EVScout2.0: Electric Vehicle Profiling through Charging Profile

ALESSANDRO BRIGHENTE, MAURO CONTI, DENIS DONADEL, and FEDERICO TURRIN, Department of Mathematics, University of Padova, Italy

EVs (Electric Vehicles) represent a green alternative to traditional fuel-powered vehicles. To enforce their widespread use, both the technical development and the security of users shall be guaranteed. Users' privacy represents a possible threat that impairs the adoption of EVs. In particular, recent works showed the feasibility of identifying EVs based on the current exchanged during the charging phase. In fact, while the resource negotiation phase runs over secure communication protocols, the signal exchanged during the actual charging contains features peculiar to each EV. In what is commonly known as profiling, a suitable feature extractor can associate such features to each EV.

In this article, we propose *EVScout2.0*, an extended and improved version of our previously proposed framework to profile EVs based on their charging behavior. By exploiting the current and pilot signals exchanged during the charging phase, our scheme can extract features peculiar for each EV, hence allowing their profiling. We implemented and tested *EVScout2.0* over a set of real-world measurements considering over 7,500 charging sessions from a total of 137 EVs. In particular, numerical results show the superiority of *EVScout2.0* with respect to the previous version. *EVScout2.0* can profile EVs, attaining a maximum of 0.88 for both recall and precision scores in the case of a balanced dataset. To the best of the authors' knowledge, these results set a new benchmark for upcoming privacy research for large datasets of EVs.

CCS Concepts: • **Security and privacy → Side-channel analysis and countermeasures;**

Additional Key Words and Phrases: Electric Vehicle, profiling, charging, Cyber-Physical Systems

ACM Reference format:
Alessandro Brighente, Mauro Conti, Denis Donadel, and Federico Turrin. 2024. *EVScout2.0*: Electric Vehicle Profiling through Charging Profile. *ACM Trans. Cyber-Phys. Syst.* 8, 2, Article 11 (May 2024), 24 pages.
https://doi.org/10.1145/3565268

1 INTRODUCTION

EVs represent one of the technologies enabling a solution for mitigating petroleum consumption and the consequent reduced emissions amount. Multiple countries have already started providing financial incentives to facilitate the purchasing and widespread of EVs ownership [46]. The EV global forecast expects a compound annual growth rate of 29% over the next 10 years, with total EV sales reaching up to 31.1 million by 2030 [17]. Among the factors decreasing the adoption

We would like to thank Cariparo Foundation and Yarix S.r.l. for supporting with a grant Federico Turrin, and Omitech S.r.l. for supporting Denis Donadel.

Authors' address: A. Brighente, M. Conti, D. Donadel, and F. Turrin, Department of Mathematics, University of Padova, Padova, Italy; emails: alessandro.brighente@unipd.it, conti@math.unipd.it, denis.donadel@studenti.unipd.it, turrin@math.unipd.it.

2378-962X/2024/05-ART11 $15.00
https://doi.org/10.1145/3565268

of EVs is consumers' concerns regarding the availability of charging infrastructures [17]. In fact, while gas stations are widely deployed in cities and rural areas, **EVSEs (Electric Vehicle Supply Equipments)** are not deployed in a sufficient number or, in some areas, are not present. Although charging stations may be deployed at house premises, the absence of publicly available EVSEs represents a limit, as a user is forced to limit the traveling distance to stay close to a charging point. To mitigate this issue and increase the interest towards EVs, current recovery plans envisioned by countries such as Germany and China designate part of their funds to the development of EV charging infrastructures [17]. Furthermore, companies are equipping their parking spots with EVSEs to serve employees EVs. For instance, the United States is incentivizing the adoption of EVSEs at companies' parking spots via the workplace charging challenge [43]. Therefore, in the next years, we expect a significant increase in the number of publicly available EVSEs, allowing users to charge their EV at any time, removing availability concerns.

EVSEs enable the charging process by bringing together multiple technologies. On the one hand, they allow the user to exchange data with the grid, providing means for authorizations towards a central entity to negotiate the service and pay the associated fees. On the other hand, they allow for an exchange of information from the EV to the infrastructure, such that the charging process is conducted by providing safety towards EV's components. The former communication process (i.e., user to infrastructure) is secured using cryptographic procedures and secure network protocols. Their use mitigates all the well-known threats to the users' security and privacy from unprotected communications. However, the latter communication process (i.e., EV to EVSE) is not secure, as signals are exchanged without encryption or aggregation techniques. The signals exchanged during the charging process can hence be exploited as a side-channel to extract information peculiar to each EV, allowing hence for their profiling and successive recognition [10]. This represents a threat to users' privacy, as the connection of their EV to an EVSE monitored by a malicious user may lead to tracking their movement as well as information regarding their driving behavior. Since the majority of publicly available EVSEs are deployed without proper physical protection, they can be accessed by anyone and hence represent favorable spots for attackers targeting the charging infrastructure [2]. Therefore, an attacker can easily install devices to collect data regarding the charging process.

In this article, we propose *EVScout2.0*, an extension of EVScout [10], where we initially showed the feasibility of profiling EVs based on the current exchanged during the charging process. In particular, we extend the proposed framework by developing an enhanced feature extractor, which allows for higher classification scores than our previous work. We then exploit a novel TS (Time Series) for feature extraction by combining the current and pilots TSs. We will explain the reasoning behind this choice and provide the details on its computation. Furthermore, we consider a larger real-world dataset, comprising up to 300 TSs of the current and pilot signals exchanged by each of the 137 considered EVs, for a total of more than 7,500 charging sessions. We perform a thorough evaluation of *EVScout2.0*, showing its profiling performance considering different training set sizes, as well as different unbalancing in the training-testing datasets. Compared to the previous work [10], we provide a more comprehensive analysis of the performance of the attack. In particular, we first show the performance of the novel feature extractor and investigate the performance of *EVScout2.0* for a varying number of features. We then compare the performance of the different classifiers that can be exploited by *EVScout2.0*. We also extend the number of classifiers and provide an in-depth description of the choice of their hyper-parameters. We then investigate the dependencies of *EVScout2.0* on the number of training TSs needed for classifying EVs with sufficient confidence. We show that the attack is already successful considering seven training examples. We then investigate the battery degradation over time and its impact on *EVScout 2.0* performances.

Last, we show the superiority of *EVScout 2.0*, comparing its performance with those in Reference [10], both on the old and new datasets.

The contribution of this article and its improvements with respect to Reference [10] can hence be summarized as follows:

- We propose an enhanced feature extraction framework, which allows for better classification performance compared to Reference [10].
- We propose the use of Delta TS, a novel TS given by the linear combination of the current and pilot TSs. Delta TS allows for the extraction of more significant features compared to those in Reference [10].
- We analyze the performance of *EVScout2.0* on a large real-world dataset, comprising more than 7,500 employable current and pilot time series from 137 EVs. Compared to the dataset in Reference [10], we consider both a larger number of TSs, as well as a larger number of EVs.
- We perform a thorough evaluation of *EVScout2.0*, showing its performance over different training set sizes, as well as its robustness towards different unbalancing amounts in the training-testing dataset sizes.
- Compared to Reference [10], we analyze the performance of a higher number of classifiers and provide an in-depth discussion on the choice of their hyper-parameters. We show that *EVScout2.0* is able to profile EVs with precision and recall up to 0.88.
- We investigate the battery degradation over time and show that *EVScout2.0* can extract features that allow for high classification scores (on average 0.8 $F1$ score) in time.
- We compare *EVScout2.0* with the framework in Reference [10], showing its superiority both in the old and in the new datasets.

The rest of the article is organized as follows: In Section 2, we review the related works. Then, in Section 3, we present the considered system and threat model in an EV charging infrastructure. In Section 4, we describe the features and the steps of *EVScout2.0*. In Section 5, we describe the evaluation framework and the new dataset. Then, in Section 6, we show the results achieved by *EVScout2.0*, while in Section 7, we propose some insight on additional analysis related to training set size and battery degradation. In Section 8, we provide a comparison between *EVScout2.0* and its previous version EVScout [10], clearly presenting the advantages of the new approach. We discuss some countermeasures to avoid profiling in Section 9, and, last, we summarize the results and draw the conclusions in Section 10.

2 RELATED WORKS

Power consumption can be exploited as a side-channel for different purposes [28]. For instance, an attacker may recognize a laptop user by measuring the current that the laptop draws from the wall socket during users' activity given [15]. The same concept can be exploited for detecting the user's presence in a smart home, where raw data can be acquired and analyzed to detect activity and hence users' presence [30]. Raw power data also provides information regarding the actions a user is performing. For instance, by analyzing raw power data exchanged via a USB cable, an attacker may be able to obtain information regarding the victim's browsing activities [48]. The power exchanged during the charging process via USB can also leak more sensitive information, which an attacker can later exploit. For instance, the power analysis may leak information regarding digits composed on a touchscreen, allowing for the deduction of users' passwords [16].

Regarding EVSEs, security and privacy research and contributions focus on the negotiation phase, as it allows the negotiation of the charging service by sharing personal users' data. This

includes both communications from the EVSE to the power distributor and from the EV to the EVSE. In the former communication scenario, the standards J1772 [13] and CHAdeMO [1] are exploited to regulate the physical standards needed for EV-to-EVSE connections, together with the signaling required for the charging process. These standards do not employ encryption or privacy measures on the exchanged information. In the latter case, the standard ISO 15118 [32] is exploited to create a secure communication link, implying that both EV and EVSE must be able to encrypt messages. In Reference [4], the authors proved that the charging cable was not shielded and data was not encrypted, leading to dangerous privacy threats. This weakness was exploited in Reference [26] to inject a signal into the cable and stop the communication. In Reference [14], the authors present a relay attack on the ISO15118 charging system. The attack enables an attacker to charge a vehicle, making a victim pay for it. The overall V2G system has been analyzed in literature from a **CPS (Cyber-Physical System)** security point of view, and several threats have been identified [2, 21]. However, security and privacy analysis should also focus on the charging phase. In fact, the signals exchanged during the charging process create a time series that can be analyzed to extract features that lead to the profiling of vehicles [10]. However, an in-depth analysis of the robustness of these types of attacks is still missing. In particular, the feasibility of the attacks has been shown for a limited dataset, and the attacker's requirements in terms of the number of samples needed for EV profiling have not been discussed yet.

3 SYSTEM AND THREAT MODEL

In this section, we introduce the scenario in which we conceived our experiments. In particular, in Section 3.1, we recall the EV charging system, which represents our system model, then in Section 3.2, we present the threat model designed for *EVScout2.0*.

3.1 EV Charging System

According to the **Vehicle-To-Grid (V2G)** paradigm [25], the charging infrastructure for EVs is a network where a central controller (power distributor) distributes power based on EVSEs demand while accounting for the maximum supported load by the electric grid. We depicted in Figure 1 the typical architecture of a V2G system. EVSEs in the network may be deployed at different sites, e.g., private customer premises, public stations, or office buildings. Each EV is both physically and logically connected to the grid via the EVSE, which manages communications between the user (i.e., the owner of the EV) and the power distributor. For public charging infrastructures and office stations, multiple EVSEs are connected to the power distributor through a Central Control that copes with the demand of a large number of connected users [2]. EVSEs are typically equipped with communication interfaces (wireless or wired) to allow communication with the user and the grid. Utilizing modules in the EV or smartphone, the user can communicate with the EVSE and, in turn, with the power supplier.

Current implementations of EVSEs are organized in three levels [21, 45]. Level 1 and 2 use a five-lead connector based on SAE J1772 standard [42], where three leads are connected to the grid via relays in the EVSE. The remaining two pins, i.e., pilot and proximity lines, are used for signaling. The proximity line indicates whether a good physical connection has been established between the EV and the EVSE, blocking the initiation of the charging process in case devices are not properly attached and hence preventing damage to both the user and the involved devices. The pilot line provides a basic communication means between the EV and the EVSE. The combination of signals collected from all the pins is used to provide the main processing unit of the EVSE information regarding the charging process, allowing for metering used to assess the charging session state. If a problem arises at one of the two sides of the charging process, then the EVSE computer hardware will remove power from the adapter to prevent injuries on both sides. Level 3

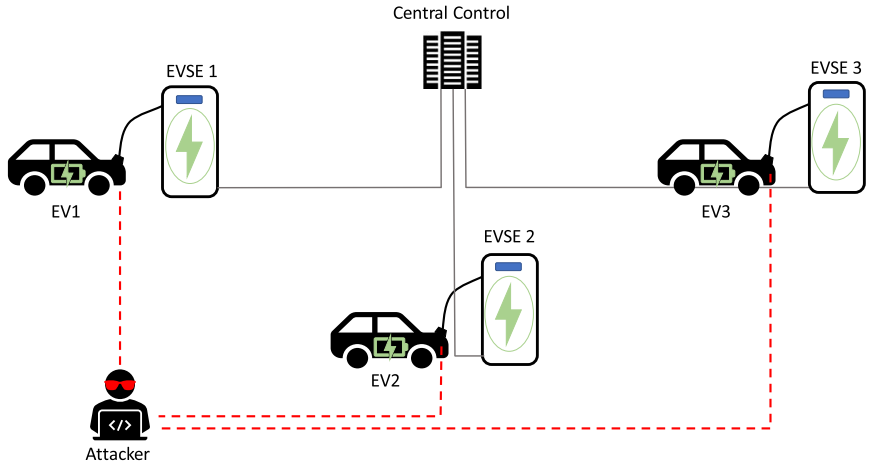


Fig. 1. System and threat model. Multiple EVSEs are connected to the central control, which provides coordination and power distribution among them. A single EV is connected to each EVSE. The attacker has access to the physical quantities exchanged by multiple EVs during the charging phase.

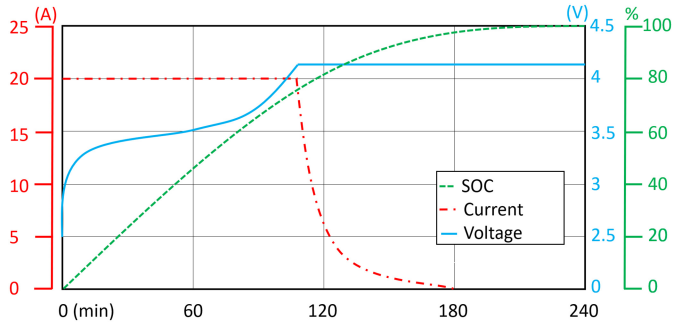


Fig. 2. Charging profile of a Li-ion battery [41]: We see that, as the SoC increases, the charging mode switches from constant current to constant voltage. We further notice that the two phases are mutually exclusive.

EVSEs are instead more complex, comprising bigger pins for power delivery and allowing power line communications via the pilot line.

Typical batteries employed for EVs belong to the class of **Li-ion (Lithium-ion)** [3, 47]. Current and voltage values exchanged during the charging process depend on the **SoC (State of Charge)** of the EV battery and can be divided into two classes: constant current/constant voltage and constant power/constant voltage [29]. In this work, we consider the first class, where the charging process can be further divided into two phases:

- *Constant current phase*, where the current level is constant while the voltage value increases;
- *Constant voltage phase*, where voltage is constant whereas current decreases.

The charging process starts with the constant current phase, and this operation mode is kept until the battery's SoC is above a certain value. After reaching the SoC switching point, the operation mode switches to constant voltage up to the full charge. Typical SoC switching values lie between 60% and 80% of the full charge. An example of a charging profile for an EV's Li-ion battery is shown in Figure 2. We here remark that constant current and constant voltage phases are mutually exclusive in time, as this will be exploited by *EVScout2.0*.

3.2 Threat Model

We consider two possible threat actors: an external attacker and the charging service provider. The first scenario considers a general malicious user who wants to target a specific vehicle. In this case, the attacker can compromise a specific EVSE that the user generally uses (e.g., in workplaces, public parking lots in industrial areas), reducing the number of EVSEs to compromise to succeed in the attack. The attacker may be equipped with a small measuring device that can be connected on one side to the EVSE plug and on the other side to the EV plug. This device measures the exchanged current at the connection point between the EV and EVSE, and we assume that it is hard to notice by users. We assume that the device provides information to the attacker via either (i) a wireless communication module or (ii) storing the values of interest to be later collected by the attacker. The device used to collect the power traces can be built in different ways. For instance, it can be composed of an Arduino board[1] and a standard power consumption monitoring module to sample both current absorbed and pilot.[2] Furthermore, the device can be battery-powered so as not to impact the normal charging behavior. Even if the integration of an external device may impact the absorbed current, we can reasonably assume that this does not affect the profiling performance significantly, as demonstrated in similar works [10, 15, 16, 48].

The second case involves the charging column provider as a threat actor (i.e., the parking spot owner with the charging columns). In this case, the attacker has a higher possibility of manipulating the charging columns; therefore, the attack scales better on more vehicles. Indeed, the V2G infrastructure is exceptionally complex nowadays, and many actors participate in the energy distribution process (e.g., the energy provider, the energy plan contractor, the charging column provider, and the parking lot owner) without access to the same data. For instance, the provider of the charging column does not have access to the user information (data are encrypted), but she can instead easily access the power traces. Therefore, she can easily track and profile users based on their power requirements without being detected.

By employing one of the strategies mentioned above, the attacker has access to the TS of the signals exchanged between the EVSE and the EV during the charging phase. These values are hence recorded for each pin of the EV charger. In this article, we assume that the attacker trains a different classifier for each target EV. To accomplish this, a sufficient number of TSs of the target EV shall be collected. The attack is hence divided into (i) the collection phase, where the attacker collects data regarding a target EV, and (ii) the exploitation phase, where the attacker exploits the previously computed features to discriminate between different EVs based on the observed time series. To collect multiple traces of a single vehicle, the attacker may exploit one or more EVSEs in a public place with regular customers (e.g., workplaces, public parking lots in industrial areas). In this way, the probability of a vehicle going there many times, and thus the attacker having access to more charging traces, is higher. To build a set with sufficient features, we assume that the attacker collects the TS of the exchanged current values and the TS of the pilot signals. Notice that, to retrieve this data, the attacker does not need to perform elaborate V2G network intrusion schemes, as signals are exchanged outside the network. Furthermore, notice that the attacker is not modifying in any way the charging process. Hence, the system cannot automatically detect the attacker's presence via intrusion/anomaly detection techniques.

Our threat model is based on the fact that the majority of publicly available EVSEs are deployed without proper physical security and hence can be accessed by any malicious actor [2]. In this case, since there is no access regulation to the EVSEs, the attacker can freely attach the measuring

[1] https://www.arduino.cc/.

[2] Example of power consumption monitoring module: http://www.sparkfun.com/datasheets/BreakoutBoards/0712.pdf.

devices. The attack is further facilitated by the fact that typical users will not modify the charging system, even though they may notice something unfamiliar. Therefore, the attacker may not only be represented by the company running the EVSEs network but can also be anyone interested in obtaining information on users' consumes and locations. However, we notice that the EVSE devices can be routinely checked by the staff of the running company.

Figure 1 shows the assumed system and threat model. In detail, multiple EVSEs communicate with the Central Control, providing coordination information and power distribution. A single EV is connected to each EVSE. As previously mentioned, the attacker gets access to the time series of the physical quantities exchanged by multiple EVs during the charging phase and exploits them for profiling. Note that if the attacker can remotely access the current exchanged in different network nodes, then it can also locate users, leading to user tracking. The knowledge of the physical signal features associated with each EV (and hence the owning user) can also be exploited for impersonation attacks. Considering EVSEs, which are automated based on the specific user needs, an attacker could steal assets from a target user by generating a signal with the same physical features such that the EVSE recognizes the attacker as the victim. Scenarios that may harm the target user include billing and misbehaving users' exclusion from the system. Therefore, the motivation behind the attack can be multiple. As an illustrative example, consider advertising: The attacker has both information on a certain user's typical movements and the amount of energy s/he consumes regularly. This information can be sold to EVSE owners, which will target their advertisement to the profiled user according to its demand. Notice that, although a single classifier is trained for each EV, the attacker collects information regarding multiple EVs, such that more than a single classifier can be implemented with the gathered data. Therefore, the attacker can also sell information about collective use of the EVSE charging stations by EVs to EVSE companies. Although profiling can be implemented using cameras, this would not allow collecting energy traces, therefore losing some of the information available with the proposed attack. Such information can be obtained utilizing *EVScout2.0*, which may be used as an alternative or a complementary solution to cameras. The possibility of tracking a user gives a further threat. In fact, thanks to *EVScout2.0*, an attacker can detect the presence of a target user in a certain place and time based on the fact that his/her EV is connected to a particular EVSE. We stress that the complexity of the overall charging infrastructure currently imposes several challenges that will be addressed in future implementations. Therefore, it is not possible to predict which actors will in the future be able to access power traces and use them for malicious purposes. Therefore, we aim to warn users and developers about the threat imposed by the cleartext exchange of power traces.

4 *EVSCOUT2.0*

In this section, we describe the *EVScout2.0* analysis methodology. We propose a high-level description of the attack configuration in Section 4.1. Then, we describe the preprocessing we apply to the dataset. Starting from Section 4.2, we present the concept of tails and outline the method we designed to extract them automatically. To improve the performance with respect to the solution in Reference [10], we propose to exploit Delta TS, i.e., the TS given by the combination of the current and pilot TSs. In Section 4.3, we present Delta TS, providing both the motivation behind our choice and the means to compute it. Last, in Section 4.4, we describe the novel and automatic feature extraction technique we employ in *EVScout2.0*.

4.1 Attack Description

As previously stated, in the context of EVs charging infrastructures, users' data are authenticated and secured. However, physical signals are generally not supposed to implement security measures

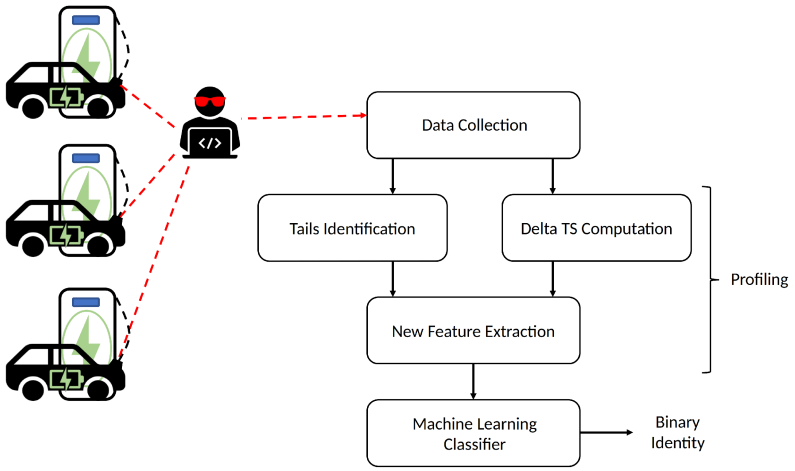


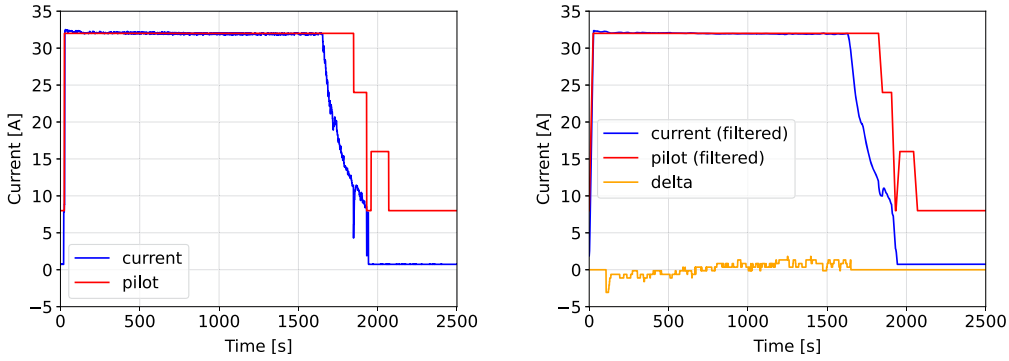
Fig. 3. Block diagram of *EVScout2.0*'s steps.

and, therefore, can be easily exploited by malicious users. Since the exchanged current during the charging phase is a user's generated data, it comprises features and recurrent behaviors useful for profiling attacks. *EVScout2.0* identifies and extracts those physical features that are representative of every single EV, such that we can assert with sufficient confidence if and when a specific user is connected to the charging grid.

Figure 3 shows the block diagram of *EVScout2.0*'s steps. *EVScout2.0* starts with data collection. To profile EVs the attacker must collect multiple charging sessions for each target EV. We will discuss this requirement in Section 7.1, assessing the number of training examples the attacker needs to collect to profile an EV with sufficient confidence. Once collected the charging TSs (i.e., the dataset), *EVScout2.0* automatically computes the features that characterize each EV. To this aim, in the following, we propose a strategy to exploit the behavior of batteries during the charging process. In particular, as proposed in EVScout [10], assuming that the attacker has access only to the ampere-based electrical quantities, we exploit the current behavior during the constant voltage phase. Leveraging the nomenclature in Reference [39], we name the current TS during the constant voltage phase as *tail*. In Section 4.2, we describe how *EVScout2.0* extracts the tails.

Notice that the choice of exploiting tails is due to the assumption that the attacker has only access to the ampere-based TS. If the attacker has access to voltage values, then the corresponding features can not be extracted from the tail, as the tail corresponds to the constant voltage phase. Therefore, features extracted from voltage values during constant voltage may be underrepresentative of the battery's behavior. If the attacker has access to both current and voltage TS, then current features can be extracted from tails, whereas voltage features can be extracted during the constant current phase.

By noticing that each battery follows the current limits imposed by the pilot differently, we generate a further TS to be used to extract more features. Together with the tail, in *EVScout2.0*, we exploit the Delta TS, i.e., the TS given by the punctual difference between the current TS and the pilot TS during the constant current phase. Delta TS hence includes all the data from the beginning of the TS up to the beginning of the tail. More precisely, since the first few seconds of the charging are generally noisy, we start our delta computation after the first few samples. We believe that this derived TS uniquely characterizes the behavior of each specific EV, as we will explain in Section 4.3.



(a) Raw current and pilot profile.

(b) Current and pilot filtered with a moving average filter with $N_{avg} = 25$.

Fig. 4. Normal current trace and filtered current trace with respect to the pilot during a sample charge. The delta is also plotted multiplied by a factor of 10 to increase understandability.

4.2 Tail Identification

Charging sessions are not necessarily comprehensive of the constant voltage phase, as a user may need to leave before the full charge is reached. In Reference [39], the authors presented a framework to cluster similar charging behavior based on the charging tail. We exploit this portion of the TS to perform more detailed profiling. Since *EVScout2.0* exploits tails during the constant voltage phase, we adopt the algorithm we proposed in Reference [10] to identify whether the considered session includes a constant voltage phase. The presence of a tail implies that the session terminates with full SoC and eventually zero-current exchanged between EV and EVSE. Tails, however, can not be uniquely identified by the presence of zeros in the current TS, as this may be due to idle phases during the power scheduling process at the grid side. Furthermore, scheduling may cause shot noise in the TS also after full SoC, leading to spikes in the TS. Therefore, we designed a suitable tail reconnaissance algorithm.

To mitigate the effects of scheduling and highlight the trends in the considered TS, we propose applying a suitable filter. In particular, we filter both the current TS and the pilot TS with a length N_{avg} moving average filter. Given time instant t and denoting the electric current value at time t as $c(t)$, the output value $y(t)$ of the moving average filter at time t is given by

$$y(t) = \frac{1}{N_{\text{avg}}} \sum_{m=0}^{N_{\text{avg}}-1} c(t-m). \tag{1}$$

The effects of the moving average filter are shown in Figure 4. We see that, with respect to the non-filtered current TS in Figure 4(a), the current TS in Figure 4(b) has a smoother behavior as the filter removes most of the noise and scheduling artifacts. Notice that different filter implementations can be considered, e.g., low pass filter. However, a low pass filter requires a more accurate design and leads to ringing effects, which may be misleading for trend and, hence, tail identification.

If the filter has a sufficient length, then its effects include spikes removal. This eases the identification of tails in the TS, as we can rely on the presence of steady zero values when the full charge is reached. In detail, if the current TS assumes zero values from t_{start} up to its end, then we can assume that full SoC has been reached. Tails are characterized by a descending trend in the TS, as shown from the current behavior during the constant voltage phase in Figure 2 and verified in Figure 4. By forward analysis of the current TS, it is difficult to identify the time instant corresponding to the beginning of the tail, as this would imply the overall TS analysis. Therefore, we propose

to proceed backward from the point where full SoC is obtained. Proceeding from t_{start} backward, we identify the tail by accounting for the number of samples in the current TS reporting an ascending trend. Notice that, even though scheduling and noise could affect the trend of the TS, its effects are mitigated by the moving average filter (see Figure 4(b)). A perfectly backward-ascending trend is given by a negative difference between the values at time t and $t-1$, i.e., $y(t)-y(t-1) < 0$. However, we notice that tails do not always exhibit a perfectly backward-ascending trend. In fact, if the non-filtered TS is affected by heavy noise, then its effects are still visible after filtering. Therefore, we relax the concept of perfect backward-ascending trend including samples for which $y(t)-y(t-1) \leq \varepsilon$, with ε being a small positive value. Furthermore, we also allow for short descending trends by accounting for $T_{\max}$ consecutive segments for which $y(t) - y(t-1) > \varepsilon$. If this is the case for $T_{\max}$ consecutive samples, then the trend is considered fully descending and hence discarded.

Based on the considerations mentioned above, the steps of the tail extraction algorithm are shown in Algorithm 1. We denote as $\mathcal{I}$ the set of EV indexes. Notice that each current TS is associated with a unique pilot TS. We denote as $\mathcal{C}(i)$ and $\mathcal{P}(i)$ the sets of the current and pilot TS associated with ID $i \in \mathcal{I}$, respectively. We assume that the two sets are ordered such that the current and pilot TS associated with a certain charging session are associated with the same index in the two sets. We define the set $\mathcal{W}(i) = \{\mathcal{C}(i), \mathcal{P}(i)\}$, whose elements (c, p) are the couples of current and pilot time series, respectively, taken from sets $\mathcal{C}(i)$ and $\mathcal{P}(i)$. For each TS $c \in \mathcal{C}(i)$, we calculate the filtered current and pilot TS, respectively, denoted as $\tilde{c}$ and $\tilde{p}$ by applying the filtering function (1). We then search for the time instant t_{start} starting from which $\tilde{c}$ is composed only by zero values. If t_{start} is found, then the current time series might contain a tail, and we hence proceed to identify the number of samples of the tail. Given our definition of backward ascending trend, we compute the number of samples for which the filtered time series is such that $\tilde{c}(t) - \tilde{c}(t-1) \leq \varepsilon$. As short descending trends are also allowed, we account for the number n of consecutive descending samples. However, if this trend is persistent, then we should discard these samples. Therefore, we set a threshold value $T_{\max}$ for the number of descending samples, after which, we stop the counter. Instead, if an ascending segment is found after a descending samples series, then the counter is reset. Given the number S of tail's samples, the tail $\tilde{c}(t_{\text{start}}, s)$ is obtained from the filtered current TS, starting from $t_{\text{start}} - s$ up to t_{start}. The tail $\tilde{p}(t_{\text{start}}, s)$ associated to the pilot TS is analogously obtained, starting from $t_{\text{start}} - s$ up to t_{start}. Both current and pilot TS are eventually added, respectively, to the set $\mathcal{T}_c(i)$ and $\mathcal{T}_p(i)$ of current and pilot TS tails associated with EV ID i.

4.3 Delta TS Computation

Aside from current tails, we compute another TS to be used for extracting features for the classifiers. Since different batteries' charging sessions have different charging parameters, the maximum current that can be absorbed is variable and characterizes the specific vehicle [10]. Furthermore, pushed by advanced charging algorithms [27], during some periods, the EV can be forced into charging at a lower current, e.g., to deal with peak leveling during peak hours. However, as visible in Figure 4, generally, the battery does not charge at the exact amount of energy expected from the pilot signal. Furthermore, the absorbed current often exhibits small variations around the maximum current deliverable by the charging column. It is particularly true when considering the behavior of the two TSs is the time preceding the tail. To capture these changes, we compute *Delta TS*, i.e., the TS given by the combination of the current TS and the control pilot TS during the constant current phase (i.e., the period preceding the tail).

To compute Delta TS, we calculate the difference between the current and pilot TSs at each time instant during the constant current phase. While the pilot TS generally is not affected by noise, the current TS instead exhibits some tiny positive and negative spikes, as shown in Figure 4(a). While

the tails generally follow a decreasing trend, the values assumed by the TSs before the tail (i.e., the ones used to compute the Delta TS) are generally more constant. Since the moving median filter provides better performance in removing noise and spikes when the data in the neighborhood of the peak are quite constant [33], we use it instead of the moving average filter used in Section 4.2.

Given time instant t, we denote the electric current value at t as $c(t)$ and the correspondent pilot value as $p(t)$. The resulting point at time t in the Delta TS can be expressed as:

$$z(t) = p(t) - median\left(c\left[t - \left\lfloor \frac{N_{\text{avg}}}{2} \right\rfloor, t + \left\lceil \frac{N_{\text{avg}}}{2} \right\rceil\right]\right), \tag{2}$$

where $c[a, b]$ represents the array of values of the TS c from $t = a$ to $t = b$, N_{avg} is the filter length, and $median(x)$ is the median value of array x (i.e., the middle value separating the greater and lower halves of x).

4.4 Improved Feature Extraction

Segmentation represents a classical approach for feature extraction in TSs [12, 15, 18]. Unfortunately, segmentation is not a viable solution, since the TSs we consider here are generally short with no stationary components. Therefore, we do not further process tails before extracting features. In our previous work [10], we computed the mean, mode, median, max value, standard deviation, auto-correlation, length of the tail, and the slope of the linear approximation for each tail. Furthermore, we used as a feature the total kW delivered and the overall session time duration, leading to a total of 18 features considering both pilot and current TSs independently. Instead, in this work, we adopted a more sophisticated feature extraction process that can extract several more features.

We analyze widely used feature extraction tools that can automatically extract features from a given TS [5, 18, 31]. We use Time Series Feature Extraction based on Scalable Hypothesis tests (`tsfresh`) [11], which is available as a Python package easily integrable with other tools such as scikit-learn [34]. `tsfresh` exploits the power of 63 TSs characterization methods to extract hundreds of features from a TS. Moreover, it contains functions to slightly reduce the number of features to remove the less meaningful ones. We use `tsfresh` to extract features from the current tails and the delta current-pilot TSs. It is worth mentioning that, with respect to the previous work [10], we decided not to employ features extracted from the tails of the control pilot TS, since the pilot tail is generally based on the optimization algorithm [27] and not on the behavior of the battery. For example, even if the battery has reached its maximum SoC, the control pilot could remain at a high value if the grid has energy available. In the same way, if many vehicles start requesting energy, then the control pilot will reduce its value independently of the SoC of our target EV. Furthermore, we removed the needs for the duration of the charge and the total energy absorbed by the battery (kW), since they can depend on the user behavior and the SoC of the battery at the beginning of the charging process.

Since `tsfresh` can generate around 800 features for each TS both from the time and frequency domains. We removed those that are not relevant to our classification problem. To reduce the number of features, from now on denoted as *NoF*, we select the most significant ones by using `SelectKBest` of scikit-learn [34] with the `chi2` function, which is suitable for classification purposes. Since the chi-squared measures the dependence between stochastic variables, we can highlight and select only the features that offer more information for the classification. We employ this strategy with respect to other more complex methods such as Random Forest feature reduction, since `SelectKBest` can achieve approximately the same results with lower computational complexity. As expected, due to the high variance in the charging tail, the most meaningful features are related to the tail TS. For example, high feature importance score is assigned to the values that

ALGORITHM 1: Tail extraction algorithm.

Data: $\mathcal{W}$, $\mathcal{I}$, $C_{\max}$, $T_{\max}$, N_{avg}
Result: $\mathcal{T}_c$, $\mathcal{T}_p$
for $i \in \mathcal{I}$ **do**
  for $(c, p) \in \mathcal{W}(i)$ **do**
    /* filter the current and pilot time series via moving average */
    compute $\tilde{c}$ and $\tilde{p}$ via (1);
    /* find the charging ending instant */
    compute t_{start};
    /* if charging reaches an end */
    if t_{start} *found* **then**
      $n = 0$, $s = 0$;
      /* detect length of backward ascending trend */
      for $t = t_{\text{start}}, t_{\text{start}} - 1, \ldots, 1$ **do**
        if $\tilde{c}(t) - \tilde{c}(t-1) > \varepsilon$ **then**
          /* increase counter of backward-ascending samples */
          $n = n + 1$;
        else
          /* reinitialize counter after descending trend */
          $n = 0$;
          $s = s + 1$;
        end
        if $n = T_{\max}$ **then**
          /* maximum length reached */
          exit loop;
        end
      end
      /* add the detected time series to the sets */
      $\mathcal{T}_c(i) = \mathcal{T}_c(i) \cup \tilde{c}(t_{\text{start}}, s)$;
      $\mathcal{T}_p(i) = \mathcal{T}_p(i) \cup \tilde{p}(t_{\text{start}}, s)$;
    else
    end
    /* move to the next time series */
    go to next c;
  end
end

are more than r times sigma distant from the mean of x, with different $r \in [3, 5, 6, 7]$, indicating the number of outliers that the moving average filtering has not eliminated. Other significant important features include the number of peaks, the standard deviation, the quantiles, and the c3 statistics [37] (a coefficient used to measure non-linearity). The features computed from the delta TS are instead less relevant, but the standard deviation has a significant importance score. More details on the features extracted can be seen in the `tsfresh` documentation [11].

5 EVALUATION FRAMEWORK DESCRIPTION

In the following, we present the experiments we use to test *EVScout2.0*. In particular, in Sections 5.1 and 5.2, we describe, respectively, the ACN Infrastructure and Dataset on which we based our

analysis. We then explain how the new version of the dataset differs from the one in the previous work. Then, in Section 5.3, we outline the machine learning classifiers we use to profile EVs.

5.1 The ACN Infrastructure and Dataset

To test *EVScout2.0*, we exploit the **ACN (Adaptive Charging Network)** proposed in Reference [27]. It consists of level two EVSEs connected with a central controller that regulates power exchanges in the grid. Employing an online optimization framework, the ACN allows adapting the power exchanged in the grid, satisfying users' power demand while coping with the grid's capacity limits. The dataset comprises $50kW$ DC charging sessions from different ACNs sites, each reporting user-specific measurements such as the arrival and departure time, the kW/h delivered, current and pilot TSs collected between the EV connection and disconnection time. Notice that, although the user may have planned for a full recharge during the selected period, this may not be reflected in the TS. In fact, due to the variable number of connected EVs, the upper power limit of the grid, and the premature departure of the user, the battery may not be fully charged at disconnection time. Notice also that, in the ACN dataset, not all TS are sampled with the same period. However, we avoid upsampling with filtering, because it can introduce statistical features that are not representative of the analyzed battery. Each user in the dataset is identified by a unique ID associated with the owned EV. We specifically focused on the biggest site, Caltech, which contains charging sessions collected from 54 different EVSEs.

5.2 New Dataset

Like in the previous work [10], we decided to use the ACN dataset containing time series sampled with an average period of 3.8 s. However, since the dataset was recently enlarged, we expanded the number of EVs considered from 22 to 187. To generate the dataset, we selected all the available EVs up to June 18, 2021, from the `caltech` site (one of the three locations available in the dataset). We also excluded by default EVs without charges and charges without any EV assigned (i.e., anonymous charges). To download the dataset, we employ the Python APIs provided by the ACN Dataset [27]. The number of charges associated with each EV ranges from one to over 300. However, not all the charges are performed up to the full SoC and thus, not always a tail is available. We, therefore, remove these TSs and the corresponding EVs, because our method is based on the presence of the tails to compute features on both current and delta. Furthermore, we consider all the EVs with more than eight employable charging processes associated. We made this choice to be able to effectively use cross-validation even for those EVs with a small number of charges. After this cleanup, 137 EVs remain in the dataset, resulting in a dataset more than six times bigger than the one used to test EVScout in Reference [10].

5.3 Classification Algorithms Comparison

The first step of *EVScout2.0* is to build a suitable dataset to be exploited for profiling, as explained in Section 5.2. It is worth mentioning that the dataset does not provide any information regarding the brand and the model of the EVs. Since the energy behavior highly depends on the chemical reactions of the single battery, we believe that *EVScout2.0* could be able to distinguish EVs of the same model. Furthermore, the batteries employed by the analyzed EVs all belong to the same class, i.e., constant current/constant voltage. However, since both classes discussed in Section 3 show particular behaviors in time, we believe that our attack could be easily extended to the constant power/constant voltage class. The effectiveness of *EVScout2.0* in these cases will be investigated in future works.

For each session, *EVScout2.0* first identifies whether a tail is present and discards all the other sessions. Then it builds a feature vector for each tail, associating it with the ID of its corresponding

EV. We test *EVScout2.0* across all EVs in the dataset by averaging the performance obtained with every single classifier. In particular, we implement a binary classifier (One-vs.-Rest strategy) for each EV, and we associate each feature vector of the *target* EV with label 1, otherwise with label 0 all the other traces (i.e., *non-target* vehicles). The overall performance of the obtained classifiers is averaged considering 100 randomly created training and testing sets, except **RF (Random Forest)** and **ADA (ADA Boost)** classifiers for which, for timing reasons, we consider 25 iterations. The overall performances of *EVScout2.0* are obtained by averaging the results obtained for each ID's classifier to mitigate too high or low results caused by a particular vehicle.

Let us denote as Q the ratio between the number of feature vectors associated with the target EV and the number of feature vectors associated with other EVs. Hence, Q measures the amount of unbalancing in the considered dataset. To further assess the performance of *EVScout2.0*, each classifier is tested for multiple Q values. Regardless of the value Q, the 80% of the dataset has been used for training and the remaining 20% for testing unless otherwise specified. As the number of feature vectors of a single EV is smaller than the overall number of feature vectors, when considering small Q values, the set of feature vectors associated with other EVs is randomly created from the overall set. Another value that can lead to different results is the number of features NoF employed in the classification. Since our feature extraction strategy returns about 1,500 features, using them all can lead to overfitting. We show in the next sections how different NoF values affect the classification performance and provide a justification for choosing a suitable NoF value that provides a good threshold to balance classification performance and overfitting.

6 *EVSCOUT2.0* PERFORMANCE: VEHICLE PROFILING

We first assess the performance of *EVScout2.0* in terms of profiling every vehicle based on its charging behavior. To this aim, we implement and compare common machine learning algorithms for classification. We describe in Section 6.1 the classifiers, and we provide a discussion on their hyper-parameters setting. Then, in Section 6.2, we present the results obtained via *EVScout2.0* with the different classifiers.

6.1 Classification Algorithms

Once features have been identified and selected, *EVScout2.0* feeds them to a binary machine learning classifier. The profiling task can be formulated as a supervised classification problem, where a two-class classifier is trained with both features from the target EV and features from all other EVs. In particular, we assume that a classifier whose input is the features vector from the target EV shall return output value 1, otherwise it shall return output value 0. In the literature, this approach is also called *One-vs.-Rest*, and more precisely, it aims at creating a specific model for a single device by using the other class, composed of other different devices, to create a decision bound around the class under consideration. If, on the one hand, this approach requires a different model for each class, on the other hand, it allows focusing on a single class, leading to a more robust model for the specific class.

We evaluate our pipeline by using and comparing six different common machine learning models that are often used in the field [7, 36], namely:

- **SVM (Support Vector Machine)** classifier [40];
- **kNN (k-Nearest Neighbors)** classifier [22];
- **DT (Decision Tree)** classifier [9];
- **LR (Logistic Regression)** classifier [23]
- RF classifier [8];
- ADA classifier [20].

Table 1. Grid Search Cross-validation Parameters for Each Model

Model	Parameter	Values
SVM	kernel	[“rbf”]
	regularization (c)	$[1, 10, 10^2, 10^3]$
	gamma (γ)	$[10^{-4}, 10^{-3}]$
kNN	n_neighbors	$[1, \ldots, 10]$
	weights	[“uniform,” “distance”]
	metric	[“euclidean,” “manhattan”]
DT	criterion	[“gini,” “entropy”]
	max_depth	[8, 10, 14, 30, 70, 110]
LR	max_iter	[5,000]
	regularization (c)	$[10^{-2}, 1, 10^2]$
RF	n_estimators	[50, 200, 1,000]
	max_depth	[10, 100, *None*]
ADA	n_estimators	[10, 100, 500, 1,000, 5,000]

Hyper-parameters optimization is obtained via grid search with cross-validation to fine-tune our models and extract the best results. Table 1 indicates the different parameters employed in the grid search for each model. The training set is suitably divided into training and validation sets, which we test on a grid of possible hyper-parameters. Notice that all six classifiers are standard machine learning algorithms without deep architectures. Although deep learning automates the feature extraction process, a large number of samples shall be used to train deep architectures effectively. The use of non-deep structures allows us to show the feasibility of *EVScout2.0* over our currently available dataset and, simultaneously, control the feature relations during the classification process. The same motivation resides behind the choice of binary classifiers. In fact, a single multi-class classifier can be designed to have a single class for each EV. However, multi-class classifiers require a larger dataset for training purposes than binary classifiers. Although we consider a larger dataset than our previous work [10], it is not big enough to provide interesting results with deep models or a multi-class scenario.

6.2 Profiling Results

We exploit the implementation of the classification algorithms employing the scikit-learn library [34]. Results are assessed in terms of precision P, recall R, and $F1$. We present numerical results assessing the validity of *EVScout2.0* as a function of Q, the amount of unbalancing in the dataset. This measure provides evidence of how the model is resistant in constrained scenarios where the attacker has few charging traces available for the target vehicle. Since traditional performance measures such as $F1$ may be misleading when considering highly unbalanced datasets, the **geometric mean (G-Mean)** between recall and specificity has been proposed as a suitable performance metric [19]. Therefore, we consider G-Mean an accurate indicator of the validity of *EVScout2.0* for large Q values. By denoting as TP, TN, FP, and FN, respectively, the number of true positive, true negative, false-positive, and false-negative outcomes, we can express the recall as $R = \frac{TP}{TP+FN}$, and specificity as $\alpha = \frac{TN}{FP+TN}$. G-Mean is hence obtained as G-Mean $= \sqrt{\alpha R}$. We recall that we present each score as the average over different vehicle-specific trained models to

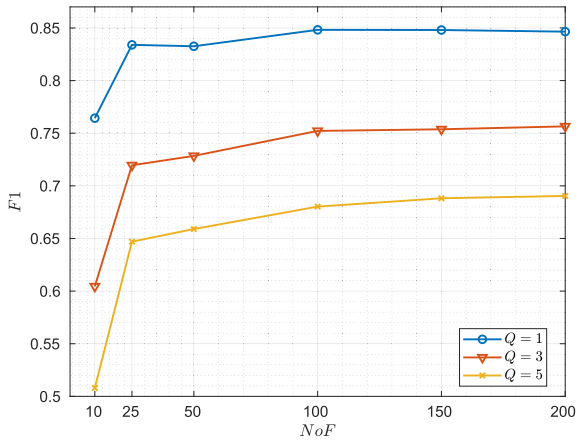


Fig. 5. $F1$ score of kNN classifier with different numbers of features NoFs.

obtain more robust scores and avoid biases due to specific vehicles with a highly diverse charging profile.

Since the tail extraction process is automated, the number of extracted tails also depends on the parameter values. Based on our previous work [10], we set N_{avg} (i.e., the size of the moving average filter) to 25, which provides the best value in terms of classification scores. A smaller filter length would fail to remove noise on the TSs, while a bigger one would filter out important data characteristics. We used the same value for the median filter to maintain coherence in the TSs, since, albeit, with the necessary differences, the two filters are quite similar. Notice that the filtering process aims to improve tail identification and classification performance. Therefore, the optimal filter length is obtained by a trial-and-error process instead of selecting a length based on, e.g., correlation analysis.

We employ the classifiers provided by the scikit-learn Python library [34]. For each model, we perform a `GridSearchCV` to tune the model using grid search with cross-validation. In the following, we present an analysis of the results with respect to different parameters and values.

6.2.1 Number of Features NoF. First, we analyze the scores based on the number of features NoF maintained after the feature extraction phase. In Figure 5, we plot the $F1$ scores for different ratios Q using the same model (kNN) but varying the numbers of features NoF from 10 to 200. We can see a significant increase in the score going from 10 to 25 features, especially for higher ratios Q. From 100 features up, the increase is instead negligible, meaning that the features already capture almost all the entropy available. For this reason, we selected $NoF = 100$ for the other experiments in this article unless otherwise specified.

6.2.2 Unbalance of the Dataset Q. To understand how our models deal with the unbalance of the dataset (i.e., the ratio between the number of feature vectors associated with the target EV and the number of feature vectors associated with other EVs), we test all the six models against different values of Q. We test Q values ranging from 1 (i.e., the same number of feature vectors for the target and the other EVs) to 5 (i.e., five times more features vectors not related to the target EV). We crafted the *non-target* class in the training set to obtain more robust scores by randomly sampling the charging traces among all the *non-target* vehicles. Furthermore, to make the classification problem more "open-world," we inserted in the test set, as *non-target* class, traces of vehicles without occurrence in the training set.

Figure 6 shows the results obtained by *EVScout2.0* for different Q values. It is possible to notice how all the scores decrease with Q for all six models. As the number of considered EVs increases with Q, the chance of two users having the same EV model or having EVs with similar charging profiles increases. This is reflected in a worsening of classifiers' performance. With no unbalancing (i.e., $Q = 1$), we reach the highest precision of 0.88 with the RF classifier. However, almost all the classifiers reached at least 0.86 in the recall, except for DT, which has lower performances almost for every indicator. We notice that the RF classifier is the most resistant to changes in Q if we look at the precision, while LR offers the best recall scores. If we look at both scores, even if the absolute values are generally lower, then kNN and ADA seem to be the most resistant to higher Q. Furthermore, we can observe that for the maximum $Q = 5$, precision and recall are, respectively, 0.77 (with RF) and 0.71 (with LR), meaning that EVs can still be profiled with sufficient confidence.

As for the other scores, also $F1$ degrades for increasing values of Q for all the models. We can see that RF is the best one for almost all the ratios Q, while for the highest two (i.e., $Q = 4.5$ and $Q = 5$) ADA performed slightly better. However, this is not the case regarding G-Mean, confirming its validity for unbalanced datasets. In particular, ADA presents a large variance in terms of G-Mean, a sign that is not a suitable algorithm for highly unbalanced datasets. Other algorithms, such as RF, kNN, and LR, are instead able to maintain high values of G-Mean for every value of Q. This shows that profiling can be achieved with good results irrespective of the amount of unbalancing in the dataset, i.e., a single user can still be profiled based on its charging profile also in largely populated networks.

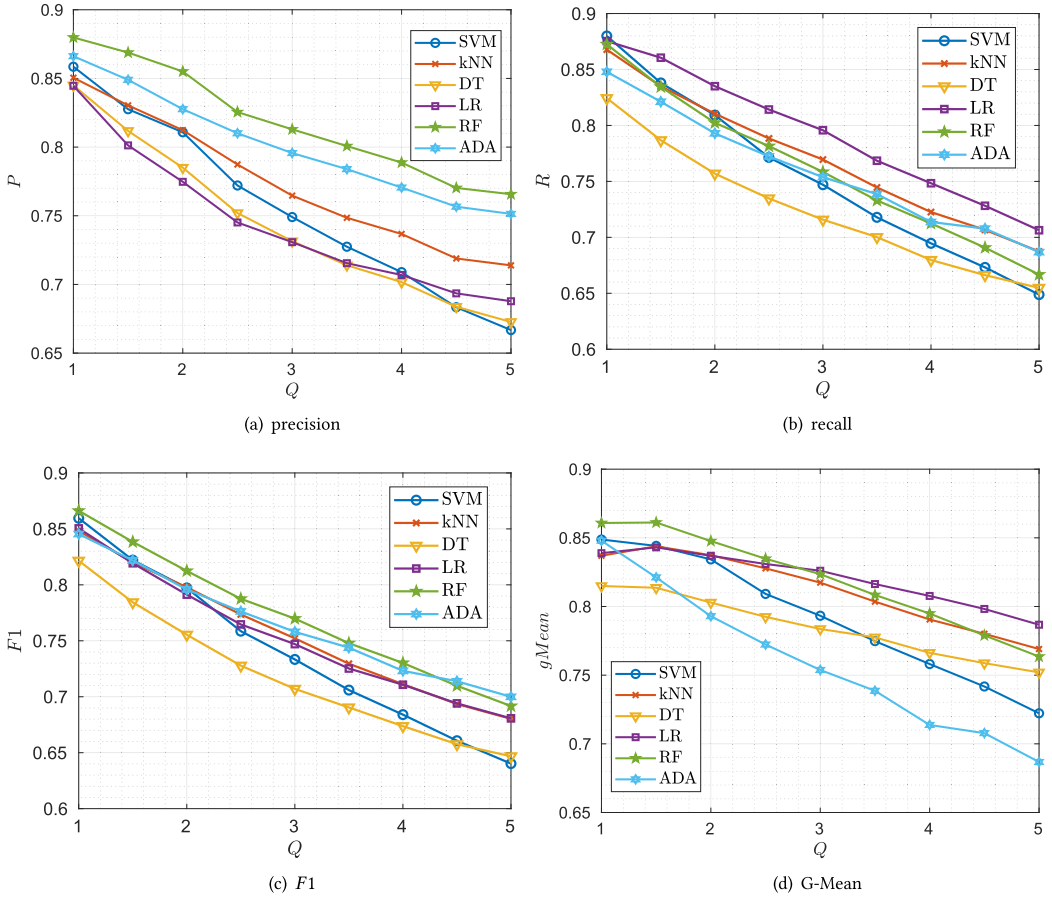
All these considerations must be considered when designing *EVScout2.0*. The best algorithm for each case can be selected if the dataset distribution is known. In particular, RF is advisable for perfectly balanced datasets, LR for highly unbalanced datasets. Instead, if the dataset is unknown and a resilient model is needed, then kNN can be a good choice. In fact, we employed kNN in many experiments from now on, also thanks to its fast training time with respect to other models such as RF or ADA.

7 *EVSCOUT2.0* PERFORMANCE: ADDITIONAL PERFORMANCE ANALYSIS

In addition to the classification-based analyses, we included additional scenarios based on the training set characteristics. Since the data conditions may be different in a real-world scenario, we propose an analysis considering different data properties in the following. In particular, in Section 7.1, we examine different training set size values to assess the minimum number of charges needed to obtain sufficiently high classification scores. In fact, a large number of labeled traces in a real-world attack may be challenging to obtain. In Section 7.2, we investigate if and how much the Li-ion battery's degradation impacts the performance of *EVScout2.0*. This analysis can be useful to understand how a model can still be precise when dealing with natural phenomena like physical battery degradation.

7.1 Training Size Variation

In a real-world scenario, an attacker may be limited by the number of labeled charges s/he can get for a target vehicle. For instance, the attacker may not want to leave its malicious device in the field too much to reduce the possibility of being detected. To simulate this scenario, we analyze the performance of *EVScout2.0* while varying the training set size. We avoid using as target EVs with less than 70 charges with tails, while to create the group with the others EVs, we employed the whole dataset. As a testing set, we always use the last 20% of the available charges. For the training set, we set fixed values from the 80% to the 10% of the min number of charges for each EV (i.e., 70). In other words, the training set size ranges from 7 to 56 feature vectors for each EV, with steps of 7 charges.



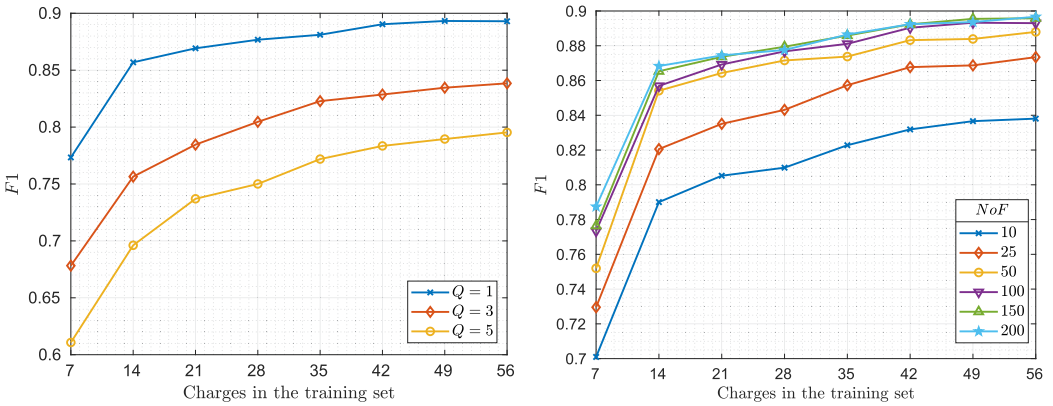
(a) precision (b) recall

(c) $F1$ (d) G-Mean

Fig. 6. Performance of *EVScout2.0* for a different amount of unbalancing in the dataset. Results are shown for the different classifier algorithms and $NoF = 100$. We see that good classification performance are obtained for all classifiers. As Q increases, some classifiers are more robust than others to increasing unbalancing. Results on G-Mean show that profiling can also be achieved in largely populated networks.

In Figure 7(a), we can see how the $F1$ decreases while decreasing the training set size. However, we can appreciate a significant fall only for the smallest values of the training set, while the increase is less relevant for training set sizes bigger than 14. Above 42 charges, the performance increase is almost negligible, especially considering the smaller Q ratios. We can also look at the absolute values of the $F1$. It is greater than 0.69 for the most unbalanced dataset when using 14 feature vectors, showing discrete classification performance even with small training sets. By considering the total number of vehicles per training set size and the imbalance ratio, we can compute the number of vehicles of the target class in the different scenarios. In particular, we can notice that with 7 charges (i.e., the training set size 14 if $Q = 1$, training set size 21 if $Q = 3$, and training set size 35 if $Q = 5$), we can obtain an F1 Score of at least 75%. With only 7 charges, we can achieve a good classification precision for the target vehicle.

Furthermore, we also analyzed the impact of both training set reduction and feature reduction. In Figure 7(b), we see the different behavior of the $F1$ score while varying the number of features and the training set size (we show the data only for $Q = 1$ for clarity). As expected, there is a



(a) Training set size changing.

(b) Changing NoF and training set size for $Q = 1$.

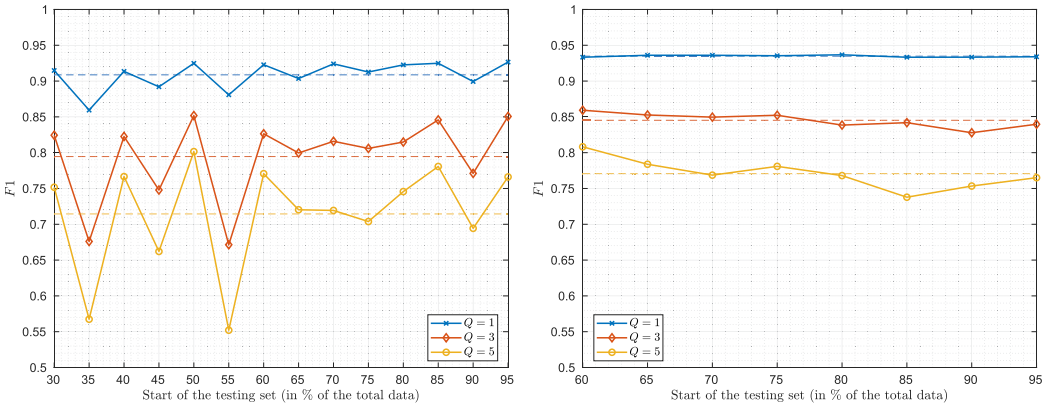
Fig. 7. $F1$ scores of kNN classifier while reducing the training set size.

clear drop in the performances for the lower training set sizes. However, this can be partially compensated by employing a bigger number of features, up to 100. Over this threshold, the gain is negligible, coherently with what is presented in Section 6.2 and Figure 5.

7.2 Battery Degradation Performance

The degradation of a Li-ion battery used in an EV is widely discussed in literature [35]. During a battery life, many aspects can adversely affect its performance depending on the number of charge cycles, aging, operating, and storing conditions. Degradation on EV batteries can lead to a shorter traveling distance and a reduced battery's available power output [6]. Generally, a battery is considered at the end of its life when it has already lost 20% of its initial capacity. Modern Li-ion batteries should have a five to ten-year calendar life, depending on the materials and how they are managed. They should be able to supply between 1,000 and 2,000 cycles of charge. However, many behaviors, such as keeping the battery at a high SoC or high temperature, or often employing fast chargers, can reduce even more the lifetime of the battery [6, 44].

Being aware of this problem, we investigate if the battery degradation affects our profiling model. Since the charge profile of an EV will always be different due to the variation of the physical properties of the battery, we analyzed how the extracted features degrade the performance of *EVScout2.0* in time. To test how our model behaves in this context, we selected the 10 EVs with the higher number of charges (i.e., more than 150 charges for each EV) spanning about two years. We train the kNN classifier in the first part of the data, and we then test it over multiple consecutive test set batches. In particular, the training set accounts for the TS measured in a specific period and represents the baseline to measure the successive battery degradation. Each testing set is given by batches of Z chronologically consecutive TSs, with $Z = 5\%$ of the total available testing data. In other words, we considered as each test set a sliding time window of consecutive TSs. Initially, we train employing only the first 30% of the data for each EV. These results are shown in Figure 8(a), where it is possible to see a small difference in scores while shifting the testing set. However, we cannot blame the physical battery degradation for these results for a series of motivations. First, the reduction is not important in absolute terms (i.e., less than 0.1 for $Q = 1$) and can be due to different behaviors of the users in different periods (e.g., detaching the EV before full SoC). Second, it is not the monotonous decrease that we would have expected. Instead, the scores seem not to follow any pattern, showing good classification results also in the last steps (i.e., testing set given



(a) Train set size 30% of the data.

(b) Train set size 60% of the data.

Fig. 8. $F1$ score for different ratios while varying the start of the testing set. Each value in the horizontal axes represents the score on a testing set composed of the 5% of the data starting from the point forward. Dashed horizontal lines represent the mean of the $F1$ for each ratio.

by more recent TSs). Third, the small size of the training set employed can be a partial motivation for the random behavior of the scores.

To remove the training set size as a variable to create strange behavior in the results, we perform a second training using the 60% of the data. By doing so, we obtain a chart with fewer points, as shown in Figure 8(b). In this case, scores are almost constants with tiny variations that are expected in the context.

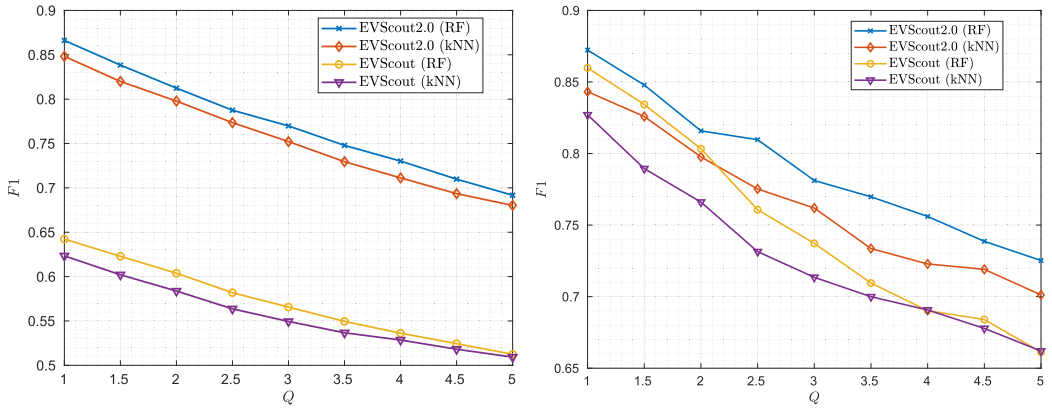
This demonstrates how *EVScout2.0* can perform well, also considering a time distance of almost two years between the training and the testing data. Although not affected by the small battery degradation that could happen in this time frame, it could still be affected by unusual and unpredictable behaviors of the EVs owners or by the ACN scheduling algorithm, especially when considering highly unbalanced datasets and a small training set. Furthermore, this experiment can be seen as a remark on the need for a big training set as presented in Section 7.1. We will provide more analysis on the degradation in future work, considering datasets that span more than two years.

8 EVSCOUT2.0 PERFORMANCE: COMPARISON WITH EVSCOUT

In our previous work [10], we performed experiments similar to those discussed in previous sections but employing a different pipeline, a different feature extraction process, and a smaller dataset composed of only 22 EVs. Since it was one of the first works on the privacy of the EV charging systems, we use the results of Reference [10] as a comparison baseline. In particular, we propose two different comparisons: one using the previous EVScout on the new and extended dataset and one using the new *EVScout2.0* on the small dataset employed in Reference [10]. It is worth mentioning that we employed the exact dataset used in Reference [10] and not simply the same EVs updated with the new charges.

8.1 EVScout on the New Dataset

We tested EVScout [10] in our new dataset. To generate results comparable with those presented in this article for *EVScout2.0*, we employed the same dataset with 137 valid EVs, even if EV with less than eight charges could also be used in EVScout. We performed the same pre-processing phases



(a) Comparison on the new dataset.

(b) Comparison on the original dataset.

Fig. 9. $F1$ values of the comparison with the previous work [10].

and assessed the performance for $N_{avg} = 25$. Results are shown in Figure 9(a) for different values of Q. We can see a clear dominance of *EVScout2.0* with respect to its previous version in the new dataset composed by 137 EVs. The difference in the classification performance ranges from about 0.15 for $Q = 5$ to more than 0.20 for $Q = 1$. We remark that the different charging traces (also for the same vehicle) belong from different EVSEs in the same site. Therefore, the results confirm that the models implemented are resistant to multiple EVSEs and that different charging stations do not significantly impact the charging behavior.

8.2 *EVScout2.0* on the Previous Dataset

Furthermore, we test our new algorithm *EVScout2.0* on the same dataset employed for the testing of EVScout. Since *EVScout2.0* needs at least eight charging sessions comprising tails for each EV, we have to discard 4 EVs, resulting in a total of 18 EVs. We show the results of this experiment in Figure 9(b). Even if the enhancement is less pronounced with respect to the new dataset scenario, we can see a clear improvement in the performance of *EVScout2.0* with respect to its previous version, especially considering high unbalancing Q.

9 POSSIBLE COUNTERMEASURES

To deal with information leakage from smart meters Reference [24] proposes to obfuscate the communication between user and supplier through rechargeable batteries. This solution is effective in modifying the demand-response correlation in the measured data. This approach can be leveraged to mitigate the effects of *EVScout2.0*, where the EV's battery drains current from a secondary battery that communicates with the EVSE, masquerading the original battery's tail behavior. However, as the number of involved batteries doubles, this approach incurs a higher implementation cost at both EV and EVSE sides. Furthermore, the attacker may be able to extract features from the secondary battery and hence still be able to perform profiling. A similar concept is exploited in Reference [38], where noise is added to smart meters data via an adversarial learning framework. This idea can be exploited to mitigate the effects of *EVScout2.0* by adding a suitable amount of noise to the current required by the EV's battery during the tail phase. However, this would imply redesigning how EVSEs manage the current required by EVs, as the added noise may mislead both the attacker and the EVSE. The risk is, therefore, that the recharging process's efficiency drops. In Reference [16], the authors successfully applied a low-pass filter to remove the information

leakage due to the high-frequency components from the signal. However, the work targets mobile devices, which are very different from vehicles, with tiny batteries and different charging patterns. Other non-technical countermeasures are also possible. For instance, EV owners can be educated to check the presence of suspicious devices attached to the EVSE. Furthermore, running companies should often inspect their equipment for illegitimate devices and install closed-circuit TV to detect the presence of such devices. As described above, several solutions in the literature prevent the leaking of power side-channel information. However, most of them have been studied for the smartphone environment and are rarely implemented in reality. In future work, we will focus on identifying ad hoc solutions to prevent the inference of sensitive information via power-side channels, specifically in communications between EVs and EVSEs.

10 CONCLUSIONS

EVs security is a novel topic that raises many novel challenges in protecting such systems. As we demonstrated, introducing a charging system that utilizes personal information can lead to privacy leakage and profiling attacks. In this article, we extended our previous work proposing *EVScout2.0*. In particular, we extended the work in Reference [10] by employing a bigger dataset, a novel feature extraction technique, and compared more algorithms for the classification task. We also show how real-world constraints such as limited training test size and battery degradation over time impact the classification quality. With respect to the previous work, we employed a bigger dataset, going from 22 to 137 EVs. We employed six models capable of reaching precision and recall of 0.88 for the balanced datasets while still offering good results (precision 0.77, recall 0.71) with high unbalanced datasets. Furthermore, we evaluate the performance loss generated by a training set size reduction, showing how *EVScout2.0* can reach good performances even with a small training set. In addition, we assess that the proposed algorithm can correctly identify EVs even if the model is trained with data two years early. Finally, we showed that *EVScout2.0* is capable of attaining good classification performance, even in challenging scenarios such as highly imbalanced datasets or small training sets, proving to be a viable and effective solution for EV profiling. We believe this work can warn all the parties involved (i.e., the users, the manufacturers, and the scientific community) about the feasibility of profilation attacks in the growing V2G infrastructure.

REFERENCES

[1] Takafumi Anegawa. 2010. Characteristics of CHAdeMO quick charging system. *World Electric Vehic. J.* 4, 4 (2010), 818–822.

[2] Joseph Antoun, Mohammad Ekramul Kabir, Bassam Moussa, Ribal Atallah, and Chadi Assi. 2020. A detailed security assessment of the EV charging ecosystem. *IEEE Netw.* 34, 3 (2020), 200–207.

[3] Ya-shuang Bai and Cheng-ning Zhang. 2014. Experiments study on fast charge technology for lithium-ion electric vehicle batteries. In *IEEE Conference and Expo Transportation Electrification Asia-Pacific (ITEC Asia-Pacific)*. 1–6.

[4] Richard Baker and Ivan Martinovic. 2019. Losing the car keys: Wireless PHY-Layer insecurity in EV charging. In *28th USENIX Security Symposium (USENIX Security'19)*. USENIX Association, Santa Clara, CA, 407–424. Retrieved from https://www.usenix.org/conference/usenixsecurity19/presentation/baker.

[5] Marília Barandas, Duarte Folgado, Letícia Fernandes, Sara Santos, Mariana Abreu, Patrícia Bota, Hui Liu, Tanja Schultz, and Hugo Gamboa. 2020. TSFEL: Time series feature extraction library. *SoftwareX* 11 (2020), 100456.

[6] Anthony Barré, Benjamin Deguilhem, Sébastien Grolleau, Mathias Gérard, Frédéric Suard, and Delphine Riu. 2013. A review on lithium-ion battery ageing mechanisms and estimations for automotive applications. *J. Power Sourc.* 241 (2013), 680–689. DOI: https://doi.org/10.1016/j.jpowsour.2013.05.040

[7] Pallav Kumar Bera and Can Isik. 2021. Identification of stable and unstable power swings using pattern recognition. In *IEEE Green Technologies Conference (GreenTech)*. 286–291.

[8] Leo Breiman. 2001. Random forests. *Mach. Learn.* 45, 1 (2001), 5–32.

[9] Leo Breiman, Jerome Friedman, Charles J. Stone, and Richard A. Olshen. 1984. *Classification and Regression Trees*. CRC Press.

[10] Alessandro Brighente, Mauro Conti, and Izza Sadaf. 2021. Tell me how you re-charge, I will tell you where you drove to: Electric vehicles profiling based on charging-current demand. In *Computer Security – ESORICS 2021*, Elisa Bertino, Haya Shulman, and Michael Waidner (Eds.). Springer International Publishing, Cham, 651–667.

[11] Maximilian Christ, Nils Braun, Julius Neuffer, and Andreas W. Kempa-Liehr. 2018. Time series feature extraction on basis of scalable hypothesis tests (tsfresh - a python package). *Neurocomputing* 307 (2018), 72–77. DOI: https://doi.org/10.1016/j.neucom.2018.03.067

[12] Maximilian Christ, Andreas W. Kempa-Liehr, and Michael Feindt. 2016. Distributed and parallel time series feature extraction for industrial big data applications. *arXiv preprint arXiv:1610.07717* (May 2016).

[13] SAE EV Charging Systems Committee et al. 2001. SAE electric vehicle conductive charge coupler. *SAE, California*. https://www.sae.org/standards/content/j1772_200111/.

[14] Mauro Conti, Denis Donadel, Radha Poovendran, and Federico Turrin. 2022. EVExchange: A relay attack on electric vehicle charging system. arXiv preprint arXiv:2203.05266. (2022).

[15] Mauro Conti, Michele Nati, Enrico Rotundo, and Riccardo Spolaor. 2016. Mind the plug! Laptop-user recognition through power consumption. In *2nd ACM International Workshop on IoT Privacy, Trust, and Security*. 37–44.

[16] Patrick Cronin, Xing Gao, Chengmo Yang, and Haining Wang. 2021. Charger-surfing: Exploiting a power line side-channel for smartphone information leakage. In *30th USENIX Security Symposium (USENIX Security'21)*. USENIX Association, 681–698. Retrieved from https://www.usenix.org/conference/usenixsecurity21/presentation/cronin..

[17] Deloitte. 2020. Electric vehicles: Setting a course for 2030. Retrieved from: https://www2.deloitte.com/us/en/insights/focus/future-of-mobility/electric-vehicle-trends-2030.html.

[18] Houtao Deng, George Runger, Eugene Tuv, and Martyanov Vladimir. 2013. A time series forest for classification and feature extraction. *Inf. Sci.* 239 (2013), 142–153.

[19] César Ferri, José Hernández-Orallo, and R. Modroiu. 2009. An experimental comparison of performance measures for classification. *Pattern Recog. Lett.* 30, 1 (2009), 27–38.

[20] Yoav Freund and Robert E. Schapire. 1997. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.* 55, 1 (1997), 119–139.

[21] Raju Gottumukkala, Rizwan Merchant, Adam Tauzin, Kaleb Leon, Andrew Roche, and Paul Darby. 2019. Cyber-physical system security of vehicle charging stations. In *IEEE Green Technologies Conference (GreenTech)*. 1–5.

[22] Gongde Guo, Hui Wang, David Bell, Yaxin Bi, and Kieran Greer. 2003. KNN model-based approach in classification. In *On the Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE*, Robert Meersman, Zahir Tari, and Douglas C. Schmidt (Eds.). Springer Berlin, 986–996.

[23] David W. Hosmer Jr, Stanley Lemeshow, and Rodney X. Sturdivant. 2013. *Applied Logistic Regression*. Vol. 398. John Wiley & Sons.

[24] Georgios Kalogridis, Costas Efthymiou, Stojan Z. Denic, Tim A. Lewis, and Rafael Cepeda. 2010. Privacy for smart meters: Towards undetectable appliance load signatures. In *1st IEEE International Conference on Smart Grid Communications*. 232–237.

[25] Willett Kempton, Jasna Tomic, Steven Letendre, Alec Brooks, and Timothy Lipman. 2001. Vehicle-to-grid power: Battery, hybrid, and fuel cell vehicles as resources for distributed electric power in California. UC Davis: Institute of Transportation Studies. Retrieved from https://escholarship.org/uc/item/0qp6s4mb.

[26] Sebastian Köhler, Richard Baker, Martin Strohmeier, and Ivan Martinovic. 2022. BROKENWIRE: Wireless disruption of CCS electric vehicle charging. *arXiv preprint arXiv:2202.02104* (2022).

[27] Zachary J. Lee, Tongxin Li, and Steven H. Low. 2019. ACN-data: Analysis and applications of an open EV charging dataset. In *10th ACM International Conference on Future Energy Systems (e-Energy'19)*. Association for Computing Machinery, New York, NY, 139–149.

[28] Hao Liu, Riccardo Spolaor, Federico Turrin, Riccardo Bonafede, and Mauro Conti. 2021. USB powered devices: A survey of side-channel threats and countermeasures. *High-conf. Comput.* 1, 1 (2021), 100007.

[29] Francesco Marra, Guang Ya Yang, Chresten Træholt, Esben Larsen, Claus Nygaard Rasmussen, and Shi You. 2012. Demand profile study of battery electric vehicle under different charging options. In *IEEE Power and Energy Society General Meeting*. 1–7.

[30] Andrés Molina-Markham, Prashant Shenoy, Kevin Fu, Emmanuel Cecchet, and David Irwin. 2010. Private memoirs of a smart meter. In *2nd ACM Workshop on Embedded Sensing Systems for Energy-efficiency in Building*. Association for Computing Machinery, New York, NY, 61–66.

[31] Fabian Mörchen. 2003. Time series feature extraction for data mining using DWT and DFT. https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.87.2037&rep=rep1&type=pdf.

[32] Marc Mültin. 2018. ISO 15118 as the enabler of vehicle-to-grid applications. In *International Conference of Electrical and Electronic Technologies for Automotive*. 1–6.

[33] N. Rajesh Kumar and J. Uday Kumar. 2015. A spatial mean and median filter for noise removal in digital images. *Int. J. Adv. Res. Electric., Electron. Instrum. Eng.* 4, 1 (2015), 246–253.

[34] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* 12 (2011), 2825–2830.

[35] Samuel Pelletier, Ola Jabali, Gilbert Laporte, and Marco Veneroni. 2017. Battery degradation and behaviour for electric vehicles: Review and numerical analyses of several models. *Transport. Res. Part B: Methodol.* 103 (2017), 158–187. DOI: https://doi.org/10.1016/j.trb.2017.01.020

[36] Christian Roth, Ngoc Thanh Dinh, and Doğan Kesdoğan. 2020. CrowdAbout: Using vehicles as sensors to improve map data for ITS. In *7th International Conference on Social Networks Analysis, Management and Security (SNAMS)*. 1–8.

[37] Thomas Schreiber and Andreas Schmitz. 1997. Discrimination power of measures for nonlinearity in a time series. *Phys. Rev. E* 55, 5 (May 1997), 5443–5447. DOI: https://doi.org/10.1103/PhysRevE.55.5443

[38] Mohammadhadi Shateri, Francisco Messina, Pablo Piantanida, and Fabrice Labeau. 2020. Real-time privacy-preserving data release for smart meters. *IEEE Trans. Smart Grid* 11, 6 (2020), 5174–5183.

[39] Chenxi Sun, Tongxin Li, Steven H. Low, and Victor O. K. Li. 2020. Classification of electric vehicle charging time series with selective clustering. *Electric Power Syst. Res.* 189 (Dec. 2020), 106695.

[40] Johan A. K. Suykens and Joos Vandewalle. 1999. Least squares support vector machine classifiers. *Neural Process. Lett.* 9, 3 (1999), 293–300.

[41] ThunderSky. 2007. *Instruction Manual for LFP/LCP/LMP Lithium Power Battery*. Technical Report. Thunder Sky.

[42] C. Troepfer. 2009. *SAE Electric Vehicle Conductive Charge Coupler, SAE J1772*. Technical Report. Society of Automotive Engineers.

[43] U.S.D.O.E. 2014. *EV Everywhere Grand Challenge: Road to Success*. Technical Report. U. S. Department of Energy.

[44] J. Vetter, P. Novák, M. R. Wagner, C. Veit, K.-C. Möller, J. O. Besenhard, M. Winter, M. Wohlfahrt-Mehrens, C. Vogler, and A. Hammouche. 2005. Ageing mechanisms in lithium-ion batteries. *J. Power Sourc.* 147, 1 (2005), 269–281. DOI: https://doi.org/10.1016/j.jpowsour.2005.01.006

[45] Lu Wang, Zian Qin, Tim Slangen, Pavol Bauer, and Thijs van Wijk. 2021. Grid impact of electric vehicle fast charging stations: Trends, standards, issues and mitigation measures-an overview. *IEEE Open J. Power Electron.* 2 (2021), 56–74.

[46] Ning Wang, Linhao Tang, and Huizhong Pan. 2019. A global comparison and assessment of incentive policy on electric vehicle promotion. *Sustain. Cities Societ.* 44 (2019), 597–603.

[47] Hao Wu, Grantham Kwok Hung Pang, King Lun Choy, and Hoi Yan Lam. 2017. An optimization model for electric vehicle battery charging at a battery swapping station. *IEEE Trans. Vehic. Technol.* 67, 2 (2017), 881–895.

[48] Qing Yang, Paolo Gasti, Gang Zhou, Aydin Farajidavar, and Kiran S. Balagani. 2016. On inferring browsing activity on smartphones via USB power analysis side-channel. *IEEE Trans. Inf. Forens. Secur.* 12, 5 (Dec. 2016), 1056–1066.

Received 1 July 2021; revised 28 May 2022; accepted 20 September 2022