
BEHAVIOR-AWARE EFFICIENT DETECTION OF MALICIOUS EVs IN V2G SYSTEMS

Ruixiang Wu, Xudong Wang

School of Data Science
The Chinese University of Hong Kong, Shenzhen
Shenzhen, Guangdong, China
{ruixiangwu, xudongwang}@link.cuhk.edu.cn

Tongxin Li

School of Data Science
The Chinese University of Hong Kong, Shenzhen
Shenzhen, Guangdong, China
litongxin@cuhk.edu.cn

ABSTRACT

With the rapid development of electric vehicles (EVs) and vehicle-to-grid (V2G) technology, detecting malicious EV drivers is becoming increasingly important for the reliability and efficiency of smart grids. To address this challenge, machine learning (ML) algorithms are employed to predict user behavior and identify patterns of non-cooperation. However, the ML predictions are often untrusted, which can significantly degrade the performance of existing algorithms. In this paper, we propose a safety-enabled group testing scheme, GTUA, which combines the efficiency of probabilistic group testing with ML predictions and the robustness of combinatorial group testing. We prove that GTUA is $O(d)$ -consistent and $O(d \log n)$ -robust, striking a near-optimal trade-off. Experiments on synthetic data and case studies based on ACN-DATA, a real-world EV charging dataset validate the efficacy of GTUA for efficiently detecting malicious users in V2G systems. Our findings contribute to the growing field of algorithms with predictions and provide insights for incorporating distributional ML advice into algorithmic decision-making in energy and transportation-related systems.

Keywords vehicle to grid, malicious user detection, group testing with distributional advice

1 Introduction

Recently, with the rising demand for green technology in modern cities, the electric vehicle (EV) industry is experiencing rapid development in smart cities [1]. Integrating EVs into smart cities presents a significant potential to cut greenhouse gas emissions [2]. One promising approach for this integration is Vehicle-to-Grid (V2G), which allows bidirectional power exchange between EVs and the smart grid [3]. This enables EVs to charge from the smart grid during off-peak hours and supply electricity to the grid during peak periods, further enhancing the flexibility of the power systems, and boosting the implementation of virtual power plant in smart cities. [4, 5]. Furthermore, V2G can provide significant economic benefits for EV owners by lowering ownership costs [6]. Collectively, these benefits position V2G as a highly promising technology for the future.

While the V2G system provides significant flexibility to the smart grid, it also demands considerable flexibility from its users [7], and its reliability and efficiency are closely related to users' behaviors [8]. The issues will arise when V2G users are either malicious or lack flexibility. These users might ignore the committed sojourn times and drive their EVs away during peak hours after getting charged during off-peak hours, preventing the V2G system from accessing the stored electricity.

Although installing meters at each charger allows for individual monitoring, the large-scale data generated introduces substantial financial and logistical burdens, and transmitting such data exposes users to privacy risks [9, 10]. Additionally, the cost for a residential electric meter ranges from \$60 to \$250 [11], this approach may not be practical for detecting malicious users in a large-scale power grid due to the high expense of equipping each user with sensors. Thus, a more efficient technique is crucial for reducing data expenses and enhancing privacy while effectively identifying malicious users.

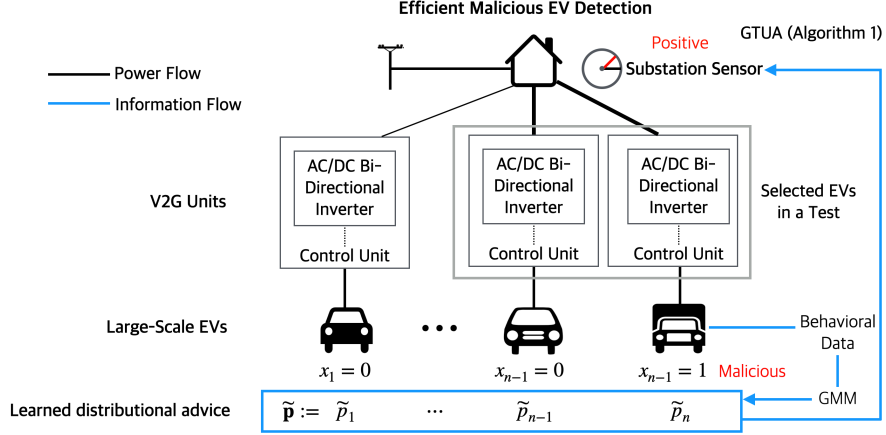


Figure 1: Illustration of Malicious User Detection in a V2G System (see the model defined in Section 2). A V2G manager is equipped with a sensor to detect malicious EVs from a subset of EVs specified by an algorithm. Machine learning models learned from grid data generate distributional advice $\tilde{\mathbf{p}}$ (denoting the probabilities of behaving maliciously). An algorithm, denoted by GTUA is presented in Section 3 for efficient malicious EV detection.

1.1 Previous Work

Previous work has explored economic incentive approaches to deter malicious behavior in V2G systems. For instance, Vickrey-Clarke-Groves (VCG) auction-based algorithms have been proposed to encourage the participation of EVs in V2G networks by using penalty payments to promote honest reporting and discourage malicious activity [12, 13]. Despite their advantages, VCG algorithms have notable limitations. They are susceptible to false-name attacks, as highlighted in [14]. Furthermore, the reliance on exact optimization renders them computationally infeasible in complex scenarios, and also, the VCG mechanism often fails to adequately address strategic behaviors that deviate from simple truth-telling [15].

To address these challenges, an effective approach is to use group testing algorithms to directly detect malicious users, offering a more robust alternative to the relatively “soft” method provided by VCG mechanisms. Group testing algorithms aim to reduce the number of tests by grouping individuals collectively [16, 17]. More recently, group testing algorithms have been applied in various areas of smart grid like fault identification and system reliability assessment. The Adaptive Binary Splitting Inspection (ABSI) algorithm [18] employs binary search techniques to efficiently identify malicious users in large-scale smart grid environments. In 2020, Xia et al. [19] proposed an inspection algorithm based on group testing to detect malicious users. This approach adaptively switches between individual and group testing strategies to achieve higher efficiency. However, the aforementioned group testing algorithms did not consider user history behavior, which is easily obtainable in a smart grid environment. In this paper, our proposed algorithm GTUA fully incorporates user history behavior, thereby improving the efficiency of the testing process.

1.2 Contributions

Our main contributions are: (1) We reformulate the detection task as a novel *group testing* problem that utilizes untrusted machine-learned *distributional advice* to detect malicious users in a V2G system, as illustrated in Figure 1; (2) Introduce GTUA, a safety-enabled testing method (Algorithm 1) combining combinatorial and probabilistic strategies; (3) We provide theoretical bounds on the expected κ -regret (Theorem 1), demonstrating robust utilization of predicted statistics; (4) We establish that our method achieves $O(d)$ -consistency and $O(d \log n)$ -robustness (Corollary 1), where d is the number of malicious drivers and n the total drivers, showing near-optimal trade-offs for group testing algorithms.

2 Problem Formulation

Consider a V2G system consisting of a large fleet of EV chargers. To detect malicious EVs, an efficient method is to remotely connect a carefully selected subset of EVs and monitor the substation power injection from the remaining chargers to see if it meets the expected setpoints (i.e., checking if the connected EVs behave normally), requiring only a single sensor installed at the substation. If the substation power injection from the fixed subset is far from the expected nominal value, it indicates that the presence of malicious EV(s). In summary, for a detection within a subset of EVs, if at least one EV behaves maliciously, the substation power injection will be abnormal (marked as 1); otherwise, it will

be normal (marked as 0). Mathematically, each run of this detecting process can be represented by an “OR” operation, as described in the following detection model.

Malicious EV Detection Model. Consider a set \mathcal{N} of n EV drivers, wherein each driver i is malicious with some probability $p_i \in [0, 1]$ independently. Let $[n] := \{1, \dots, n\}$. The states of the n drivers are represented by an n -length vector $\mathbf{X} = (X_i : i \in [n])$ such that $X_i = 1$ (or $X_i = 0$) implies that the i -th driver is malicious (or not). Our goal is to asymptotically detect malicious drivers by conducting a minimal number of tests. Each test selects a subset of drivers and performs an “OR” operation, which is defined as follows: The test outcome is positive if the subset contains at least one malicious EV (i.e., there exists at least one $X_i = 1$ within the subset); otherwise, it is negative.

2.1 Untrusted ML Predictions

With data available in many real-world scenarios, prior statistics of the probabilities $\mathbf{p} := (p_1, \dots, p_n)$ can often be predicted via black-box ML models. The problem considered in this paper, which sets it apart from the classic probabilistic group testing models [17, 20], involves incorporating a set of predictions $\tilde{\mathbf{p}} = (\tilde{p}_1, \dots, \tilde{p}_n)$ of \mathbf{p} . These predictions are assumed to be provided by an ML agency whose reliability is not fully trusted. The potential for error in $\tilde{\mathbf{p}}$ arises from various factors, such as the inherent variability in learning algorithms [21] or challenges in generalizing beyond the distribution of the training data [22], etc. Our model takes into account these uncertainties in $\tilde{\mathbf{p}}$.

Let $\mathcal{P} := \{\mathbf{p} : 0 \leq p_i \leq 1, i \in [n]\}$ be a set consisting of all \mathbf{p} ’s. Denote by $d := \sum_{i=1}^n p_i$. According to Byzantine agreement [23], for a system to achieve reliable consensus in the presence of malicious nodes, the number of faulty nodes must satisfy the condition above. Accordingly, we focus on the sub-linear regime $d = o(n)$ aligning with the standard sparsity assumption in the classic group testing model. The *distributional advice* given by an untrusted ML model is denoted by $\tilde{\mathbf{p}} := (\tilde{p}_i : i \in [n]) \in \mathcal{P}$ such that $\sum_{i=1}^n \tilde{p}_i = d$.¹ To characterize the prediction error, we use the following accumulated pseudo KL-divergence (logarithm base e) between \mathbf{p} and $\tilde{\mathbf{p}}$:

$$D_{\text{KL}}(\mathbf{p}, \tilde{\mathbf{p}}) := \sum_{i=1}^n \left(p_i \log \frac{p_i}{\tilde{p}_i} \right). \quad (1)$$

In this work, we focus on designing *adaptive* group testing algorithms, with access to untrusted predictions $\tilde{\mathbf{p}}$, i.e., the next test can be determined after observing previous test outcomes. We revisit the classic adaptive probabilistic group testing problem within the framework of online optimization. This perspective allows us to assess our adaptive group testing algorithm against established performance benchmarks commonly used in online optimization contexts. The specifics of the benchmark used in this paper are detailed next.

2.2 Performance Benchmarks

Our algorithm is evaluated based on a novel benchmark known as the κ -*regret*. This benchmark hinges on an external parameter κ that measures the performance or “tightness” of the current best algorithms in probabilistic group testing. This parameter is defined as the *tightness gap* below. Consider any deterministic group testing algorithm, denoted as ALG . Let $\bar{T}(\text{ALG}; \mathbf{p})$ represent the best-known upper bounds on the expected number of tests needed by ALG . We then define $\underline{T}^*(\mathbf{p})$ as the minimum expected number of tests required, considering the same prior information \mathbf{p} .

Definition 1 (Tightness Gap). Given a deterministic group testing algorithm ALG , the tightness gap, denoted by $\kappa(\text{ALG})$, is defined as the smallest ratio between $\bar{T}(\text{ALG}; \mathbf{p})$ and $\underline{T}^*(\mathbf{p})$, i.e.,

$$\inf \left\{ \kappa \in [1, \infty) : \sup_{\mathbf{p} \in \mathcal{P}} \bar{T}(\text{ALG}; \mathbf{p}) \leq \kappa \sup_{\mathbf{p} \in \mathcal{P}} \underline{T}^*(\mathbf{p}) \right\}. \quad (2)$$

This tightness gap highlights the optimality of the considered group testing algorithm, which measures how closely an algorithm’s performance aligns with its theoretical optimal. This gap is crucial not only in presenting our main result regarding the consistency and robustness trade-off but also in applications such as large-scale energy systems, where near-optimal performance grants significant operational savings. A typical example related to our algorithm design is the Laminar Algorithm (LA) in [24], whose *tightness gap* κ_{LA} is less than $2 + 6/\log(n/d)$. In the sequel, we introduce a key performance benchmark for our novel probabilistic group testing model with ML predictions, by considering adaptive group testing as an instance of online optimization.

¹It is known that standard concentration inequalities imply that with high probability, the actual number of malicious users can be bounded by δd for some $\delta > 1$ [17, 24]. Thus, WLOG, we assume the predictions are normalized such that $\sum_{i=1}^n \tilde{p}_i = d$.

Definition 2 (Expected κ -Regret). *The expected κ -regret, denoted by $\text{ER}(\text{ALG}; \varepsilon)$ with $D_{\text{KL}}(p, \tilde{p}) \leq \varepsilon$ for a given deterministic group testing algorithm ALG is defined as*

$$\text{ER}(\text{ALG}; \varepsilon) := \sup_{\mathbf{p}, \mathbf{P} \in \mathcal{P}(\varepsilon)} \mathbb{E} [T(\text{ALG}; \mathbf{p}) - \kappa(\text{LA})T^*(\mathbf{p})] \quad (3)$$

where $\mathcal{P}(\varepsilon) := \{(p, q) \in \mathcal{P} \times \mathcal{P} : D_{\text{KL}}(p, q) \leq \varepsilon\}$; $\kappa(\text{LA})$ denotes the tightness bound of the LA in [24]; and the expectation is taken over randomly sampled instances of \mathbf{X} distributed according to \mathbf{p} .

The notion of regret defined above characterizes the performance of ALG by comparing the *worst-case* difference between the expected number of tests required by ALG , given untrusted ML predictions $\tilde{\mathbf{p}}$ that may contain some error, and a fundamental limit on the minimal expected number of tests required, normalized by the tightness gap.

3 Detecting Malicious Users Efficiently from Behavioral Data

In this section, we provide a detailed implementation of GTUA, which is designed to equip with both the advantages of the probabilistic group testing with (untrusted) ML predictions and the classical group testing for worst-case situations and achieves near-optimal performance. First, PROCEDURE 2 summarizes the Laminar Algorithm (LA) in [24], which is an adaptive group testing algorithm that only works well with accurate prior statistics, i.e., the unknown exact \mathbf{p} in our model. Second, PROCEDURE 3 summarizes the Generalized Binary Splitting (GBS) algorithm [25], as a classic adaptive combinatorial group testing algorithm that does not incorporate any ML predictions. Therefore, even if the learned ML advice from the EV drivers' behavioral data is wrong, PROCEDURE 3 remains unaffected, while PROCEDURE 2 is sensitive to the error induced by the ML model. Both PROCEDURE 2 and PROCEDURE 3 are provided in Appendix A. Later in Section 5 and Appendix F, we compare both LA and GBS with our proposed robust detection algorithm.

3.1 Warm-Up: Why Don't Trust All Predictions?

Prior to delving into the GTUA algorithm, it is crucial to understand the impact of untrusted ML predictions on established probabilistic group testing algorithms. A notable example is the LA algorithm detailed in [24]. Theorem 1 from [24] presents an upper bound on the expected number of tests required by LA, denoted as $\mathbb{E}[T(\text{LA})]$, which is bounded from above by $2\mathbb{H}(\mathbf{X}) + 6d$, with $\mathbb{H}(\mathbf{X})$ representing the binary entropy of \mathbf{X} . Building upon this, we introduce a lemma that expands the theorem to more practical scenarios involving ML predictions $\tilde{\mathbf{p}}$, and offers a generalized upper bound on $\mathbb{E}[T(\text{LA})]$ accordingly.

Lemma 1 (Generalization of Theorem 1 in [24]). *When trusting the ML predictions $\tilde{\mathbf{p}}$, the expected number of tests used by the LA satisfies*

$$\mathbb{E}[T(\text{LA})] \leq 2 \sum_{i=1}^n p_i \log \frac{1}{\tilde{p}_i} + 6d. \quad (4)$$

The proof of Lemma 1 follows the same argument of the proof of Theorem 1 in [24], where the term p_i in the summation of (4) follows by taking expectation according to the true distribution of X and the term \tilde{p}_i appears by constructing testing sub-trees based on the ML input $\tilde{\mathbf{p}}$. A key consequence of Lemma 1 is observed when the predicted probabilities \tilde{p}_i are significantly low. Under such circumstances, the LA algorithm tends to combine users corresponding to these negligible probabilities with a large set of users. This can substantially degrade the LA's efficiency, thereby the upper bound in (4) is not tight. Such a scenario highlights the need for a design that ensures robustness even when confronted with untrustworthy ML predictions.

3.2 Our Solution: Incorporating a Safety Threshold

To mitigate the vulnerability issues raised by small \tilde{p}_i 's, we now introduce a safety-enabled testing scheme, GTUA, which combines both probabilistic and classical group testing methods. We consider a *safety threshold*, denoted $\eta \geq \frac{1}{n}$, which divides the users \mathcal{N} into two disjoint pools $\mathcal{N}_p(\eta)$ and $\mathcal{N}_c(\eta)$ such that $\mathcal{N}_p(\eta) \cup \mathcal{N}_c(\eta) = \mathcal{N}$, based on the associated ML predictions $\tilde{\mathbf{p}}$. For the subset of users within $\mathcal{N}_p(\eta)$, the predicted probabilities ($\tilde{p}_i : i \in \mathcal{N}_p(\eta)$) are considered reliable and are therefore utilized accordingly. Conversely, for users in the set $\mathcal{N}_c(\eta)$, we disregard the less reliable predictions and instead employ combinatorial group testing techniques to identify any potentially malicious behavior. The two disjoint user pools are defined formally in the following:

$$\mathcal{N}_p(\eta) := \{i \in \mathcal{N} : \tilde{p}_i \geq \eta\}, \quad (5)$$

$$\mathcal{N}_c(\eta) := \mathcal{N} \setminus \mathcal{N}_p(\eta). \quad (6)$$

Algorithm 1 Group Testing with Untrusted ML Advice (GTUA)

```

1: Input: Population set  $\mathcal{N}$ , distributional advice  $\tilde{\mathbf{p}}$ 
2: Initialize a safety threshold  $\eta \geq \frac{1}{n}$ 
3: Partition  $\mathcal{N}$  into  $\mathcal{N}_p(\eta)$  and  $\mathcal{N}_c(\eta)$  based on  $\eta$  and  $\tilde{\mathbf{p}}$  according to (5) and (6)
4: for  $i$  in  $\mathcal{N}_p(\eta)$  do
5:   Implement PROCEDURE 2 using the predicted  $\tilde{p}$  on  $\mathcal{N}_p(\eta)$ 
6: end for
7: for  $i$  in  $\mathcal{N}_c(\eta)$  do
8:   Implement PROCEDURE 3 on  $\mathcal{N}_c(\eta)$ 
9: end for
    
```

Enabled by a tunable safety threshold, the framework summarized in Algorithm 1, demonstrates a generic approach that integrates any recognized probabilistic group testing algorithm with predictions for users in $\mathcal{N}_p(\eta)$, as well as combinatorial group testing algorithms for users in $\mathcal{N}_c(\eta)$. In the sequel, we provide theoretical guarantees of GTUA, assuming LA and GBS are implemented as inner procedures in Algorithm 1. It is worth noting that the algorithm we present functions as a meta-testing scheme. It is designed to integrate and accommodate a variety of testing procedures – a probabilistic one that ensures *consistency* when prediction errors are low, and a combinatorial one that guarantees *robustness* against increasing prediction inaccuracies. The procedures can be arbitrary, as long as each is selected in its capacity as satisfying one of these crucial properties, allowing the overall scheme to maintain performance across varying levels of prediction reliability. In PROCEDURE 2 and PROCEDURE 3 we exemplify the LA and the Generalized Binary Splitting (GBS) as two particular instances, provided in Appendix A.

4 Theoretical Results

In this section, we provide theoretical analysis on GTUA and show that it achieves near-optimal performance, in terms of the consistency and robustness trade-off (see Definition 3). Fixing the inner procedures as LA and GBS, respectively, we bound the expected regret of GTUA, with a tightness gap $\kappa(\text{LA})$.

4.1 Expected Regret Analysis

Theorem 1 (Expected $\kappa(\text{LA})$ -Regret). *Fix $D_{\text{KL}}(p, \tilde{p}) \leq \varepsilon$, when $n \geq d \geq e$. The expected $\kappa(\text{LA})$ -regret of GTUA satisfies*

$$\begin{aligned} \text{ER}(\text{GTUA}; \varepsilon) &\leq 2 \min \{d \log(1/\eta), \varepsilon_p\} \\ &\quad + 2 \min \{d \log n, \varepsilon_c + d \log \eta n\}, \end{aligned}$$

where ε_p and ε_c are upper bounds on the pseudo KL-divergence for the two pools, $\mathcal{N}_p(\eta)$ and $\mathcal{N}_c(\eta)$ respectively, such that $\sum_{i \in \mathcal{N}_p(\eta)} (p_i \log p_i / \tilde{p}_i) \leq \varepsilon_p$ and $\sum_{i \in \mathcal{N}_c(\eta)} (p_i \log p_i / \tilde{p}_i) \leq \varepsilon_c$.

To characterize the overall performance of GTUA, following the standard definitions in the online algorithms community [20, 26], we adapt the following definitions to evaluate the consistency and robustness of the algorithm with respect to two error regimes of the expected regret.

Definition 3 (Consistency and robustness). *A testing algorithm ALG is γ -consistent if its expected $\kappa(\text{LA})$ -regret satisfies $\text{ER}(\text{ALG}; \varepsilon) \leq \gamma$ when $\varepsilon = 0$, and β -robust if $\text{ER}(\text{ALG}; \varepsilon) \leq \beta$ for any $\varepsilon \geq 0$.*

Here, the values of γ and β depend on system parameters such as d and n . Theorem 1 above implies the following result.

Corollary 1. *With a fixed threshold parameter $\eta \geq \frac{1}{n}$, the GTUA algorithm is $(d \log(\eta n))$ -consistent and $(4d \log \frac{n}{\eta})$ -robust.*

This result highlights that when $\eta = \Theta(1/n)$, the proposed GTUA is $O(d)$ -consistent, implying that it uses at most $O(T(\text{LA}))$ number of tests, which is optimal when the ML predictions $\tilde{\mathbf{p}}$ are perfect. On the other hand, for any ε , GTUA is $O(d \log n)$ -robust. Note that any group testing algorithm must have a lower bound $\Omega(d \log n)$ on its expected number of tests, as implied by the information-theoretic lower bound [16, 17]. Furthermore, the following result verifies that the consistency and robustness trade-off achieved by GTUA in Corollary 1 is near-optimal.

Theorem 2. *For any deterministic group testing algorithm ALG, if it is $(\frac{1}{2}d \log(n\eta))$ -consistent, then it is at least $O\left(d \log \frac{n}{\eta}\right)$ -robust.*

Note that GTUA does not use any randomness in the algorithm implementation; it is therefore a deterministic algorithm.

5 Best-of-Both-Worlds Performance of GTUA

In this part, we will use synthetic data to test the performance of the aforementioned three group testing algorithms against various values of error (in terms of the KL divergence between p and \tilde{p} , denoted by $D_{\text{KL}}(p, \tilde{p})$) to verify the efficacy of GTUA and the robustness of its performance, regardless of the prediction error of the ML model. The detailed setting is deferred to Appendix 5. A simulated V2G system used to validate the applicability and effectiveness of GTUA in real-life scenarios is deferred to Appendix F.

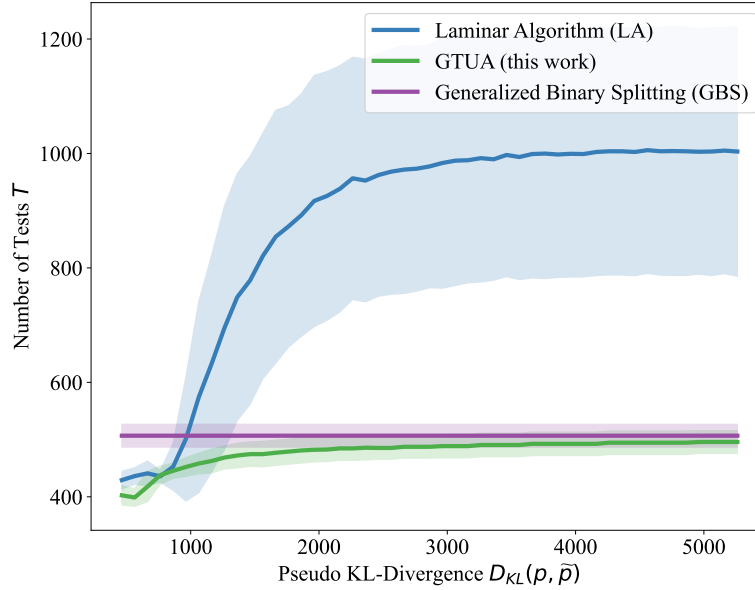


Figure 2: **Comparison of the average number of tests** for LA [24], GBS [25], and the proposed GTUA (Algorithm 1) in this work. Shadow areas depict the magnitudes of standard deviations.

Experimental Results. The result in Figure 2 highlights that when the divergence between \mathbf{p} and $\tilde{\mathbf{p}}$ is small (implying high predictive accuracy), the LA [24] exhibits high efficiency as desired. However, when prediction error increases, its efficacy degrades significantly, as evidenced by the escalating number of required tests and high variance in the figure (see the blue curve). In contrast, GBS [25]’s performance (see the purple curve) remains unaffected regardless of the prediction error of the distributional advice $\tilde{\mathbf{p}}$. As suggested by our theoretical guarantees in Theorem 1 and Corollary 1, the proposed algorithm shows its reliability in practical scenarios, where ML predictions cannot be fully trusted.

6 Conclusion

In this paper, we addressed the critical issue of detecting malicious EV drivers in vehicle-to-grid systems using group testing with untrusted ML predictions. We proposed GTUA, a safety-enhanced group testing scheme that combines the efficiency of probabilistic group testing with ML predictions and the robustness of combinatorial group testing. Theoretically, we proved that GTUA achieves a near-optimal consistency and robustness trade-off. We validated the effectiveness of GTUA through experiments on synthetic data and simulations on real-world EV charging data, demonstrating its superior performance compared to existing group testing algorithms.

Built upon the results presented in this work, there are various aspects that are worth future investigation. For example, it would be interesting to study if more advanced ML models can further improve the detection efficiency in practice, even though our algorithm has a theoretically guaranteed performance regardless of the ML prediction accuracy. The proposed GTUA algorithm offers a promising solution to adaptively detect malicious EVs in a V2G system with learned distributional advice from the EVs' behavioral data. Extending GTUA to non-adaptive settings, where all tests are designed offline beforehand, is a potential direction for future research.

References

- [1] Uthman Opeyemi Abdullahi and Adnan Adnan. Hybrid and electric vehicle (ev) integration in public transport: Challenges and opportunities. 2024.
- [2] Meriem Azari, Jabir Arif, Fouad Jawab, and Hassane Moustabchir. Smart city implementation of electric vehicles: Challenges and opportunities. In *2024 IEEE 15th International Colloquium on Logistics and Supply Chain Management (LOGISTIQUA)*, pages 1–6, 2024.
- [3] Weifeng Zhong, Rong Yu, Shengli Xie, Yan Zhang, and David K. Y. Yau. On stability and robustness of demand response in v2g mobile energy networks. *IEEE Transactions on Smart Grid*, 9(4):3203–3212, 2018.
- [4] Antonio Pepiciello and Alfredo Vaccaro. The role of vehicle to grid technology for enhancing power distribution system flexibility. In *2022 IEEE International Workshop on Metrology for Automotive (MetroAutomotive)*, pages 24–29, 2022.
- [5] Tasneim Aldhanhani, Anuj Abraham, Wassim Hamidouche, and Mostafa Shaaban. Future trends in smart green iov: Vehicle-to-everything in the era of electric vehicles. *IEEE Open Journal of Vehicular Technology*, 5:278–297, 2024.
- [6] Apostolos Filippas, John J. Horton, and Richard J. Zeckhauser. Owning, Using, and Renting: Some Simple Economics of the “Sharing Economy”. *Management Science*, 66(9):4152–4172, September 2020.
- [7] Yudi Qin, Yiran Rao, Zhoucheng Xu, Xianping Lin, Kegang Cui, Jiuyu Du, and Minggao Ouyang. Toward flexibility of user side in china: Virtual power plant (vpp) and vehicle-to-grid (v2g) interaction. *eTransportation*, 18:100291, 2023.
- [8] Gwang-Su Shin, Ho-Young Kim, Jean Mahseredjian, and Chul-Hwan Kim. Smart vehicle-to-grid operation of power system based on ev user behavior. *Journal of Electrical Engineering & Technology*, pages 1–12, 2024.
- [9] Zhuoqun Xia, Li Zhang, and Arun Kumar Sangaiah. An efficient data aggregation solution for smart meters based on cloud-edge collaboration. In Min Luo and Liang-Jie Zhang, editors, *Services Computing – SCC 2023*, pages 95–107, Cham, 2024. Springer Nature Switzerland.
- [10] Yuwen Chen, Shisong Yang, José-Fernán Martínez-Ortega, Lourdes López, and Zhen Yang. A resilient group-based multisubset data aggregation scheme for smart grid. *IEEE Internet of Things Journal*, 10(15):13649–13661, 2023.
- [11] EVVR. What is a smart meter for electricity & electric meter cost?, 2024. Accessed: 2024-08-04.
- [12] Ifiok Anthony Umoren, Muhammad Zeeshan Shakir, and Hamed Ahmadi. Vcg-based auction for incentivized energy trading in electric vehicle enabled microgrids. *IEEE Access*, 11:21117–21126, 2023.
- [13] Ming Zeng, Supeng Leng, Sabita Maharjan, Yan Zhang, and Stein Gjessing. Group bidding for guaranteed quality of energy in v2g smart grid networks. In *2015 IEEE International Conference on Communications (ICC)*, pages 5266–5271, 2015.
- [14] Yotam Gafni, Ron Lavi, and Moshe Tennenholtz. Vcg under false-name attacks: a bayesian analysis, 2021.
- [15] Shahar Dobzinski and Noam Nisan. Limitations of vcg-based mechanisms. *Combinatorica*, 31(4):379–396, 2011.
- [16] Dingzhu Du, Frank K Hwang, and Frank Hwang. *Combinatorial group testing and its applications*, volume 12. World Scientific, 2000.
- [17] Chun Lam Chan, Pak Hou Che, Sidharth Jaggi, and Venkatesh Saligrama. Non-adaptive probabilistic group testing with noisy measurements: Near-optimal bounds with efficient algorithms, 2011.
- [18] Xiaofang Xia, Yang Xiao, and Wei Liang. Absi: An adaptive binary splitting algorithm for malicious meter inspection in smart grid. *IEEE Transactions on Information Forensics and Security*, 14(2):445–458, 2019.
- [19] Xiaofang Xia, Yang Xiao, W. Liang, and M. Zheng. Gthi: A heuristic algorithm to detect malicious users in smart grids. *IEEE Transactions on Network Science and Engineering*, 7:805–816, 2020.

- [20] Matthew Aldridge, Oliver Johnson, Jonathan Scarlett, et al. Group testing: an information theory perspective. *Foundations and Trends® in Communications and Information Theory*, 15(3-4):196–392, 2019.
- [21] Benjamin Recht. A tour of reinforcement learning: The view from continuous control. *Annual Review of Control, Robotics, and Autonomous Systems*, 2:253–279, 2019.
- [22] Florian Wenzel, Andrea Dittadi, Peter Gehler, Carl-Johann Simon-Gabriel, Max Horn, Dominik Zietlow, David Kernert, Chris Russell, Thomas Brox, Bernt Schiele, et al. Assaying out-of-distribution generalization in transfer learning. *Advances in Neural Information Processing Systems*, 35:7181–7198, 2022.
- [23] Leslie Lamport, Robert Shostak, and Marshall Pease. The byzantine generals problem. *ACM Transactions on Programming Languages and Systems*, pages 382–401, July 1982.
- [24] Tongxin Li, Chun Lam Chan, Wenhao Huang, Tarik Kaced, and Sidharth Jaggi. Group testing with prior statistics. In *2014 IEEE International Symposium on Information Theory*, pages 2346–2350, 2014.
- [25] F. K. Hwang. A method for detecting all defective members in a population by group testing. *Journal of the American Statistical Association*, 67(339):605–608, 1972.
- [26] Alexander Wei and Fred Zhang. Optimal robustness-consistency trade-offs for learning-augmented online algorithms. *arXiv preprint arXiv:2010.11443*, 2020.
- [27] Zachary J. Lee, Tongxin Li, and Steven H. Low. ACN-Data: Analysis and Applications of an Open EV Charging Dataset. In *Proceedings of the Tenth International Conference on Future Energy Systems*, e-Energy '19, June 2019.
- [28] Bruce G Lindsay. Mixture models: theory, geometry, and applications. Ims, 1995.

A Probabilistic and Combinatorial Group Testing Procedures

Procedure 2 PROCEDURE PGT [24]

- 1: **Procedure** PGT($\mathcal{N}_p(\eta), \tilde{\mathbf{p}}$) \triangleright This procedure implements the Maximum Entropy-based LA [24] for group testing
- 2: **Laminar Algorithm (LA):**
First Stage
- 3: Partition \mathcal{N} into subsets $\{S_{1,1}^r\}_{r=1}^m$ greedily: $\triangleright S_{k,l}^r(k, l$ are parameters indexing the child nodes and r indexes the partitions)

$$\min_{S_{1,1}^r} \left| \prod_{X_i \in S_{1,1}^r} (1 - p_i) - \frac{1}{2} \right|,$$

$$\text{subject to } S_{1,1}^r \subseteq \mathcal{N} \setminus \bigcup_{j=1}^{r-1} S_{1,1}^j.$$

The k -th Stage ($k \in \{2, 3, 4, \dots\}$)

- 4: **for** subgroups that tested positive in the previous stage **do**
- 5: Partition the set $S_{k-1,l}^r$ according to:

$$\min_{S_{k,l}^r} \left| \frac{1 - \prod_{X_i \in S_{k,l}^r} (1 - p_i)}{1 - \prod_{X_i \in S_{k-1,l}^r} (1 - p_i)} - \frac{1}{2} \right|,$$

$$\text{subject to } S_{k,l}^r \subseteq S_{k-1,l}^r.$$

- 6: **end for**
 - 7: This strategy is designed to maximize the entropy, which is optimal for probabilistic group testing problem.
 - 8: **end procedure**
-

Procedure 3 PROCEDURE CGT [25]

- 1: **Procedure** CGT($\mathcal{N}_c(\eta)$) \triangleright This procedure implements the GBS [25] for group testing
 - 2: **Generalised Binary-Splitting (GBS) Key Steps:**
 - 3: Set $\alpha = \lceil \log_2 \frac{n-d+1}{d} \rceil$.
 - 4: Perform group testing with groups of size 2^α .
 - 5: **if** a group is tested positive **then**
 - 6: Use binary search to find a defective.
 - 7: **end if**
 - 8: Adjust n and d after identifying defectives and repeat.
 - 9: **end procedure**
-

B Useful Lemmas

In this appendix, we provide lemmas that will be used in the proofs of our main results.

Lemma 2 (Theorem 1 in [24]). *For any group testing algorithm operating on the user set \mathcal{N} with the probability vector \mathcal{P} , and allowing a maximum error probability P_e , the lower bound on the number of tests is given by:*

$$T \geq (1 - \mathbb{P}_e) H(\mathbf{X}), \quad (7)$$

where $H(\mathbf{X}) := \left(\sum_{i \in \mathcal{N}} p_i \log \frac{1}{p_i} + \sum_{i \in \mathcal{N}} (1 - p_i) \log \frac{1}{1 - p_i} \right)$ denotes the summation of entropy of random variable $X_i \in \mathbf{X}$.

The next lemma provides the upper bound of the expected number of tests required by LA [24] (see Procedure 2).

Lemma 3 (Theorem 2 in [24]). *A user i with probability p_i to be malicious needs at most $\log \frac{1}{p_i}$ number of tests, and the expected number of tests $\mathbb{E}(T(\mathbf{LA}))$ for the Laminar Algorithm with a user set \mathcal{N} and ground truth probability vector \mathbf{p} can be bounded as:*

$$\begin{aligned}\mathbb{E}[T(\mathbf{LA})] &\leq 2 \sum_{i \in \mathcal{N}} p_i \left\lceil \log \left(\frac{1}{p_i} \right) \right\rceil + 4 \sum_{i \in \mathcal{N}} p_i \\ &\leq 2 \sum_{i \in \mathcal{N}} p_i \log \left(\frac{1}{p_i} \right) + 6 \sum_{i \in \mathcal{N}} p_i.\end{aligned}$$

In the next lemma, we characterize the tightness bound for the Laminar Algorithm (\mathbf{LA}), as a critical benchmark in probabilistic group testing.

Lemma 4. *The tightness gap $\kappa(\mathbf{LA})$ for (adaptive) Laminar Algorithm (\mathbf{LA}) in [24] is at most $2 + 6/\log(n/d)$.*

Proof. By definition, $\kappa(\mathbf{LA})$ can be computed as:

$$\begin{aligned}\kappa &= \lim_{n \rightarrow \infty} \frac{\max_{\mathcal{P}} \left(2 \sum_{i \in \mathcal{N}} p_i \log \left(\frac{1}{p_i} \right) + 6 \sum_{i \in \mathcal{N}} p_i \right)}{\max_{\mathcal{P}} (1 - \mathbb{P}_e) H(\mathbf{X})} \\ &\leq \lim_{n \rightarrow \infty} \frac{\max_{\mathcal{P}} \left(2 \sum_{i \in \mathcal{N}} p_i \log \left(\frac{1}{p_i} \right) + 6 \sum_{i \in \mathcal{N}} p_i \right)}{\max_{\mathcal{P}} (1 - \mathbb{P}_e) \left(\sum_{i \in \mathcal{N}} p_i \log \left(\frac{1}{p_i} \right) \right)} \\ &= \frac{2d \log \left(\frac{n}{d} \right) + 6d}{d \log \left(\frac{n}{d} \right)} \\ &= 2 + \frac{6}{\log(n/d)},\end{aligned}\tag{8}$$

where (8) follows that in probabilistic group testing, $\mathbb{P}_e \rightarrow 0$ when $n \rightarrow \infty$. Furthermore, since $d = o(n)$, when $n \rightarrow \infty$, $\kappa \rightarrow 2$. \square

Next, we consider the number of tests required by the generalized binary splitting (GBS) algorithm in [25].

Lemma 5 (Theorem 1 in [25]). *Consider a population of n users among which d users are malicious, and suppose $d = o(n)$. In this context, the required number of tests of GBS, denoted by $T(\mathbf{GBS})$, can be bounded from above as follows:*

$$T(\mathbf{GBS}) \leq d \left(\left\lfloor \log \frac{n-d+1}{d} \right\rfloor + 2 \right) + p - 1,\tag{9}$$

where p is a non-negative integer satisfying $0 \leq p < d$.

Therefore, the number of tests $T(\mathbf{GBS})$ satisfies $T(\mathbf{GBS}) \leq 2d \log n$ for $n \geq d \geq e$.

C Proof of Theorem 1

Proof of Theorem 1. Applying lemma 3, the expected number of tests for probabilistic group testing algorithm $\mathbb{E}[T_{i \in \mathcal{N}_p(\eta)}]$ can be bounded as:

$$\begin{aligned}\mathbb{E}[T_{i \in \mathcal{N}_p(\eta)}] &\leq 2 \sum_{i \in \mathcal{N}} p_i \left\lceil \log \left(\frac{1}{\tilde{p}_i} \right) \right\rceil + 4 \sum_{i \in \mathcal{N}} p_i \\ &\leq 2 \sum_{i \in \mathcal{N}} p_i \log \left(\frac{1}{\tilde{p}_i} \right) + 2 \sum_{i \in \mathcal{N}} \tilde{p}_i + 4 \sum_{i \in \mathcal{N}} p_i.\end{aligned}\tag{10}$$

Therefore, combining (10), (9), (7), and using Lemma 1, the expected $\kappa(\mathbf{LA})$ -regret (see Definition 1) can be bounded as:

$$\begin{aligned}\text{ER}(\text{GTUA}) &\leq \underbrace{\left(\sum_{i \in \mathcal{N}_p(\eta)} p_i \log \frac{1}{\tilde{p}_i} + \sum_{i \in \mathcal{N}_p(\eta)} p_i + \sum_{i \in \mathcal{N}_p(\eta)} 2\tilde{p}_i \right)}_{\text{Lemma 1}} + \underbrace{\left(\sum_{i \in \mathcal{N}_c(\eta)} 3p_i \log \left(\frac{|\mathcal{N}_c(\eta)|}{\sum_{i \in \mathcal{N}_c(\eta)} p_i} \right) \right)}_{\text{Lemma 5}} - \kappa H(\mathbf{X}).\end{aligned}\tag{11}$$

Rearranging the terms in (11) above and applying Lemma 5, the expected regret satisfies

$$\begin{aligned}
 \text{ER}(\text{GTUA}) &\leq \sum_{i \in \mathcal{N}_p(\eta)} 2p_i \log \frac{1}{\tilde{p}_i} + \sum_{i \in \mathcal{N}_c(\eta)} 2p_i \log(|\mathcal{N}_c(\eta)|) - \sum_{i \in \mathcal{N}} 2p_i \log \frac{1}{p_i} \\
 &\leq \sum_{i \in \mathcal{N}_p(\eta)} 2p_i \log \frac{p_i}{\tilde{p}_i} + \sum_{i \in \mathcal{N}_c(\eta)} 2p_i \log(p_i |\mathcal{N}_c(\eta)|) \\
 &= \sum_{i \in \mathcal{N}_p(\eta)} 2p_i \log \frac{p_i}{\tilde{p}_i} + \sum_{i \in \mathcal{N}_c(\eta)} 2p_i \log \frac{p_i}{\tilde{p}_i} + \sum_{i \in \mathcal{N}_c(\eta)} 2p_i \log(\tilde{p}_i |\mathcal{N}_c(\eta)|)
 \end{aligned} \tag{12}$$

Since the threshold parameter η partitions the population set \mathcal{N} based on the rules in (5)-(6), (12) yields

$$\begin{aligned}
 \text{ER}(\text{GTUA}) &\leq 2 \min \left\{ \sum_{i \in \mathcal{N}_p(\eta)} p_i \log \frac{1}{\eta}, \varepsilon_p \right\} + 2 \min \left\{ \sum_{i \in \mathcal{N}_c(\eta)} p_i \log n, \varepsilon_c + \sum_{i \in \mathcal{N}_c(\eta)} \log \eta |\mathcal{N}_c(\eta)| \right\} \\
 &\leq 2 \min \left\{ d \log \frac{1}{\eta}, \varepsilon_p \right\} + \min \{ d \log n, \varepsilon_c + d \log \eta n \}
 \end{aligned}$$

where the terms ε_p and ε_c are defined in Theorem 1. \square

D Proof of Theorem 2

Proof of Theorem 2. Suppose that a probabilistic group testing algorithm ALG is $\frac{1}{2} \log n \lambda$ -consistent. Assuming that the true probability vector \mathbf{p} and the predicted probability vector $\tilde{\mathbf{p}}$ are the same, then we need at least d tests to find out all the defective cases. Therefore, the algorithm ALG needs at least $d + \frac{1}{2} \log n \lambda$ number of tests to find out all defective individuals. Next, we assume that $\tilde{\mathbf{p}}$ and \mathbf{p} have totally different distributions. Thus, the robustness constant β can be bounded as follows.

$$\begin{aligned}
 \beta &\geq \log \left(\frac{\binom{n-d}{d}}{2^{d+\frac{1}{2} \log n \lambda}} \right) = \log \left(\frac{\binom{n-d}{d}}{(2\sqrt{n\lambda})^d} \right) \\
 &= O \left(\log \left(\frac{\sqrt{2p_i(n-d)} \left(\frac{n-d}{e}\right)^{n-d}}{\sqrt{2p_i(n-2d)} \left(\frac{n-2d}{e}\right)^{n-2d} \sqrt{2p_i d} \left(\frac{d}{e}\right)^d (2\sqrt{n\lambda})^d} \right) \right),
 \end{aligned} \tag{13}$$

where we have applied the Stirling's approximation to derive (13). Continuing from above,

$$\beta \geq O \left(\log \left(\frac{(n-d)^{n-d+\frac{1}{2}}}{\sqrt{2p_i} d^{d+\frac{1}{2}} (n-2d)^{n-2d+\frac{1}{2}} (2\sqrt{n\lambda})^d} \right) \right), \tag{14}$$

with the term inside $O(\cdot)$ in (14) bounded from above by

$$\frac{(n-d)^{n-d+\frac{1}{2}}}{\sqrt{2p_i} d^{d+\frac{1}{2}} (n-2d)^{n-2d+\frac{1}{2}} (2\sqrt{n\lambda})^d} = O \left(\log \frac{n}{\lambda} \right),$$

implying the theorem. \square

E Numerical Experiment Settings.

Recall all the parameters in GTUA, n denotes the total EV users under monitoring, \mathbf{p} is the true probability vector for each EV user to be defective, and $\tilde{\mathbf{p}}$ is the machine-learned prediction for \mathbf{p} . Setting $n = 1000$, we fix \mathbf{p} and vary the distributional advice $\tilde{\mathbf{p}}$, which results in a sequence of pseudo KL-divergence $D_{\text{KL}}(p, \tilde{p})$ values (x -axis in Figure 2) defined in (1).

F Case Studies of Real-World ACN-DATA

In this section, we will use authentic EV driver behavior data to validate the effectiveness of the proposed algorithm. We will first explain how we learn EV driver behavior from ACN-DATA [27], which contains “userID”, “connectionTime”, “doneChargingTime”, “disconnectTime”, “requestedDeparture”. Then we will elaborate our simulation process. Finally, simulation results will be displayed to show that GTUA can efficiently detect malicious EV drivers in a V2G system.

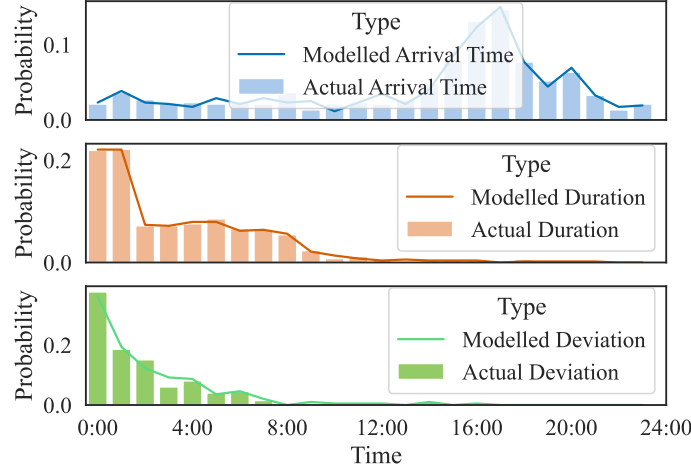


Figure 3: Comparison of model distributions with actual collected data.

Learn EV driver behavior from real world EV charging dataset. Using the similar idea in [27], we first utilize Gaussian Mixture Models (GMMs) [28] to approximate an distribution of EV drivers' behavior. The dataset, which includes the data from Jan. 1, 2019 to Sep. 1, 2021, can be modelled into a GMM model based on the following assumption: for each EV driver, their behavior profiles is finite [27]. To be more specific, for each charging session i in the dataset, we denote the corresponding user profile using a triple $x_i = (a_i, d_i, e_i)$ in \mathbb{R}^3 , where a_i stands for the arrival time, d_i denotes the duration ("disconnectTime" - "connectionTime"), and e_i is the deviation ("disconnectTime" - "requestedDeparture"). Therefore, each data point x_i can be viewed as an user profile with a certain probability. The GMM model is trained on the whole data set, and we generate 100,000 samples from the trained GMM model to evaluate how well this model fits the underlying distribution. Figure 3 shows that the trained GMM model matched the real world data well. The trained GMM model is also used to predict whether an EV driver is likely to be malicious based on its prediction of the deviation. If the deviation is greater than 2 hours, it indicates a higher likelihood of the user not adhering to the schedule, which represents the trustworthiness of the user. This metric can be used to infer whether the user will be malicious or non-cooperative in the V2G system.

Simulation setting We first sample 100,000 copies of EV drivers' profile from the trained GMM model, with each profile including arrival time, duration, and deviation. The simulated V2G system starts with no EVs connected, and EVs are added to the system gradually based on their arrival time at each time step t . For those users with deviation greater than 2 hours, we classify them as malicious users. These users may leave before the scheduled time, causing the total energy injection to be lower than the expectation. To ensure timely detection of malicious EV drivers, we conduct testing every hour. During the testing process, we implement GTUA by disconnecting a subset of EV each time, which allows us to perform the grouping operations necessary for identifying non-cooperative drivers.

Results The result in Figure 4 illustrates that the number of tests conducted each hour is significantly lower than number of EVs being tested. Compared with individually testing all the EVs in the V2G system, GTUA reduces the number of tests by about 40 to 80 percent. This observation further validate the efficiency of GTUA in real world scenario. By minimizing the number of tests required to accurately identify malicious EV drivers, GTUA reduced computational overhead. The reliable performance of GTUA will enhance the overall efficiency and reliability of the V2G system. In total we conducted five experiments using different random seeds, and the average results are shown in the table 1.

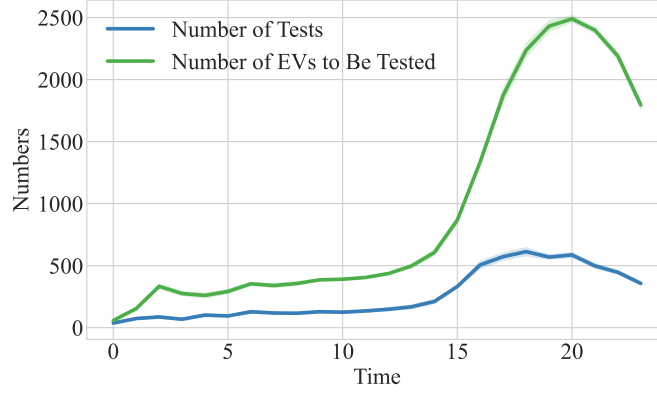


Figure 4: Comparison of number of tests each hour using GTUA and the number of EVs to be tested each hour.

	1	2	3	4	5	6	7	8	9	10	11	12
Ratio	0.64	0.48	0.26	0.25	0.39	0.32	0.36	0.35	0.33	0.33	0.32	0.33
Number of Tests	37.2	73.8	86.6	68.0	101.6	94.8	128.2	118.6	116.4	128.8	125.0	135.0
Number of Users	58.2	153.6	333.6	275.4	260.0	292.2	353.8	340.0	356.2	385.6	390.8	404.8
	13	14	15	16	17	18	19	20	21	22	23	24
Ratio	0.34	0.34	0.35	0.38	0.38	0.31	0.27	0.23	0.24	0.21	0.20	0.20
Number of Tests	148.4	168.0	211.2	332.2	506.6	572.8	613.0	569.4	586.4	498.2	446.6	356.8
Number of Users	436.6	497.6	606.0	867.6	1337.8	1873.4	2236.6	2430.8	2489.6	2398.4	2190.0	1795.0

Table 1: Number of tests each hour using GTUA and the number of EVs to be tested each hour.