Resource Clogging Attacks in Mobile Crowd-Sensing: AI-based Modeling, Detection and Mitigation

by

Yueqian Zhang

Thesis Supervisor : Dr. Burak Kantarci

A thesis
presented to the University of Ottawa
in fulfillment of the
thesis requirement for the degree of
Master of Computer Science

School of Electrical Engineering and Computer Science
Faculty of Engineering
University of Ottawa

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Mobile Crowdsensing (MCS) has emerged as a ubiquitous solution for data collection from embedded sensors of the smart devices to improve the sensing capacity and reduce the sensing costs in large regions. Due to the ubiquitous nature of MCS, smart devices require cyber protection against adversaries that are becoming smarter with the objective of clogging the resources and spreading misinformation in such a non-dedicated sensing environment. In an MCS setting, one of the various adversary types has the primary goal of keeping participant devices occupied by submitting fake/illegitimate sensing tasks so as to clog the participant resources such as the battery, sensing, storage, and computing. With this in mind, this thesis proposes a systematical study of fake task injection in MCS, including modeling, detection, and mitigation of such resource clogging attacks.

We introduce modeling of fake task attacks in MCS intending to clog the server and drain battery energy from mobile devices. We creatively grant mobility to the tasks for more extensive coverage of potential participants and propose two take movement patterns, namely Zone-free Movement (ZFM) model and Zone-limited Movement (ZLM) model. Based on the attack model and task movement patterns, we design task features and create structured simulation settings that can be modified to adapt different research scenarios and research purposes.

Since the development of a secure sensing campaign highly depends on the existence of a realistic adversarial model. With this in mind, we apply the self-organizing feature map (SOFM) to maximize the number of impacted participants and recruits according to the user movement pattern of these cities. Our simulation results verify the magnified effect of SOFM-based fake task injection comparing with randomly selected attack regions in terms of more affected recruits and participants, and increased energy consumption in the recruited devices due to the illegitimate task submission.

For the sake of a secure MCS platform, we introduce Machine Learning (ML) methods into the MCS server to detect and eliminate the fake tasks, making sure the tasks arrived at the user side are legitimate tasks. In our work, two machine learning algorithms, Random Forest and Gradient Boosting are adopted to train the system to predict the legitimacy of a task, and Gradient Boosting is proven to be a more promising algorithm. We have validated the feasibility of ML in differentiating the legitimacy of tasks in terms of precision, recall, and F_1 score. By comparing the energy-consuming, effected recruits, and impacted candidates with and without ML, we convince the efficiency of applying ML to mitigate the effect of fake task injection.

Acknowledgements

I would like to express my sincere gratitude to my supervisor Dr. Burak Kantarci for his continuous support throughout my Master study. He opened the door for me to a novel research field, helped me to find a suitable research topic, and guided me to complete my work successfully. I remember how Dr. Burak Kantarci encouraged me when he advised me to switch to thesis-based Masters. Thanks for this decision, I harvest more than a series of papers that contribute to this thesis. The honest, careful and hard working spirit of Dr. Burak Kantarci will always inspire me.

I would also like to thank Dr. Murat Simsek for his help and contributions to our papers and all the NEXTCON lab members for their helping and accompanying.

I would thank Dr. Andrew Walenstein and Dr. Andrew Malton from BlackBerry for giving the opportunity to gain experience during my internship at Blackberry. I would also thank Mr. Hossain Ghadimi for his helping during my co-op terms in Persistent System.

Finally, I must express my very profound gratitude to my parents for supporting me in every step of my life and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them. Thank you.

Table of Contents

List of Tables

List of Figures

Abbreviations

AI Artificial Intelligence 3, 6, 12, 18, 82, 83, 85, 86

ARCing Adaptive Reweighting and Combining 72

ATM Adaptive Transmission Map 9

AUC area under curve 19

BMU Best matching unit 48–51

CNN convolutional neural network 18

CPU central processing unit 2, 7

DAP DoS-Resistant Authentication Protocol 12

DoS Denial of Service x, 2–4, 10–12, 19–22, 25, 28, 34, 83

FN False Negative 29, 30, 39, 75

FP False Positive 29, 30, 39, 75

GB Gradient Boosting iii, 4, 5, 19, 25, 26, 28–30, 32, 34, 38, 71–73, 75, 76, 81–84

GLOSA Green Light Optimal Speed Advisory 8

GPS Global Positioning System 1, 8

HMAC Hash-based Message Authentication Code 10

IoT Internet of Things 1, 7, 83

LCR legitimate task completion rate 42

LSTM-CNN Long Short Term Memory-Convolutional Neural Network 11

MCS Mobile Crowdsensing iii, x, 1–23, 27, 34, 35, 37, 41–45, 47, 49, 51, 52, 55–57, 59, 60, 65, 68, 71, 72, 76, 81–86

ML Machine Learning iii, 3, 4, 6, 20, 21, 25, 28, 40, 42, 43, 83–86

PCS Piggyback Crowdsensing 16

PSI Pollutants Standard Index 9

QoI Quality of Information 16

RF Random Forest iii, 4, 25, 26, 28–30, 34, 83

SOFM self-organizing feature map iii, vii, 3, 5, 6, 43–46, 48–52, 55–57, 59, 60, 62–68, 71, 81, 84, 85

SONATA Social Network-Aided Trustworthiness Assurance 16

TN True Negative 29, 30, 39, 75

TP True Positive 29, 30, 39, 75

TSCM Trustworthy Sensing For Crowd Management 14–16

UCLA University of California, Los Angeles 17

ZFM Zone-free Movement iii, 4, 35, 52, 66, 68, 74, 75, 77, 80, 84

ZLM Zone-limited Movement iii, 4, 35, 40, 52, 67, 68, 74, 75, 80, 84

List of Symbols

$Dist(X, W)$ Distance function for two dimensional inputs X and 2D node's weight vector W. 49

$L(y, f(x))$ The loss function in Gradient Boosting algorithm. 73

PN_d Pre-defined neuron distance. xi, 57, 58

$W^{(i+1)}$ Updated weight vector when adjusting weight vectors of BMU's neighbourhood in SOFM . 50

W The neuron's weight vector in SOFM. 50

X The current 2D input vector in SOFM. 50

F_1 An overall measure of accuracy, which is calculated as the harmonic mean of precision and recall. iii, 3, 5, 25, 39, 75, 82, 85

$f(x)$ A goal function in Gradient Boosting. 73

Max_epoch Maximum epoch, also called iteration limit. 50

$P_{Corrected}$ The corrected position in the algorithm to determine the real next movement destination with consideration of task movement patterns. 35

$P_{Estimated}$ Estimated position in the algorithm to determine the next movement destination without considering any limit. 35

$\phi_{(x)}$ A surrogate loss in Gradient Boosting. 73

$\Phi(i)$ The effect of a node's learning distance from the BMU in SOFM. 50

r_{rec} Recruitment radius of a task 21, 24, 39

$R(i)$ The learning rate which decays over time in SOFM. 50

r_{mov} Movement radius of a task ix, 21, 24, 29, 31, 32, 39

σ_0 The initial value of radius when determine the width of the BMU's neighbourhood. 50

$\sigma(i)$ The width of the neighbourhood function in self organizing feature map. 50

σ_m The step length determined in line search procedure in Gradient Boosting. 73

$w/0$ Without. 42

$w.$ With. 42

Publications of the Candidate During MCS Studies

Publications that are the direct outcomes of the thesis:

- Zhang, Y, et al. "Adversarial Region Based-Self Organizing Feature Map for Modelling of Fake Task Attacks in Mobile Crowdsensing." **(Submitted to ACM Transactions on Cyberphysical Systems)**
- Zhang, Yueqian, and B. Kantarci. "AI-based Security Design of Mobile Crowdsensing Systems: Review, Challenges and Case Studies." In 2019 IEEE International Conference on Service-Oriented System Engineering (SOSE), pp. 17-1709. IEEE, 2019.
- Zhang, Y, M. Simsek, and B. Kantarci. "Machine Learning-based Prevention of Battery-oriented Illegitimate Task Injection in Mobile Crowdsensing." In Proceedings of the ACM Workshop on Wireless Security and Machine Learning, pp. 31-36. ACM, 2019.
- Zhang, Y, M. Simsek, and B. Kantarci. "Ensemble Machine Learning for the Mitigation of Adversarial AI-driven Fake Task Submission in Mobile Crowdsensing." In 2020 IEEE International Conference on Communications (ICC), IEEE, 2020. **(Submitted)**
- Zhang, Y, M. Simsek, and B. Kantarci. "Self Organizing Feature Map for Fake Task Attack Modellingin Mobile Crowdsensing." In 2019 IEEE Global Communications Conference (GLOBECOM), IEEE, 2019.

Publications that are contributed and related to the thesis:

- Z. Chen, Zhang, Y, M. Simsek, and B. Kantarci. "Deep Belief Network-based Fake Task Mitigation for Mobile Crowdsensing under Data Scarcity." IEEE International Conference on Communications (ICC), IEEE, 2020. **(Submitted)**
- Ankkita Sood, M. Simsek, Zhang, Y, M., and B. Kantarci.. "Deep Learning-Based Detection of Fake Task Injection in Mobile Crowdsensing." In 2019 IEEE Global Conference on Signal and Information Processing (GlobalSIP), IEEE, 2019. **(Accepted)**

Chapter 1

Introduction

The evolution of communication technologies during the past ten years fuels the emergence and development of mobile devices. As a typical representation, smartphones are acting as a daily necessity of citizens, providing convenience to message interchange, social contact, and entertainment. Reported by Newzoo. [1], there are approximately three billion smartphone users around the world in 2019, and the number is expected to reach 3.8 billion by 2021. Utilizing the information obtained from multi-functional sensors (e.g., camera and GPS) embedded in mobile devices, a paradigm called Mobile Crowdsensing (MCS) has emerged as a new form of Internet of Things (IoT). As the footstone of MCS system, the mobile device owners shuttle in the cities, providing wide coverage and rich varieties of data in the sensing process using human intelligence. Besides, the participation of human saves the system from expensive cost on the establishment and maintains when comparing with the traditional sensing networks.

A complete MCS system contains three necessary parts, including a service platform, a group of end-users who request data by submitting tasks to the MCS platform, and participants who use their mobile devices to collect data. A task usually specifies what information needed, the location of the task, the information collection time, task duration, and some other requirements. For instance, if a community wants to study the noise pollution in a city, it can assign tasks to the users around the city to collect noise level data by exploiting the smartphone's built-in microphones. The MCS platform has some recruitment policies to determine how to assign tasks to mobile device owners for data collection. Once mobile device owners accept tasks, they collect sensing data and send it back to the platform for compensation. In the thesis, each task has a location, and it

[1] A leading global provider of games and mobile analytics.

moves a little for broader coverage of potential participants, and mobile devices owners need to move to the area of 100 meters from the task to accomplish it.

The application of MCS in the smart city has been studied in a lot of work, including traffic monitoring and management, noise detection, and air quality monitoring [72, 18, 75, 78, 12, 81, 94, 56]. Specifically, data harvested from crowdsensing enables detection of free parking spots in smart cities [78, 12] and traffic management [18, 75]. Noise obtained from crowdsensing facilitate the construction of noise maps in urban areas, which helps authorities (e.g., government) to learn about the air conditions. Similarly, MCS plays a fundamental role in air quality monitoring, where air quality maps can be created based on the information from participants.

Although MCS is considered to be a popular topic, the current issues hamper the full acceptance of MCS. From the standpoint of users, privacy concerns, and the energy limitation of their devices may decrease their interest to participate in MCS campaigns. Specifically, data assembling from personal devices may reveal sensitive information such as location, action, and even identity. Moreover, the mobile devices naturally have a limited battery that needs recharging from time to time, constrained computing capability due to the CPU, and bandwidth related to network states.

When the users are enlisted to the MCS campaigns, the data trustworthiness is still a problem. Firstly, due to the dynamic movement of users, the diverse performance of sensors, and the uncertain stability of data exchange channels, the accuracy of data cannot be guaranteed. Secondly, many incentive mechanisms have been applied to MCS and among which the typical one is the payment-motivated mechanism where the system pays for collected data from users[77, 55, 32, 49, 122]. Driven by self profit, dishonest users gain personal revenue by submitting untrustworthy data, which further compromises the reliability of knowledge analyzed from the sensing information. Besides, other kinds of attacks (e.g., Denial of Service (DoS) attack, Sybil attack) are emerging, which makes the security issues become a primary concern in MCS system establishment.

To this end, authentication is widely used in privacy-preserving mechanisms[47, 40, 14], where the data is encrypted before transferring to the third parties. Another promoted approach to protect participants' privacy is anonymization mechanism[76, 93, 70, 50], which aims to discard the information of users' identities and avoid sensitive data revealing to any other users. To deal with resource limits of mobile devices, studies focusing on energy-efficient MCS solutions have been proposed from different perspectives [96, 51, 110, 13, 79, 42, 89, 104, 51, 120, 63]. As solutions to suppress unreliable data from users (also called fake sensing attacks), the authors introduced game theory and reputation of users in [105, 11, 107]. It is also noticeable that researchers put efforts into addressing attacks,

among which plenty of work is on solutions for DoS attacks[92, 115, 16, 117, 34, 27, 65].

1.1 Motivation

As attacks are common security issues existing in MCS due to the openness and dynamics of MCS. Although researchers have been working on solutions, variations and brand new attacks are appearing. What is more, the limited resource of mobile devices, especially battery constraint adds another challenge awaiting for resolutions. In a MCS setting, a type of adversary initiates a DoS-like attack to keep participant devices occupied by submitting fake/illegitimate sensing tasks to clog the participant resources such as the battery, sensing, storage, and computing. In this resource clogging attacks, malicious end users act as attackers who inject fake tasks through the MCS platform towards MCS servers and participants with aims of clogging the resources of MCS servers and leading resource famine in participants' mobile devices. If this kind of resource clogging attack exists in the MCS system for a long time, the platform will start losing participant willingness to participate in MCS campaigns. To maintain the durability of the MCS system, the system should detect illegitimate tasks before assigning them to any participant. Recently, AI-assisted or AI-driven cybersecurity has become an attractive topic with the advent of artificial intelligence and particularly the widespread use of machine learning techniques in various areas, including security [73, 2]. As AI-based techniques have shown to be effective under certain contexts, this thesis aims to tackle the possibility of the AI-based adversarial model and AI-based security provision for MCS.

In this thesis, we investigate and address the resource clogging attack in MCS. Our motivation to use AI is two-fold: 1) to forecast the legitimacy of tasks, and 2) to model an ambitious adversary. Specifically, we design the attack models and introduce two movement patterns of tasks with consideration of the attack areas. For the sake of secure MCS platform, we add ML methods into MCS server to eliminate the fake tasks and make sure the tasks arrived at the user side are legitimate tasks. We have proven the feasibility of ML in differentiating the legitimacy of tasks in terms of precision, recall, and F_1 score (also called F-score). By comparing the energy-consuming and effected recruits with and without ML, we convince the efficiency of applying ML to mitigate the effect of fake task injection. Since the development of a secure sensing campaign highly depends on the existence of a realistic adversarial model. With this in mind, this thesis introduces self-organizing feature map (SOFM) for fake task attack modeling from the standpoint of attackers. Through numerical studies, to maximize the number of impacted participants and recruits according to the user movement pattern of these cities. All the simulations

are completed based on Crowdsensim simulator, which is a practical tool using real street maps and applying user mobility patterns on the street networks [19]. The contributions of the thesis are listed in detail in the following Section 1.2.

1.2 Contributions

We have accomplished an integrated study on a severe security problem (i.e. fake task injection) in MCS in the thesis. By exploring the vulnerabilities and challenges existing in MCS, we reconsider the possibility of DoS-like attack with additional purposes, which we formed the fake task injection attack. From the standpoints of both defenders and attackers, we designed the defense mechanism and attack strategy, respectively. Our contributions in this thesis can be summarized as follows.

1. **Investigation of ensemble machine learning against fake task injection:** We have investigated machine learning-based defenses against empirically designed fake task injection attacks in MCS. This empirical energy-oriented model for clogging attacks that inject fake tasks has two malicious motivations: 1) Clogging the MCS server (similar to a DoS attack), and 2) Draining the battery of mobile smart devices of MCS participants. Based on the attack model, we design task features and create structured simulation settings that can be modified to adapt different researching scenarios and research purposes. In our work, two machine learning algorithms, Random Forest and Gradient Boosting are adopted to train the system to predict the legitimacy of a task, and we proved Gradient Boosting to be a more promising algorithm with the performance of up to 0.95 overall accuracy and 0.9 F-scores for both illegitimate tasks and legitimate tasks. Besides, we have investigated two ML strategies and proven that day-after-day learning can provide higher performance in the task legitimacy decision. In contrast, per-day learning will require less storage due to the size difference of training data stored in the system.

2. **Introduce and study the impact of different fake task movement models:** We creatively grant mobility to the tasks for broader coverage of potential participants and propose two take movement patterns, namely Zone-free Movement (ZFM) model and Zone-limited Movement (ZLM) model. Under the two scenarios, we test the impact of ML-based detection and mitigation of attack impact by analyzing energy savings and the decreased size of impacted recruit population with the existence of ML-based prevention of illegitimate task submission. The results confirm

the improvement in the battery saving and affected recruit at the expense of a slight decrease in the completion rate of legitimate tasks.

3. **Adversarial AI-based design of fake task injection:** From the standpoint of adversaries, we have introduced adversarial AI-based modeling of fake task attacks and evaluate their impacts. We apply self-organizing feature map (SOFM) to find the locations of the aggregated population around the cities using users' routine as input. Our simulation results verify the SOFM-based attack is more potent than randomly-selected attack regions. To find the optimal attack effect, we analyze the impact of adversarial tasks under different SOFM topology by comparing the affected participants and battery drain of the participated device, by which we validate the magnified effect of SOFM-based fake task injection. More importantly, we present the emergency of defense mechanisms against the fake task attacks by analyzing the significant impact on candidates of MCS campaigns. The simulation results have shown the SOFM structured attacks can affect up to 46% of the participants and up to 37% of the recruits under various SOFM topologies. Furthermore, SOFM-based attack models can increase the energy consumption in the recruited devices by up to 39% due to the illegitimate task submission.

4. **Machine Learning-based mitigation of adversarial AI-driven fake task attacks:** We present machine learning-driven mitigation of SOFM-based fake task injection attacks in MCS systems. To this end, we model and train Gradient Boosting, which is validated as a promising algorithm in the investigation of ensemble machine learning against fake task injection, to identify and filter out illegitimate tasks injected/submitted to the MCS platform. Specifically, Gradient Boosting-based ensemble learning can achieve up to 98.9% overall accuracy, 99.4% legitimate score, and 94.6% illegitimate score. With Gradient Boosting embedded in the MCS system, the majority of the fake tasks can be eliminated, which leads to around 20% battery capacity savings for the participants. The experiments validated the efficiency, effectiveness, and emergency of an integrated task legitimacy assessment module in terms of mitigation of the impact on the candidate's population and recruit.

1.3 Structure of the Thesis

Organization of this thesis is as follows.:

Chapter 2 represents the background and literature study: Section 2.1 gives an overview of the origin and backgrounds of mobile crowdsensing. Section 2.2 discusses MCS applica-

tions in different areas. We summarize privacy threats and privacy-preserving solutions in Section 2.3. In the following Section 2.4 and Section 2.5, we present secure MCS solutions and solutions for MCS data trustworthiness respectively. Section 2.6 explains the current research on resource limits in MCS and Section 2.7 presents some incentive mechanism in MCS. In Section 2.8, we present a brief overview of AI-based cybersecurity solutions.

In Chapter 3, we propose a machine learning-assisted prevention of fake task injection in MCS. Section 3.1 introduces the existed attack and proposes the solution. Section 3.2 explains the machine learning embedded MCS system in details. Section 3.3 describes the simulation settings in our experiments and machine learning performance under different scenarios. Section 3.4 introduces two different task mobility models and investigate their impacts on the mitigation.

In Chapter 4, we design a self-organizing feature map (SOFM) modeling fake task injection to optimize the attack effect: Section 4.1 introduces the motivation and design of the modeling. Section 4.2 explains the simulation process, providing a better understanding of the steps taken in the simulations. Section 4.3 evaluates the attack under fixed number of neuron-based SOFM model in terms of energy consumption, impacted recruit and users. Section 4.4 presents an extended work on attack evaluation under different SOFM topology.

In Chapter 5, we present an ensemble machine learning for the mitigation of adversarial AI-driven fake task submission which is introduced in Chapter 4. Section 5.1 introduces the problem and proposes the solution. Section 5.2 reviews the ML-driven mechanism against fake task injection. Section 5.3 explains the mitigation effect of adversarial AI-driven fake task submission.

Finally, in Chapter 6, we wrap up the thesis with our final comments, discuss the issues that still need to be addressed and give research directions for the researches in this field.

Chapter 2

Background and Literature study

2.1 Mobile Crowdsensing Background

Mobile Crowdsensing (MCS) emerges as a combination concept of participatory sensing[7] and opportunistic sensing[52], which is formed by Ganti et al. in [22] to represent a extensive concept of community sensing paradigms. The mobile devices and their owners are the basis for the construction of MCS systems. Driven by the increasing performance requirements from consumers (e.g., high-megapixel cameras, multi-core processors), modern mobile devices are becoming advanced minicomputers. As a typical representative, smartphones are embedded with precise sensors for information assembling, high-performance CPU for computing, and extending storage space. Besides, the development of communication technologies, especially the rapidly developing cellular network technology, accelerates the real-time data exchange between mobile devices and the Internet. All the smart features of mobile devices combined with human intelligence, pave the path for MCS to grow up to a popular branch of Internet of Things (IoT)[9].

From the first time coining of MCS, it has been applied to data assembling in many areas in real life, especially these applications need data of a large area and of timeliness. However, the adoption of MCS has not always been smooth. In the last years, research communities pointed out the challenges and opportunities MCS confronts. Privacy, security, and trust consist of three main concerns of establishing a stable MCS system. Resource limits in mobile devices (e.g., battery, computing) also hamper the wide adoption of MCS applications. Due to the above factors, the users may lose interest in participating in MCS campaigns, which lead the MCS to find the appropriate incentive mechanisms to attract users' active involvement.

2.2 Mobile Crowdsensing Applications

In this section, we classify the MCS applications into different categories according to their purposes, and we describe some most related works in each application category.

2.2.1 Traffic Monitering

In [74], the authors present a system named Nericell, which uses the information collected from sensing components(e.g., accelerometer, GPS) installed in mobile smartphones to estimate the condition of cars such as the collision, honking, and braking.

MCS in traffic monitoring has two challenges, including energy consumption required by sensing the locations of cars and imprecise position sampling. To this end, Thiagarajan et al. present a real-time traffic estimating system called VTrack in [100], in which a hidden Markov model is used to build a model for traffic on road segments and thus find the possible location given inaccurate sensing coordinates.

SignalGuru is a service designed in [48] to detect traffic signals using cameras embedded in the mobile phones from crowdsensing. SignalGuru aggregates the camera-sensing information and predicts the traffic signal schedule according to the patterns retrieved from the sensing data, which facilitates Green Light Optimal Speed Advisory (GLOSA), assisting drivers in controlling vehicle speed and avoiding sudden brakes.

In the article [31], the authors propose a system SmartRoad, a system based on GPS sensing data from smartphones in the vehicles to achieve accurate detection of traffic lights and stop signs on the roads around the city. Their experiments on volunteer vehicles show their system performs with high efficiency, robustness, and effectiveness besides its feasibility to be adapted to some application requirements.

2.2.2 Noise Detection

Noise pollution in the urban area has been drawing global attention. Obtaining noise maps in the urban area helps authorities to know about the air pollution condition and propose responding solutions to mitigate the noise issue. In the paper [94], the authors present an Android participatory sensing application to collect noise data from users' locations. All the data will be transferred to da_sense, a platform where all the data is processed to noise maps. Rana et al. design a system to build noise maps based on insufficient and

casual data obtained from participatory sensing in [90]. However, the users only act as data collectors in these works.

In the research [119], the authors present an MCS-based application on noise monitoring, collecting data via smartphones owned by students. What's interesting in this work is the integration of the teaching purpose that students can use the application to complete some experiments on acoustics courses. The work in [118] introduced a Web application capable of providing suggestions on the mitigation of noise when applying different Noise Reduction Interventions.

2.2.3 Air Quality Monitoring

It is noticeable that air quality has been a topic discussed not only by governments and meteorological associations but also by the public because it is related to every human's health. MCS-based air quality monitoring is a more feasible solution comparing with traditional air pollution monitoring.

In the paper [56], the authors present SecondNose, a crowdsensing service for air-condition monitoring running on the volunteers' smartphones coupled with air pollution sensors to collect multiple proxies (e.g., Carbon Monoxide, air press) in an Italian city. However, the approach requires users to carry an additional sensor with them. Besides, the accuracy and quality of the sensing data are not guaranteed. An MCS-based mobile application proposed in [80] only relies on the smartphone itself as long as it is equipped with the camera. The application, namely AirTick, takes advantage of deep learning assistant image recognition to judge air pollution. They adopted Adaptive Transmission Map (ATM) proposed by Levin et al. [57] to extract haziness from a photo taken by users, which is used as input of the deep neural network to predict Pollutants Standard Index (PSI). Their experiments provide approximately 80% accuracy for the photo taken during daytime and night.

2.3 Privacy in Mobile Crowdsensing Systems

Threats concerning privacy, reliability, and availability lead to the main security challenges in MCS, and related studies span all of these areas [116]. In this section, we investigate each of these areas individually, report existing works, and point out future directions and opportunities. It is worth noting that a typical threat model of an MCS system considers internal and external attackers: the former involves participants aiming at disinformation

at the MCS servers whereas the latter considers jamming and Denial of Service (DoS) type of attacks that aim at resource clogging in the network as stated in [24].

2.3.1 Privacy threats in Mobile crowdsensing

Layla et al. classify the adversarial entities into two groups, including passive (semi-honest) entities and malicious entities, and investigate privacy threats when participants sharing information with these entities [84]. The passive entities are considered as these entities exploit data from participants for knowledge assembling, and malicious entities are defined as these entities with the malicious purpose of breaching the privacy of participants actively. According to their classification, the process of a participant accepting a task, completing the task, and reporting the sensing data may reveal information such as location, time, and participant's preference. It leads to a task-tracing attack where a passive entity trace a specific task accepted by a participant to excavate the private information [97] and location-based attack where location-based de-anonymization attacks are involved in disclosing one's identity [21].

Besides these two types of attacks, malicious attackers also initiate malicious tasking attacks and collusion attacks [84]. If vicious entities create tasks that require specific participants (e.g., the device has a unique sensor), then the tasking itself potentially disclose sensitive information of the user, which is also called narrow tasking [97]. In collusion attacks, several vicious end-users cooperate unlawfully to de-anonymize participants' identities and deceive their private information. As another format of the collusion attack, one entity might arrange the data from multiple applications and combine them to find valuable information [29].

2.3.2 Privacy-aware, privacy-preserving and privacy-respecting Mobile Crowdsensing solutions

The authors in [40] proposed AnonySense, a privacy-preserving system with a framework that enables anonymous tasks receiving and sensing data reporting. Before report aggregation, users can report their locations as a geographical area instead of exact coordinates using the tessellation-based technique. Besides, the time is also automatically blurred via statistical k-anonymity.

In [58], an efficient and secure protocol is designed to obtain statistics in mobile sensing while ensuring the privacy of the participants. Crowdsensed data is encrypted by Hash-based Message Authentication Code (HMAC) through additive homomorphic encryption

before being submitted to the aggregator nodes. Thus, although there are no pre-existing trust mechanism between the MCS platform, aggregator, and the users, an aggregator node is prevented from obtaining the original readings or intermediate information from participants.

To acquire usable data from data owners while preserving their privacy in mobile crowdsensing campaigns, Lyu et al. [67] presented a two-stage randomization-based scheme against Bayesian estimation and independent component analysis attacks in each stage. The authors also proposed a Long Short Term Memory-Convolutional Neural Network (LSTM-CNN) model to distinguish activities, which relatively improved the performance of standalone LSTM, CNN, and a Deep Belief Network. Indeed recovery resistance of the proposal under varying probability distribution of the original data can be further investigated, and further attack types can also be considered in addition to the Bayesian estimation and independent component analysis attacks.

2.4 Secure Mobile Crowdsensing solutions

Security solutions for MCS vary depending on the considered threats. In this subsection, we group security solutions under three categories: 1) solutions against Denial of Service (DoS) attacks in MCS and large scale sensing systems, 2) security mechanisms against Sybil attacks in MCS systems, and 3) security solutions against other attacks (e.g. tag forgery collusion attacks, etc.), as shown in Figure 2.1.

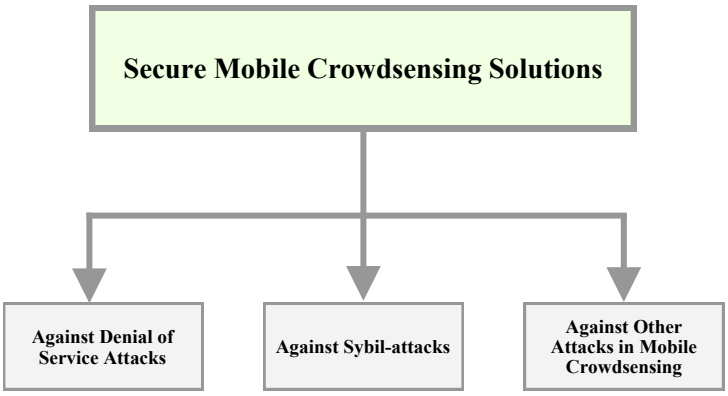


Figure 2.1: Secure mobile crowdsensing solutions against different kinds of attacks

2.4.1 Security Mechanisms against Denial of Service Attacks

In a DoS attack, attackers aim to flood the server with overloaded illegitimate service requests and thus make the resource unavailable. Solutions targeting DoS attacks in MCS specifically confront the challenge of limited resources. The first lightweight user authentication protocol to mitigate DoS attack is called μTESLA [83], and based on it the authors' proposed DoS-Resistant Authentication Protocol (DAP) [92], where a hash function is built to shorten messages and thus reduce the space requirement. Two protocols have been proposed in [59]: 1) Protocol for fault-tolerance, and 2) DoS-resistant protocol as countermeasures against DoS attacks.

The jamming attack, as a typical type of DoS attack, injects fake traffic patterns to interrupt service provided to legitimate users. To this end, game theory is widely introduced [115, 16, 117, 34]. Among these analyses, the authors in [34] extend the Stackelberg game to build a decision-making framework, and two case studies are discussed for potential applications based on that. These two studies highlight the solutions on incomplete information and dense wireless networks, respectively. Recently, Artificial Intelligence (AI) techniques are becoming alternative solutions to control jamming. For example, Han et al. [27] develop an anti-jamming communication scheme with two dimensions based on a deep convolutional neural network. The schema helps the secondary user to find the optional policy for communicating without any knowledge of jamming. In [65], the authors design a deep reinforcement learning algorithm that utilizes the spectrum waterfall for obtaining the best anti-jamming strategy. In [99], the authors consider DoS-like attacks that aim at interrupting the routing procedures and interfere with the communication on the network layer and a multi-dataflow topology scheme is proposed by the authors to address the attacks.

2.4.2 Security mechanisms against Sybil-attacks

By manipulating fake identities, Sybil attack emerges as a new topic in MCS. In the study of Chang et al. [10], a cloud-based schema (CbTMS) was formed to assess and manage trust score and catch suspicious Sybil identities. A series of features involving topology and traffic of MCS nodes are monitored for checking whether the node is compromised. The simulation results proved the efficiency of Sybil nodes detection with promising lower energy consumption comparing with another routing protocol.

Jiang et al. [35] introduced two MCS system models by pushing crowdsensing tasks into a social network to improve participation. By aiming at delay-sensitive campaigns with

effective Sybil detection, the authors designed two incentive mechanisms that formulated and ran reverse auctions. The proposal was shown to perform with promising truthfulness and efficiency of the diffusion. On the other hand, the authors showed that these benefits were at the expense of run time and social cost.

2.4.3 Security solutions against other attacks in Mobile Crowdsensing

To address the location-related security threats in indoor MCS, where reference tags enable accurate location information, Xu et al. categorized attacks under three classes: tag forgery, tag misplacement, and tag removal [112]. Upon introducing location-based fingerprints, a truth discovery algorithm is called to detect tag forgery. Through the truth discovery algorithm, which uses visiting patterns, tag misplacement, and removal can be identified. The proposed approach can be further improved under specific sophisticated scenarios.

To prevent collusion attacks towards MCS participants, Ji et al. studied whether an incentive mechanism can cope with collusion, or it can be protected against profit trading [33]. In a situation where participants are considered to be incentivized by the MCS platform, the authors proposed a new incentive method to meet these objectives by formulating a game. The proposed game requires all users to claim their costs truthfully when participating in MCS campaigns, which is a possible improvement over the proposed promising incentive mechanism.

2.5 Solutions for Mobile Crowdsensing data trustworthiness

2.5.1 Solutions against Fake Sensing Attacks

Fake sensing attack refers to the injection of unreliable and even faked sensing information from selfish users who want to gain benefits since the reward mechanism is applied to MCS with aiming to stimulate users' participation into MCS campaign.

To suppress fake sensing attack, Wang et al. [105] design a dynamic reputation management protocol embedded with a trust assessment algorithm which calculates the level of reliability and correctness of sensing data from a user based on his or her reputation

level and contextual factors without compromising the user's anonymity. In [11], an online auction mechanism based on truthful incentive is reported. The mechanism tries to solve the ex-post service quality in the stage of winner selection since the quality of information for users is unknown before they submit the sensing data. Taking dynamic user arriving into account, the designed system choose the winner by the knowledge of expected service qualities and auction price of upcoming users.

Another persuasive study is reported by Xiao et al. in [108], the authors designed an incentive mechanism by formulating a Stackelberg game in MCS where an MCS server was assigned the lead role. Furthermore, the authors provided a deep Q-network-based MCS payment strategy that combines deep learning with Q-learning to ensure speeding up the learning process. The proposed strategy was designed for a use case on real-time traffic monitoring through MCS campaigns. Here, the fundamental assumption was that the data mining algorithms on the server side were of high quality, and thus sensing accuracy evaluation was accurate in spite of noise due to the involvement of a huge number of users and sensor arrays.

Most recently, the authors in [3] create an architecture for anti-fake sensing by taking advantage of Blockchain concepts, which enables an analysis of data quality history submitted by participants and their behaviors. Besides, the award is adjusted according to the analysis results (e.g., behavioral score) to encourage trustworthy data uploading and restrain fake sensing.

2.5.2 Solutions against data poisoning

In real-time applications, users may submit sensed data at various quality levels. With this in mind, Jin et al. [36] proposed a reputation-based multi-auditing algorithm that exploits temporal reputation based on tasks alongside a truth inference technique (which is also reputation-driven) coupled with the performance-driven payment process. The study considers both rational and irrational participants and introduces certain methods to determine the rational workers' updating of strategies. The multi-auditing algorithm with reputation awareness can be extended to multi-task crowdsensing campaigns where no preconditions exist that necessitate the requesters to be always trustworthy.

For public safety use cases, Kantarci and Mouftah presented a crowd management scheme called Trustworthy Sensing For Crowd Management (TSCM). TSCM maintains a dynamic database of the reputation scores of mobile users [39], and it builds on the user-centric MSensing auction in [114] to incentivize participants. TSCM extends its predecessor by introducing the reputation-awareness and trustworthiness of the users of the

mobile devices. The performance study carried out by the authors was based on highly accurate readings of the built-in sensors of smartphones. The authors also introduced the disinformation probability metric, which stands for the probability of a malicious participant get rewards while the participant has sent false sensor readings to the MCS platform. It was shown by the results that TSCM could degrade disinformation probability and significantly reduce the rewards/payments issued to malicious users when compared to reputation-unaware crowdsensing. User utility that represents the difference between total payments made to the winners and their total sensing costs has been shown to decrease despite the benefits of the proposal slightly. Indeed, this can be an interesting future direction to tackle.

2.5.3 Solutions on participant/sensor reliability

Ma et al. [69] presented a time-series data model via a dynamic Bayesian network as a new solution to ensure the effectiveness, efficiency, and trustworthiness of data collection in MCS. Furthermore, the proposed solution is also committed to addressing the challenges due to the non-deterministic mobility behavior and low reliability of crowdsensing participants. In the presented system, the Expected Maximization (EM) algorithm is an essential component to learn the trustworthiness of a mobile user. An interesting extension to that work can be applying the learning model to an MCS context that works with delay-sensitive data and investigating the likelihood of over-fitting on long-time windows.

Malicious users in a smart city crowdsensing setting can attack the crowdsourcer while aiming to avoid being identified as an adversary. With this in mind, Pouryazdan et al. introduced vote and anchor-based approaches [86]. In the proposed scheme aggregate votes denote community recommendation on the trustworthiness of a participant whereas an anchor node is analogous to a trusted entity in the system that has the highest level of trustworthiness/reputation and full vote capacity as opposed to a regular participant that may have vote capacity less than one. The authors have shown that the trustworthiness of crowdsensing data can be improved if the number of anchor nodes is selected properly. Since physical constraints in the wireless communication medium were not fully considered, the impact of these constraints on the trustworthiness of crowdsensing data could be incorporated into the trustworthiness assessment mechanism of the related work.

As a more generalized solution, Pouryazdan et al. proposed a game incentive framework with three steps, including user recruitment, collaborative decision making, and badge rewarding [85]. Their method makes a compromise between decentralized and centralized reputation/trustworthiness assessment while ensuring the highest possible level of trustworthiness in user recruitment in mobile crowdsensing systems. User selection phase of

in TSCM as well as Social Network-Aided Trustworthiness Assurance (SONATA) [38] is adopted in the proposed approach. A repeated Sub-game Perfect Equilibrium (SPE)-based voting system was also introduced to build and assess users' reputation, which helps improve the rewarding phase of the recruitment. In comparison to fully distributed, community-driven, and user-centric trustworthy crowdsensing (i.e., SONATA), a promising improvement was achieved in terms of platform and user utility. More trustworthiness features could be considered to evaluate the game.

2.6 Resource limitations in Mobile Crowdsensing Systems

Despite the wide coverage and rapid knowledge updating of mobile devices, the battery and computation ability constraints remain and yield security vulnerabilities.

Multiple studies have proposed various methods for reducing energy usage in MCS campaigns from various aspects. From the standpoint of participating devices, most of them follow the steps, including data acquisition and data transfer to complete MCS campaign where every step requires energy. The existing approaches for lower energy expenditure at the data acquisition step mainly focus on adopting less energy-consuming sensors [13] and dynamically sensor work time scheduling [79, 42]. Besides, the authors in [89] investigate the effect of using lower-power processors to achieve a reduction in energy consumption in processing. To reduce the energy expended on data transferring, the authors in [104] present a data uploading framework called effSense, which enables devices to determine the proper time and network for data transition based on two planted decision-making schemes called cold-start and prediction-based scheme. Besides, the authors in [51] propose a system called Piggyback Crowdsensing (PCS). PCS makes it possible for devices to discover the best time to upload data among sporadic opportunities by predicting application usage patterns.

From the standpoint of recruitment, several studies report mechanisms for choosing a small number of participants and guaranteeing sufficient coverage at the same time [120, 63, 111, 98, 43]. Their proposals avoid useless battery consumption from redundant participants in a specific sensing area. For instance, in [98], the authors present a Quality of Information (QoI)-satisfied participatory sensing selection scheme where a greedy algorithm is applied for picking up participants in a real trace scenario. Experimental results validate the satisfying trade-off between energy consumption and the quantity of sensing data.

2.7 Incentive Mechanism in Mobile Crowdsensing Systems

Incentive mechanism plays a key role in MCS campaigns for the guarantee of sufficient participants. According to the survey [121], the incentives are mainly composed of three classes including entertainment as incentives [37, 28, 4], service as incentives [66, 25, 15] and monetary incentives [122, 55, 32, 114].

In [37], the authors proposed a game named Ostereiersuche to collect the physical location of players in a city park. With the similar spirit as the traditional game "Easter egg hunt", the game is designed to drive the players to find the hidden coupons spreading around the park. Via analyzing the data, they aim at identifying landmarks with distinguishable features comparing to their surroundings. Although the data reliability is not satisfied because of tessellation, the game theory behind the work provides a good example for MCS incentive. Similarly, the authors present an outdoor game in [28], where 50 volunteers from University of California, Los Angeles (UCLA) try to observe plants and record their status to gain points.

The authors apply game theory from a different perspective in [30], where they combine game-embedded ideas and monetary incentives. Specifically, the game platform determines the deposit, refund, and rewards and maximizes the personal gain of the users, thereby encourages participants to provide high-quality data.

In [66], the authors use services as an incentive for participants. The service providers request and assemble data from users and then process the data for valuable information that provides to users as compelling services. To achieve optimal fairness and social welfare, they proposed two schemes, namely Incentive with Demand Fairness and Iterative Tank Filling, based on demand.

In the monetary incentive mechanism, how to dynamically adjust the price of the user's sensing data based on the real valuation of the provided information is a challenge. To this end, the authors in [55] propose a reverse auction where users bid for selling their sensing data and the service provider checks the data quality and choose valuable users who have lower valuation than others. Then the selected users receive the money they bid for, and the losers in the campaign get the information containing maximum bid price. Their approach decreases the incentive cost with the guarantee of data quality.

2.8 A Brief Overview of Artificial Intelligence and Cybersecurity

AI-assisted or AI-driven cybersecurity has become an attractive topic with the advent of artificial intelligence and particularly the widespread use of machine learning techniques in various areas including security [73, 2]. In [109], the authors proposed a new solution for the MCS server by using Q-learning to ensure the security of the crowdsensing campaigns. On top of this, the authors built a deep neural network-based solution to ensure a secure MCS system to meet the following targets: authenticity, privacy, veracity of crowdsensing data, anti-jamming, and intrusion detection [106]. The authors explored the feasibility of applying stack autoencoder and convolutional neural networks to cope with various types of attacks in MCS systems.

Following the similar methods, Li et al. presented an offloading model in the context of a mobile cloud to detect malware in smartphones contacting with the same server but without being aware of system parameters [60]. In the event of limited network bandwidth, Q-learning is adopted to offload strategies to accelerate offloading speed and thus facilitate detecting malware. As an extension, Wan et al. improved the malware detection rate and accuracy by employing a deep convolutional neural network [103].

Zhou et al. present a robust MCS framework that combines deep learning and edge processing for security in [125]. The proposed scheme exploits a convolutional neural network (CNN), which was introduced in [17], to validate the data submitted by participants and prune deceptive data at the network edge nodes prior to being transmitted to the cloud. By doing so, self-interest participants could be efficiently prevented from transmitting useless data, and transmission latency due to data traffic could be reduced.

In a fog computing network, wireless links between fog nodes and end-users are vulnerable to attacks such as impersonation attacks on the physical layer. To deal with typical impersonation attacks on the physical layer, Tu et al. introduced Q-learning for the efficient detection of these attacks in a dynamic environment [101].

As AI-based techniques have shown to be effective under certain contexts, we aim to tackle the possibility of AI-based security provision for MCS. In the next Chapter 3, we investigate the ensemble machine learning against resource clogging attacks (also named fake task injection in the thesis).

Chapter 3

Machine Learning Assisted Prevention of Fake Task Injection in Mobile Crowdsensing Systems

3.1 Introduction

The non-dedicated nature of Mobile Crowdsensing (MCS) results in vulnerabilities in the presence of malicious participants to compromise the availability of the MCS components, particularly the servers and participants' devices. In this chapter, we focus on Denial of Service attacks in MCS, where malicious participants submit illegitimate task requests to the MCS platform to keep MCS servers busy while having sensing devices expend energy needlessly. After reviewing Artificial Intelligence-based security solutions for MCS systems, we focus on a typical location-based and energy-oriented DoS attack, and present a security solution that applies ensemble techniques in machine learning to identify illegitimate tasks and prevent personal devices from pointless energy consumption to improve the availability of the whole system. Through simulations, we show that ensemble techniques are capable of identifying illegitimate and legitimate tasks while Gradient Boosting appears to be a preferable solution with a area under curve (AUC) performance higher than 0.88 in the precision-recall curve. We also investigate the impact of environmental settings on the detection performance to provide a clearer understanding of the model. Our performance results show that MCS task legitimacy decisions with high F-scores are possible for both illegitimate and legitimate tasks.

Building on machine learning-based detection of illegitimate tasks, we investigate the

impact of machine learning-based prevention of battery-oriented illegitimate task injection in MCS campaigns. To this end, we introduce two different attack strategies and test the impact of ML-based detection and elimination of fake tasks on task completion rate, as well as the overall battery drain of participating devices. Simulation results confirm that up to 14% battery power can be saved at the expense of a slight decrease in the completion rate of legitimate tasks.

3.2 Proposed Machine Learning Embedded Mobile Crowdsensing System

Attacks can target mobile devices' battery and other resources. DoS attack is an example of attacks where attackers inject many fake (i.e., illegitimate) tasks with the motivation of keeping the MCS servers busy and draining the battery power of user devices. Fig. 3.1 presents a minimalist illustration of this type of attack where malicious end users submit sensing tasks.

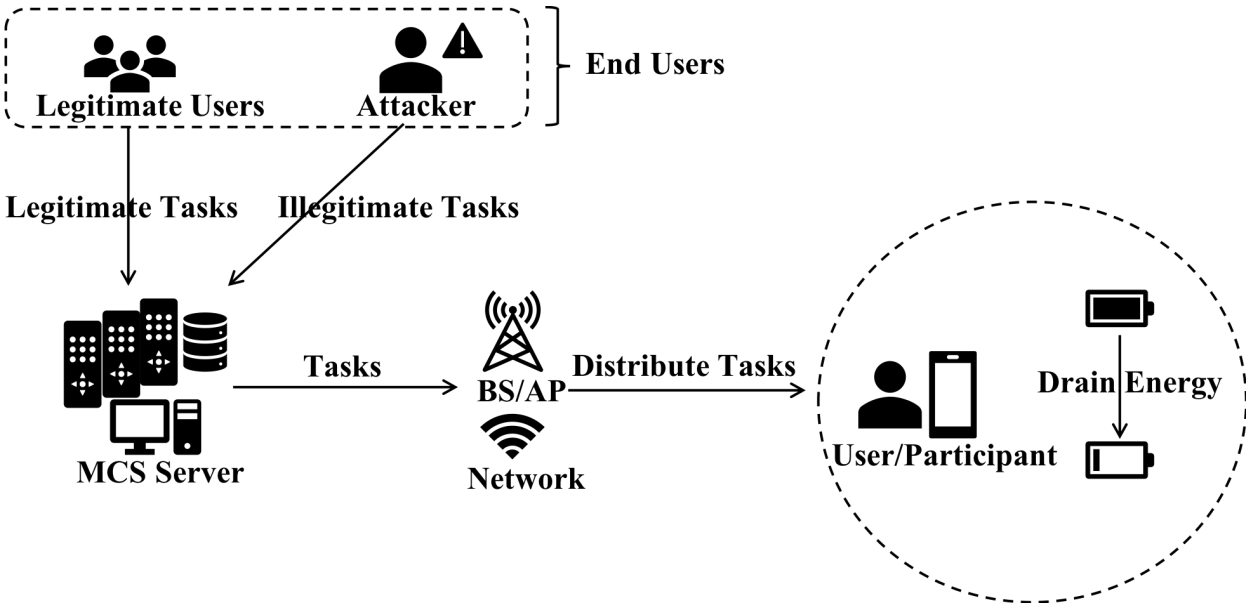


Figure 3.1: A minimalist illustration of a DoS attack in an MCS system. The target of a DoS attack is two-fold: 1) Clogging the resources of MCS servers, 2) Leading to resource (battery, processing power, bandwidth, storage) famine in participants' mobile devices.

Our system model is built on the common structure of MCS with the addition of an ML module prior to the task assignment by the MCS platform. A minimalist illustration of the considered setting is presented in Figure 3.2. In this environment, we characterize the attackers as malicious end-users who launch DoS attacks through the MCS platform towards MCS servers and participants. It is worth noting that we consider two types of tasks in this environment: legitimate tasks that are submitted by regular end-users, and illegitimate tasks that are submitted by malicious users who aim at inefficient resource usage at the servers and user devices. To prevent legitimate tasks from being fulfilled, the attacker aims to flood the targets with superfluous tasks which are also named illegitimate tasks in our system, as mentioned above. Here, a fundamental assumption (that is also presented in our performance evaluation section) is the following: illegitimate tasks tend to initiate during peak hours of the day. We align our peak hour definition with that of the utilities.

If the DoS attack remaining in the MCS system for a long time, the platform will start losing participant willingness to participate in MCS campaigns. To maintain the durability of the MCS system, an illegitimate task should be forecast before being assigned to any participant.

It is reasonable to assume that all illegitimate tasks initiate from specific zones around the city. Hereafter, we use the terms attack areas, attack zones, and attack regions interchangeably. For legitimate tasks, their locations are randomly selected; thus, there may be some tasks to be legitimate, although they are in the area of attack regions. Each task has a recruitment radius (r_{rec}) and a movement radius (r_{mov}). Users are recruited for sensing the tasks based on the distance between the users and the tasks, which means a task will be assigned to the nearest participant within the recruitment radius as the participant. The task moves to another location every ten minutes within its movement pattern following a pre-determined probability distribution, since each task has a different duration, the number of slots can be various. For instance, if the duration of a task is 30 minutes, it will move twice and will possess three time slots.

3.3 Machine Learning Performance Study

3.3.1 Simulation Settings

Our simulations are implemented on Crowdsensim simulator, which is an open source platform to introduce realistic MCS campaigns on real street maps [19]. In this case study,

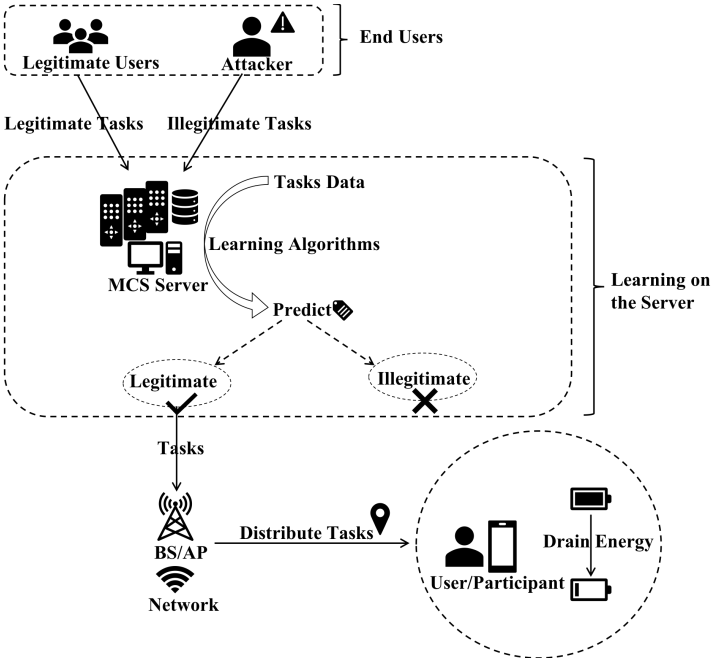


Figure 3.2: MCS System with DoS attackers targeting the participants' devices through the server. In order to prevent needlessly increased resource utilization at the MCS servers and/or participating devices, machine learning algorithms are utilized at the MCS platform to make a decision on the legitimacy of an arriving MCS task.

illegitimate tasks form 10% of all submitted tasks. A task is represented by a set of features as follows:

$task$ = ['ID', 'Latitude', 'Longitude', 'Day', 'Hour', 'Minute', 'Duration', 'Remaining time', 'Energy', 'Range', 'Legitimacy'].

The definition of these features are listed in Table A.2. Table 3.2 briefly presents how global variables and features of illegitimate and legitimate tasks are set in our simulations. Task-specific settings of illegitimate tasks also shape the details of the threat model in the MCS system.

We consider the peak hours to be coherent with the electricity demands on a weekday, as reported by Ontario hydro [1], and we classify the hours for one day into three groups:

[1]We use the winter schedule of Ontario Energy Board to make peak/off-mean time judgment. We assume that this also aligns with the times when users are actively using information and communication services.

Feature	Description
Latitude Longitude	Location of the task.
Day Hour Minute	The time when the task initiates.
Duration	Total active duration of the task.
Remaining time	The remaining time to complete the whole task.
Energy (%)	Battery consumption of the mobile device to complete the task.
Range	The recruitment radius of the task.
Legitimacy	The legitimacy of the task: legitimate or illegitimate.

Table 3.1: Task features and their definition.

- On-peak hours:7am - 11am
- Mid-peak hours: 12pm - 5pm
- Off-peak hours: 6pm - 6am

With these in mind, we add a new Boolean feature *OnPeakHours* into the data set to indicate whether a task is executed in on-peak hours or not.

Each task moves to another random location within the movement radius every ten minutes for wider coverage of possible participants. The tasks' movements are also used in the training and testing of the legitimacy of tasks. Our purpose is to recognize illegitimate tasks even if tasks move in arbitrary directions.

We choose five cities of different sizes in Canada for MCS terrains. Their map sizes depend on the data size when the map is downloaded as shown in Table 3.3.

3.3.2 Data pre-processing

Data pre-processing involves data cleansing and location issues. As task locations are not fixed, a task on the border or near the boundary of a city could move out of the city as

Table 3.2: Global and class-specific simulation settings for task generation

(a) Global Parameters

Number of tasks	{1000, 2000, 3000, 4000, 5000, 6000}
Number of attacked regions	{5, 10, 15, 20, 25}
Attack region radius	200m
Simulation Duration	6 days

(b) Class-specific settings. Attacker tasks target certain locations, tend to last longer, mostly require more battery capacity, and mostly target peak hours.

	Illegitimate Tasks	Legitimate Tasks
Day	Uniformly distributed in [1, 6]	
Hour (task)	80% : 7am to 11am; 20%: 12pm to 5pm	Uniformly distributed throughout the day
Duration (task)	70% in {40, 50, 60} [minutes]; 30% in {10, 20, 30} [minutes]	Uniformly distributed over {10, 20, 30, 40, 50, 60} mins
Battery usage (smartphone)	50% in {7%-10%}; 50% in {1% -6%}	Uniformly distributed in {1%-10%}
Recruitment Radius / r_{rec} (task)	Uniformly distributed in {30m, 100m}	
Movement Radius / r_{mov} (task)	{60m, 80m, 100m}	

Table 3.3: Downloaded map data size of cities. The street network is converted to a multi-digraph with multiple edges. The data size of a city denotes the number of edge sets.

City	Map size
Dryden	2101
Powell-River	3726
Clarence-Rockland	4776
Brant	10678
Timmins	12771

simulation time elapses. We delete all the task data outside the city area in the data preprocessing step because we don't consider the tasks beyond the boundary of a city in our simulations. To facilitate location tracking, we partition the original map of the city into a grid of multiple cells and number each grid cell from west to east and from north to south, as in Figure 3.3. Any point (i.e., longitude & latitude) located in the same grid cell is given the number of the grid. In our scenario, the initial distance of each grid cell is set as 600m, which is double the distance of the recruitment radius plus the maximum movement radius. We choose this number to maximize the probability of grouping the illegitimate tasks around an attack location into the same grid cell or at least over adjacent cells. However, the width and length of a city are not necessarily to be integral multiples of 600 meters. Thus, we divide the map into multiple grid cells of approximately 600 meters according to the minimum longitude and latitude and the maximum longitude and latitude of the city area.

3.3.3 Machine Learning Algorithm Selection

Although various ML algorithms can be used in the DoS prevention module, in this subsection, we consider a flexible ensemble classifier, Random Forest (RF) and another ensemble technique Gradient Boosting (GB) [62, 71, 41, 6]. In the simulation experiments, we tried single classifiers like Naive Bayes and Decision Trees; however, their performance is unsatisfactory. We present the machine learning performance (i.e., F_1 score) of these two

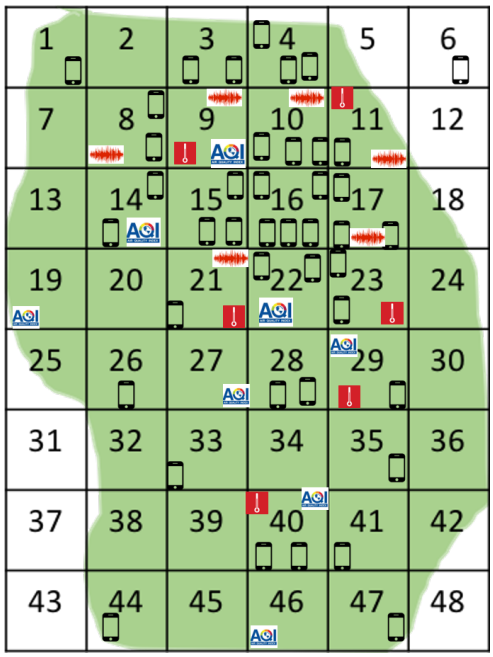


Figure 3.3: An illustration of partitioning a city map into multiple grid cells. Three sensing tasks (sound level, air quality index and temperature) are illustrated along with a number of participating mobile devices. Partitioning the city map into grid cells enables easy tracking of task and participant locations that need to be incorporated in the decision making process as a part of the inputs.

algorithms in AppendixA. Thus, we evaluate two ensemble machine learning algorithms based on decision trees: Random Forest and Gradient Boosting to represent averaging methods and boosting methods, respectively. The principle behind the averaging methods is to average the predictions of several independent estimators and thus reduce the variance. The learning procedure of boosting methods is sequentially adding predictors to the combined estimator, each one correcting its predecessor by gradually minimizing the loss function.

Specifically, RF uses a random sample of the data to build decision trees independently where different features are used. Then it aggregates their results to make a decision, which reduces the risk of over-fitting and improves the overall performance. The critical motivation behind it is to reduce variance. Different from RF, GB also generates an ensemble of trees; however, the idea is to follow the direction of improvement. According to the algorithm, one tree at a time is built, and based on the gradient computed on

previously trained trees, the new tree is trained to reduce the bias of the model and improve classification performance.

During the learning and prediction steps, the influence of different parameters, and the size of the crowdsensing city on the learning algorithms' performance may vary. The design and application of machine learning algorithms for an MCS system is a challenging problem. Therefore, in the results part, we run a feasibility study of two candidate algorithms under diverse settings.

3.3.4 Machine Learning Strategy Design

Based on storage constraints, we introduce two learning strategies into our model. One is learning from the updated information on the previous day to predict the legitimacy of tasks of the following day. The other requires more storage space to continuously accumulate data collected in the past few days to forecast the tasks on the next day. For both strategies, the MCS server starts predicting on the second day. If a task is predicted as an illegitimate task, the MCS platform filters out the task without attempting to assign it to a participant. Thus, by doing so, the MCS platform is expected to avoid energy-wasting while keeping the MCS servers available.

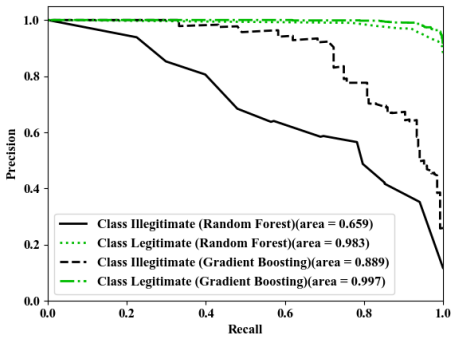
Per-Day learning:

Per-Day learning is the strategy that predicts the legitimacy of tasks based on the learning of the day before. For instance, when edge devices are used for learning purposes as opposed to a cloud-based MCS server, due to possible storage limitations of the MCS server, this method applies to the MCS system as the database will not be large enough to store data in big volume. Thus, the data could be updated every day to save the latest information about tasks collected on that day.

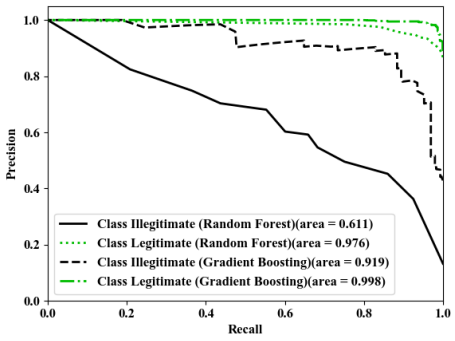
Day-after-Day learning:

Day-after-Day learning, also called incremental learning, is the strategy that combines all the data before one day for training and then makes the prediction of tasks on that day. Although more training data usually results in better prediction accuracy, vast storage space is required on the server-side.

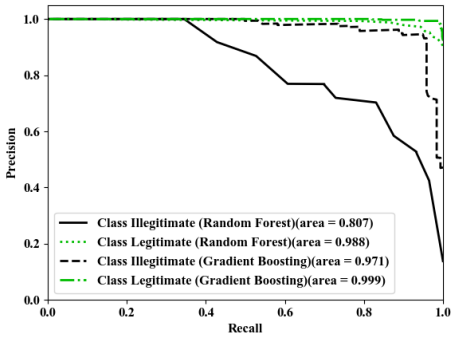
3.3.5 Machine Learning Performance Results



(a) Precision-recall curve (Dryden)



(b) Precision-recall curve (Clarence-Rockland)



(c) Precision-recall curve (Timmins)

Figure 3.4: Precision-recall performance under two learning algorithms. Gradient Boosting outperforms Random Forest especially for illegitimate task classes in both cities.

Regardless of the uniformly distributed features, the DoS prevention module uses the following features in the learning: *Hour, Duration, Energy, OnPeakHours,* and *GridNumber*. In the simulations, we evaluate two kinds of ensemble machine learning algorithms: Random Forest (RF) and Gradient Boosting (GB) to represent averaging methods and boosting methods. In the simulation, *GiniImpurity* is used as a criterion in RF to evaluate the probability that we classify the data incorrectly. The number of trees in the RF is 100. For GB, we use logistic regression as the loss function and set the number of boosting stages as 100. Our experiments show that when we use data set with 4000 tasks to train and test, the random ten runs of the ML code need around 4.98 CPU time when using

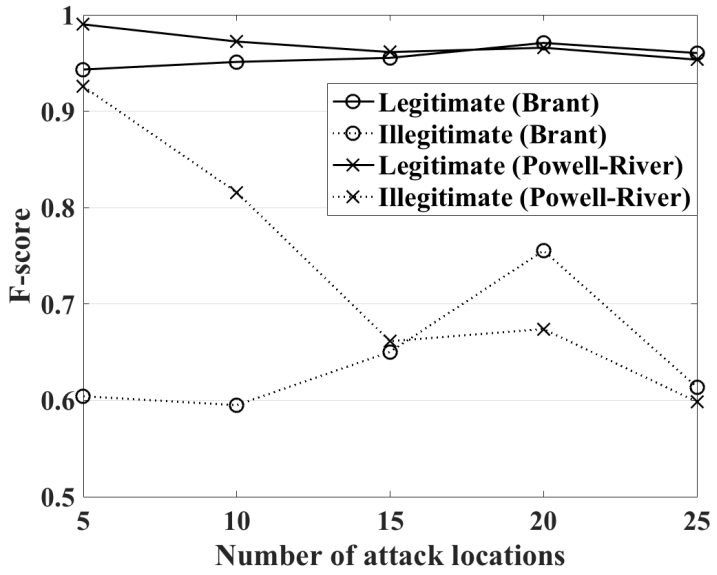


Figure 3.5: F-scores with the number of attack locations. Performance is highly related to the number of attack locations in big and small cities.

Gradient Boosting and 1.92 CPU time when using Random Forest. The system in the computer is 64-bit Ubuntu 18 with Intel Core i5 CPU @3.40$GHz \times 4$.

Having fixed the number of tasks to 4000, the number of attack regions to 10, and moving radius of a task (r_{mov}) to 80m, Figure 3.4 illustrates the precision-recall performance when applying the two ML algorithms in three cities. Dryden, Clarence-Rockland, and Timmins are the smallest, the medium, and the biggest city, respectively in our scenarios.

To evaluate the classifier, we use the mean precision-recall curve as it is appropriate to use in our study as the classes in our scenario are imbalanced. The precision-recall curve stands as a critical plot for interpretation of the performance when the model aims at predicting the positive class, and it summarizes the true positive rate versus the sum of true positive rate and false positive rate. Thus, true positive denotes the case when a legitimate (illegitimate) task has been classified as legitimate (illegitimate) whereas false positive denotes the case where an illegitimate (legitimate) task has been classified as legitimate (illegitimate). In the other word, when we calculate precision and recall of each class, we consider this class as positive class, which is illustrated in confusion matrix in Table 3.4 where TP, FP, FN and TN refer to true positive, false positive, false negative and true negative respectively. According to the simulation results, Gradient Boosting

outperforms Random Forest regardless of the size of the city.

Table 3.4: Each class acts as positive class when drawing precision-recall curve.

(a) The legitimate as positive class.

		Predicted	
		legitimate	illegitimate
Actual	legitimate	TP	FN
	illegitimate	FP	TN

(b) The illegitimate as positive class

		Predicted	
		illegitimate	legitimate
Actual	illegitimate	TP	FN
	legitimate	FP	TN

Since the simulation duration is set at 6 days, the learning process could vary using diverse strategies. We test the use of two alternate strategies Per-Day Learning and Day-after-Day Learning.

We evaluate the average performance results of these two strategies in the 6-day-simulation process, as shown in Table 3.5. Incremental learning exhibits improvement over per-day learning; however, the strategy should be chosen in terms of the requirements and effective storage size of the MCS system.

To study the correlation between the density of attack areas and the size of the city, we set the total number of tasks at 4000 and movement radius of tasks at 80m, whereas the only variable parameter is the number of attack locations.

For a small city (which is Powell River in the simulations), high performance can be achieved unless the number of attack locations does not exceed five. As the number of attack locations increases, the performance significantly degrades. For a bigger city (which is Brant in the simulations), the best performance is obtained when there are 20 attack locations, as seen in Figure 3.5. The reason for this behavior is that in a bigger city, the illegitimate tasks are more likely to be scattered over multiple grids (i.e., more popular locations).

Under ten attack locations for illegitimate tasks, the classification performance in larger cities outperforms that under smaller cities, as seen in Figure 3.6. Indeed, one can expect that smaller cities would certainly have fewer attack locations, whereas bigger cities have more attack locations due to having more popular spots. Thus, in a real-life setting, since the number of attack locations is expected to have a direct relation with the size of a city, illegitimate tasks can be detected with high performance in both small cities and big cities when Gradient Boosting is adopted.

In the next step, we study the impact of the density of tasks and the size of the city on task legitimacy detection performance. To this end, we set the number of attack locations

Table 3.5: Per-day and day-after-day performance under Gradient Boosting in different cities (4000 tasks, 10 attack locations and r_{mov} = 80m)

City	Metric	Per-Day (Avg)	Day-after-Day (Avg)
Dryden	Overall Accuracy	0.9212	0.9318
	Illegitimate F-score	0.6706	0.7279
	Legitimate F-score	0.9551	0.9609
Powell-River	Overall Accuracy	0.9428	0.9525
	Illegitimate F-score	0.7674	0.8092
	Legitimate F-score	0.9673	0.9735
Clarence-Rockland	Overall Accuracy	0.9530	0.9624
	Illegitimate F-score	0.8000	0.8398
	Legitimate F-score	0.9733	0.9787
Brant	Overall Accuracy	0.9612	0.9744
	Illegitimate F-score	0.8399	0.8975
	Legitimate F-score	0.9779	0.9853
Timmins	Overall Accuracy	0.9398	0.9754
	Illegitimate F-score	0.7238	0.8950
	Legitimate F-score	0.9704	0.9860

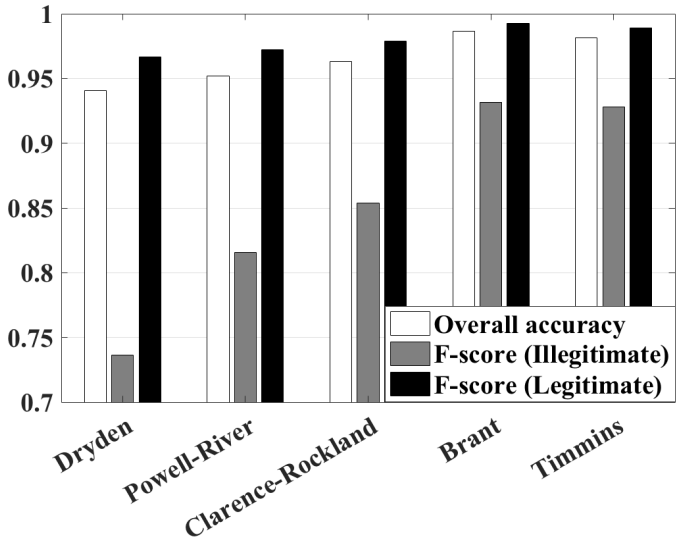
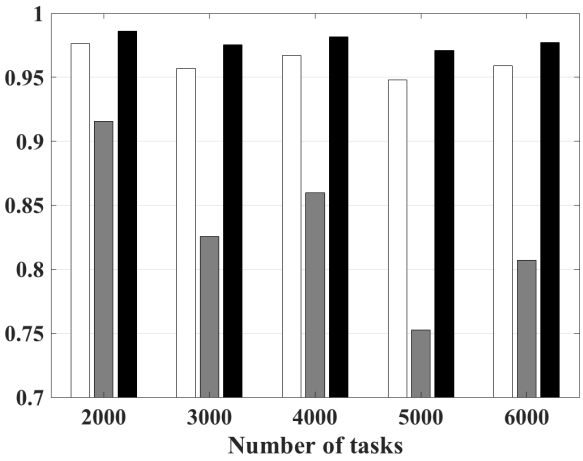


Figure 3.6: Final (last-day) performance of different cities under 10 attack locations. Better performance can be obtained in larger cities. However, small cities are unlikely to have many attack locations as in this scenario.

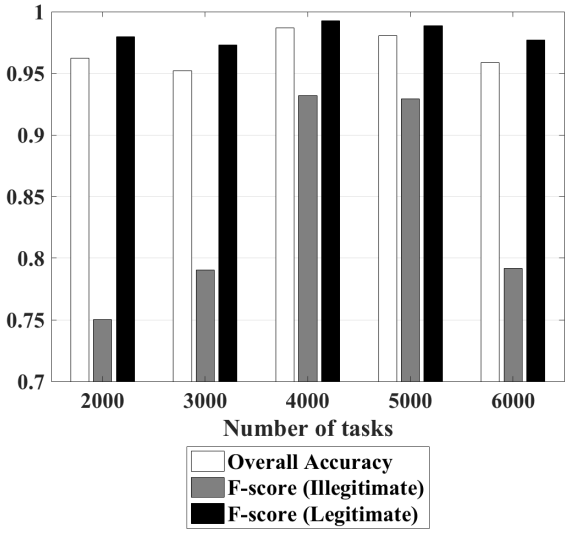
at ten and task movement radius (r_{mov}) at 80m. For a small city (which is Dryden in the simulation), we run our simulations by varying the number of tasks from 2000 to 6000 and run a 6-day simulation under five attack locations under Gradient Boosting. As seen in Figure 3.7a, the best classifier performance is under 2000 tasks. For a bigger city (which is Brant in the simulation), the best performance of the classifier can be achieved under a higher number of tasks (i.e., 4000 tasks), as seen in Figure 3.7b.

Nevertheless, similar to the previous comment, if the density of tasks is appropriate for the city size (i.e., a smaller city would have fewer tasks than a bigger city), it is possible to predict the legitimacy of MCS tasks accurately in both small and big cities.

In the last step of the simulations, we study the impact of the movement radius on the detection performance. As seen in Figure 3.8, the smaller the range of movement, the more concentrated the illegitimate task groups, and consequently, the illegitimate task clusters become easily distinguishable from legitimate ones. Here, the medium city Clarence-Rockland is chosen as an example.



(a) Dryden



(b) Brant

Figure 3.7: The impact of the number of tasks on MCS task legitimacy decision. Performance degrades beyond a certain number of tasks.

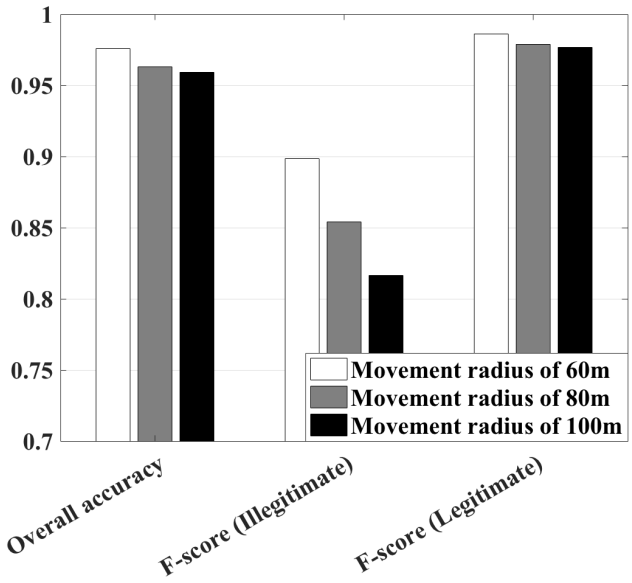


Figure 3.8: The impact of the movement radius on MCS task legitimacy decision (Clarence-Rockland). When illegitimate tasks move on large circles, it becomes harder to detect them.

3.3.6 Concluding Remarks

In this section, we have presented a feasibility study of machine learning-based solutions to overcome the Denial of Service (DoS) attacks in MCS. We focus on the differentiation of illegitimate and legitimate tasks so that the system can predict the legitimacy of a task to avoid needlessly consuming computational power, battery power, bandwidth, and/or cellular data. In our system model, attacks launch from specific locations to inject illegitimate tasks into the system. To address this security vulnerability, we have adopted two machine learning algorithms, Random Forest (RF) and Gradient Boosting (GB), to train the system to predict the legitimacy of a task. Our simulation results illustrate that Gradient Boosting outperforms Random Forest in terms of precision and recall parameters. Furthermore, under realistic assumptions regarding the density of tasks and attack locations, Gradient Boosting exhibits promising performance up to 0.95 accuracy and 0.9 F-scores for both illegitimate and legitimate tasks.

3.4 The Impact of Task Movement

3.4.1 Task Movement Models

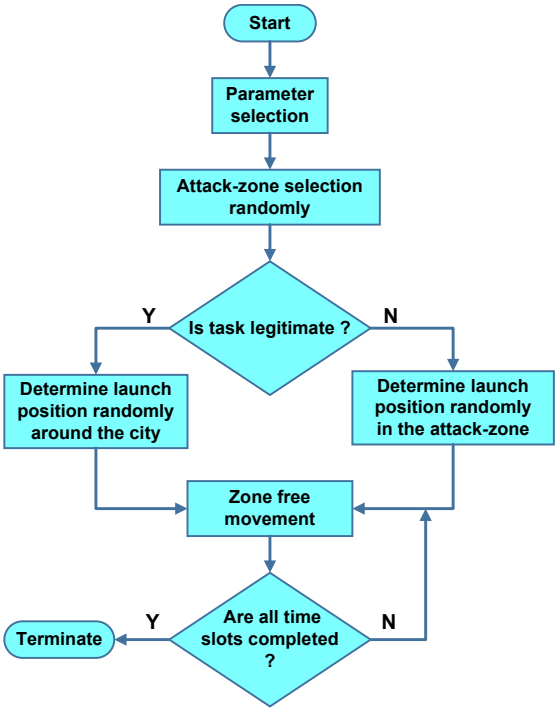
In this subsection, we design two take movement models based on the different behaviors of tasks. We consider the boundary of attack regions carefully and propose the boundary limits on the movement to form a zone-limited task movement model. The details are described as follows:

Zone-free Task Movement (ZFM)

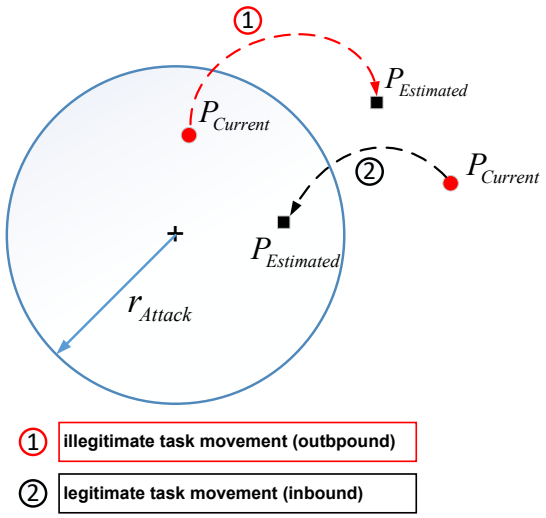
The center of attack zone, as well as the zone radius together, determine the location and area of an attack zone where the attacker submits illegitimate tasks to the MCS platform. All illegitimate tasks are generated in pre-determined attack zones in the city area. The initial positions for legitimate tasks are set up without restriction, and as a result, they are possibly located in the attack zones too. Zone-free Movement (ZFM) is defined by unrestricted movement principle for both illegitimate tasks and legitimate tasks, as shown in Figure 3.9a. Illegitimate tasks can move to the outside of the selected attack zones, and meanwhile, legitimate tasks are able to get into the attack zones without any restriction, as illustrated in Figure 3.9b.

Zone-limited Task Movement (ZLM)

Zone-limited Movement (ZLM) approach is developed to add an unreachable zone concept to introduce boundary control to the illegitimate tasks in MCS campaigns. The zone-limited task movement algorithm indicates how to generate tasks according to the zone-limited definition, as shown in Figure 3.10. The legitimate tasks never appear in the attack zones even in the initiation stage, and that is the reason why their launch positions are selected out of the attack zones. With the existence of attack zones' boundary control, these two different types of tasks adopt two separate movement rules. According to the zone-limited mobility model, illegitimate tasks remain in the attack zones while legitimate tasks are prevented from moving inside of the attack regions. Figure 3.11b and Figure 3.11a illustrate details on how to determine the next destination for two kinds of tasks in our simulation. To achieve the principles of zone-limited movements, we firstly propose a destination position as the estimated position ($P_{Estimated}$) and then modify it to determine the corrected position ($P_{Corrected}$), which will be the final position of the movement decision.



(a) Task location under Zone-free Movement



(b) Zone-free task movement according to zone region

Figure 3.9: Zone-free Movement

3.4.2 Participant Recruitment

In our simulations, all participants have their motion trail, which is composed of a series of data of latitude, longitude, and time. We assume each user owns only one mobile device. As the tasks and users both update their locations on the routines as well as active timestamps, the recruitment policy involves the following three factors regarding space, time, and energy conditions:

- The user should be located within 100 meters of a task to be considered as a participant since the sensing distance is limited;
- The user's active time is during the task's recruitment stage;
- The battery level of the user's mobile device needs to be higher than the energy usage for completing the task.

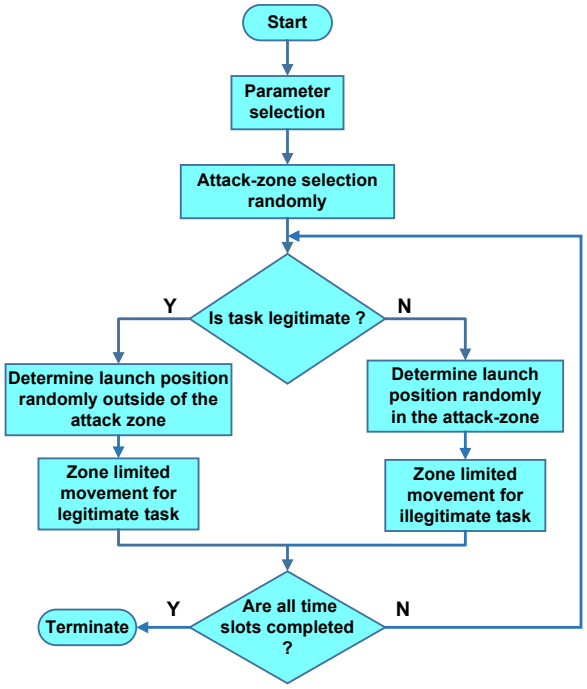
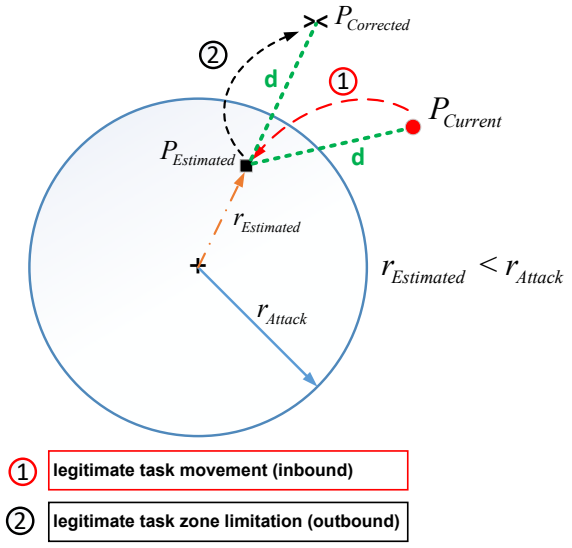


Figure 3.10: Task location under Zone-limited Movement

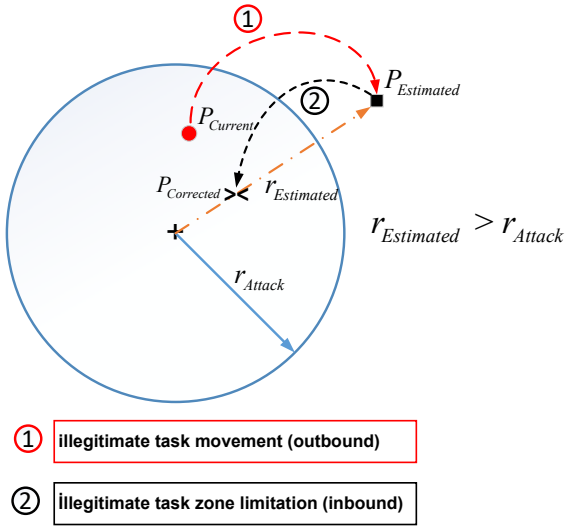
Only the participants that meet all three requirements could be considered for recruitment in the MCS campaigns. Based on the settings and the recruitment policy, our simulation results evaluate the performance of the proposed machine learning-based illegitimate task prevention mechanism.

3.4.3 Performance Evaluation

We evaluate the performance of the MCS system when applying the two task mobility models under the simulation parameters presented in Table 4.1. Simulations are carried out in Crowdsensim simulator [19]. Each data point in the simulation results presents an average of ten runs within a confidence interval of 95%. We consider two cities, namely Dryden and Brant, which represent a small and a big city, respectively. 10,000 mobile device users walk around the cities with their actives time randomly distribute between 60 minutes and 180 minutes. In addition, the battery level of a mobile device is assigned as 20% to 100% with the possibility of 80% and 1% to slightly less than 20% for the other 20% possibility.



(a) Zone-limited legitimate task movement



(b) Zone-limited illegitimate take movement

Figure 3.11: Mobility of legitimate and illegitimate tasks under zone-limited movement scenario

Machine Learning Performance

Table 3.7 presents the machine learning performance using Gradient Boosting with 150 boosting stages based on average of ten simulation runs for two movement models. We

Table 3.6: Simulation settings for task generation

	Illegitimate Tasks	Legitimate Tasks
Day	Uniformly distributed in [1, 6]	
Hour (task)	80% : 7am to 11am; 20%: 12pm to 5pm	Uniformly distributed throughout the day
Duration (task)	70% in {40, 50, 60} [minutes]; 30% in {10, 20, 30} [minutes]	Uniformly distributed over {10, 20, 30, 40, 50, 60} mins
Battery usage (smartphone)	50% in {7%-10%}; 50% in {1% -6%}	Uniformly distributed in {1%-10%}
Recruitment Radius / (r_{rec})	Uniformly distributed in {30m, 100m}	
Movement Radius / (r_{mov})	[10m, 80m]	
Movement Bearing (θ)	[0°, 360°]	
Number of tasks	{500, 1000, 2000}	

calculate legitimate task F_1 score and illegitimate task F_1 score separately to explore the ability of machine learning on recognizing two types of tasks. F_1 score denotes the harmonic mean of precision and recall, and can be broken down as seen in Equation (3.1). In addition, accuracy is also presented as an overall judgment of ML performance which is calculated as shown in Equation (3.2) where TP, TN, FP and FN are true positive, true negative, false positive and false negative, respectively.

$$F_1 = \frac{2TP}{(2TP + FP + FN)} \tag{3.1}$$

$$Accuracy = \frac{(TP + TN)}{(TP + FP + FN + TN)} \tag{3.2}$$

Table 3.7: Machine Learning Performance

City	Task Number	Legitimate F1		Illegitimate F1		Accuracy	
		ZFM	ZLM	ZFM	ZLM	ZFM	ZLM
Dryden	500	0.9568	0.9617	0.6289	0.6631	0.9221	0.9314
	1000	0.9678	0.9755	0.6981	0.7795	0.9418	0.956
	2000	0.9777	0.9863	0.8137	0.8757	0.9602	0.9754
Brant	500	0.9698	0.9739	0.7371	0.7752	0.946	0.9533
	1000	0.9838	0.9861	0.8635	0.8817	0.9712	0.9752
	2000	0.9913	0.9929	0.9261	0.9347	0.9844	0.9873

Since the illegitimate tasks are constrained within the attack zones in the ZLM scenario while they are not restricted to relocate from one position to another in the zone-free movement, it is reasonable that machine learning performs better under ZLM. On the other hand, it is also possible that sometimes ML in ZLM cannot guarantee an better performance due to the machine learning limits. It is noticeable when it comes to a bigger city (such as Brant in this example) with the higher volume of tasks, the results of machine learning are similar for two models, which can be seen in Brant with 1000 tasks and 2000 tasks.

Energy Savings

The simulation results illustrate energy savings under the zone-free movement of tasks, as seen in Figure 3.12. Energy consumption on the y-axis indicates the sum of the battery unit percentage of all recruits drained for the total recruitment. For instance, if a task needs a 6% battery level and another one requires 5%, then the energy consumption of these two tasks will be 11. As seen in the results, as the city size gets larger, energy savings increase. Furthermore, as the number of sensing tasks increases, battery savings can reach 14%. The second set of tests has been carried out under the ZLM movement of the tasks, which is also shown in Figure 3.12. Energy savings can be achieved in the range of approximately 6.8% to 10%, which both occur with 1000 tasks in both cities.

Due to the non-deterministic nature of the users' movement, there is no necessary relationship between the energy-saving rate and movement models. Based on 10% existence of illegitimate tasks and unwarrantable successful recruitment of these tasks, it is fair to

have the energy-saving rate fluctuate around 10%.

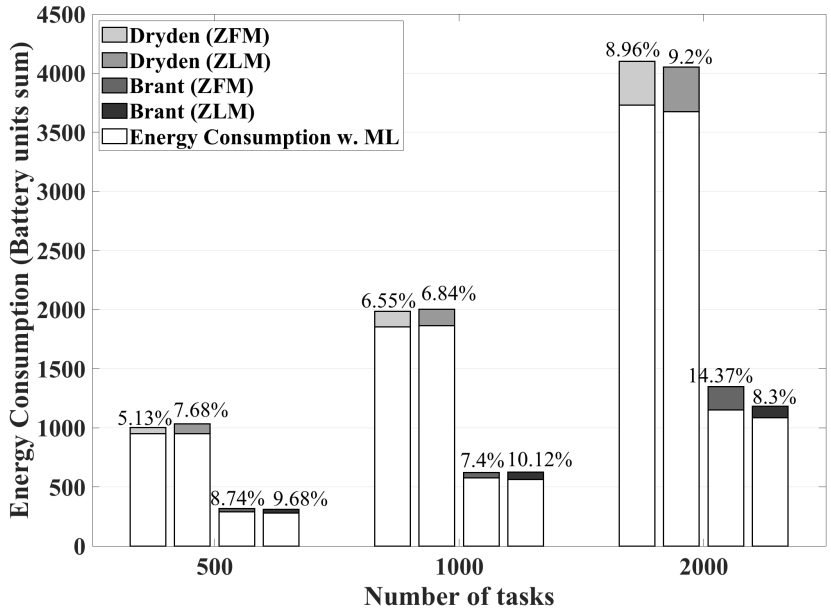
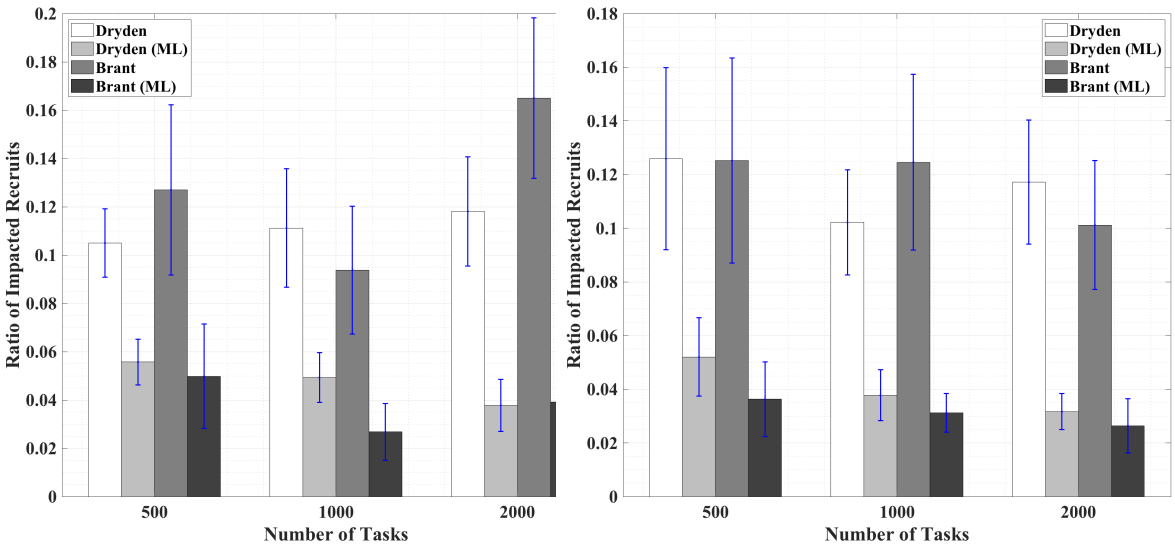


Figure 3.12: Energy saving conditions with saving rate. Energy consumption denotes total drained battery unit percentage.

Affected Recruits

In this subsection, we explore how machine learning relieves the influence of battery-oriented illegitimate tasks on MCS players. As shown in Figure 3.13a and Figure 3.13b, the impacted recruits by illegitimate tasks decrease with the existence of machine learning. It is noticeable that the influence of illegitimate tasks drops from 16.5% to 3.9% in Brant with 2000 tasks under the zone-free scenario. Even in the worst case, when machine learning cannot achieve good performance, the simulation results indicate that in the small city Dryden with a low volume of 500 tasks, the effect on the participants also drops down from approximately 10% to 5%. The confidence interval in the charts is calculated using the 95% confidence level. The error bars may look large due to two reasons: 1) the attack zones in ten runs are generated randomly while the users' movements are the same, and 2) the plotted images are zoomed in.



(a) ZFM (b) ZLM

Figure 3.13: Impacted recruits with and without applying ML in two movement models.

Legitimate Task Completion

Machine Learning cannot offer prediction with 100% precision in MCS, therefore introducing machine learning into the MCS will potentially misjudge some legitimate tasks as illegitimate tasks, and to some extent, it affects normal tasks completion. We investigate the legitimate tasks completion rate with and without machine learning to investigate the feasibility of ML. Table 3.8 shows the results, where LCR stands legitimate task completion rate and influence rate is calculated as the ratio of incomplete legitimate tasks when ML is applied, as formulated in Equation (3.3). In this equation, $w/0$ stands "without" while $w.$ represents "with".

$$Influence\ Rate = \frac{(LCR\ \ w/o\ ML - LCR\ \ w.\ ML)}{(LCR\ \ w/o\ ML)} \tag{3.3}$$

The results show that machine learning leads to 3.68% of the legitimate tasks unfinished in the worst case, which is acceptable in a real mobile sensing campaign based on the promising improvement in the energy savings of the participants.

Table 3.8: Legitimate Tasks Completion

	City	Number of tasks	LCR w/o ML	LCR w.ML	Influence Rate
ZFM	Dryden	500	0.3872	0.3723	0.0386
		1000	0.3793	0.3717	0.0199
		2000	0.389	0.3793	0.0251
	Brant	500	0.1035	0.0997	0.0368
		1000	0.1091	0.1081	0.0092
		2000	0.1079	0.1063	0.0145
ZLM	Dryden	500	0.3795	0.3673	0.0323
		1000	0.3795	0.3723	0.019
		2000	0.382	0.3779	0.0107
	Brant	500	0.1013	0.0994	0.0196
		1000	0.1018	0.1004	0.0132
		2000	0.1014	0.1007	0.0065

3.4.4 Concluding Remarks

This subsection studies the impact of machine learning-based prevention of illegitimate task injection in MCS campaigns. To this end, we first introduce two possible behaviors of illegitimate task injection. Under the two scenarios, we simulated an MCS system in two cities with different sizes and fixed population 10,000. Through simulations, we demonstrate that when ML-based prevention of illegitimate tasks is achieved, up to 14% energy savings can be achieved, and up to 13% of participants can be protected across the MCS network. Indeed, since it is yet not possible to reach 100% precision of ML performance, the task completion rate decreases slightly (i.e., 0.6-4%), which is acceptable on the basis of the improvement in the battery saving and impacted population.

As an improvement on the empirical design of fake task injection in this chapter, in the next chapter, we introduce a self-organizing feature map (SOFM) modeled fake task injection and analyze the magnified impacts under different SOFM topology.

Chapter 4

Self Organizing Feature Map Modeling Fake Task Injection: From Standpoint of Attackers

4.1 Introduction

Clogging attacks in Mobile Crowdsensing (MCS) denote the injection of fake sensing tasks into MCS campaigns to interfere with serviceability and user participation in sensing campaigns. Due to the lack of a realistic location-based and energy-oriented clogging attack model in MCS, this type of attack has not been well investigated. To this end, for the first time, we introduce a self-organizing feature map (SOFM)-based clogging attack model that aims at maximizing the number of affected participants according to the location of attack zones. These zones are identified by clustering 2-D coordinates of all potential participants and finding out aggregation areas of their mobile devices. We evaluate and verify the introduced attack model via simulations by comparing it to an attack model that relies on random mobility of illegitimate tasks over the attack zones. Our simulation results demonstrate that SOFM-based modeling of clogging attacks in MCS results in a significant impact with almost 50% affected participant population, 24% affected recruitment decisions, and up to 28% energy overhead introduced by illegitimate tasks injected to the MCS campaigns. This section presents a SOFM to model the clogging attacks in MCS systems by configuring multiple features of the fake tasks to maximize the number of affected participants. The presented attack model is compared to the formerly presented attack model in Section 3.3, where task features were static except for their locations.

Simulation results show that MCS-based clogging attacks can affect almost 50% of the participants while increasing the energy consumption in the devices by up to 28%.

Uniformly and randomly initialized neurons are designed with fixed and adaptive quantities that are determined based upon the affected area on the covered terrain. Through numerical studies, we show that the impact of the SOFM structures can affect up to 46% of the participants and up to 37% of the recruits under various SOFM topologies. Furthermore, SOFM-based attack models can increase the energy consumption in the recruited devices by up to 39% due to the illegitimate task submission.

4.2 User and Task Data Generation

We choose Crowdsensim [19] as the simulation tool because it enables us to validate the results based on the real city maps. Using Python package OSMnx introduced in Crowdsensim, we download geometrical street networks of cities in Canada to represent cities of different sizes. Specifically, we choose "walk" street networks, which gives us all the paths in the cities where the pedestrians can walk along. We divided the whole simulation process into sections based on their functions, which are described in detail, as shown in Figure 4.1.

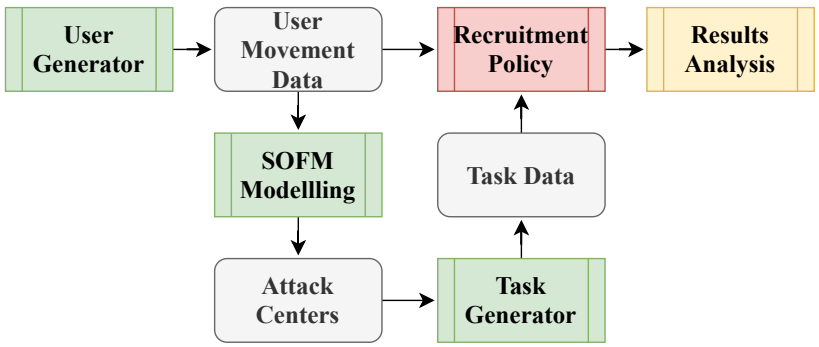


Figure 4.1: Data generation, processing and analysis.

Table 4.1a shows a basic global setting in our simulations. We set the simulation continuation as 6 days and specify the number of users to be 10,000. The attack areas are settled as circles with a radius of 200 meters from the attack centers, which are generated randomly or based on SOFM results.

In the following three subsections, we introduce four process modules in the simulation process in detail, presenting how to generate users' routine, how to apply SOFM modeling, what kind of task data look like, and the principles to recruit participants to tasks respectively.

Table 4.1: Simulation settings in data generation

(a) Global Parameters

Simulation Duration	6 days
Number of users	10000
Number of tasks	{500, 1000, 2000}
Num. of attacked regions	6
Attack region radius	200m
Attack region center	{Random, SOFM-based}

(b) Distinctive settings for two classes. Illegitimate tasks own features of longer duration, more energy consumption and peak hours aggregation.

	Illegitimate Tasks	Legitimate Tasks
Hour (task)	80% : 7am to 11am; 20%: 12pm to 5pm	20%: 0pm to 5am; 80%: 6am to 23pm
Duration (task)	70% in {40, 50, 60} [minutes]; 30% in {10, 20, 30} [minutes]	Uniformly distributed over {10, 20, 30, 40, 50, 60} [minutes]
Battery usage of smartphone for a whole task	80% in {7%-10%}; 20% in {1% -6%}	Uniformly distributed in {1%-10%}

4.2.1 User Generation

Users' movement data is generated in the format {'ID', 'latitude', 'longitude', 'day', 'hour', 'minute' }. Each user moves over some nodes on the paths with the speed that is uniformly distributed between [1, 1.5] m/s. Meanwhile, their active duration is uniformly distributed between 1 hour and 3 hours. For achieving a more realistic user routine model, we set

approximately 8% users to arrive from 0 pm to 5 pm, which can be considered as the night-shift workers.

Every user owns a smartphone, and among all the smartphones, 80% of their battery levels randomly distributed between [20%, 100%] while the other 20% smartphones' battery levels are randomly distributed between [1%, 20%].

4.2.2 Task Generation

For a wider coverage of the players, we give mobility to the tasks, which has been discussed in the previous Section under the task movement models.

Then we generate tasks data in the following format: $task$ = {'ID', 'latitude', 'longitude', 'day', 'hour', 'minute', 'duration', 'remaining time', 'battery requirement (%)', 'legitimacy' }. We set 10% of the total tasks to be illegitimate tasks while others to be legitimate tasks. Since we simulate a 6-day MCS campaign, the "day" feature is uniformly distributed in $[1, 6]$. The battery requirement for each sub-task is the average of the whole energy requirement for a task divided by the number of its sub-tasks. According to the motivation of fake task injection, the illegitimate tasks tend to drain more battery levels with longer duration and attempt to impact as many as possible players. Therefore, the two groups of tasks have different settings, as shown in Table 4.1b.

As reported by Ontario Hydro, the charging rate is mapped onto three bands known as on-peak hours, mid-peak hours, and off-peak hours. We assume that the periods when smartphone owners using communication services follow similar patterns as hydro usage. Thus, we classify the time in a day into three groups: more-active hours (7 am - 11 am), medium-active hours (12 pm - 5 pm), and less-active hours (6 pm - 6 am). Therefore, with the aim to aggregate in the users' most active hours, 80% illegitimate tasks are in more-active hours, and 20% of them are in medium-active hours in our setting.

4.2.3 Adversarial Self Organizing Feature Map Artificial Neural Network (SOFM-ANN)-Based Attack Modeling

Fake tasks can be created by adding typical properties to simulate realistic attack scenario. Still, the scenario should also attain adequate positioning knowledge to obtain a wide coverage for the extended effect on the victims. After finding the best position for broad coverage, fake tasks influence more users than randomly determined position-based attacks. Since this attack can cover more users, battery level of potential recruits will remain less

than other attack scenarios. Consequently, illegitimate task attacks affect the completion of legitimate tasks due to battery level constraints in the recruited devices.

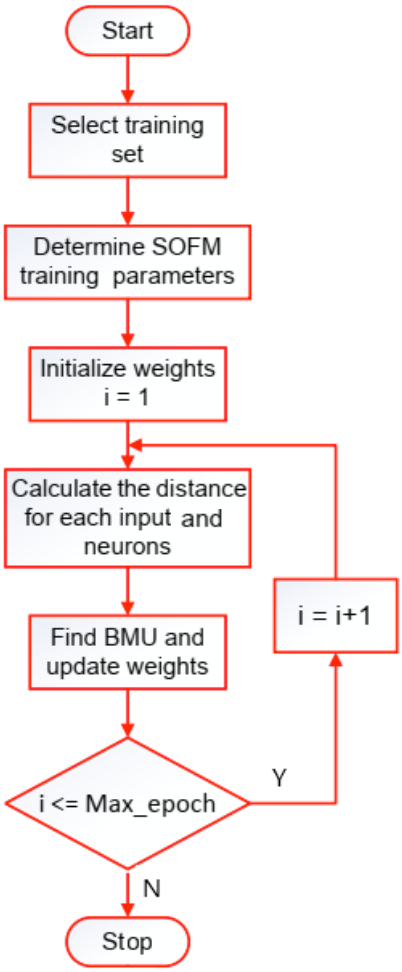


Figure 4.2: Adversarial Self Organizing Feature Map steps

SOFM is an unsupervised learning method [44, 45, 46, 102] for clustering multi-dimensional data into two-dimensional neurons which connect through a pre-defined topology. SOFM is based on competitive learning, similar to winner-take-all networks. The additional feature of SOFM different from winner-take-all is the winner neuron, which is also called the Best matching unit (BMU), is not only updated to be closer to the input coordinates but also

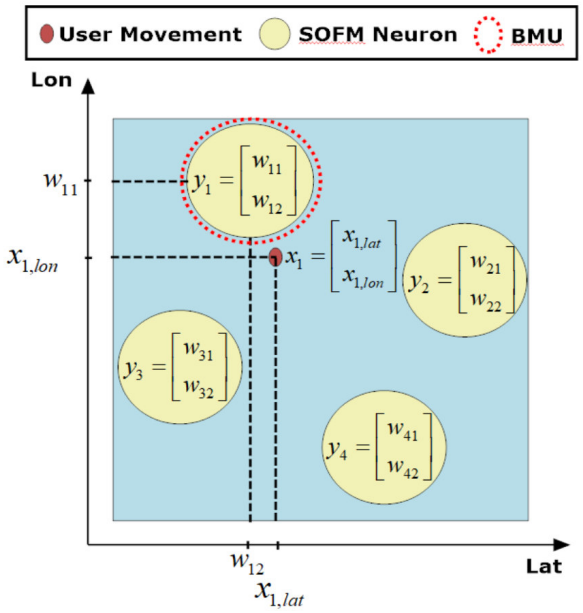


Figure 4.3: Self Organizing Feature Map training and BMU selection according to user movement

connected neurons of the BMU are updated. Therefore BMU and its neighbors are closer to the input coordinates than the other neurons. This process repeats for all inputs for each epoch, and finally, a maximum number of epochs is used as the stopping condition for SOFM, as summarized in Figure 4.2.

Since MCS platforms pursue data aggregation from multiple and various sensory sources, SOFM is one of the best approaches to find the proper locations to increase the impact ratio of the illegitimate tasks on the participants in an MCS campaign. In this work, coordinates of user movements are used as the inputs for a SOFM, as shown in Figure 4.3. The weight of each neuron consists of coordinates (i.e., latitude and longitude) in meters.

The training process in SOFM is very similar to the winner-take-all algorithm, which utilizes distance-based competitive learning. The Euclidean distance is calculated between each neuron's weight vector and the input training vector. Distance function $Dist(X, W)$ for 2 dimensional input vector X and the neuron's weight vector W can be indicated in

the following Equation(4.1),

$$Dist(X, W) = \sqrt{\sum_{j=1}^{2} (X_j - W_j)^2} \tag{4.1}$$

where X is the current 2D input vector, and W is the node's weight vector. The winner neuron of the competitive learning process, also called BMU, has the closest weight vector to the input vector, as shown in Equation(4.2).

$$BMU_X = \arg\min Dist(X, W^k) \quad for \quad k = 1, .., n \tag{4.2}$$

The next step is to determine the BMU's local neighborhood. The weights of the BMU and its neighboring neurons in the SOFM 2D-grid are adjusted to be closer to the input vector. During the learning phase of SOFM, the area of the neighborhood shrinks over time (iteration of the loop), which is represented by the radius in Equation (4.3). Here, σ_0 denotes the initial value of radius, λ denotes a time constant, i is the current time step, and Max_epoch is the number of iteration. The time constant λ is calculated by the Equation(4.4).

$$\sigma(i) = \sigma_0 e^{\left(-\frac{i}{\lambda}\right)} \quad i = 1, 2, 3, ...Max\_epoch \tag{4.3}$$

$$\lambda = \frac{Max\_epoch}{\log_{\sigma_0}} \tag{4.4}$$

After the BMU's neighbourhood area is determined, BMU and its every neighbour neuron need to adjust their weight vector according to the following Equation (4.5):

$$W^{(i+1)} = W^i + \Phi\,(i)\, R\,(i)\, \left(X^i - W^i\right) \tag{4.5}$$

In Equation (4.5), W^i is the old weight vector at time step i and $W^{(i+1)}$ is the updated weight vector at time step $i + 1$. $R(i)$ indicates learning rate which decays over time, as shown in Equation (4.6). It is the same as function in Equation (4.3) Since the farther a neuron is from the BMU, the lower influence its learning should be, here is a parameter $\Phi(i)$ to represent the effect of its learning. $\Phi(i)$ is given by Equation (4.7) where *dist* which is the distance from a neuron to the BMU, and $\sigma(i)$ is the radius of the BMU's local neighbourhood area as calculated by Equation (4.3).

$$R(i) = R_0 e^{\left(-\frac{i}{\lambda}\right)} \quad i = 1, 2, 3, ...Max\_epoch \tag{4.6}$$

$$\Phi(i) = e^{\left(-\frac{dist^2}{2\sigma(i)^2}\right)} \tag{4.7}$$

Hexagonal updating after BMU selection and 2D coordinates of user movement are depicted in Figure 4.3. Latitudes and longitudes of 10,000 users are applied to the adversarial SOFM algorithm. After the training process completes, the maximum value of the user coverage is obtained by SOFM. Therefore, the illegitimate task position, which satisfies maximum user coverage, has the highest impact on the participant mobile devices to drain the devices' battery.

4.2.4 Recruitment Policy

As a crucial procedure in an MCS campaign, recruitment policy determines the mobile device users who really participate in the campaign. In our simulations, we simply consider that the users meet the following three rules to be the players in an MCS campaign.:

$Rule1$: Time of the users' passing by is within the time span of the sub-task.

$Rule2$: Due to the sensing radius limit, the user should be within the range of 100m from the task.

$Rule3$: The device's battery level is sufficient to finish the task.

On the grounds that a task is composed of sub-tasks, we define the energy requirement of a sub-task to be the average of the whole energy requirement for a task divided by the number of its sub-tasks. Besides, if more than half of sub-tasks belonging to a task are complete, we mark the task as completed. Otherwise, we mark the task as an incomplete task.

4.3 Attack Evaluation Under Fixed Number of Neuron-based SOFM

We verify the effectiveness of the SOFM-based clogging attack model for MCS systems through simulations under Crowdsensim simulator [19] in terms of the affected participants, affected recruits, and the energy consumption of the participant devices.

We evaluate two movement models (ZFM and ZLM), initially look into the energy expenditure conditions of fake tasks, and then the recruitment impacted by illegitimate tasks. Based on our recruitment policy, multiple users may get recruited; thus, the population affected by fake tasks is also worth investigation, which is defined as impacted participants. We investigate the impact of the attack model with and without SOFM on recruitment and users separately. All results in this section are the average of ten runs.

4.3.1 SOFM Results

We select two cities in Canada, namely Dryden and County of Brant (Brant as abbreviation), to be coherent with the previous work in [123]. SOFM results that are obtained by Matlab SOM Toolbox are presented in Figure 4.5 for Brant. 16 neurons are selected for SOFM based on the simulation results. We illustrate the final coordinates of SOFM neurons in Figure 4.5a maximum coverage in terms of user movement in Figure 4.5b, and neighbourhood connections of all neurons in Figure 4.5c for Brant. The same outputs from Dryden, which exhibit similar behavior to that in Brant, are shown in Figure 4.4.

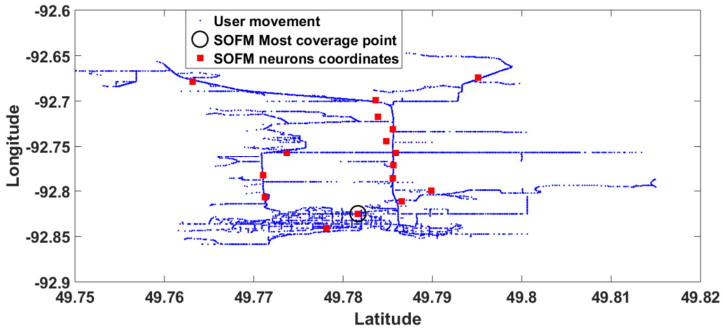
4.3.2 Mitigation on Energy consumption

As seen in Figure 4.6 and Figure 4.7, the ratio of energy consumption under a clogging attack by applying SOFM is significantly higher than that without SOFM, regardless of the city with different volumes of tasks. In most cases, the illegitimate tasks approximately double the battery drain when the attack is based on SOFM.

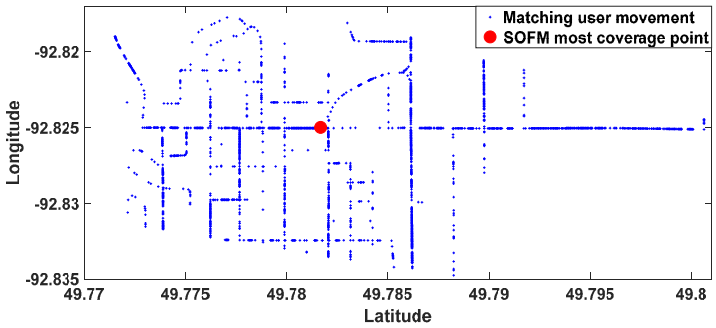
4.3.3 Mitigation on Impacted Recruitment and Users

The ratio of impacted recruitment and participants in both cities increases significantly when SOFM is applied for both movement models. In terms of affected recruitment, the results confirm that SOFM-based modeling of clogging attacks in MCS improves the influence by up to 14%, which can be observed under ZFM with 1000 tasks in Brant. Even in the worst case, the impact ratio increases by roughly 7% from 16.5% to 23.42% in Brant with 500 tasks when attacks are based on the SOFM. As for the impact on participants, the results also yield a clear view that the introduced adversarial model dramatically extends the effect on players in MCS campaigns.

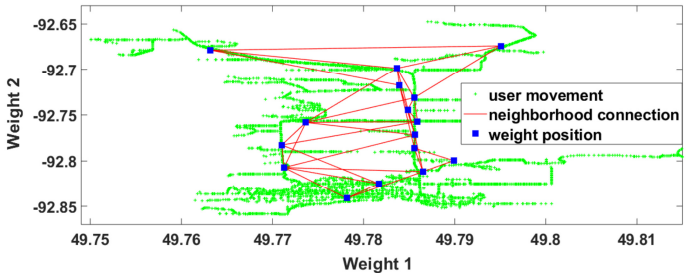
Since Brant is significantly larger than Dryden, the same number of tasks spread around the city with a higher density in Dryden when compared to the task distribution in Brant.



(a) SOFM neurons



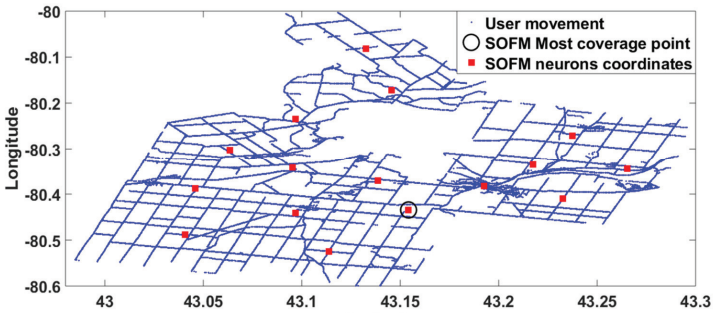
(b) SOFM most user cluster



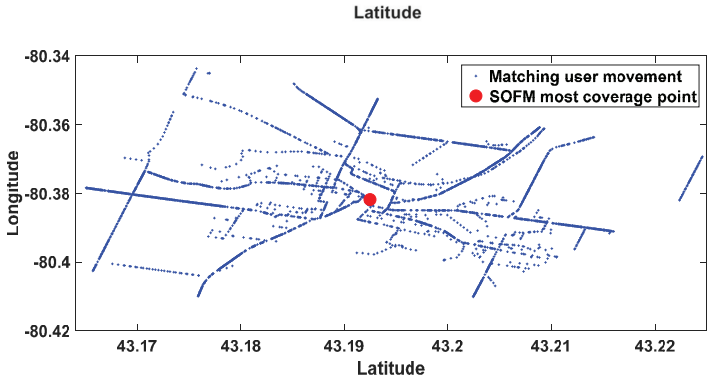
(c) SOFM neuron neighbourhoods

Figure 4.4: Self Organizing Future Map for Dryden

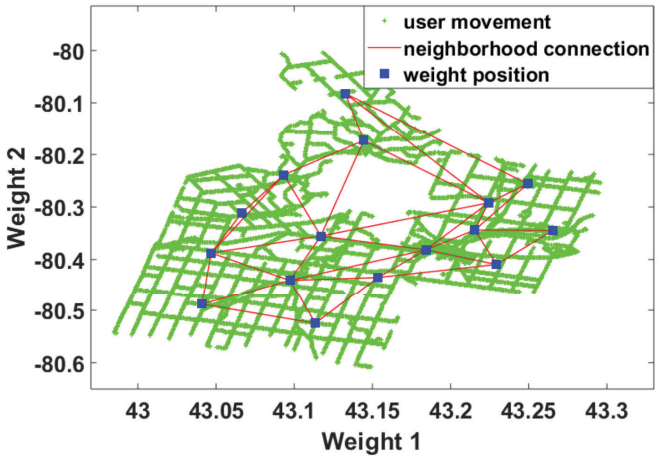
With the fixed user population of 10,000, the impact on users in Brant is reasonably smaller than that in Dryden. This phenomenon yields the simulation results presented in Figure 4.8 and Figure 4.9.



(a) SOFM neurons



(b) SOFM most user cluster



(c) SOFM neuron neighbourhoods

Figure 4.5: Self Organizing Future Map for Brant

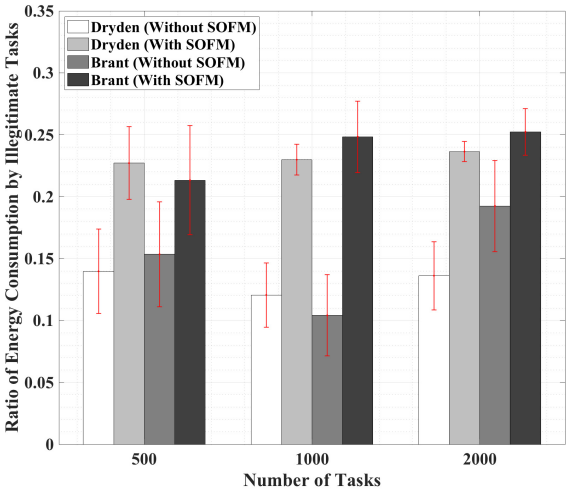


Figure 4.6: Energy consumption by illegitimate tasks (ZFM).

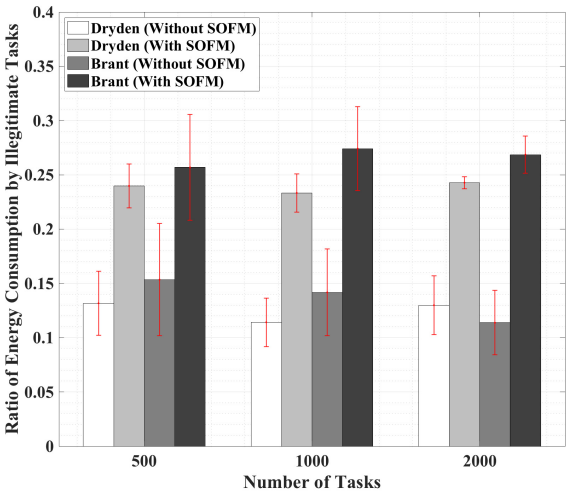


Figure 4.7: Energy consumption by illegitimate tasks (ZLM).

4.3.4 Concluding Remarks

We have introduced an attacker design model for illegitimate task injection (i.e., clogging attacks) by leveraging a self-organizing feature map (SOFM) to increase the affected user population of Mobile Crowdsensing (MCS) campaigns. Illegitimate (fake) task attack zone coordinates are determined by the SOFM through the inputs of mobile device users' mobility patterns. Thus, more participants are affected by the illegitimate task submissions

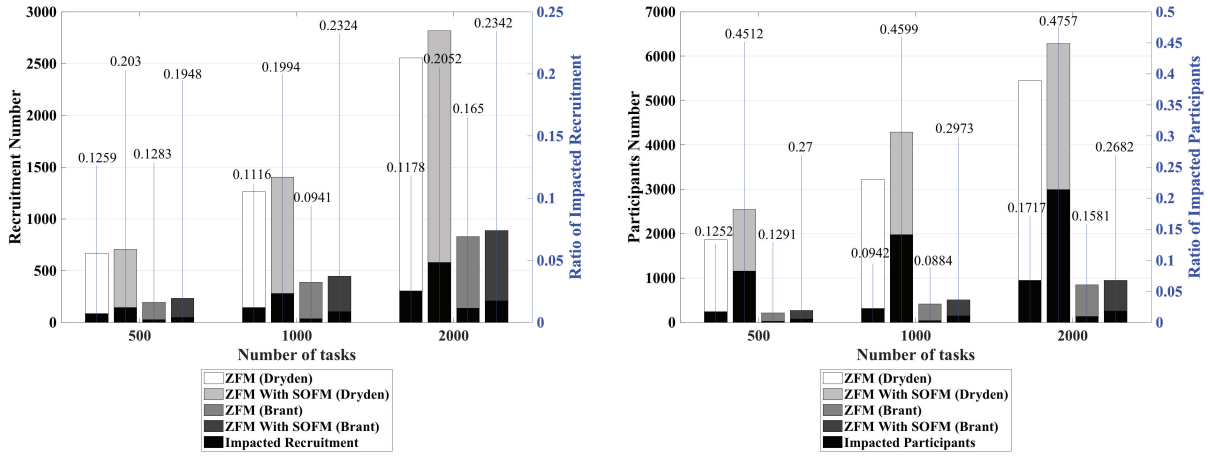


Figure 4.8: ZFM impacts: Recruits (top), Participants (bottom)

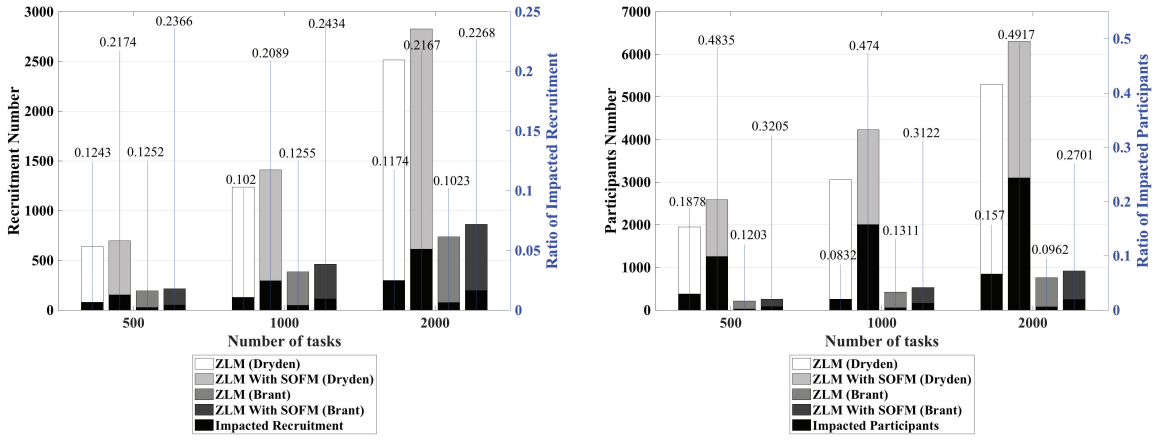


Figure 4.9: ZLM impacts: Recruits (top), Participants (bottom)

when compared to the randomly selected attack zones. Accordingly, the energy levels of participating devices increase significantly, as well. We have shown that when illegitimate task positions are selected through the machine learning-based SOFM clustering method, almost half of the participants are affected with an energy overhead up to 28%. Since all results demonstrate that intelligent attack scenario increases the impact of the attacks, attack detection, and the elimination of attacks are crucial to improve the performance of MCS campaigns, which are addressed by the work presented in Chapter 5.

4.4 Attack Evaluation Under Different SOFM Topologies

In this section, we firstly present the extended SOFM modeling of fake task attacks with various topology, following by the extended SOFM modeling results for two cities. We apply four different SOFM topologies with the diverse number and the type of initialization of the neurons to each city. The number of neurons consists of fix and adaptive, which is calculated from the user movement pattern of each city. Initialization consists of two options, such as random and uniform. According to attack locations coming from four different topologies, we deal with the effect of each attack scenario in terms of three parameters. These parameters include total energy consumption in terms of smartphone battery level, total impacted participants, and total impacted recruits.

4.4.1 SOFM Modeling with Different Topology

The number of neurons is fixed in the previous Section 4.3, and its neuron positions were randomly initialized for the fake task attacks in the MCS scenario. In this work, neuron positions of SOFM are determined by the geometry of the target location. The maximum and minimum values of latitude and longitude are considered to find an effective area that is used to determine the initial neuron positions according to the pre-defined distance among all neurons. We choose pre-defined distance according to the effective area and the maximum number of neurons. If the pre-defined distance is shorter, the required number of neurons should be high; thus SOFM needs more training time. The entire process to determine the number of neurons according to the city coordinates is summarized in Figure 4.10.

4.4.2 SOFM Modelling Results for Clarence-Rockland and Timmins

SOFM based results are divided into two main parts concerning their location: Clarence-Rockland and Timmins. In this work, the different pre-defined neuron distance PN_d as shown in Figure 4.10 is selected to determine the number of neurons for SOFM. PN_d= 0.02 for Clarence-Rockland and PN_d= 0.05 for Timmins are selected for both uniform and randomly initialized SOFM neurons before the training process. After applying PN_d values, the total number of neurons is equal to 11×12 (132) and 9×18 (162) for Clarence-Rockland and Timmins, respectively. This adaptive approach assures satisfactory results

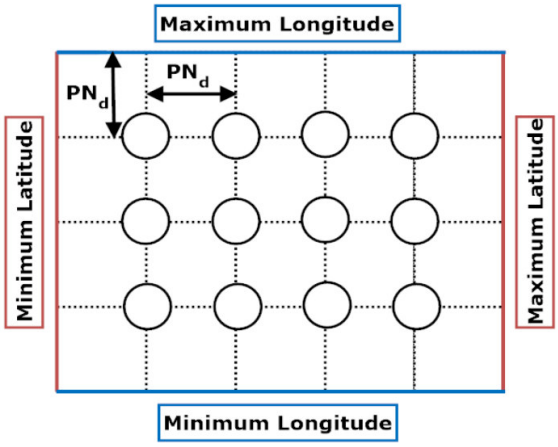
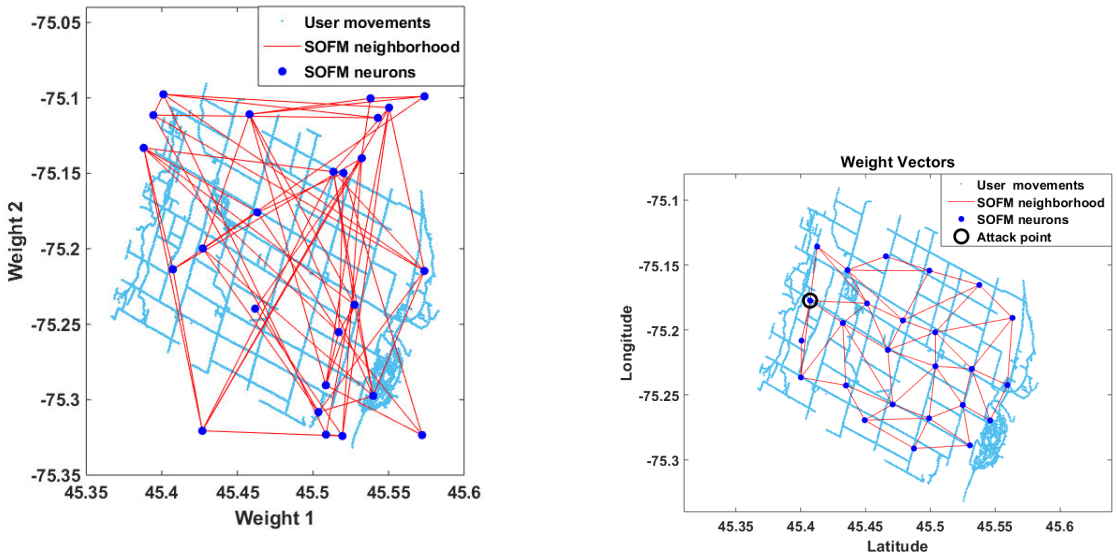


Figure 4.10: Self Organizing Feature Map: Uniform distribution according to pre-defined neuron distance PN_d.

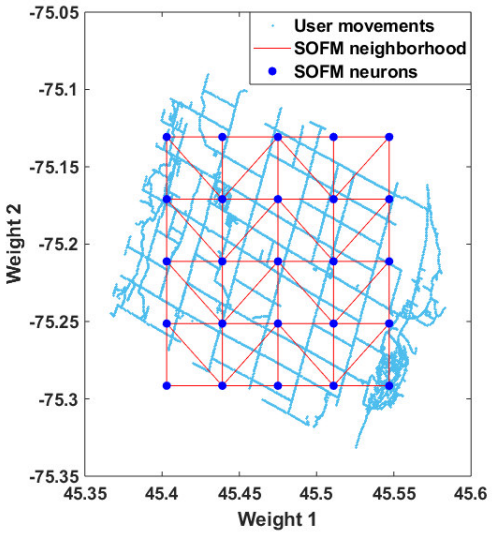


(a) Initial locations of 25 neurons

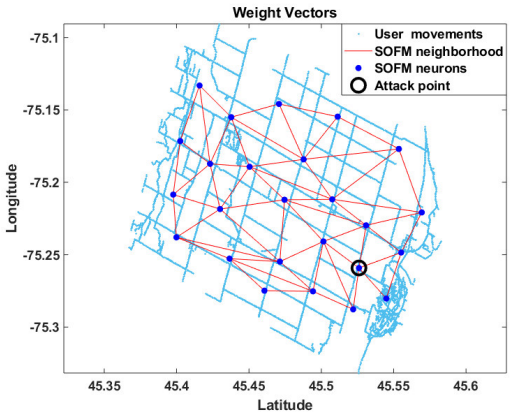
(b) Final and most vulnerable locations of 25 neurons

Figure 4.11: Results of SOFM with randomly generated 25 neurons for Clarence-Rockland

corresponding to reasonable training time. The maximum number of epochs is used as



(a) Initial locations of 25 neurons



(b) Final and most vulnerable locations of 25 neurons

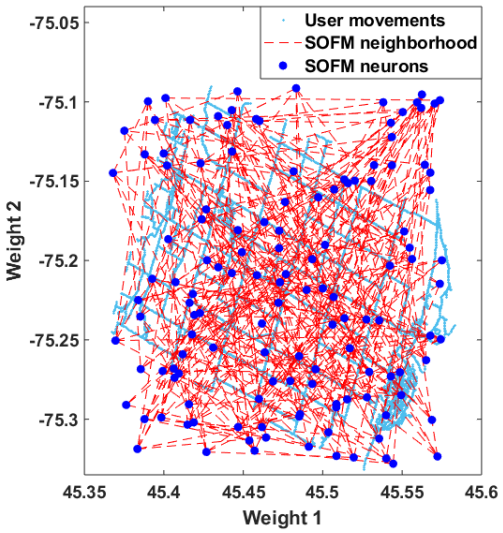
Figure 4.12: Results of SOFM with uniformly generated 25 neurons for Clarence-Rockland

200, and this number is also stopping condition for SOFM, as shown in Figure 4.2.

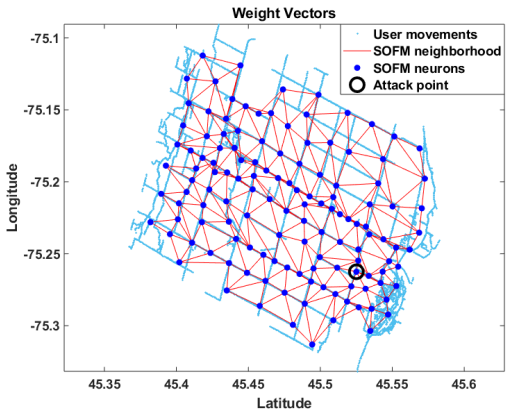
Training process of SOFM with 25 neurons provides attack point as depicted in Figure 4.11b for initialized random location as depicted in Figure 4.11a according to effective area in terms of maximum and minimum values of latitude and longitude for Clarence-Rockland. Instead of randomly determined neurons position, uniformly initialized neurons as depicted in Figure 4.12a provide an attack point as depicted in Figure 4.12b according to pre-defined distance inside the effective area.

Considering the SOFM results on the Clarence-Rockland, random and uniform options for 25 neurons give different attack positions in the MCS campaign. Since the user movement pattern spreads a large area for this region, random and uniform initialization can provide different attack locations.

Training process of SOFM with 132 neurons provides attack point as depicted in Figure 4.13b for initialized random location as depicted in Figure 4.13a according to effective area in terms of maximum and minimum values of latitude and longitude for Clarence-Rockland. Instead of randomly initialized neurons position, uniformly initialized neurons as depicted in Figure 4.14a provide an attack point as depicted in Figure 4.14b according to pre-defined distance inside the effective area. Considering the SOFM results in Fig-



(a) Initial locations of (11 × 12) 132 neurons



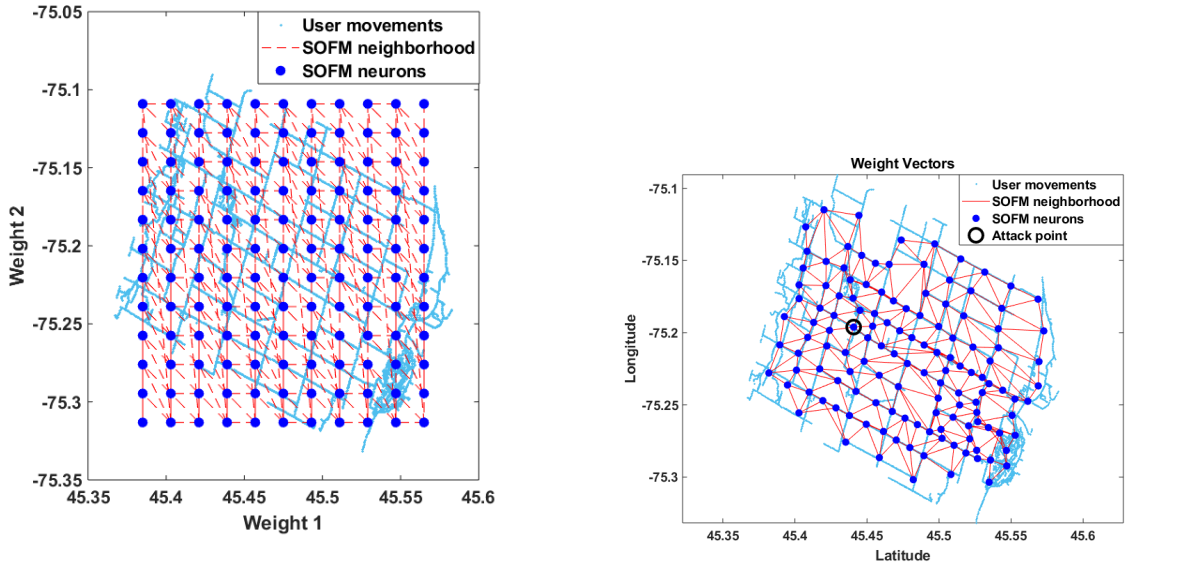
(b) Final and most vulnerable locations of (11 × 12) 132 neurons

Figure 4.13: Coordinate based results of SOFM with randomly generated (11 × 12) 132 neurons for Clarence-Rockland

ure 4.13 and Figure 4.14, random and uniform options for 132 neurons generate different attack positions in the MCS campaign even different than the random and uniform options for 25 neurons as depicted in Figure 4.11 and Figure 4.12, respectively. As a result, user movement pattern can be efficiently covered by 132 neurons both randomly and uniformly initialized neurons of SOFM.

Training process of SOFM with 25 neurons provides attack point as depicted in Figure 4.15b for initialized random location as depicted in Figure 4.15a according to effective area in terms of maximum and minimum values of latitude and longitude for Timmins. Instead of randomly determined neuron position, uniformly initialized neurons as depicted in Figure 4.16a provide the attack point as depicted in Figure 4.16b according to pre-defined distance inside the effective area.

Training process of SOFM with 162 neurons provides attack point as depicted in Figure 4.17b for initialized random location as depicted in Figure 4.17a according to effective area in terms of maximum and minimum values of latitude and longitude for Timmins. Instead of randomly initialized neuron position, uniformly initialized neurons as depicted in Figure 4.18a provide an attack point as depicted in Figure 4.18b according to pre-defined



(a) Initial locations of (11 × 12) 132 neurons

(b) Final and most vulnerable locations of (11 × 12) 132 neurons

Figure 4.14: Coordinate based results of SOFM with uniformly generated (11 × 12) 132 neurons for Clarence-Rockland

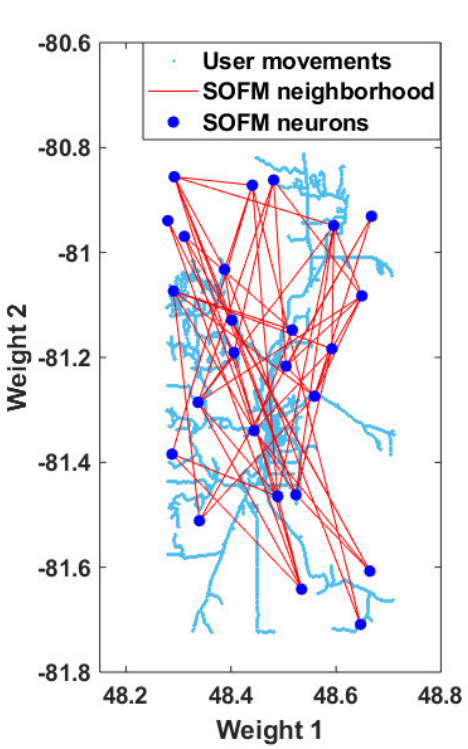
distance inside the effective area.

Since user movement pattern of Timmins as depicted in Figures 4.15-4.18 is denser in the large area than the user movement pattern of Clarence-Rockland, 25 and 162 neurons generate quite close coordinates for illegitimate task location for Timmins. Moreover, user movement is quite dense on the specific area, that's why randomly initialized neurons depicted in Figure 4.15, 4.17 and uniformly initialized neurons in Figure 4.16, 4.18 emerge similar attack location without depending on number of neurons and coordinates.

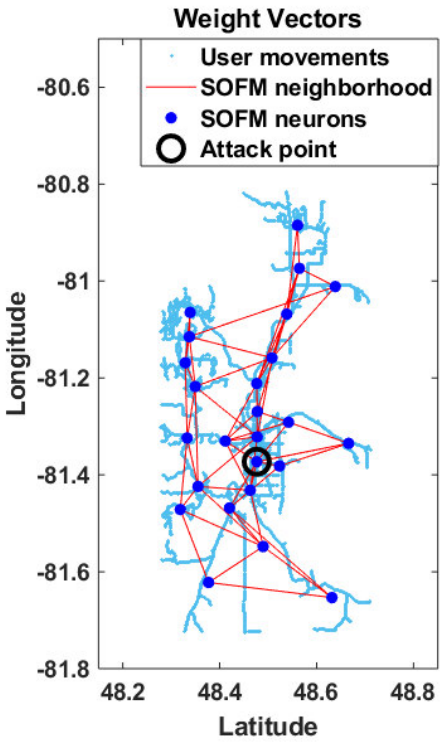
4.4.3 Simulation Examples

We use Folium [1] introduced in Crowdsensim [19] to visualize the simulations. We give two examples that represent the task movement models in the simulation, as shown in Figure 4.19a and Figure 4.19b. In each figure, the red circle is the attack zone, and dark blue popups are the positions of legitimate tasks while red popups are markers of fake

[1] https://pypi.org/project/folium/



(a) Initial locations for 25 neurons



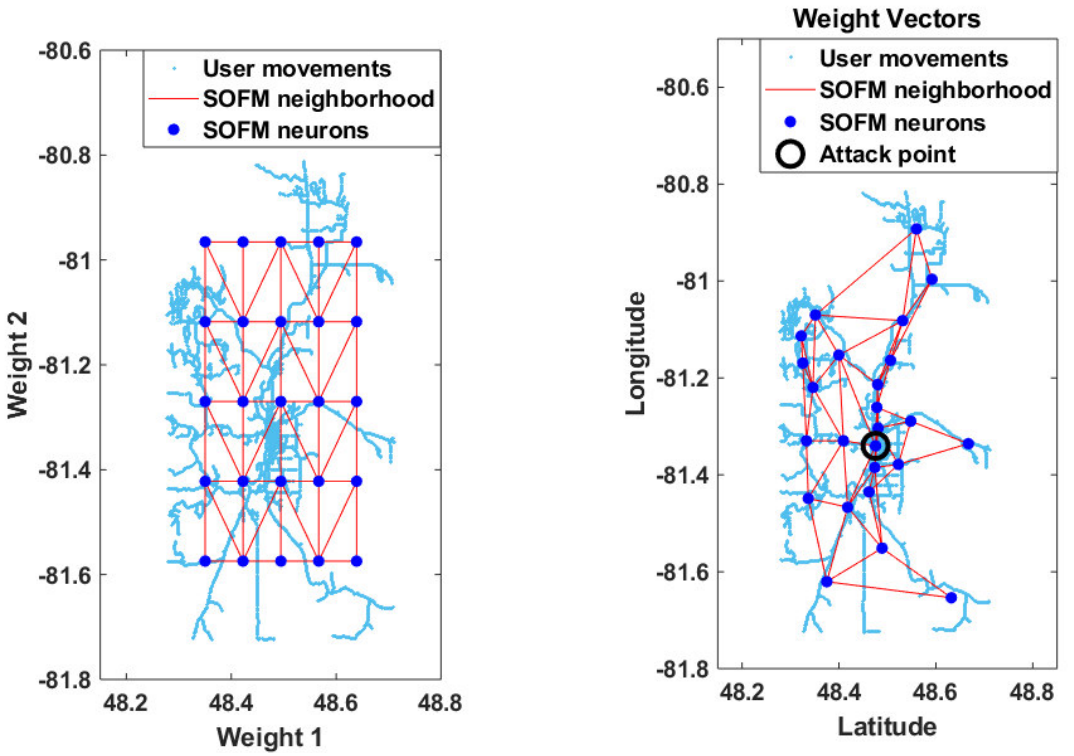
(b) Final and most vulnerable locations for 25 neurons

Figure 4.15: Results of SOFM with randomly generated 25 neurons for Timmins

tasks' locations. The arrows between these points give a clear shifting pattern of the tasks. As shown in Figure 4.19a, a fake task moves beyond the attack area at the left corner. On the other hand, the fake tasks are controlled to move inside of the attack zone, which is noticeable in Figure 4.19b.

Figure 4.20 shows an example of a SOFM-based attack zone (red circle), a randomly-selected attack zone (dark blue circle) and their surrounding user densities (which is calculated by clustering the points on users' routine). As seen from the Figures, SOFM-based illegitimate task location increases the number of potentially impacted recruits compared to the randomly selected illegitimate task location.

Figure 4.21 and Figure 4.22 present a clear view on the two SOFM topologies-based attack zones in Clarence-Rockland and Timmins respectively. According to user movement,



(a) Initial uniform locations for 25 neurons

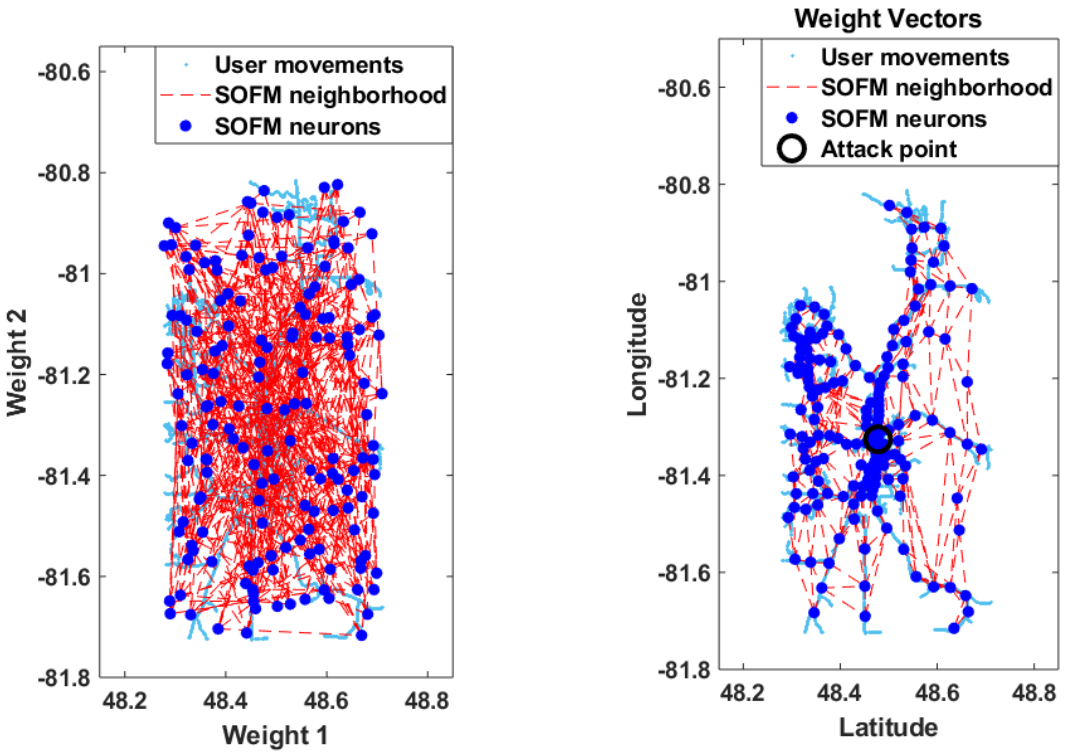
(b) Final and most vulnerable locations for 25 neurons in *(b)*

Figure 4.16: Results of SOFM with uniformly generated 25 neurons for Timmins

Clarence-Rockland is a smaller area than Timmins, but Timmins has more dense user movement in the limited area than Clarence-Rockland. As a result of two different user movement patterns, SOFM positions the tasks on farthest locations from each other in the wide area for Clarence-Rockland, and positions tasks on the closest possible locations to each other in the narrow area under both initialization types and the number of neurons.

4.4.4 MCS results for Battery Consumption

The energy consumed by fake tasks increases significantly in both cities, as shown in Figure 4.23a and Figure 4.24a. Here, it is worth to note that one battery drain unit corresponds to 1% of a fully charged battery. In Clarence-Rockland, the attack based

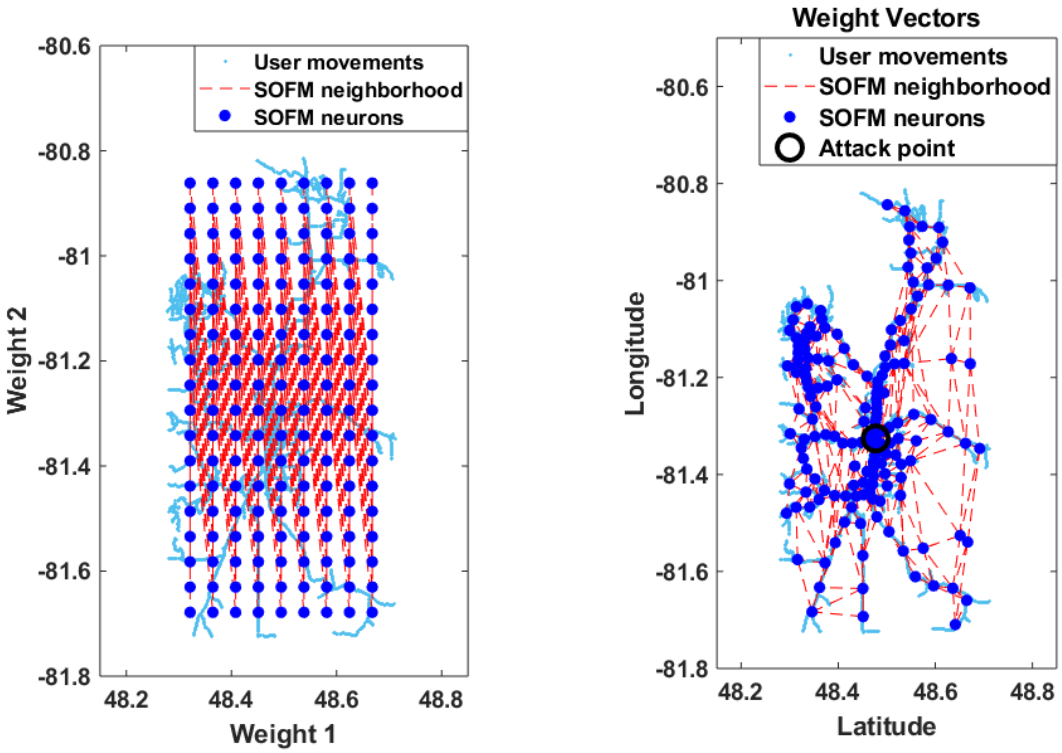


(a) Initial random locations for (9×18) 162 neurons

(b) Final and most vulnerable locations for (9×18) 162 neurons in *(a)*

Figure 4.17: Coordinate based results of SOFM with randomly generated (9×18) 162 neurons for Timmins

on an 11×12 random SOFM performs the best with a total battery drain of about 407 units from all recruits, which is the double of the consumption without SOFM modeling. While in Timmins, 5×5 Uniform modeling makes the attack achieve the highest energy expenditure of almost 400 units, which is four times the battery drain without SOFM. Comparing with Clarence-Rockland, the SOFM seems to be more effective in Timmins because of a more notable rising range. When comparing the two movement models, no significant differences can be observed in battery consumption.



(a) Initial uniform locations for (9×18) 162 neurons

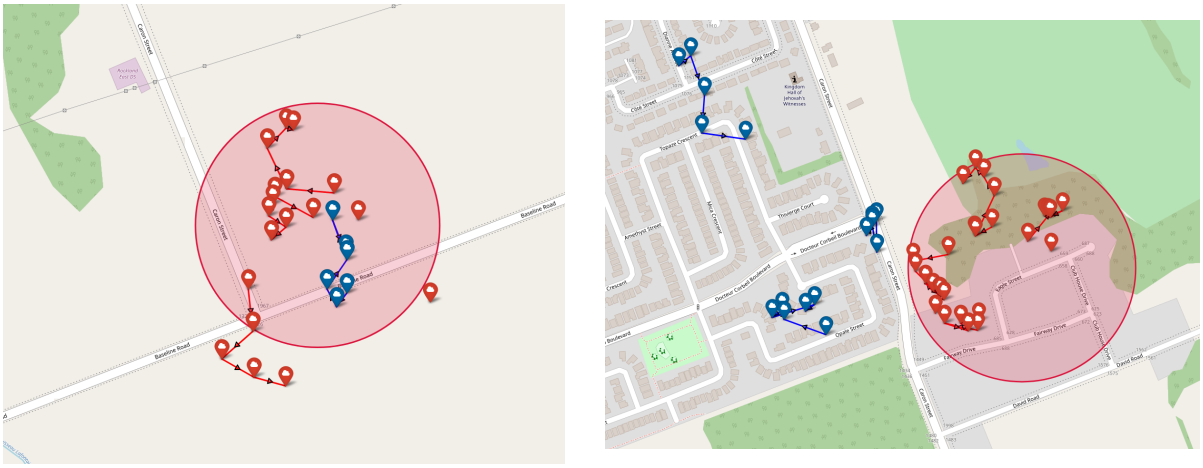
(b) Final and most vulnerable locations for (9×18) 162 neurons in *(b)*

Figure 4.18: Coordinate based results of SOFM with uniformly generated (9×18) 162 neurons for Timmins

4.4.5 MCS results for impacted participants and recruits

Although there are multiple users that meet recruitment policy, only a limited number of participants can suffice a task's recruitment requirements. In the simulation, how to choose the one who finally participates in an MCS campaign is left to our future research agenda. We assume that all candidates can be the potential victims of a fake task injection; hence, we use impacted participants to quantify the effect of such adversarial behavior. Besides, we still investigate the ratio of impacted recruits to find out the practical influence on the task recruitment stage.

With adversarial SOFM-based modeling, the percentage of impacted participants and recruits grows significantly in both cities for two movement patterns. In Clarence-Rockland,



(a) ZFM Task Movement example in simulation (Location: Clarence-Rockland)

(b) ZLM Task Movement example in simulation (Location: Clarence-Rockland)

Figure 4.19: Two taks movement examples in Clarence-Rockland

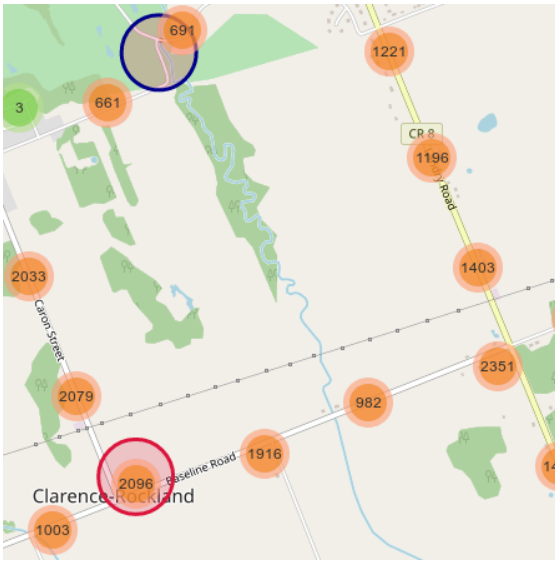


Figure 4.20: Two attack zones and the surrounding user density. Red circle: SOFM-based attack zone. Dark blue circle: random-selected attack zone.

the ratio of affected participants rises from 16% (without SOFM) to 34% in ZFM when

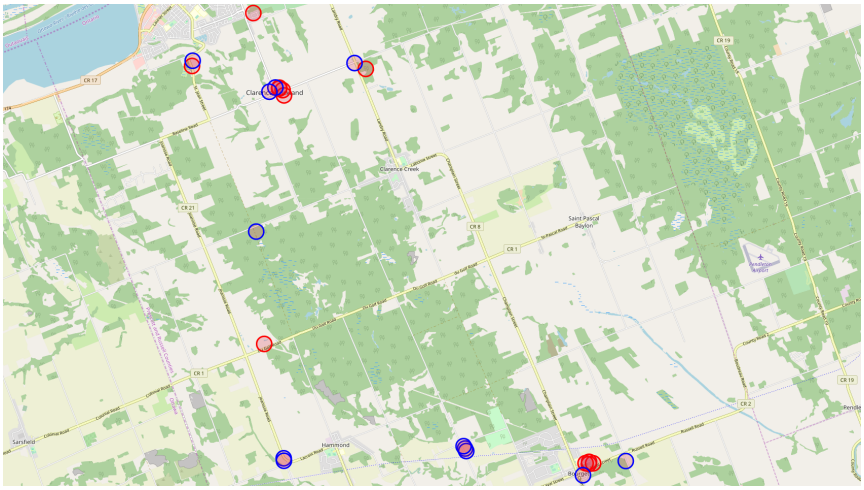


Figure 4.21: All SOFM-based attack zones in Clarence-Rockland. Red circles represent 5×5 topology-based attack areas while blue ones are 11×12 topology modelled attack areas.

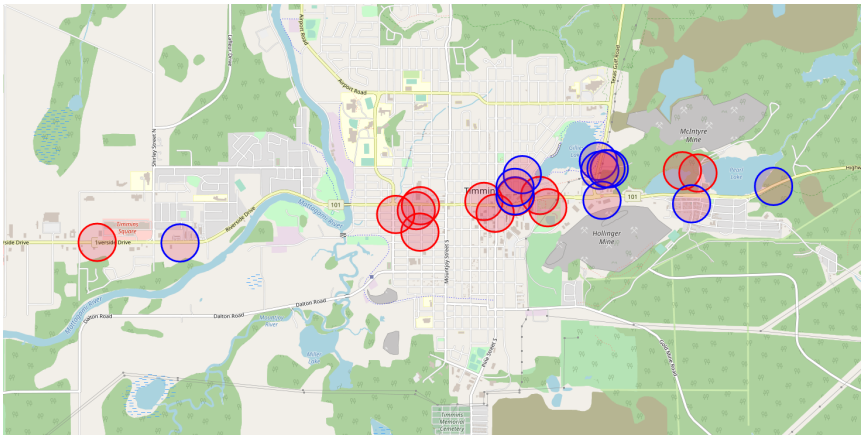


Figure 4.22: All SOFM-based attack zones in Timmins. Red circles represent 5×5 topology-based attack areas while blue ones are 9×18 topology modelled attack areas.

applying 11×12 random SOFM to model the clogging attack (i.e., fake task injection). Correspondingly, the number of participants within the attack range increases from 174 to 435. No apparent difference is noticed in ZLM where 11×12 random SOFM improves the effect to 32% (i.e. 423 participants are potential victims) as presented in Figure 4.23b. Comparing with Clarence-Rockland, the results prove that adversarial SOFM-based modeling can impact up to 46% of the overall participants by using 5×5 uniform SOFM while

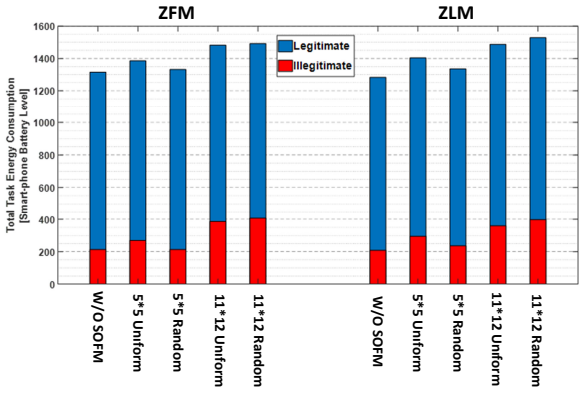
randomly selected attack regions can only affect up to 15% participants in ZLM.

The impacted recruits have the same pattern as the affected participants. When the zone-free movement model is applied, totally 222 out of 914 (24%) recruits suffer from the fake tasks modeled by 11×12 random SOFM in Clarence-Rockland whereas 194 out of 558 (35%) are victims under 5×5 uniform SOFM-based attack in Timmins. When ZLM model is used, the same ratio of 24% influence on recruits is caused by 11×12 random SOFM in Clarence-Rockland, and 37% of the recruits are impacted by 5×5 Uniform SOFM in Timmins, which is 2% higher than the corresponding value in ZFM.

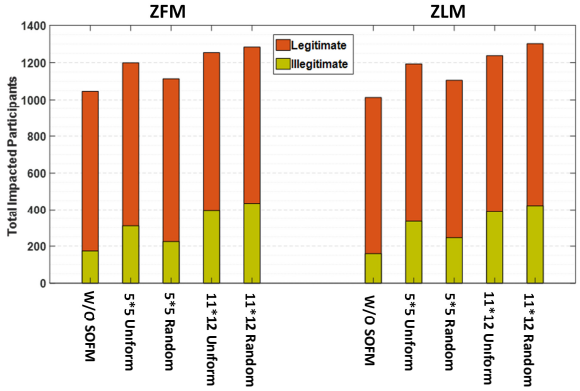
In brief, the results in two different cities show the magnified attack effect with SOFM modeling. It is also noticeable that different topologies of the SOFM models perform diversely in these cities, more specifically, 11×12 Random-based SOFM modeling attack is the most powerful in city Clarence-Rockland while 5×5 Uniform-based SOFM fake task injection impacts the most participants and consume the most energy of smartphones.

4.4.6 Concluding Remarks

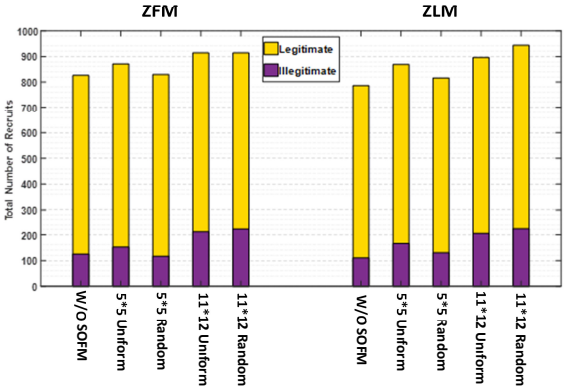
In this chapter, for the first time in literature, we have introduced a new model for illegitimate task injection (i.e., clogging attacks) in MCS by leveraging an adversarial SOFM to increase the affected user population of MCS campaigns. Illegitimate (fake) task attack zone coordinates are determined by the SOFM through the inputs of mobile device users' mobility patterns. Thus, more participants are affected by the illegitimate task submissions when compared to the randomly selected attack zones. Through numerical results, we have shown that different SOFM topologies can significantly affect the performance of intelligently designed fake task injection attacks. Consequently, the battery usage of participating devices is significantly increased as well. We have shown that when illegitimate task positions are selected through the adversarial machine learning-based SOFM clustering method, almost half of the participants are affected with an energy overhead up to 28%. Since numerical results also demonstrate that intelligently designed attack scenarios can increase the impact of the adversaries. The SOFM-based modeling can serve as a reliable tool to be used in the detection and mitigation of adversarial behavior in MCS campaigns to reduce the roadblocks against wide adoption of MCS as a large scale monitoring solution.



(a) Total energy consumption in terms of smartphone battery level

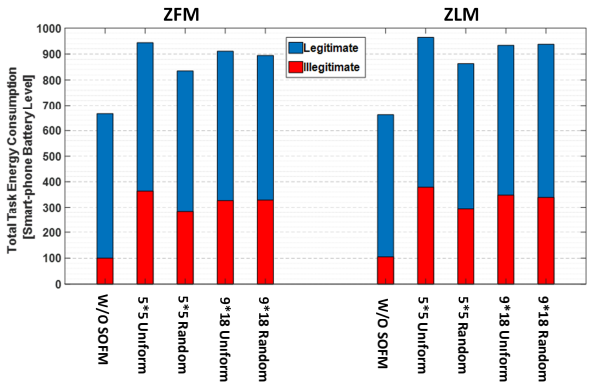


(b) Total impacted participants

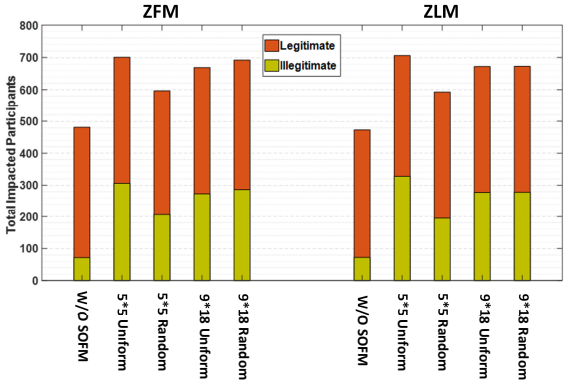


(c) Total impacted recruits

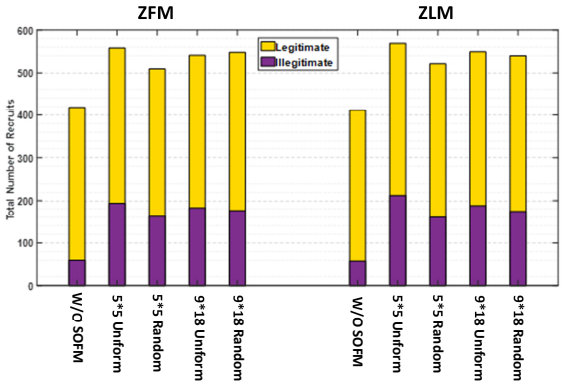
Figure 4.23: MCS simulation results of Clarence-Rockland for adversary modeling with and without SOFM



(a) Total energy consumption in terms of smartphone battery level



(b) Total impacted participants



(c) Total impacted recruits

Figure 4.24: MCS simulation results of Timmins for adversary modeling with and without SOFM

Chapter 5

Ensemble Machine Learning Against Adversarial AI-driven Fake Task Submission in Mobile Crowdsensing

5.1 Introduction

In this chapter, we present machine learning-driven mitigation of self-organizing feature map (SOFM)-based fake task injection attacks in Mobile Crowdsensing (MCS) systems. To this end, we model and train an ensemble classifier, namely Gradient Boosting, so to identify and filter out illegitimate tasks injected/submitted to the MCS platform. According to our numerical results obtained under the dataset generated by the realistic crowdsensing scenarios under Crowdsensim simulator [19], up to 23% energy saving can be achieved through ensemble classification of submitted tasks, and impacted candidates can be reduced from ∼47.6% to ∼12% compared to a baseline scenario where no pre-assessment of submitted tasks exists.

5.2 Machine Learning-driven Mechanism against Fake Task Injection

Under normal conditions, a typical MCS platform is expected to follow the steps listed below:

- The legitimate end-users initiate tasks into the MCS server to request sensing data on specific phenomena.
- The users are recruited to complete these tasks according to the recruitment policies.
- The selected participants collect sensed data and submit them to the sensing servers at the MCS platform.
- The participants get rewarded based on the quality and value of the data they have sensed.

Indeed, due to various vulnerabilities in the MCS platforms, the above mentioned four steps are not always straightforward. Thus, as reported in [123], malicious end users can submit fake tasks into the MCS platform so to drain the batteries of the participating devices, as well as to clog the sensing servers at the MCS platform. Indeed, while the malicious end users inject fake tasks into the MCS platform, cooperative end users submit legitimate tasks. If the system is unable to detect the fake tasks, the tasks will reach the mobile devices, drain their batteries, and even with further malicious motivations (e.g., stealing private information of users). With the existence of such threats, users would consider an MCS system as entrusted and would refuse to join the MCS campaigns. To this end, we aim to investigate a solution to detect illegitimate tasks, and therefore design an MCS mechanism embedded with machine learning, which is illustrated in Figure. 5.1. Before assigning the tasks, a machine learning module is applied to differentiate the illegitimate tasks from legitimate tasks. Therefore, the task that is marked as the illegitimate will not be assigned to any participant. To guarantee that the tasks received by the participants are legitimate, the performance of machine learning becomes the key to achieve the success of the proposed mechanism. Gradient Boosting is chosen in this work to determine fake tasks as it has been proven to be a more satisfactory algorithm in Chapter 3.

Boosting builds on the concept of weak learners, which can be adjusted to improve the overall accuracy as follows: A weak learner is considered to perform slightly better than the random process in the worst case. Thus, new weak learners are added sequentially to learn more difficult patterns. Decisions are taken concerning majority voting, which is based upon the predictions of the weak learners, and weighted by their individual accuracy. AdaBoost and related algorithms were recast in a statistical framework first by Breiman [5], calling them Adaptive Reweighting and Combining (ARCing) algorithms. This framework was further improved by Friedman and called Gradient Boosting Machines [20]. The statistical framework transforms boosting to a numerical optimization problem. Therefore, the objective is to minimize the loss function by adding weak learners and the

gradient descent approach, which determines the direction of the loss function. To satisfy this condition, one new weak learner is added at a time, and existing weak learners in the model are frozen and remain unchanged.

Gradient Boosting is an effective ensemble learning algorithm that is used in various use cases including energy theft detection [87], road condition monitoring [95] and traffic flow forecast [61]. The algorithm is built on the basis of gradient descent which operates to find a goal function, $f(x)$. Given a training set $\{(x_i, y_i)\}_{i=1}^{n}$, at m-th iteration, the negative gradient of the loss function ($L(y, f(x))$) determines the direction of the steepest descent as shown in Eq. 5.1.

$$-\left[\frac{\partial L\left(y, f\left(x\right)\right)}{\partial f\left(x\right)}\right]_{f(x)=f_{m-1}(x)} \tag{5.1}$$

In addition to the direction, line search procedure is applied to determine the step length σ_m. Friedman algorithm generates N distinct predictions on σ_m for each region R_1 ,..., R_N. Combining the learning rate λ, the model at the m-th iteration can be formulated as in Eq. 5.2 [88] where $\phi_{(x)}$ denotes a surrogate loss. Then, the final aim model can be formulated as in Eq. 5.3 where M stands for the number of iterations.

$$f_m\left(x\right) = \lambda\sigma_m\phi_{(x)} \tag{5.2}$$

$$f\left(x\right) = argmin_\gamma \sum_{i=1}^{n} L\left(y_i, \gamma\right) + \sum_{m=1}^{M} \lambda\sigma_m\phi_m\left(x\right) \tag{5.3}$$

In this chapter, we use deviance as the loss function in Gradient Boosting. We set the 150 boosting stages, learning rate as 0.1, and the maximum tree depth as three. Meanwhile, instead of training once, we design a continuous learning strategy. In the beginning, the data obtained on the first day is used as a learning set to predict the legitimacy of tasks on the second day. Then the data during the first two days are combined to estimate the legitimacy of tasks on the third day. The process works by continuously combining all the data before one day to predict the data coming on that day until the last day.

Crowdsensim simulator [19] is introduced as the base of our simulations. In the simulation, we follow the process and adopt the simulation settings discussed in Chapter 4. Specifically. we use the task data generated in Section 4.3, and we generate user movement data which involves the routines of 10000 users during six days in format {'ID', 'latitude', 'longitude', 'day', 'hour', 'minute' }.

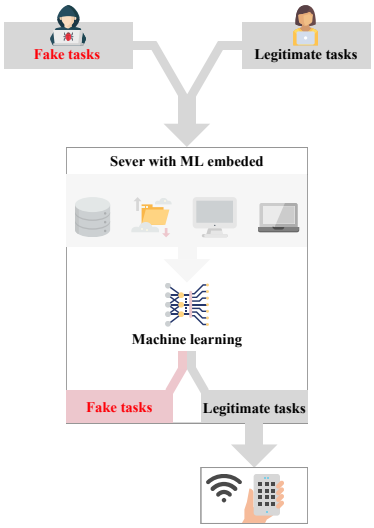


Figure 5.1: Machine learning embedded MCS mechanism: the fake tasks can be eliminate before spreading out to participants.

Before we apply the machine learning algorithm, we abstract another two features from *task*: 'OnPeakHours' and 'GridNumber' as we have discussed in Chapter 3.

We use the same recruitment policy as described in Section 3.4, which is spatial, push tasking schema according to the classifications in [84]. It requires a group of participants at specific locations within the 100 meters radius from the task. The server chooses the users passing by the tasks when the tasks are active, and their device battery is enough for finishing the task. If there is more than one user meeting the requirements, the server automatically selects the one who is the nearest from the task.

We also adopt the same task movement modeling we have designed in Section 3.3 to increase the coverage of tasks and achieve higher successful recruitment possibilities. The modeling consists of two movement patterns, namely Zone-free Movement (ZFM) model and Zone-limited Movement (ZLM) model. In the ZFM model, the legitimate tasks and the illegitimate tasks are free to move without any limit while the legitimate tasks have constrained the ability to move within the attack zones, and legitimate tasks are never permitted to shift into the attack zones.

5.3 Mitigation of Adversarial AI-driven Fake Task Submission

5.3.1 Machine Learning Performance

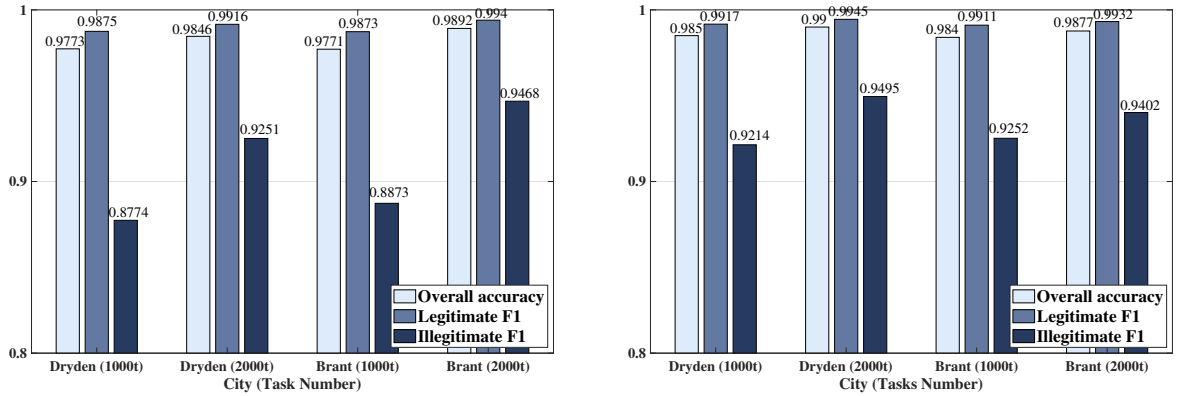


Figure 5.2: Machine learning performance: Zone-free Movement (top), Zone-limited Movement (bottom)

We use overall accuracy, legitimate F_1 score and illegitimate F_1 score to assess the performance of machine learning algorithm. The calculations are shown in Equation 3.2 and Equation 3.1 where TP, TN, FP, FN refer to true positive, true negative, false positive and false negative respectively.

As shown in Figure. 5.2, Gradient Boosting presents its feasibility in both task movement models. The overall accuracy is always high than 97.7% in all cases. Even in the worst case, the overall accuracy, the legitimate F_1 score, and the illegitimate F_1 score can reach 97.73%, 98.75%, and 87.74%, respectively, which can be seen in Dryden with 1000 tasks in ZFM model. It is noticeable in most cases, the machine learning performance in the ZLM is better than that in the ZFM, which is understandable because the illegitimate tasks in ZFM are controlled to be staying in the attack regions.

5.3.2 Saved Energy from Devices

In this sub-section, we present the energy savings when machine learning is applied. The tasks without any participant are out of scope because it will not be able to drain the

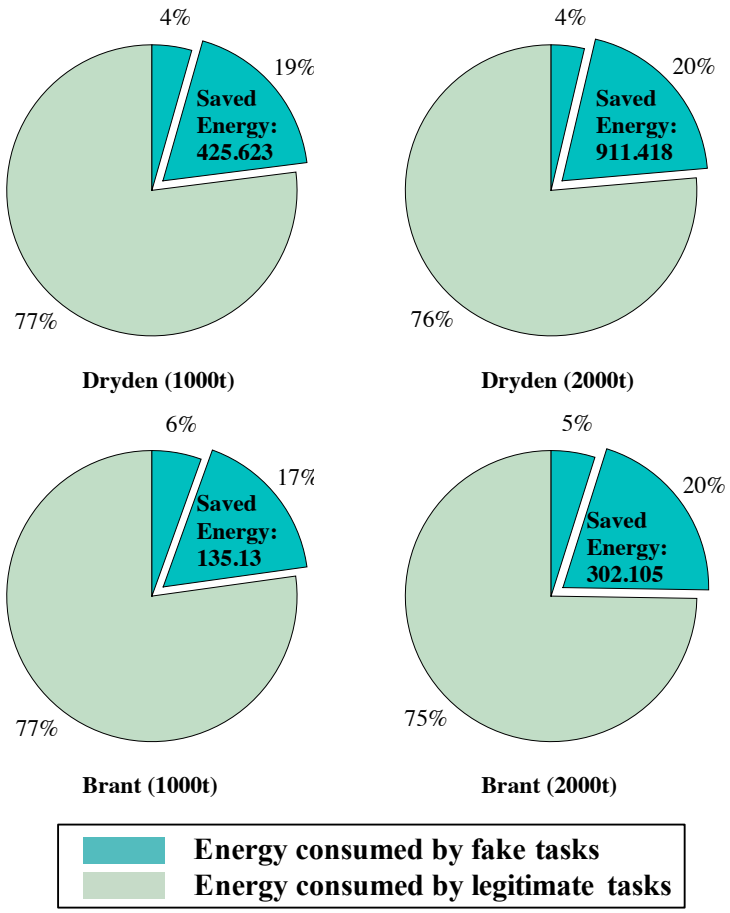


Figure 5.3: Energy savings in ZFM model

battery from devices if no user is recruited to complete the task, even if it is an illegitimate task. We use the battery percentage level(%) to quantify energy consumption. For instance, if a task requires a 2% battery drain from the devices, we define energy usage as 2.

As we can see from the pie charts, as shown in Figure. 5.3 and Figure. 5.4, the darker parts (dark cyan and dark orange) are energy drained by fake tasks, and the exploded parts are saved energy when machine leaning recognizes the fake tasks and eliminates them from MCS system.

Energy savings in two task movement models don't show too much difference due to the uncertainty of successful recruits due to the non-deterministic movement of users. Gradient

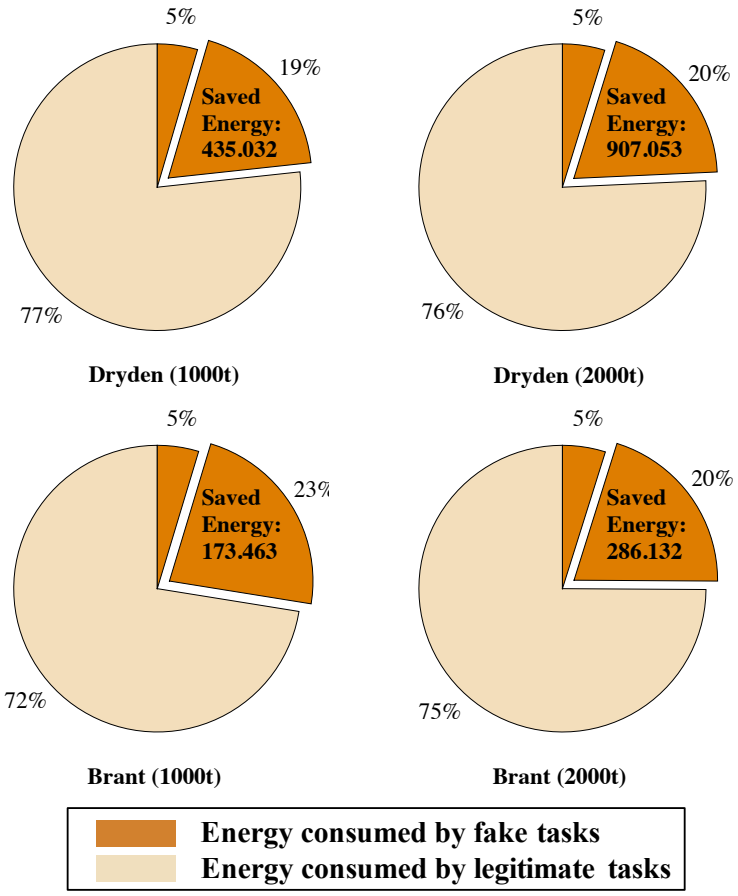


Figure 5.4: Energy savings in ZLM model

Boosting-based machine learning mechanism saves around 20% (i.e., 17% - 23%) energy consumption in all the cases. In the ZFM model, the best saving happens in Dryden with 2000 tasks where the saved energy is 911.418, taking up 20% of total energy utilization. The energy conservation is even better in Brant with 1000 tasks performing the zone-limited movement, which is approximately 23%.

5.3.3 Protected Candidates and Recruits

In our recruitment policy, one task may have a group of mobile device users to be candidates. To be thoughtful, we consider the candidates to be the users potentially affected by

Table 5.1: Saved candidates with ML from fake task injection

(a) Zone-free Movement Model

City	**Dryden**		**Brant**	
Num of Task	1000	2000	1000	2000
Total Candidates	4290	6287	505	947
Impacted Candidates Without ML	1976	2992	150	254
Tendency (↓ OR ↑)	↓	↓	↓	↓
Impacted Candidates With ML	569	756	45	11
Saved Candidates	1407	2236	105	234
Saved Candidates Ratio	32.75%	35.59%	20.95%	19.26%

(b) Zone-limited Movement Model

City	**Dryden**		**Brant**	
Num of Task	1000	2000	1000	2000
Total Candidates	4231	6307	527	915
Impacted Candidates Without ML	2007	3102	165	248
Tendency (↓ OR ↑)	↓	↓	↓	↓
Impacted Candidates With ML	550	932	45	61
Saved Candidates	1457	2170	120	187
Saved Candidates Ratio	34.41%	34.4%	22.81%	20.39%

Table 5.2: Protected recruit with ML from fake task injection

(a) Zone-free Movement Model

City	**Dryden**		**Brant**	
Num of Task	1000	2000	1000	2000
Total Recruit	1402	2815	446	887
Impacted Recruit Without ML	280	578	104	208
Tendency (↓ OR ↑)	↓	↓	↓	↓
Impacted Recruit With ML	75	131	29	57
Saved Recruit	205	447	75	151
Saved Recruit Ratio	14.62%	15.88%	16.82%	16.97%

(b) Zone-limited Movement Model

City	**Dryden**		**Brant**	
Num of Task	1000	2000	1000	2000
Total Recruit	1409	2825	460	862
Impacted Recruit Without ML	294	612	113	196
Tendency (↓ OR ↑)	↓	↓	↓	↓
Impacted Recruit With ML	78	169	30	43
Saved Recruit	216	443	83	153
Saved Recruit Ratio	15.38%	15.7%	18.05%	17.76%

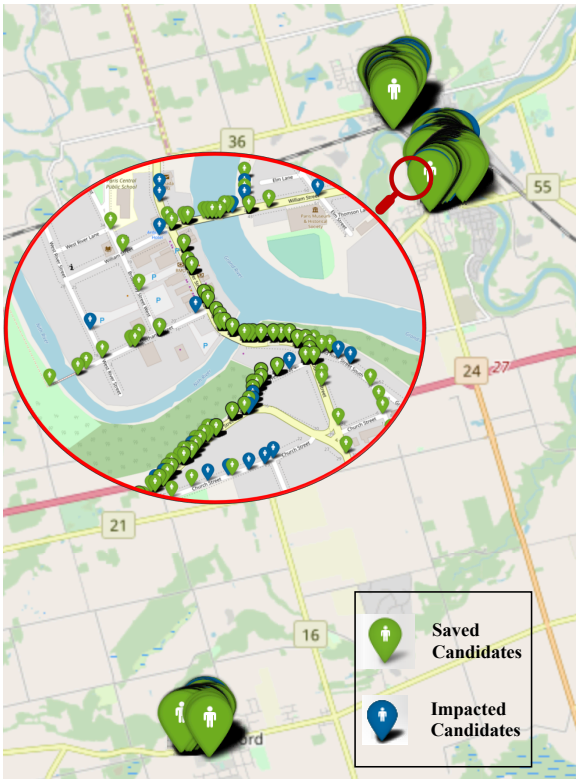


Figure 5.5: The impacted population after applying Machine Learning. Task movement model: ZFM. City: Brant. Task number: 1000. (Sum of ten runs)

the attacks and analyze the impacted candidates with and without machine learning. We also care about the number of users been recruited into tasks, which we define as "recruit" in this part.

We summarize the saved candidates and recruit when machine learning is applied in Table 5.1 and 5.2. For a better understanding of the impact before and after machine learning, we insert a tendency row in the table, which is filled with grey. Obviously, the impacts are alleviated with machine learning.

As shown in Table 5.1, under the Zone-free Movement model, 2236 candidates are protected from fake task injection in Dryden with 2000 tasks, which contribute approximately 35.59% of total candidates. And under the Zone-limited Movement model, around 34.41% of candidates are saved in Dryden with 1000 tasks. Even in the worst case, there are still 234 (19.26%) candidates protected from attacks.

Figure 5.6: The impacted population after applying Machine Learning. Task movement model: ZFM. City: Dryden. Task number: 1000. One time run.

We visualize the effect on candidates in the simulation, as shown in Figure 5.5 and Figure 5.6. The number of impacted candidates in Dryden is about ten times as the number in Brant. Thereby, for a better visual effect, we visualize the sum of ten times' simulation in Brant and one time run in Dryden. In the two maps, the green popups are the locations of saved candidates, while the blue ones are locations of still impacted candidates. We amplify partial areas in red circles, which are clearer in detail.

Obviously, the number of candidates is certainly higher than the number of recruits. In Table 5.2, we can notice the average protected recruit with machine learning is around 15%. As the best case, 83 recruits are spared from illegitimate tasks in Brant with 1000 tasks, which is 18.05% of the total recruit. Even in the worst case, there is still 14.62% of recruits are saved in Dryden with 1000 tasks.

5.3.4 Concluding Remarks

In this chapter, we have proposed an ensemble machine learning-based mitigation against self-organizing feature map modeled fake task attacks designed in Chapter 4 for Mobile Crowdsensing (MCS) platforms. Gradient Boosting is chosen as an efficient algorithm

with competitive performance on differentiating two categories of tasks: legitimate tasks and illegitimate tasks. Specifically, Gradient Boosting-based ensemble learning can achieve up to 98.9% overall accuracy, 99.4% legitimate F_1 score, and 94.6% illegitimate F_1 score. With ensemble machine learning embedded in the MCS system, majority of the fake tasks can be eliminated, which leads to up to 23% energy saving, and impacted candidates can be reduced from ∼47.6% to ∼12% compared to a baseline scenario where no pre-assessment of submitted tasks exists. The simulation results integrated with ensemble machine learning demonstrate the efficiency, effectiveness, and emergency of an integrated task legitimacy assessment module that utilizes a machine learning-based model to identify fake task injection even if the attacks were launched by adversarial Artificial Intelligence (AI).

Chapter 6

Conclusion and Future Directions

Mobile Crowdsensing (MCS) systems are becoming important parts of Internet of Things (IoT)-based services and applications as they offer flexibility, convenience, and low-cost benefits as a result of the ubiquitous availability and non-dedicated deployment of built-in sensors in smartphones, personalized devices, and even intelligent vehicles. As mentioned earlier, despite their numerous benefits, MCS platforms face various security vulnerabilities. In this thesis, we have investigated the possibility of using Artificial Intelligence (AI)-based solutions in securing the MCS campaigns. We have reviewed the state of the art in MCS systems from the standpoint of security, privacy, and trust, followed by an overview of AI-based cybersecurity solutions.

Among AI-based solutions, we have focused on Machine Learning (ML) algorithms to detect adversaries that inject malicious tasks as state of the art reports the strength of machine intelligence in smart environments where various sensing paradigms are employed [26]. We have particularly focused on location-based and task-initiated DoS attacks that consume sensing, communication, processing, and storage resources of participating devices in an MCS system with the ultimate goal of leaving fewer resources for legitimate and critical crowdsensing tasks.

Two ensemble approaches (i.e., random forest and gradient boosting) based on ML have been considered to train the system and then to forecast the legitimacy of tasks before the participants are seeking to sense the task collaboratively. We have generated all the data in real geographical maps of five different cities with Crowdsensim simulator. Through simulations, we have shown that Gradient Boosting can outperform Random Forest while distinguishing legitimate tasks from illegitimate (i.e., maliciously injected) ones. The promising performance is observed, especially under the appropriate density of

tasks and attack regions with respect to the size of the cities where MCS campaigns are launched. We have also investigated the viability of two learning strategies: per-day and day-after-day learning strategies. We have shown that day-after-day learning can provide higher performance in the task legitimacy decision, whereas per-day learning will require less storage due to training the system based on the most recent observations.

We creatively proposed two take movement models, namely Zone-free Movement (ZFM) model and Zone-limited Movement (ZLM) model. And based on the attack model and task movement patterns, we have designed task features and created structured simulation settings that can be modified to adapt different researching scenarios and research purposes. We have investigated the impact of ML-based prevention of illegitimate task submissions in MCS systems. We have shown their impacts on the battery levels of participating devices, as well as the task completion rate. Our simulation results have been carried out under two different task arrival and mobility scenarios. They have shown that when illegitimate tasks are filtered out based on a ML classifier at the MCS servers, up to 14% of the battery power of participating devices can be saved at the expense of a slight reduction in the task completion rate. Moreover, the ratio of impacted recruits can decrease from 16.5% to 3.9% by ML-based illegitimate task prevention.

From the standpoint of adversaries, we have introduced an attacker design model for illegitimate task injection (i.e., clogging attacks) by leveraging a self-organizing feature map (SOFM) to increase the affected user population of Mobile Crowdsensing (MCS) campaigns. Illegitimate (fake) task attack zone coordinates are determined by the SOFM through the inputs of mobile device users' mobility patterns. Thus, more participants are affected by the illegitimate task submissions when compared to the randomly selected attack zones. Accordingly, the energy levels of participating devices are significantly increased as well. We have shown that when illegitimate task positions are selected through the ML-based SOFM clustering method, almost half of the participants are affected with an energy overhead up to 28%. In addition, uniformly and randomly initialized neurons are designed with fixed and adaptive quantities that are determined based upon the affected area on the covered terrain. Through numerical studies, we have shown that the impact of the SOFM structures can affect up to 46% of the participants and up to 37% of the recruits under various SOFM topologies. Furthermore, SOFM-based attack models can increase the energy consumption in the recruited devices by up to 39% due to the illegitimate task submission.

We also proposed an ensemble ML-based mitigation against SOFM modeled fake task attacks. Gradient Boosting is chosen as an efficient algorithm with competitive performance on differentiating two categories of tasks: legitimate tasks and illegitimate tasks. Specifically, Gradient Boosting-based ensemble learning can achieve up to 98.9% overall

accuracy, 99.4% legitimate F_1 score, and 94.6% illegitimate F_1 score. With ensemble ML embedded in the MCS system, the majority of the fake tasks can be eliminated, which leads to around 20% battery capacity savings for the participants. We have also validated the mitigation of the impact on the candidates' population and recruit. We have shown that up to 35.6% of candidates have been saved from the attacks. The simulation results integrated with ensemble ML demonstrate the efficiency, effectiveness, and emergency of an integrated task legitimacy assessment module that utilizes a ML-based model to identify fake task injection even if the attacks were launched by adversarial artificial intelligence.

Our future work involves the following six objects:

- We are working on AI assisted adversary attacks with higher impacts. Currently, the SOFM is used to cluster the locations of users to find the spots where users aggregate with the highest density. However, the routines of users also include the time they pass by the coordinates, which introduces another feature on the time dimension. The possible approach to find a more powerful resource clogging attack is considering commuting, so to get different attack regions at different time intervals during a day.

- Our ongoing work involves running simulations on other recruitment policies. The recruitment policy we have applied so far is a spatial, push tasking schema according to the classifications in [84], which considers the time, distance, and battery level as essential factors. However, we still need to analyze the adversarial effects on different types of recruitment policies — our aim of adversary modeling to design effective attacks no matter how recruitment works.

- We still need to analyze other ensemble techniques on differentiating legitimate and illegitimate tasks. Besides random forest and gradient boosting, there are some other candidates, for instance, bagging methods, voting classifiers, and AdaBoost. Although they may have similar motivations as the methods we have adopted in this thesis, we are not clear what the exact performance of these ensemble techniques, which is awaiting for experiments.

- It is necessary to address the data limitations of our data sets as future work. Two top limitations are 1) random task initiation location generation without considering the area of the crowd, and 2) peak hours when the communication is frequent are defined roughly according to Ontario hydro. Thus, our future work to address data limitations involves: 1) To design a more realistic task initiation location model based on user density, and 2) To refine the definition of peak hours according to the users' active time.

- Our ongoing work also involves running deep learning networks to improve accuracy in the presence of adversarial AI-based fake task injection. Furthermore, integrating mobile edge-based computing infrastructure with the MCS system to accelerate ML-based decisions is also in our short term agenda.
- Another interesting topic would be investigating the impact of different user movement patterns on task recruitment, which might cause various attack effects.

References

[1] Mohamed Abomhara and Geir Køien. Cyber security and the internet of things: Vulnerabilities, threats, intruders and attacks. *Journal of Cyber Security*, 4(1):65–88, 05 2015.

[2] Fazel Anjomshoa, Moayad Aloqaily, Burak Kantarci, Melike Erol-Kantarci, and Stephanie Schuckers. Social behaviometrics for personalized devices in the internet of things era. *IEEE Access*, 5:12199–12213, 2017.

[3] Mohamad Arafeh, May El Barachi, Azzam Mourad, and Fatna Belqasmi. A blockchain based architecture for the detection of fake sensing in mobile crowdsensing. In *2019 4th International Conference on Smart and Sustainable Technologies (SpliTech)*, pages 1–6. IEEE, 2019.

[4] Marek Bell, Stuart Reeves, Barry Brown, Scott Sherwood, Donny MacMillan, John Ferguson, and Matthew Chalmers. Eyespy: supporting navigation through play. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 123–132. ACM, 2009.

[5] Leo Breiman. Prediction games and arcing algorithms. *Neural Computation*, 11:1493–1517, 1999.

[6] Iain Brown and Christophe Mues. An experimental comparison of classification algorithms for imbalanced credit scoring data sets. *Expert Systems with Applications*, 39(3):3446–3453, 2012.

[7] Jeffrey A Burke, Deborah Estrin, Mark Hansen, Andrew Parker, Nithya Ramanathan, Sasank Reddy, and Mani B Srivastava. Participatory sensing. 2006.

[8] Andrea Capponi, Claudio Fiandrino, Burak Kantarci, Luca Foschini, Dzmitry Kliazovich, and Pascal Bouvry. A survey on mobile crowdsensing systems: Challenges, solutions and opportunities. *IEEE Communications Surveys & Tutorials*, 2019.

[9] Claudio Fiandrino Burak Kantarci Luca Foschini Dzmitry Kliazovich Capponi, Andrea and Pascal Bouvry. A survey on mobile crowdsensing systems: Challenges, solutions and opportunities. *IEEE Communications Surveys & Tutorials*, 2019.

[10] Shih-Hao Chang and Zhi-Rong Chen. Protecting mobile crowd sensing against sybil attacks using cloud based trust management system. *Mobile Information Systems*, 2016, 2016.

[11] Xiao Chen, Min Liu, Yaqin Zhou, Zhongcheng Li, Shuang Chen, and Xiangnan He. A truthful incentive mechanism for online recruitment in mobile crowd sensing system. *Sensors*, 17(1):79, 2017.

[12] Jim Cherian, Jun Luo, Hongliang Guo, Shen-Shyang Ho, and Richard Wisbrun. Parkgauge: Gauging the occupancy of parking garages with crowdsensed parking characteristics. In *2016 17th IEEE International Conference on Mobile Data Management (MDM)*, volume 1, pages 92–101. IEEE, 2016.

[13] Gabe Cohn, Sidhant Gupta, Tien-Jui Lee, Dan Morris, Joshua R Smith, Matthew S Reynolds, Desney S Tan, and Shwetak N Patel. An ultra-low-power human body motion sensor using static electric field sensing. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, pages 99–102. ACM, 2012.

[14] Cory Cornelius, Apu Kapadia, David Kotz, Dan Peebles, Minho Shin, and Nikos Triandopoulos. Anonysense: privacy-aware people-centric sensing. In *Proceedings of the 6th international conference on Mobile systems, applications, and services*, pages 211–224. ACM, 2008.

[15] Linda Deng and Landon P Cox. Livecompare: grocery bargain hunting through participatory sensing. In *Proceedings of the 10th workshop on Mobile Computing Systems and Applications*, page 4. ACM, 2009.

[16] Salvatore d'Oro, Laura Galluccio, Giacomo Morabito, Sergio Palazzo, Lin Chen, and Fabio Martignon. Defeating jamming with the power of silence: A game-theoretic analysis. *IEEE transactions on wireless communications*, 14(5):2337–2352, 2014.

[17] Zubair Fadlullah, Fengxiao Tang, Bomin Mao, Nei Kato, Osamu Akashi, Takeru Inoue, and Kimihiro Mizutani. State-of-the-art deep learning: Evolving machine intelligence toward tomorrow's intelligent network traffic control systems. *IEEE Comm. Surv. & Tutorials*, 19/4:2432–2455, 2017.

[18] Karoly Farkas, Gabor Feher, Andras Benczur, and Csaba Sidlo. Crowdsensing based public transport information service in smart cities. *IEEE Communications Magazine*, 53(8):158–165, 2015.

[19] Claudio Fiandrino, Andrea Capponi, Giuseppe Cacciatore, Dzmitry Kliazovich, Ulrich Sorger, Pascal Bouvry, Burak Kantarci, Fabrizio Granelli, and Stefano Giordano. Crowdsensim: a simulation platform for mobile crowdsensing in realistic urban environments. *IEEE Access*, 5:3490–3503, 2017.

[20] Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29:1189–1232, 2000.

[21] Sébastien Gambs, Marc-Olivier Killijian, and Miguel Núñez del Prado Cortez. De-anonymization attack on geolocated data. *Journal of Computer and System Sciences*, 80(8):1597–1614, 2014.

[22] Raghu K Ganti, Fan Ye, and Hui Lei. Mobile crowdsensing: current state and future challenges. *IEEE Communications Magazine*, 49(11), 2011.

[23] Amrita Ghosal, Subir Halder, Sreerupa Chatterjee, Jaydeep Sen, and Sipra DasBit. Estimating delay in a data forwarding scheme for defending jamming attack in wireless sensor network. In *Next Generation Mobile Applications, Services and Technologies, 2009. NGMAST'09. Third International Conference on*, pages 351–356. IEEE, 2009.

[24] Stylianos Gisdakis, Thanassis Giannetsos, and Panagiotis Papadimitratos. Security, privacy, and incentive provision for mobile crowd sensing systems. *IEEE Internet of Things Journal*, 3(5):839–853, 2016.

[25] Aakar Gupta, William Thies, Edward Cutrell, and Ravin Balakrishnan. mclerk: enabling mobile crowdsourcing in developing regions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1843–1852. ACM, 2012.

[26] Hadi Habibzadeh, Andrew Boggio-Dandry, Zhou Qin, Tolga Soyata, Burak Kantarci, and Hussein T Mouftah. Soft sensing in smart cities: Handling 3vs using recommender systems, machine intelligence, and data analytics. *IEEE Communications Magazine*, 56(2):78–86, 2018.

[27] Guoan Han, Liang Xiao, and H Vincent Poor. Two-dimensional anti-jamming communication based on deep reinforcement learning. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2087–2091. IEEE, 2017.

[28] Kyungsik Han, Eric A Graham, Dylan Vassallo, and Deborah Estrin. Enhancing motivation in a mobile participatory sensing project through gaming. In *2011 IEEE Third International Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third International Conference on Social Computing*, pages 1443–1448. IEEE, 2011.

[29] Daojing He, Sammy Chan, and Mohsen Guizani. User privacy and data trustworthiness in mobile crowd sensing. *IEEE Wireless Communications*, 22(1):28–34, 2015.

[30] Baik Hoh, Tingxin Yan, Deepak Ganesan, Kenneth Tracton, Toch Iwuchukwu, and Juong-Sik Lee. Trucentive: A game-theoretic incentive platform for trustworthy mobile crowdsourcing parking services. In *2012 15th International IEEE Conference on Intelligent Transportation Systems*, pages 160–166. IEEE, 2012.

[31] Shaohan Hu, Lu Su, Hengchang Liu, Hongyan Wang, and Tarek F Abdelzaher. Smartroad: Smartphone-based crowd sensing for traffic regulator detection and identification. *ACM Transactions on Sensor Networks (TOSN)*, 11(4):55, 2015.

[32] Luis G Jaimes, Idalides Vergara-Laurens, and Miguel A Labrador. A location-based incentive mechanism for participatory sensing systems with budget constraints. In *2012 IEEE International Conference on Pervasive Computing and Communications*, pages 103–108. IEEE, 2012.

[33] Shiyu Ji and Tingting Chen. On designing collusion-resistant incentive mechanisms for mobile crowdsensing systems. In *2017 IEEE Trustcom/BigDataSE/ICESS*, pages 162–169. IEEE, 2017.

[34] Luliang Jia, Yuhua Xu, Youming Sun, Shuo Feng, and Alagan Anpalagan. Stackelberg game approaches for anti-jamming defence in wireless networks. *IEEE Wireless Communications*, 25(6):120–128, 2018.

[35] Lingyun Jiang, Xiaofu Niu, Jia Xu, Yuchan Wang, Yongqi Wu, and Lijie Xu. Time-sensitive and sybil-proof incentive mechanisms for mobile crowdsensing via social network. *IEEE Access*, 6:48156–48168, 2018.

[36] Xing Jin, Mingchu Li, Xiaomei Sun, Cheng Guo, and Jia Liu. Reputation-based multi-auditing algorithmic mechanism for reliable mobile crowdsensing. *Pervasive and Mobile Computing*, 2018.

[37] Klaas Jordan, Iaroslav Sheptykin, Barbara Grüter, and Heide-Rose Vatterrott. Identification of structural landmarks in a park using movement data collected in a location-based game. In *COMP@ SIGSPATIAL*, pages 1–8, 2013.

[38] Burak Kantarci, Kevin G Carr, and Connor D Pearsall. Sonata: Social network assisted trustworthiness assurance in smart city crowdsensing. *International Journal of Distributed Systems and Technologies (IJDST)*, 7(1):59–78, 2016.

[39] Burak Kantarci and Hussein T Mouftah. Trustworthy sensing for public safety in cloud-centric internet of things. *IEEE Internet of Things Journal*, 1(4):360–368, 2014.

[40] Apu Kapadia, Nikos Triandopoulos, Cory Cornelius, Daniel Peebles, and David Kotz. Anonysense: Opportunistic and privacy-preserving context collection. In *International Conference on Pervasive Computing*, pages 280–297. Springer, 2008.

[41] Taghi M Khoshgoftaar, Moiz Golawala, and Jason Van Hulse. An empirical study of learning from imbalanced data using random forest. In *19th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2007)*, volume 2, pages 310–317. IEEE, 2007.

[42] Mikkel Baun Kjærgaard, Sourav Bhattacharya, Henrik Blunck, and Petteri Nurmi. Energy-efficient trajectory tracking for mobile devices. In *Proceedings of the 9th international conference on Mobile systems, applications, and services*, pages 307–320. ACM, 2011.

[43] Haneul Ko, Sangheon Pack, and Victor CM Leung. Coverage-guaranteed and energy-efficient participant selection strategy in mobile crowdsensing. *IEEE Internet of Things Journal*, 6(2):3202–3211, 2018.

[44] Teuvo Kohonen. Self-organized formation of topologically correct feature maps. *Biological cybernetics*, 43(1):59–69, 1982.

[45] Teuvo Kohonen. The self-organizing map. *Proceedings of the IEEE*, 78(9):1464–1480, 1990.

[46] Teuvo Kohonen. Exploration of very large databases by self-organizing maps. In *Proceedings of International Conference on Neural Networks (ICNN'97)*, volume 1, pages PL1–PL6, 1997.

[47] Divyan Munirathnam Konidala, Robert H Deng, Yingjiu Li, Hoong Chuin Lau, and Stephen E Fienberg. Anonymous authentication of visitors for mobile crowd sensing at amusement parks. In *International Conference on Information Security Practice and Experience*, pages 174–188. Springer, 2013.

[48] Emmanouil Koukoumidis, Li-Shiuan Peh, and Margaret Rose Martonosi. Signalguru: leveraging mobile phones for collaborative traffic signal schedule advisory. In *Proceedings of the 9th international conference on Mobile systems, applications, and services*, pages 127–140. ACM, 2011.

[49] Ioannis Krontiris and Andreas Albers. Monetary incentives in participatory sensing using multi-attributive auctions. *International Journal of Parallel, Emergent and Distributed Systems*, 27(4):317–336, 2012.

[50] Ioannis Krontiris and Nicolas Maisonneuve. Participatory sensing: the tension between social translucence and privacy. In *Trustworthy Internet*, pages 159–170. Springer, 2011.

[51] Nicholas D. Lane, Yohan Chon, Lin Zhou, Yongzhe Zhang, Fan Li, Dongwon Kim, Guanzhong Ding, Feng Zhao, and Hojung Cha. Piggyback crowdsensing (pcs): Energy efficient crowdsourcing of mobile sensor data by exploiting smartphone app opportunities. pages 7:1–7:14, 2013.

[52] Nicholas D Lane, Emiliano Miluzzo, Hong Lu, Daniel Peebles, Tanzeem Choudhury, and Andrew T Campbell. A survey of mobile phone sensing. *IEEE Communications magazine*, 48(9):140–150, 2010.

[53] Yee Wei Law, Pieter H Hartel, Jerry den Hartog, and Paul JM Havinga. Link-layer jamming attacks on s-mac. In *EWSN*, pages 217–225, 2005.

[54] Yee Wei Law, Marimuthu Palaniswami, Lodewijk Van Hoesel, Jeroen Doumen, Pieter Hartel, and Paul Havinga. Energy-efficient link-layer jamming attacks against wireless sensor network mac protocols. *ACM Transactions on Sensor Networks (TOSN)*, 5(1):6, 2009.

[55] Juong-Sik Lee and Baik Hoh. Dynamic pricing incentive for participatory sensing. *Pervasive and Mobile Computing*, 6(6):693–708, 2010.

[56] Chiara Leonardi, Andrea Cappellotto, Michele Caraviello, Bruno Lepri, and Fabrizio Antonelli. Secondnose: an air quality mobile crowdsensing system. In *Proceedings of the 8th Nordic Conference on Human-Computer Interaction: Fun, Fast, Foundational*, pages 1051–1054. ACM, 2014.

[57] Anat Levin, Dani Lischinski, and Yair Weiss. A closed-form solution to natural image matting. *IEEE transactions on pattern analysis and machine intelligence*, 30(2):228–242, 2007.

[58] Qinghua Li, Guohong Cao, and Thomas F La Porta. Efficient and privacy-aware data aggregation in mobile sensing. *IEEE Transactions on Dependable and Secure Computing*, 11(2):115–129, 2014.

[59] Xiang Li, Na Ruan, Fan Wu, Jie Li, and Mengyuan Li. Efficient and enhanced broadcast authentication protocols based on multilevel μtesla. In *Performance Computing and Communications Conference (IPCCC), 2014 IEEE International*, pages 1–8. IEEE, 2014.

[60] Yanda Li, Jinliang Liu, Qiangda Li, and Liang Xiao. Mobile cloud offloading for malware detections with learning. In *Computer Communications Workshops (INFOCOM WKSHPS), 2015 IEEE Conference on*, pages 197–201. IEEE, 2015.

[61] Zili Li, Zuduo Zheng, and Simon Washington. Short-term traffic flow forecasting: a component-wise gradient boosting approach with hierarchical reconciliation. *IEEE Transactions on Intelligent Transportation Systems*, 2019.

[62] Andy Liaw, Matthew Wiener, et al. Classification and regression by randomforest. *R news*, 2(3):18–22, 2002.

[63] Chi Harold Liu, Bo Zhang, Xin Su, Jian Ma, Wendong Wang, and Kin K Leung. Energy-aware participant selection for smartphone-enabled mobile crowd sensing. *IEEE Systems Journal*, 11(3):1435–1446, 2015.

[64] Donggang Liu and Peng Ning. Multilevel μtesla: Broadcast authentication for distributed sensor networks. *ACM Transactions on Embedded Computing Systems (TECS)*, 3(4):800–836, 2004.

[65] Xin Liu, Yuhua Xu, Luliang Jia, Qihui Wu, and Alagan Anpalagan. Anti-jamming communications using spectrum waterfall: A deep reinforcement learning approach. *IEEE Communications Letters*, 22(5):998–1001, 2018.

[66] Tie Luo and Chen-Khong Tham. Fairness and social welfare in incentivizing participatory sensing. In *2012 9th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, pages 425–433. IEEE, 2012.

[67] Lingjuan Lyu, Xuanli He, Yee Wei Law, and Marimuthu Palaniswami. Privacy-preserving collaborative deep learning with application to human activity recognition. In *ACM on Conference on Information and Knowledge Management*, CIKM '17, pages 1219–1228, New York, NY, USA, 2017. ACM.

[68] Huadong Ma, Dong Zhao, and Peiyan Yuan. Opportunities in mobile crowd sensing. *IEEE Communications Magazine*, 52(8):29–35, 2014.

[69] Xiao Ma, Zhenzhe Zheng, Fan Wu, and Guihai Chen. Trust-based time series data model for mobile crowdsensing. In *2017 IEEE International Conference on Communications (ICC)*, pages 1–6. IEEE, 2017.

[70] Ashwin Machanavajjhala, Johannes Gehrke, Daniel Kifer, and Muthuramakrishnan Venkitasubramaniam. l-diversity: Privacy beyond k-anonymity. In *22nd International Conference on Data Engineering (ICDE'06)*, pages 24–24. IEEE, 2006.

[71] Llew Mason, Jonathan Baxter, Peter L Bartlett, and Marcus R Frean. Boosting algorithms as gradient descent. In *Advances in neural information processing systems*, pages 512–518, 2000.

[72] Thomas J Matarazzo, Paolo Santi, Shamim N Pakzad, Kristopher Carter, Carlo Ratti, Babak Moaveni, Chris Osgood, and Nigel Jacob. Crowdsensing framework for monitoring bridge vibrations using moving smartphones. *Proceedings of the IEEE*, 106(4):577–593, 2018.

[73] Patrick McDaniel, Nicolas Papernot, and Z Berkay Celik. Machine learning in adversarial settings. *IEEE Security & Privacy*, 14(3):68–72, 2016.

[74] Prashanth Mohan, Venkata N Padmanabhan, and Ramachandran Ramjee. Nericell: rich monitoring of road and traffic conditions using mobile smartphones. In *Proceedings of the 6th ACM conference on Embedded network sensor systems*, pages 323–336. ACM, 2008.

[75] M Victoria Moreno, Antonio F Skarmeta, and Antonio J Jara. How to intelligently make sense of real data of smart cities. In *2015 International Conference on Recent Advances in Internet of Things (RIoT)*, pages 1–6. IEEE, 2015.

[76] Reza Mortazavi, Saeed Jalili, and Hojjat Gohargazi. Multivariate microaggregation by iterative optimization. *Applied intelligence*, 39(3):529–544, 2013.

[77] Mohamed Musthag, Andrew Raij, Deepak Ganesan, Santosh Kumar, and Saul Shiffman. Exploring micro-incentive strategies for participant compensation in high-burden studies. In *Proceedings of the 13th international conference on Ubiquitous computing*, pages 435–444. ACM, 2011.

[78] Sarfraz Nawaz, Christos Efstratiou, and Cecilia Mascolo. Parksense: A smartphone based sensing system for on-street parking. In *Proceedings of the 19th annual international conference on Mobile computing & networking*, pages 75–86. ACM, 2013.

[79] Jeongyeup Paek, Joongheon Kim, and Ramesh Govindan. Energy-efficient rate-adaptive gps-based positioning for smartphones. In *Proceedings of the 8th international conference on Mobile systems, applications, and services*, pages 299–314. ACM, 2010.

[80] Zhengxiang Pan, Han Yu, Chunyan Miao, and Cyril Leung. Crowdsensing air quality with camera-enabled mobile devices. In *Twenty-Ninth IAAI Conference*, 2017.

[81] Victor Pankratius, Frank Lind, Anthea Coster, Philip Erickson, and Joshua Semeter. Mobile crowd sensing in space weather monitoring: the mahali project. *IEEE Communications Magazine*, 52(8):22–28, 2014.

[82] Adrian Perrig, Ran Canetti, J Doug Tygar, and Dawn Song. Efficient authentication and signing of multicast streams over lossy channels. In *Security and Privacy, 2000. S&P 2000. Proceedings. 2000 IEEE Symposium on*, pages 56–73. IEEE, 2000.

[83] Adrian Perrig, Robert Szewczyk, Justin Douglas Tygar, Victor Wen, and David E Culler. Spins: Security protocols for sensor networks. *Wireless networks*, 8(5):521–534, 2002.

[84] Layla Pournajaf, Daniel A Garcia-Ulloa, Li Xiong, and Vaidy Sunderam. Participant privacy in mobile crowd sensing task management: A survey of methods and challenges. *ACM Sigmod Record*, 44(4):23–34, 2016.

[85] Maryam Pouryazdan, Claudio Fiandrino, Burak Kantarci, Tolga Soyata, Dzmitry Kliazovich, and Pascal Bouvry. Intelligent gaming for mobile crowd-sensing participants to acquire trustworthy big data in the internet of things. *IEEE Access*, 2017.

[86] Maryam Pouryazdan, Burak Kantarci, Tolga Soyata, and Houbing Song. Anchor-assisted and vote-based trustworthiness assurance in smart city crowdsensing. *IEEE Access*, 4:529–541, 2016.

[87] Rajiv Punmiya and Sangho Choe. Energy theft detection using gradient boosting theft detector with feature engineering-based preprocessing. *IEEE Transactions on Smart Grid*, 10(2):2326–2329, 2019.

[88] Qu-Tang Cai, Yang-Qui Song, and Chang-Shui Zhang. Cost-sensitive boosting algorithms as gradient descent. In *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2009–2012, March 2008.

[89] Moo-Ryong Ra, Bodhi Priyantha, Aman Kansal, and Jie Liu. Improving energy efficiency of personal sensing applications with heterogeneous multi-processors. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, pages 1–10. ACM, 2012.

[90] Rajib Kumar Rana, Chun Tung Chou, Salil S Kanhere, Nirupama Bulusu, and Wen Hu. Ear-phone: an end-to-end participatory urban noise mapping system. In *Proceedings of the 9th ACM/IEEE international conference on information processing in sensor networks*, pages 105–116. ACM, 2010.

[91] Vandana Milind Rohokale, Neeli Rashmi Prasad, and Ramjee Prasad. Cooperative jamming for physical layer security in wireless sensor networks. In *Wireless Personal Multimedia Communications (WPMC), 2012 15th International Symposium on*, pages 458–462. IEEE, 2012.

[92] Na Ruan, Lei Gao, Haojin Zhu, Weijia Jia, Xiang Li, and Qi Hu. Toward optimal dos-resistant authentication in crowdsensing networks via evolutionary game. In *Distributed Computing Systems (ICDCS), 2016 IEEE 36th International Conference on*, pages 364–373. IEEE, 2016.

[93] David Sánchez, Josep Domingo-Ferrer, Sergio Martínez, and Jordi Soria-Comas. Utility-preserving differentially private data releases via individual ranking microaggregation. *Information Fusion*, 30:1–14, 2016.

[94] Immanuel Schweizer, Roman Bärtl, Axel Schulz, Florian Probst, and Max Mühläuser. Noisemap-real-time participatory noise maps. In *Second international workshop on sensing applications on mobile phones*, pages 1–5. Citeseer, 2011.

[95] Peng Sheng, Li Chen, and Jing Tian. Learning-based road crack detection using gradient boost decision tree. In *2018 13th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, pages 1228–1232. IEEE, 2018.

[96] Xiang Sheng, Jian Tang, and Weiyi Zhang. Energy-efficient collaborative sensing with mobile phones. In *INFOCOM, 2012 Proceedings IEEE*, pages 1916–1924. IEEE, 2012.

[97] Minho Shin, Cory Cornelius, Dan Peebles, Apu Kapadia, David Kotz, and Nikos Triandopoulos. Anonysense: A system for anonymous opportunistic sensing. *Pervasive and Mobile Computing*, 7(1):16–30, 2011.

[98] Zheng Song, Bo Zhang, Chi Harold Liu, Athanasios V Vasilakos, Jian Ma, and Wendong Wang. Qoi-aware energy-efficient participant selection. In *2014 Eleventh Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, pages 248–256. IEEE, 2014.

[99] Hung-Min Sun, Shih-Pu Hsu, and Chien-Ming Chen. Mobile jamming attack and its countermeasure in wireless sensor networks. In *Advanced Information Networking and Applications Workshops, 2007, AINAW'07. 21st International Conference on*, volume 1, pages 457–462. IEEE, 2007.

[100] Arvind Thiagarajan, Lenin Ravindranath, Katrina LaCurts, Samuel Madden, Hari Balakrishnan, Sivan Toledo, and Jakob Eriksson. Vtrack: accurate, energy-aware road traffic delay estimation using mobile phones. In *Proceedings of the 7th ACM conference on embedded networked sensor systems*, pages 85–98. ACM, 2009.

[101] Shanshan Tu, Muhammad Waqas, Sadaqat Ur Rehman, Muhammad Aamir, Obaid Ur Rehman, Zhang Jianbiao, and Chin-Chen Chang. Security in fog computing: A novel technique to tackle an impersonation attack. *IEEE Access*, 6:74993–75001, 2018.

[102] Juha Vesanto, Esa Alhoniemi, et al. Clustering of the self-organizing map. *IEEE Transactions on neural networks*, 11(3):586–600, 2000.

[103] Xiaoyue Wan, Geyi Sheng, Yanda Li, Liang Xiao, and Xiaojiang Du. Reinforcement learning based mobile offloading for cloud-based malware detection. In *GLOBECOM 2017-2017 IEEE Global Communications Conference*, pages 1–6. IEEE, 2017.

[104] Leye Wang, Daqing Zhang, Zhixian Yan, Haoyi Xiong, and Bing Xie. effsense: A novel mobile crowd-sensing framework for energy-efficient and cost-effective data uploading. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 45(12):1549–1563, 2015.

[105] Xinlei Oscar Wang, Wei Cheng, Prasant Mohapatra, and Tarek Abdelzaher. Artsense: Anonymous reputation and trust in participatory sensing. In *2013 Proceedings IEEE INFOCOM*, pages 2517–2525. IEEE, 2013.

[106] Liang Xiao, Donghua Jiang, Dongjin Xu, and Ning An. Secure mobile crowdsensing with deep learning, arXiv, eprint: 1801.07379, 2018.

[107] Liang Xiao, Yanda Li, Guoan Han, Huaiyu Dai, and H Vincent Poor. A secure mobile crowdsensing game with deep reinforcement learning. *IEEE Transactions on Information Forensics and Security*, 13(1):35–47, 2017.

[108] Liang Xiao, Yanda Li, Guoan Han, Huaiyu Dai, and H Vincent Poor. A secure mobile crowdsensing game with deep reinforcement learning. *IEEE Transactions on Information Forensics and Security*, 13(1):35–47, 2017.

[109] Liang Xiao, Jinliang Liu, Qiangda Li, and H Vincent Poor. Secure mobile crowdsensing game. In *Communications (ICC), 2015 IEEE International Conference on*, pages 7157–7162. IEEE, 2015.

[110] Haoyi Xiong, Daqing Zhang, Leye Wang, and Hakima Chaouchi. Emc 3: Energy-efficient data transfer in mobile crowdsensing under full coverage constraint. *IEEE Transactions on Mobile Computing*, 14(7):1355–1368, 2015.

[111] Haoyi Xiong, Daqing Zhang, Leye Wang, J Paul Gibson, and Jie Zhu. Eemc: Enabling energy-efficient mobile crowdsensing with anonymous participants. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 6(3):39, 2015.

[112] Jia Xu, Zhengqiang Rao, Lijie Xu, Dejun Yang, and Tao Li. Mobile crowd sensing via online communities: Incentive mechanisms for multiple cooperative tasks. In *2017 IEEE 14th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, pages 171–179. IEEE, 2017.

[113] Wenyuan Xu, Wade Trappe, Yanyong Zhang, and Timothy Wood. The feasibility of launching and detecting jamming attacks in wireless networks. In *Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*, pages 46–57. ACM, 2005.

[114] Dejun Yang, Guoliang Xue, Xi Fang, and Jian Tang. Crowdsourcing to smartphones: incentive mechanism design for mobile phone sensing. In *Proceedings of the 18th annual international conference on Mobile computing and networking*, pages 173–184. ACM, 2012.

[115] Dejun Yang, Guoliang Xue, Jin Zhang, Andrea Richa, and Xi Fang. Coping with a smart jammer in wireless networks: A stackelberg game approach. *IEEE Transactions on Wireless Communications*, 12(8):4038–4047, 2013.

[116] Kan Yang, Kuan Zhang, Ju Ren, and Xuemin Shen. Security and privacy in mobile crowdsourcing networks: challenges and opportunities. *IEEE communications magazine*, 53(8):75–81, 2015.

[117] Fuqiang Yao, Luliang Jia, Youming Sun, Yuhua Xu, Shuo Feng, and Yonggang Zhu. A hierarchical learning approach to anti-jamming channel selection strategies. *Wireless Networks*, 25(1):201–213, 2019.

[118] Marco Zappatore, Antonella Longo, and Mario A Bochicchio. Using mobile crowd sensing for noise monitoring in smart cities. In *2016 international multidisciplinary conference on computer and energy science (Splitech)*, pages 1–6. IEEE, 2016.

[119] Marco Zappatore, Antonella Longo, Mario A Bochicchio, Daniele Zappatore, Alessandro A Morrone, and Gianluca De Mitri. Mobile crowd sensing-based noise monitoring as a way to improve learning quality on acoustics. In *2015 International Conference on Interactive Mobile Communication Technologies and Learning (IMCL)*, pages 96–100. IEEE, 2015.

[120] Daqing Zhang, Haoyi Xiong, Leye Wang, and Guanling Chen. Crowdrecruiter: selecting participants for piggyback crowdsensing under probabilistic coverage constraint. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 703–714. ACM, 2014.

[121] Xinglin Zhang, Zheng Yang, Wei Sun, Yunhao Liu, Shaohua Tang, Kai Xing, and Xufei Mao. Incentives for mobile crowd sensing: A survey. *IEEE Communications Surveys & Tutorials*, 18(1):54–67, 2015.

[122] Xinglin Zhang, Zheng Yang, Zimu Zhou, Haibin Cai, Lei Chen, and Xiangyang Li. Free market of crowdsourcing: Incentive mechanism design for mobile sensing. *IEEE transactions on parallel and distributed systems*, 25(12):3190–3200, 2014.

[123] Yueqian Zhang and Burak Kantarci. Ai-based security design of mobile crowdsensing systems: Review, challenges and case studies. In *2019 IEEE International Conference on Service-Oriented System Engineering (SOSE)*, pages 17–1709. IEEE, 2019.

[124] Yueqian Zhang, Murat Simsek, and Burak Kantarci. Machine learning-based prevention of battery-oriented illegitimate task injection in mobile crowdsensing. In *Proceedings of the ACM Workshop on Wireless Security and Machine Learning*, pages 31–36. ACM, 2019.

[125] Zhenyu Zhou, Haijun Liao, Bo Gu, Kazi Mohammed Saidul Huq, Shahid Mumtaz, and Jonathan Rodriguez. Robust mobile crowd sensing: When deep learning meets edge computing. *IEEE Network*, 32(4):54–60, 2018.

APPENDICES

Appendix A

Machine learning performance of Naive Bayes and Decision Trees

A.1 Using Naive Bayes

City	Legitimate F-score	Illegitimate F-score
Dryden	0.8635	0.2742
Powell-River	0.7539	0.3294
Clarence-Rockland	0.8583	0.3006
Brant	0.9086	0.2049
Timmins	0.8092	0.3352

Table A.1: Naive Bayes performance in different cities with 4000 tasks, 10 attack regions and 80m moving radius of a task.

A.2 Using Decision Trees

City	**Legitimate F-score**	**Illegitimate F-score**
Dryden	0.9391	0.6211
Powell-River	0.9537	0.6775
Clarence-Rockland	0.942	0.7239
Brant	0.955	0.7538
Timmins	0.9607	0.7358

Table A.2: Decision Trees performance in different cities with 4000 tasks, 10 attack regions and 80m moving radius of a task.