of AEON store credit. Similarly, GPS spoofing has been used ride share systems to gain economic advantages. [11] reports that Uber drivers in Lagos have been spoofing GPS locations to trick the Uber platform into calculating an inflated fare for the rider. An alternative strategy is presented in [12] where a driver uses two phones. Airports tend to yield high-profit rides for ride-sharing drivers; however, because of this there is often a long queue of drivers waiting for passengers. On one phone, they spoof the location of the local airport, so that they can enter the queue, then they use another phone to perform normal rides only to the airport just in time to be at the head of the queue.

We present the development of a spatio-temporal model that uses a context proof from a user to determine how likely they are to be in the location they claim at the time they claim. The context-proof system is designed so that it can integrate into existing crowdsensing frameworks such as the AWARE Framework[13]. The primary contribution of this work is to raise the cost to an attacker, so that these currently used attacks are no longer viable. Additionally, we describe two more sophisticated, but also more expensive, attacks that adversaries might move to in the future and demonstrate their effectiveness to motivate future work in this area.

II. Background

A. Security in Crowdsensing

Secure crowdsensing consists of 3 main research challenges that extend beyond traditional security research. The first of those is ensuring that the data that is received from the users is trustworthy. The second is to ensure that the privacy of both the participants and adjacent non-participants is preserved. The third, which is the focus of this work, is to ensure that users do not exploit crowdsensing incentive systems to receive rewards without contributing useful information. Incentives are used in crowdsensing experiments and applications to motivate users to continue participating. They can take many forms such as financial, service, or entertainment. Incentive security differs from data security in that an attacker could replay real data that is completely valid. This data would not damage any of the analysis that is done on the dataset, but it would come at an unnecessary cost to the system organizer.

B. Device Analyzer Dataset

The Device Analyzer project [14] by the University of Cambridge collects data from the mobile phones of volunteers. The project collects data about approximately 300 different events including information about alarms, applications, audio settings, various types of network connectivity, contacts, location, telephony status, power, sensors, system settings, storage, and much more, making it generic enough for use with any crowdsensing application[1]. The data has been collected from hundreds of models of phones in over 175 countries across many years.

III. Problem Statement

Most work in incentive mechanisms focus on creating incentive mechanisms that are game-theoretically sound and elicit honest valuation of the work from the participants. Little work has been done on ensuring that the data that itself is submitted is adding value to the system. Some work has focused on incentivizing quality of information by recruiting reliable users [15]; however, these approaches lack a scalable and secure approach to validating QoI of sensed information[16] when the attacker voluntarily submits low quality information. As a result, the problem in that we address in this paper is **how can a crowdsensing system detect that a participant is submitting information that they did not actually collect at the time and location they claim.**

A. System Model

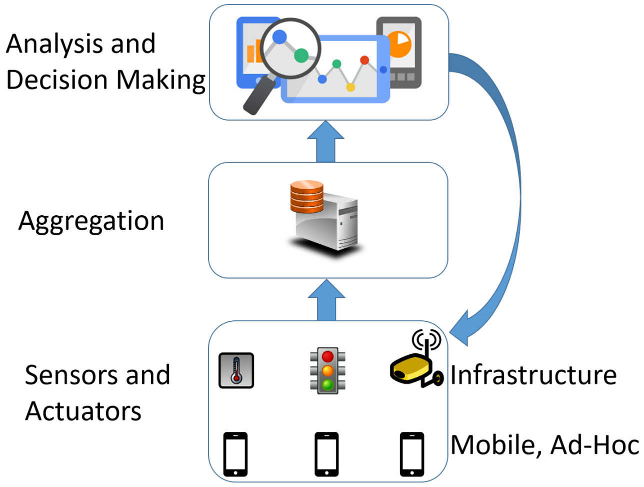

Fig. 1: Typical Crowdsensing System

For this project, we assume a generic crowdsensing system that leverages the data collecting capabilities of large groups of participants. The goal of the system is to derive meaningful information for its owner from data collected voluntarily by participants through aggregate analytics. Incentives are provided by the system to participants in order to compensate them for their time and resources used. Figure 1 shows a generic architecture of a crowdsensing system.

B. Attacker Model

We identify three main attacker models that might be used against crowdsensing incentive systems. In this work, we take a first step into preventing them by addressing the **Indiscriminant Data** attacker model as it is currently the only one that has been observed in the wild and is the cheapest for an attacker to use. In our future work, we will address the remaining attacker models, so we include them in this section to motivate discussion.

1) Indiscriminant Data.: In this model, the attacker submits arbitrary data to the system to receive rewards even though this data is not valid. This is the most simplistic of the attacker models, but it is also the lowest cost to the attacker.

2) Realistic Data.: In this model, the attacker has some domain knowledge of the context that is submitted. As a result, the attacker can submit data that could realistically fall within the range of values for the time and location. An example of this would be to report that the temperature was 20°C in Seattle on September 20. This comes at an increased cost to the attacker in that they must have some knowledge of the context values for the attack to be successful.

3) Replayed/Generated Data.: In this model, the attacker submits data that is technically correct, but does not contribute additional value to the system because this is data that has either been previously submitted or it has been derived from existing data. For example, if a system is trying to create a fine-grained temperature map in a city, an attacker could check a weather website for that city and use the general temperature to generate realistic looking data for very specific locations in that city to collect incentives without actually measuring the temperature in those locations.

IV. Design

We approach the problem outlined in III by building a machine learning model, CAPP, to verify the legitimacy of context packages submitted by crowdsensing participants. CAPP is initially trained on contextual data previously known to be malicious or legitimate and once deployed, will be able to reject crowdsensing submissions based on whether or not their accompanying context package fits into its existing model. We evaluate this approach using data supplied by the Device Analyzer (DA) project II-B.

A. Feature Extraction

In this subsection, we describe the features extracted from the DA datasets and how they will be processed.

For CAPP, an "environment" is indexed by a location-time 2-tuple and is characterized by features such as temperature

[1] https://deviceanalyzer.cl.cam.ac.uk/keyValuePairs.htm

TABLE I: Features extracted from the Device Analyzer datasets

Index	Description
Time:	time when data was collected, recorded in complete ISO 8601 format
Location:	location where data was collected, recorded in latitude and longitude
Numerical Features (Type I)	**Description**
Ambient Temperature:	surrounding ambient temperature measured in Celsius
Relative Humidity:	surrounding relative ambient air humidity measured in percentage
Pressure:	surrounding atmospheric pressure measured in millibar
Light:	surrounding ambient light levels measured in lux
Magnetic Field:	surrounding ambient magnetic field measured in microTesla on three axes (x, y, z)
Gravity:	records magnitude and direction of gravity in m/s^2 on three axes (x, y, z)
Nominal Features (Type II)	**Description**
Wifi:	records hashes of SSIDs of all discovered Wifi access points
Bluetooth:	records hashes of MAC addresses of all discovered Bluetooth devices

recorded within that location over an extended period of time. The set of features chosen from the DA dataset that best characterizes any given environment are summarized in table I. Combined with the 2-tuple index, this results in an initial dataset dimensionality of 14.

The features chosen can be categorized into two different types: (1) those that help establish location outdoors, and (2) those that help establish location indoors. Type 1 features are numerical, and are those that are representative of the weather of a location at a given time. However we acknowledge that the weather might not always be sufficient on its own in establishing the location of a user, especially when they are indoors. For instance, it is not improbable that an office in Houston, Texas, has the same ambient temperature, light levels, and humidity as an office in Seattle, Washington for most of the day. For that reason, we include type 2 features. These are nominal events that are representative of what smartphones can "physically observe", and includes detected Wifi access points and Bluetooth devices.

Each of these features are then grouped into 30-minute time bins to create consistent time intervals and ease computational costs during testing and training. Groupings for numerical features are done by averaging observations within the 30-minute interval whereas groupings for nominal features are done by concatenating observations into a single list. This set of data will be referred to as our **ground-truth dataset**.

B. Fake Data Generation

In this subsection, we discuss how fake contextual data will be generated to augment ground-truth observations. Fake data will be generated in three different ways to simulate attacks from the different attacker models outlined in section III-B.

1) Indiscriminant Data: Contextual data is aribtrarily generated without respecting bounds. Examples include an ambient temperature of -9999°C or randomly generated Wifi SSIDs.

2) Realistic Data: Contextual data is generated within realistic bounds. This means that generated numerical data will be bounded by the lowest and highest observed value for an environment, whereas a combination of observed values will be randomly selected for nominal features. For example, a temperature of 25°C in Seattle on the 25th of December.

3) Replayed/Generated Data: Contextual data is generated by a predictive long short term memory (LSTM model) that has been trained using a subset of the ground-truth observations to, given a continuous observations for an environment, predict the next set of observations for it.

TABLE II: Different types of augmented datasets used to evaluate CAPP

Dataset	Description
Indiscriminant (IDS)	arbitrarily fake data
Realistic (RDS)	realistic fake data
Generated (GDS)	predictively generated data
All-inclusive (ADS)	mix of all categories of fake data

C. Data Augmentation

The augmented datasets that will be used to train and test CAPP are created by evenly combining our ground-truth dataset with fake data. Four different kinds of augmented datasets, whose descriptions are summarized in table II, will be created to evaluate CAPP against the different attacker models in III-B.

In all cases, fake data is introduced into our ground-truth datasets in the following manner. Given a location-time 2-tuple, a random combination of features are selected and their values are replaced with falsified data. For instance, the following datapoint:

$$\text{Date, time, location, temp, humidity}$$
$$20130904,\ 1730,\ 58.775|8.863,\ 21.12,\ 63.469$$

could be falsified by replacing either temperature, humidity or both observations with falsified data. Introducing data this way allows us to better simulate real-life scenarios in which an attacker may not have all feature observations and thus chooses to generate the other missing ones.

V. Evaluation

A. Experimental Settings

In this section, we discuss CAPP's configurations. CAPP uses long short term memory (LSTM) networks in order to be able to recognize dependencies across a dozen environmental features over an extended period of time. Furthermore, we use the sigmoid activation function and binary cross-entropy loss function as we are dealing with a binary classification problem.

To evaluate CAPP's classification performance, we measure resulting F1 and AUC scores. F1 provides insight into how good CAPP is at correctly identifying malicious users while at the same time not incorrectly flagging legitimate context packages as false. Similarly, AUC allows us to determine how good CAPP is at distinguishing between genuine and falsified context data.

Initial test runs with CAPP reveals that increasing the number of neurons in the hidden layer improves the resulting AUC score. However increasing this to beyond 100 provided diminishing returns with respect to performance. We find the optimal number of epochs to be 10, and the optimal batch size to be 16 using the same process.

B. Experimental Decisions

We build three different versions of CAPP to evaluate how the number of features used to model an environment will affect its classification performance:

- CAPP-1 uses only temperature
- CAPP-2 uses temperature and humidity
- CAPP-3 uses temperature, humidity, and pressure

Each version of CAPP will be trained and tested with a 70-30 split of their corresponding ADS dataset variants, then evaluated on their corresponding IDS, RDS, and GDS dataset variants to see how well they hold up against each of our outlined attacker models.

1) Evaluation Hardware: The preprocessing phase as well as the testing and training of CAPP was performed on a system with the following hardware:

- Intel(R) Core(TM) i7-6850K CPU @ 3.60GHz, 12 cores
- 64 GB System memory
- 4 × NVIDIA TU102 [GeForce RTX 2080 Ti]

2) Computational Costs: In this section, we summarize the computational costs of preparing datasets, training CAPP, and asking CAPP to verify a context-package.

The preparation of datasets for use with CAPP was the most intensive process of the project. The total size of raw data supplied by the DeviceAnalyzer project was in the order of 10TB over several thousand compressed files. The feature extraction, normalization, and binning process took approximately 2 weeks, and resulted final augmented datasets within the order of 100MB each.

In contrast, training CAPP only took several minutes for each scenario. Similarly, querying CAPP with a context package yielded results near instantaenously.

VI. Results and Discussion

In this section, we discuss the results of evaluating each CAPP version against all attacker models (i.e. the IDS, RDS, and GDS datasets).

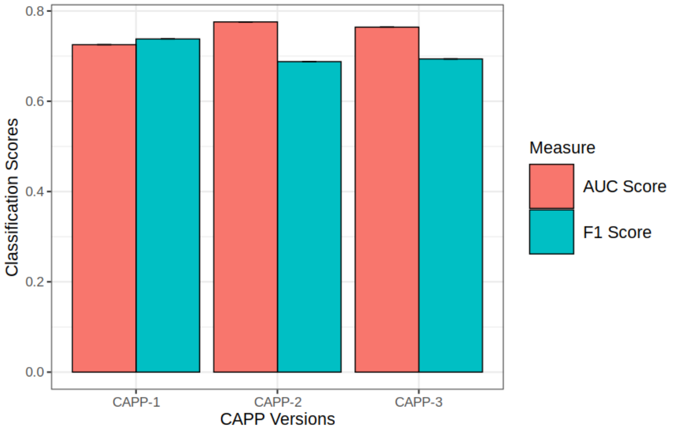

Fig. 2: AUC and F1 scores after evaluating all CAPP versions on ADS

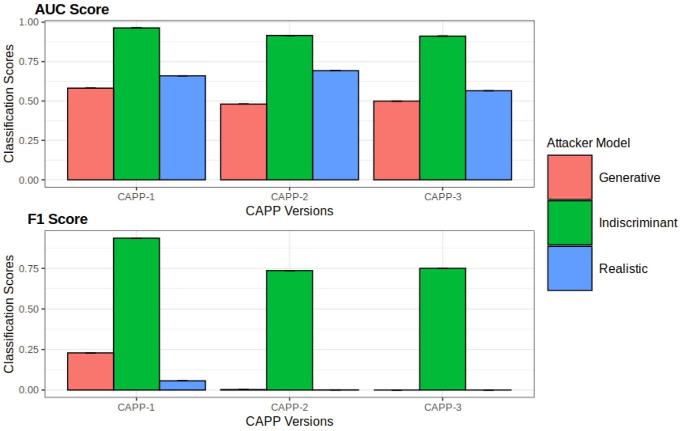

Fig. 3: AUC and F1 scores after evaluating all CAPP versions on IDS, RDS, and GDS

Figure 2 compares the AUC and F1 scores obtained by CAPP-1, CAPP-2, and CAPP-3 after evaluating them on their respective versions of the ADS dataset. The upper-left plot of figures 4, 5, 6 shows their corresponding confusion matrices. We observe that although using more features to describe a particular environment results in CAPP's ability to more reliably distinguish between falsified and legitimate data, it also reduces CAPP's F1 score.

Figure 3 depicts the comparison of two metrics: the upper graph compares the AUC scores each of the CAPP versions obtained evaluating them on their respective variants of the IDS, RDS, and GDS datasets, while the lower graph compares the F1 scores. The remaining plots in figures 4, 5, 6 shows their corresponding matrices.

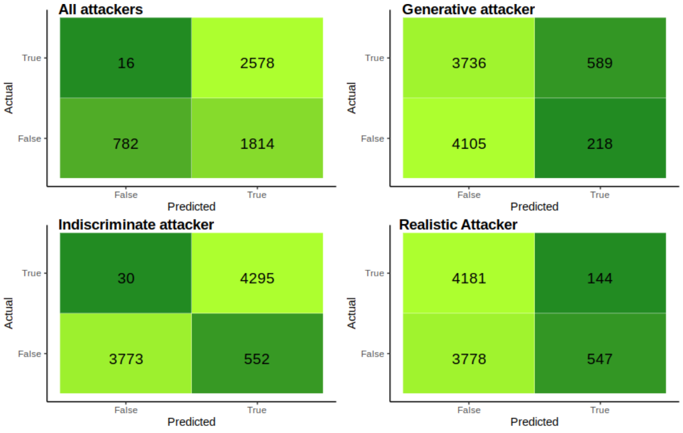

Fig. 4: Confusion matrix for CAPP-1

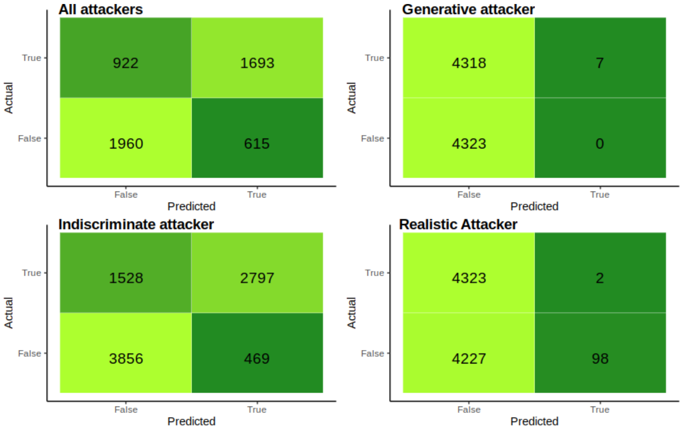

Fig. 5: Confusion matrix for CAPP-2

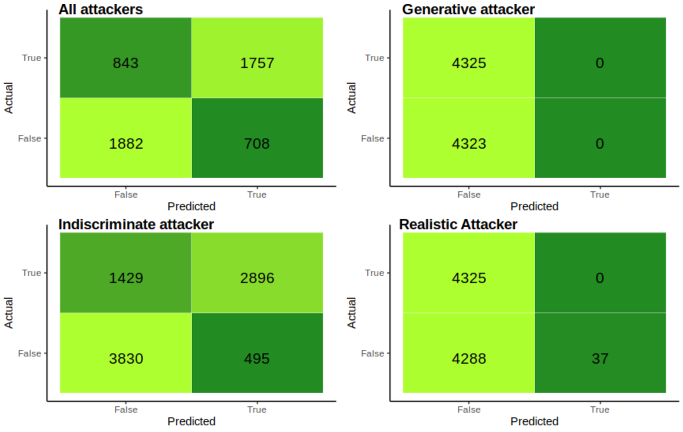

Fig. 6: Confusion matrix for CAPP-3

All three CAPP versions are able to reliably distinguish between indiscriminately falsified data from legitimate data, with the worst AUC score obtained being 0.91 by CAPP-2 and the best score being 0.96 by CAPP-1. In contrast, all CAPP versions performed poorly against the realistic attacker, with the best AUC score being only 0.69 for CAPP-2. These results could be attributed to the low variance for the chosen features. The legitimate temperature observations found in our prepared datasets for instance, had a range of only -7 to 36°C, making the probability of getting realistically randomed data within ±1 of ground truth around 6%. Similarly, neither CAPP versions were able to defeat the generative attacker, with AUC scores ranging between 0.48 and 0.58. However, this is to be expected as it is trivial for a predictive machine learning model to generate data that looks identical to legitimate data; however, it does add a cost to the attacker in that they must be able to train such a generative model for the time and location they are attacking.

The F1 results obtained by all CAPP versions follow a similar trend. CAPP-1 performed well against the indscriminant attacker, obtaining a score of 0.93, whereas CAPP-2 and CAPP-3 performed decently, obtaining scores of 0.73 and 0.75 respectively. However when tested against the generative and realistic attacker, all CAPP versions were unable to defeat the more sophisticated attacks, with both CAPP-2 and CAPP-3 obtaining a score of 0.0 in both cases. Recall that given a legitimate datapoint, we falsify it by only changing the values of some of its features. It is possible that these datapoints consisting of a mix of falsified and true feature observations are confusing CAPP during the training phase, causing it to misidentify input values during the evaluation phase.

VII. Related Work

While there is an abundance of work exploring the area of effective incentive mechanisms, most of it is focused on designing better ways to stimulate participation through the use of mechanisms such as dynamic pricing or monetary coupons [17], [18], [19], [20].

Most efforts to ensure that incentives are distributed fairly tend to focus more on rewarding the right user with their deserved amount rather than prevent dishonest users from accepting rewards. For instance, Zhu et al. proposes using a quality checking module to ensure that only the highest quality submission gets rewarded [21]. However, they neglect to provide a tangible set of metrics for what the module considers "high quality". Similarly, SPPEAR rejects and does not reward submissions that deviate too far from the norm, but does not provide any examples of how that might happen [22]. In both these cases, CAPP has the potential to augment the discussed quality checks by additionally proving that a user has truthfully collected the information via proof of presence.

The notion of using contextual information to verify and establish the location of a user is not new. For instance, Truong et al. proposes using the co-presence of a user and a verifier in order to facilitate zero-interaction authentication [23]. Lawrence et al. proposes using the co-presence of multiple users in order to create a community that can effectively share information [24]. Though works discuss using contextual information as a means to prove copresence, but neither have obtained concrete results to show that this actually works.

VIII. Conclusion and Future Work

In this paper, we demonstrate the feasibility of using a context-aware approach to preventing malicious users from exploiting incentives provided by a crowdsensing system based on attacks that have been observed in real systems. CAPP requires that all users wanting to participate in a crowdsensing system submit, along with the required information, a package containing contextual data that is representative of their surroundings. Using this package, we show that CAPP is able to reliably distinguish between a legitimate participant and a malicious participant whose context package consists of arbitrarily generated data. We note that although CAPP fails against the realistic and generative attacker models, the cost to create context packages in these scenarios for an attacker can be reasoned to be sufficiently high:

- **Realistic Data:** the attacker would have to collect environmental data of a particular location for a sufficient amount of time in order to know what a reasonably generated context package would look like.
- **Generated Data:** in addition to collecting environmental data for a sufficient amount of time, the attacker would also have to train a predictive machine learning model to output a reasonably looking context package.

In either case, given the computational costs of data processing, the cost of obtaining data is high enough to prevent most attacks from occurring.

For future iterations of this work, we plan on incorporating nominal data described in table I to see if that could improve CAPP's classification performance. Furthermore, the version of CAPP described in this paper only looks at a single shot evaluation. We intend to expand the system with a reputation system that monitors users' reported context over time, thus further magnifying the effort required to reliably generate realistic fake data.

IX. Acknowledgments

The work presented in this paper was partially supported by the US/UK Fulbright Cyber Security Award and the CAE Cybersecurity Grant H98230-15-1-0284.

References

[1] "What's Invasive!" 2019. [Online]. Available: https://www.scientificamerican.com/citizen-science/whats-invasive/

[2] N. Maisonneuve, M. Stevens, M. E. Niessen, and L. Steels, "NoiseTube: Measuring and mapping noise pollution with mobile phones," in *Information Technologies in Environmental Engineering*, 2009, pp. 215–228.

[3] P. Mohan, V. N. Padmanabhan, and R. Ramjee, "Nericell: Rich Monitoring of Road and Traffic Conditions Using Mobile Smartphones," in *Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems*, ser. SenSys '08. New York, NY, USA: ACM, 2008, pp. 323–336. [Online]. Available: http://doi.acm.org/10.1145/1460412.1460444

[4] X. Zhang, Z. Yang, W. Sun, Y. Liu, S. Tang, K. Xing, and X. Mao, "Incentives for Mobile Crowd Sensing: A Survey," *IEEE Communications Surveys Tutorials*, vol. 18, no. 1, pp. 54–67, 2016.

[5] L. G. Jaimes, I. J. Vergara-Laurens, and A. Raij, "A Survey of Incentive Techniques for Mobile Crowd Sensing," *IEEE Internet of Things Journal*, vol. 2, no. 5, pp. 370–380, Oct. 2015.

[6] F. Restuccia, S. K. Das, and J. Payton, "Incentive Mechanisms for Participatory Sensing: Survey and Research Challenges," *ACM Trans. Sen. Netw.*, vol. 12, no. 2, pp. 13:1–13:40, Apr. 2016. [Online]. Available: http://doi.acm.org/10.1145/2888398

[7] K. C. Zeng, Y. Shu, S. Liu, Y. Dou, and Y. Yang, "A Practical GPS Location Spoofing Attack in Road Navigation Scenario," in *Proceedings of the 18th International Workshop on Mobile Computing Systems and Applications - HotMobile '17*. ACM Press, 2017, pp. 85–90. [Online]. Available: http://dl.acm.org/citation.cfm?doid=3032970.3032983

[8] A. J. Kerns, D. P. Shepard, J. A. Bhatti, and T. E. Humphreys, "Unmanned Aircraft Capture and Control Via GPS Spoofing," *Journal of Field Robotics*, vol. 31, no. 4, pp. 617–636, 2014. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21513

[9] C. Shepherd, I. Gurulian, E. Frank, K. Markantonakis, R. N. Akram, E. Panaousis, and K. Mayes, "The Applicability of Ambient Sensors as Proximity Evidence for NFC Transactions," in *2017 IEEE Security and Privacy Workshops (SPW)*, May 2017, pp. 179–188.

[10] Steven Le Blanc and Masami M, "Hokkaido man arrested for pretending to visit Aeon shopping centers 2.7 million times," Nov. 2018. [Online]. Available: https://soranews24.com/2018/11/13/hokkaido-man-arrested-for-pretending-to-visit-aeon-shopping-centers-2-7-million-times/

[11] Y. Adegoke, "Uber drivers in Lagos are using a fake GPS app to inflate rider fares." [Online]. Available: https://qz.com/africa/1127853/uber-drivers-in-lagos-nigeria-use-fake-lockito-app-to-boost-fares/

[12] "GPS Spoofing A Growing Problem for Uber." [Online]. Available: http://SolidDriver.com/GPS-Spoofing-A-Growing-Problem-for-Uber

[13] D. Ferreira, V. Kostakos, and A. K. Dey, "AWARE: Mobile Context Instrumentation Framework," *Frontiers in ICT*, vol. 2, Apr. 2015. [Online]. Available: http://journal.frontiersin.org/article/10.3389/fict.2015.00006/abstract

[14] D. T. Wagner, A. Rice, and A. R. Beresford, "Device Analyzer: Large-scale Mobile Data Collection," *SIGMETRICS Perform. Eval. Rev.*, vol. 41, no. 4, pp. 53–56, Apr. 2014. [Online]. Available: http://doi.acm.org/10.1145/2627534.2627553

[15] S. Reddy, D. Estrin, and M. Srivastava, "Recruitment Framework for Participatory Sensing Data Collections," in *Proceedings of the 8th International Conference on Pervasive Computing*, ser. Pervasive'10. Springer-Verlag, 2010, pp. 138–155. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-12654-3_9

[16] F. Restuccia and S. K. Das, "FIDES: A trust-based framework for secure user incentivization in participatory sensing," in *Proceeding of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks 2014*, Jun. 2014, pp. 1–10.

[17] I. Krontiris and A. Albers, "Monetary incentives in participatory sensing using multi-attributive auctions," *International Journal of Parallel, Emergent and Distributed Systems*, vol. 27, no. 4, pp. 317–336, Aug. 2012. [Online]. Available: https://doi.org/10.1080/17445760.2012.686170

[18] B. Kantarci and H. T. Mouftah, "Trustworthy Sensing for Public Safety in Cloud-Centric Internet of Things," *IEEE Internet of Things Journal*, vol. 1, no. 4, pp. 360–368, Aug. 2014.

[19] E. Androulaki, S. G. Choi, S. M. Bellovin, and T. Malkin, "Reputation Systems for Anonymous Networks," in *Privacy Enhancing Technologies*, ser. Lecture Notes in Computer Science, N. Borisov and I. Goldberg, Eds. Springer, 2008, pp. 202–218.

[20] D. Peng, F. Wu, and G. Chen, "Pay As How Well You Do: A Quality Based Incentive Mechanism for Crowdsensing," in *Proceedings of the 16th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, ser. MobiHoc '15. New York, NY, USA: ACM, 2015, pp. 177–186. [Online]. Available: http://doi.acm.org/10.1145/2746285.2746306

[21] X. Zhu, J. An, M. Yang, L. Xiang, Q. Yang, and X. Gui, "A Fair Incentive Mechanism for Crowdsourcing in Crowd Sensing," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 1364–1372, Dec. 2016.

[22] S. Gisdakis, T. Giannetsos, and P. Papadimitratos, "SPPEAR: Security & Privacy-preserving Architecture for Participatory-sensing Applications," in *Proceedings of the 2014 ACM Conference on Security and Privacy in Wireless & Mobile Networks*, ser. WiSec '14. New York, NY, USA: ACM, 2014, pp. 39–50. [Online]. Available: http://doi.acm.org/10.1145/2627393.2627402

[23] H. T. T. Truong, X. Gao, B. Shrestha, N. Saxena, N. Asokan, and P. Nurmi, "Using contextual co-presence to strengthen Zero-Interaction Authentication: Design, integration and usability," *Pervasive and Mobile Computing*, vol. 16, pp. 187–204, Jan. 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1574119214001771

[24] J. Lawrence, T. R. Payne, and D. D. Roure, "Co-Presence Communities: Using Pervasive Computing to Support Weak Social Networks," in *15th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE'06)*, Jun. 2006, pp. 149–156.