Hindawi
Mobile Information Systems
Volume 2017, Article ID 1367064, 14 pages
https://doi.org/10.1155/2017/1367064

Research Article

Leveraging Battery Usage from Mobile Devices for Active Authentication

Jan Spooren, Davy Preuveneers, and Wouter Joosen

imec-DistriNet, Department of Computer Science, KU Leuven, Leuven, Belgium

Correspondence should be addressed to Davy Preuveneers; davy.preuveneers@cs.kuleuven.be

Received 7 September 2016; Revised 16 January 2017; Accepted 14 February 2017; Published 12 March 2017

Academic Editor: Daniele Riboni

Active authentication is the practice of continuously verifying the identity of users, based on their context, interactions with a system, and information provided by that system. In this paper, we investigate if battery charge readings from mobile devices can be used as an extra factor to improve active authentication. We make use of a large data set of battery charge readings from real users and construct two computationally inexpensive machine learning classifiers to predict if a user session is authentic: the first one only based on the battery charge at a certain time of day; the second one predicts the authenticity of the user session when a previous, recent battery charge reading is available. Our research shows that a simple two-figure battery charge value can make a useful albeit minor contribution to active authentication.

1. Introduction

Mobile devices allow users to access online information resources and services anywhere, anytime, and anyhow. A cornerstone to secure access to these resources and services is effective user authentication. Several methods are available to authenticate a user to a remote system. The most well-known username and password authentication is still widely used due to its low cost and ease of implementation. However, this authentication method exhibits some severe security threats, such as password reuse on different services, easily guessable passwords, leaking passwords, or even entire password databases leaking from badly implemented online systems. State-of-practice multifactor authentication [1] combines inherence, possession, and knowledge factors, that is, something the user *is*, *has*, and *knows*, leveraging hardware tokens, smart cards or biometric devices [2] to strengthen authentication, but such solutions are often perceived as cumbersome or too expensive to roll out to users.

In this work, we investigate to what extent information provided by mobile information systems themselves can be leveraged as additional authentication factors to address some of the above security threats.

The focus of our work is on multifactor authentication with the objective to offer user-friendly means of authentication, that is, beyond typing passwords on small or inconvenient user interfaces, using the capabilities of a mobile device. The goal is not an enhanced smart device locking feature that would protect resources on the device itself, although this would also be feasible with the techniques presented in this paper, but rather having the ability with a mobile device to securely authenticate an individual against an online web application or service that is usually protected by an identity and access management system (such as ForgeRock's OpenAM [3]). The challenges we aim to address are twofold: (1) how can we first conveniently and reliably authenticate the identity of a user and (2) how can we then continuously assess the confidence in the user's identity during the application session. Indeed, in the past decade we have observed a growing interest in *Active Authentication*, also known as *Context-aware* [4], *Continuous* [5], or *Implicit* [6] *Authentication*. These authentication systems try to use information about the user's context or the user's behavior [7, 8] within that context to assess the likelihood of an authentic user session. The proliferation of mobile devices and the resulting abundance of new sensors carried by users have fueled a renewed interest

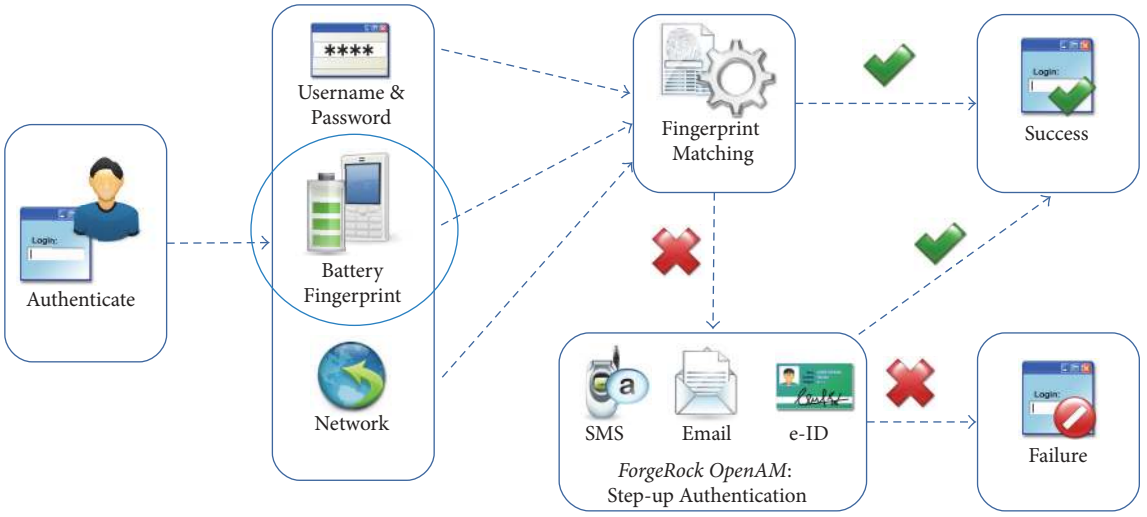

Figure 1: Integrating battery-based fingerprinting for multifactor authentication in contemporary identity and access management systems.

in the idea of being able to authenticate a user, simply by her context and interactions with the device. As depicted in Figure 1, the features provided by the mobile device are then continuously and seamlessly leveraged as additional authentication factors to validate (1) the user's identity and (2) the user's session.

One criticism of behavior-based authentication systems is that, similar to biometric authentication, the authentication material cannot be revoked. However, in the event of compromised behavioral data, privacy requirements will typically allow the user to revoke the use of "behaviometrics" at the expense of usability. An alternative solution is to split the computation using cryptographic schemes, such as secure multiparty computation [9] or homomorphic encryption [10], for privacy preserving machine learning. However, from a practical point of view, such solutions might be too resource demanding for the mobile client device and effectively jeopardize the effectiveness of using battery charge values as a behaviometric.

A common approach to realize active authentication is to employ a risk-based methodology [11, 12]: an existing authentication technology is used for initial authentication, after which the authenticated session is maintained as long as the confidence in the authenticity of that session is high. This results in potentially long lasting sessions; only when the confidence in the session drops below a certain threshold, the user is asked to reauthenticate. The choice of the threshold for reauthentication allows balancing usability with security.

To make predictions on a session's authenticity, suitable context *features* are selected, as well as matching *classifiers*. Features are measurable properties of the system and of the user's interaction with the system. Classifiers are algorithms that map the input of one or more features to a prediction into classes, such as {*authentic*, *compromised*}, typically after training the classifier with real (historical) data. Examples of features that can be used for classification are location [6, 13, 14], other devices nearby, application usage [13], telephone calls placed and received [6, 14], websites browsed and network traffic [15], stylometry [13], keystroke dynamics [16–19], ambient sound [20], and gait recognition [21, 22]. In active authentication, typically one context feature and its matching classifier are not sufficient to reliably predict the authenticity of a user session. Instead, multiple features and classifiers are considered, and their decision is fused to obtain acceptable error rates.

In this paper, we investigate if mobile device battery charge measurements can be used as a feature contributing to predictions for session authenticity. To the best of our knowledge, no one has studied the use of battery charge data as a novel information source for active authentication. Battery charge is an interesting information property, since its value is largely determined by the user's use of the mobile device and therefore constitutes a user *behavior* fingerprint. In that sense, behavioral authentication factors (or behaviometrics) are closely related to *inherence* or biometric factors. Whereas the latter are unique features that reflect the person *you are*, the former tries to distinguish a user based on something *you do*.

The advantage of using battery charge information as a behaviometric factor is that biometric factors, such as fingerprints, voice, or iris patterns, usually require additional hardware. While some mobile devices do have fingerprint scanners, this feature is not yet ubiquitously available. Additionally, fingerprint scanners are cumbersome to the end-user. They are impractical for continuous authentication, and rather privacy sensitive as a user cannot retract his fingerprints in a way he can change his password if needed. Using battery charge information as a behaviometric does not have these drawbacks. Furthermore, it is a feature which can easily be obtained: Native apps can usually query this value and web applications can obtain a mobile device's battery charge without the user's knowledge through the HTML5 draft *Battery Status API*, which (at the time of writing) is available in the Chrome and FireFox browsers. Battery charge is therefore a convenient feature to add to an active authentication system, thereby decreasing the overall error rate of such a system.

To ascertain and validate the authenticity of the user's digital identity and session, we investigate two novel binary classifiers based on battery charge measurements as a source of behavioral information:

(1) *User identity*: the first classifier is based on histograms, used to estimate the probability for a given user to detect a particular battery charge at a particular time of the day.

(2) *User session*: the second classifier is using a stored recent battery charge measurement to assess the likelihood of measuring a particular battery charge after a certain time interval.

These classifiers are both computationally and storage-wise inexpensive and scale well for large numbers of users. We also explore the use of kernel density estimators as well as other conventional machine learning algorithms which may classify more accurately at the cost of being more computationally intensive compared to our histograms. Fortunately, these techniques and algorithms can be parallelised to run on several machines if need be. This is a key advantage whenever the data and risk analysis must be carried out for multiple users in parallel on the authentication platform, rather than on the mobile device. Indeed, only the authentication platform (on the server side) is able to analyze, process, and compare the behavioral data of all its registered users and devices.

To obtain battery charge data from real users, we explored existing data sets such as Carat [23, 24] and the data set collected by the Device Analyzer Android App for large-scale mobile data collection [25]. The usefulness of Carat for our research turned out to be limited due to insufficient information about battery charge levels, as well as lacking timestamps and unique identifiers per user. We carried out our evaluation on 28 days of battery charge measurements of 645 devices from the Device Analyzer Dataset. While it was possible for some devices to tap into months of battery charge measurements, we limit the period on which we train the classification algorithms, therefore allowing real-life implementation to use a sliding window approach to collecting training data. This way, our solution can handle changing user behavior without affecting the user.

From a practical point of view, our solution only needs a few battery charge measurements to become useful. When a particular value is known at a given point in time, our algorithms can ascertain the likelihood that a certain measurement is genuine minutes later during authentication by taking into consideration the physical limitations on the maximum charging and discharging rate of a mobile phone. Once more measurements are collected, the likelihood can be further tuned to match the specific behavior of the user, both in terms of obtaining a genuine measurement at any given time of the day and when a recent prior battery charge measurement is known. Depending on the mobile phone usage, our solution typically needs to collect a few days of data before it becomes effective.

Last but not least, we also evaluate possible security threats with respect to using battery information as a behaviometric for authentication purposes. Indeed, we must avoid having a well-informed attacker being able to impersonate the user by observing his behavior, as well as even ascertain the possibility of an attacker to reliably predict and subsequently spoof the battery charge at any moment in time without any further information.

The key contributions of this work are (1) two classifiers for predicting identity and session authenticity based on battery draining and charging behavior, (2) the confirmation that battery charge measurements can contribute to active authentication systems, and (3) a practical integration into a state-of-practice identity and access management platform.

We have integrated our solution in OpenAM (https://www.forgerock.com/platform/identity-management/) 13, a contemporary identity and access management system. OpenAM offers device fingerprinting and matching capabilities using client-side and server-side JavaScript technology. As shown in our previous work [26], the built-in fingerprinting code is not well suited for mobile devices. In this work, we adapted the JavaScript code to call our service to process battery data. The additional benefit of this integration (illustrated in Figure 1) is that OpenAM and our solution can be independently scaled out.

After reviewing related work in Section 2, we describe our approach to collecting real user's data and building usable classifiers in Section 3. In Section 4, we evaluate the proposed classifiers and discuss the results of our experimental evaluation. We formulate conclusions and topics for future work in Section 5.

2. Related Work

Several studies have investigated the concept of Active Authentication, using various different sources of information, like app usage [13], stylometry [13], keystroke dynamics [16–18, 27, 28], mouse movement [27], smart phone touch screen dynamics [29–31], phone calls placed [6, 14], GPS location [6, 13, 14], ambient sound [20], and so forth. We refer to an extensive survey on behavioral biometrics [32] for a detailed comparison of accuracy rates when verifying users with different behavioral biometric approaches.

Traore et al. [27] combined keystroke dynamics and mouse movement and showed that they can be used for risk-based authentication on a web page.

To the best of our knowledge, no one has studied the use of battery charge data as a behavioral information source for active authentication. Furthermore, most studies are based on information that must be harvested by a dedicated app on the mobile device, while (similarly to [27, 33]) our work can also be used (through the HTML5 draft Battery Status API) by web services without requiring any dedicated monitoring software on the mobile device.

One can construct a more accurate classifier and improve the confidence in a particular user session by fusing the output of different features and their individual classifiers. Fridman et al. [13] studied the combination of several different behavioral context elements, selecting suitable classifiers for each of them and combining their decision outputs using Chair and Varshney's [34] *Optimal Fusion Rule*, to combine the decisions of multiple detectors [1], minimizing the overall

error probability [35]. Bailey et al. [36] used *Ensemble Based Decision Level* (EBDL) fusion to combine classifiers for keyboard, mouse, and GUI interaction features and concluded that EBDL fusion significantly outperformed each individual modality as well as feature fusion.

Preuveneers and Joosen [37] presented a contextual authentication framework built upon an existing identity and access management platform (OpenAM 11). They used IP address range, geolocation, time of access, and a number of user agent fingerprints (language, color depth, screen resolution, etc.).

In a paper by Olejnik et al. [38], battery capacity is used to fingerprint mobile devices. However, the focus of this research was not on battery usage as a behavior fingerprint, but on battery capacity as a *device fingerprint*, which is not suitable for authentication purposes (since it can easily be obtained and then replayed by an adversary to falsely impersonate a user), as illustrated in our previous work [26] on the use of device and browser fingerprints [39] for authentication. We also believe that web browser manufacturers can easily modify the estimated device capacity to be less unique without affecting usefulness of the Battery Status API, while this is much less likely for the battery charge percentage, which we use in this work.

Our work goes beyond the state of the art by researching the applicability of battery charge information as a (weak) behaviometric, both as a means for authenticating the user's identity and as a continuous assessment of the confidence of a user's identity during the application session. The advantage of our proposal is that relevant information is readily available and accessible, even within a browser context by tapping into the HTML 5 Battery Status API. Compared to the related work, this behaviometric does not require any explicit user interaction, and is therefore much more convenient to leverage.

3. Methodology

The goal of this work is to investigate the usefulness of battery charge information for active authentication. A typical scenario might be that of a user who consults a website using her mobile phone (also see Figure 1). As part of the authentication mechanism, the website can request the current battery charge level from the mobile device, using the HTML5 draft Battery Status API. This information (merged with several other authentication elements) can then be used by the website to verify the authenticity of the user.

The approach to establishing the user's identity can be twofold: Firstly, the website can use a classifier that uses the battery charge at a certain time of the day as a feature to predict if this is the expected user. Secondly, if another battery charge reading was recently collected by the same website, then a different classifier can be built, predicting the likelihood of the user session to be authentic. Clearly, a mobile device that was reporting a battery charge of 20% is very unlikely to be reporting 95% battery charge only 20 minutes later. By keeping track of how battery charges are distributed throughout the day, we can ascertain not only whether a particular battery charge is likely or not for a particular user or device, but also if that measurement is probable or even technically feasible given a previous battery charge measurement for that same device.

We call the first classifier, based on the battery charge at a certain time of the day the *User Verification Classifier*. It is presented in Section 3.2. The second classifier, which uses a battery charge and another, recently collected battery charge from the same device, t minutes earlier, is called the *User Session Classifier* and is presented in Section 3.3.

The overall methodology of our active authentication solution is as follows. The enrollment of new users starts with an on-boarding phase in which the user registers his mobile device to the authentication platform. In our solution, we build on top of ForgeRock's OpenAM authentication platform (see Figure 1). This way, both users and mobile devices have a unique identity. Using its push authentication mechanism, the OpenAM platform can interact with the mobile device in the background to collect any behavioral information, including battery charge information. An alternative is to have a mobile application continuously collect and forward behavioral data to the OpenAM authentication platform.

By leveraging statistical features in the collected data and machine learning techniques, we build a profile for each user and his mobile device that characterizes the former's interaction behavior by tapping into the battery consumption of the mobile device. This profile is continuously updated, and significant deviations from this profile may indicate that the device (or the person using the device) is not genuine.

3.1. Battery Charge Data Acquisition. To study the feasibility of using device battery data for authentication purposes, actual battery data is needed from mobile devices used by real users. Several methods are available to collect this battery data: it can be obtained by a native application on the mobile or by adding JavaScript code to web pages of an existing, frequently used web service. Using the HTML5 draft Battery Status API, a device's battery charge can be queried and recorded for all users visiting the web pages. Contrary to a native application on the mobile, the web page will not be able continuously collect battery information at runtime, as it only operates from within a browser context whenever the user visits the instrumented web page. While the training phase needs sufficient data to build an accurate model to avoid bias in the measurements as well as a lack of information in particular circumstances, we only need occasional samples during the testing phase to analyze the confidence in the user's identity, making the HTML5 Battery Status API a perfect candidate for a practical realization in a concrete online application.

For training purposes, a dedicated device monitoring app can be installed, which will record the device's battery charge with regular intervals. We opted for the latter solution and made use of the Device Analyzer data set, which was collected by the University of Cambridge [25]. It contains over 100 billion records of Android smartphone and tablet usage from over 20,000 devices across the globe (http://deviceanalyzer.cl.cam.ac.uk), collected from volunteers who installed an Android app, which gives them insights in their own usage

data in return. We filtered a subset of this data, based on the following criteria:

(i) A device has at least 28 consecutive days of battery charge measurements.

(ii) These measurements are collected in intervals of less than 15 minutes.

In total, 645 mobile devices were retained, of which we used the last 28 days of battery data recorded.

3.2. User Verification Classifier: Histogram-Based Classification on Battery Data. In this first classifier, we will try to use solely the reported battery charge at a particular time of the day to predict if a user session is authentic.

For each user, we construct a classifier for classification of the battery charge C at time t into two classes: the first class (H_1) is trained on the battery charge measurements and measurement times for the valid user and the second class (H_0) is trained on the charge measurements and measurement times of all users.

As a simple binary classifier, we determine the maximum likelihood of finding charge C at time t (as a minute-of-the-day offset) for the valid user or for the average user:

$$H^* = H_i \mid \arg\max_{i\in\{0,1\}} P\left(C_t \mid H_i\right). \tag{1}$$

Differently put, we estimate the user session of user u to be valid, when the probability $P_u(C_t)$ for this user of finding battery charge C at time t is higher than the probability $P(C_t)$ of finding battery charge C at time t for the average user.

To estimate these probabilities, we create a battery histogram matrix $B^u_{i,j}$ for each user u. Each measurement $m = (m_C, m_t)$ in the collection of measurements M_u for user u is tallied into a 101×1440 matrix, providing 101 charge slots (for charges recorded from $0\%, \ldots, 100\%$ in one percent increments in one dimension and 1440 minute per day slots for the other dimension):

$$B^u_{i,j} = \sum_{m\in M_u} \delta_{i,m_C}\delta_{j,m_t}, \tag{2}$$

where m_t is the time of the measurement represented in minutes since midnight and $\delta_{i,j}$ is the Kronecker delta.

Then, we normalize each of the columns as follows:

$$\overline{B}^u_{i,j} = \frac{B^u_{i,j}}{\sum_k B^u_{k,j}}. \tag{3}$$

For each column (corresponding to minute-of-the-day j), the sum of the elements for all possible charges now equals $\sum_i \overline{B}^u_{i,j} = 1$ and the matrix element $\overline{B}^u_{i,j}$ of the normalized battery histogram matrix therefore contains an estimation of the probability of finding a charge i, at minute-of-the-day j for user u, based on past measurements.

We can write the estimated probability of finding a charge C at time t as follows:

$$P_u\left(C \mid t\right) = \overline{B}^u_{C,t} \tag{4}$$

A graphical representation of the probability densities provided by the normalized battery histogram is shown in Figure 2(a). The red cells indicate high probability and the blue and purple cells indicate low probability for finding a particular battery charge at a particular time slot.

Each of the columns in the grids of Figure 2 represents a histogram for one particular one minute time slot.

To estimate the probability for the invalid class, we create a normalized global battery histogram for all of the users in the system:

$$\overline{B}_{i,j} = \frac{\sum_u \sum_{m\in M_u} \delta_{i,m_C}\delta_{j,m_t}}{\sum_u \sum_k B^u_{k,j}}. \tag{5}$$

The normalized global battery histogram for our measurement data is shown in Figure 3.

3.3. User Session Classifier: Histogram-Based Classification on Battery Data with Known Prior Battery Charges. Similar to the first classifier, this second classifier too uses histograms to calculate the probability of a user's mobile device having a specific battery charge percentage at a certain time. It differs from the one proposed in Section 3.2 in that it establishes the probability of measuring a specific battery charge reading given *another*, prior and recent battery charge measurement.

Let $P_u(C \mid C'_t)$ be the probability of measuring a battery charge C for user u, t minutes after a previous battery charge C' was recorded. This probability is interesting, because it can be used to detect hijacked user sessions: Battery charge is a continuously evolving property, and both the physical properties of the device and the typical usage patterns of the user dictate boundaries within which the battery charge can evolve during a certain time span. Battery charge readings outside of these boundaries can be an indication of a compromised user session. We can estimate the probability $P_u(C \mid C'_t)$ by using past battery charge measurements.

Let U be the collection of users and M_u the collection of battery measurement samples for user $u \in U$. Each $m \in M_u$ is a (battery charge, time) tuple: $m = (m_C, m_t)$. For each user u, timespan $t \in \{5, 10, 15, 20, \ldots, 140\}$, and initial charge C', we can now create a

$$D^{u,C'}_{C,t} = \sum_{m\in M_u} \sum_{m'\in M_u} \delta_{C',m'_C}\delta_{t,(m_t-m'_t)}\delta_{C,m_C} \tag{6}$$

and normalize $D^{u,C'}_{C,t}$ as follows:

$$\overline{D}^{u,C'}_{C,t} = \frac{D^{u,C'}_{C,t}}{\sum_{C'\in\{0,\ldots,100\}} D^{u,C'}_{C',t}} \tag{7}$$

such that

$$\sum_{C\in\{0,\ldots,100\}} \overline{D}^{u,C'}_{C,t} = 1, \qquad \forall u \in U,\ C' \in \{0,\ldots,100\},\ t \in \{0,5,10,\ldots,140\}. \tag{8}$$

$\overline{D}^{u,C'}_{C,t}$ can now be regarded as an estimation based on past measurements of the probability $P_u(C \mid C'_t)$ to detect a battery

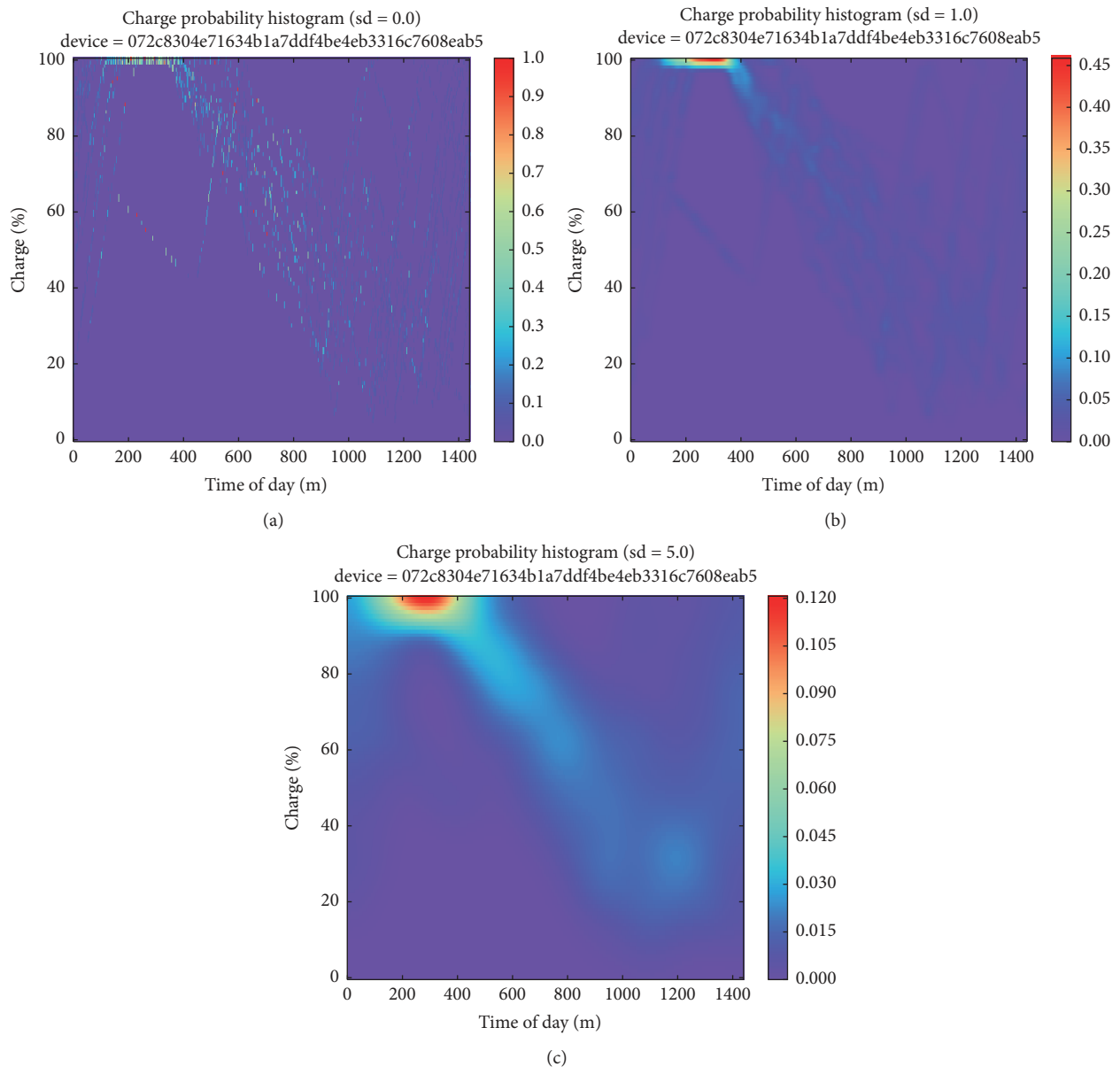

Figure 2: Graphical representation of normalized battery histograms, (a) without Gaussian smoothing applied and (b and c) with Gaussian smoothing applied, with, respectively, $\sigma = 1.0$ and $\sigma = 5.0$.

charge C and a timespan t after having detected an earlier battery charge C'.

The approach is illustrated in Figure 4, which shows the probability distribution $\overline{D}^{u,70}_{C,60}$ for user u, for measuring a particular battery charge C, 60 minutes after a battery charge of 70% was measured. In this case, the figure indicates that the highest probability is at 66%. Clearly, for this user, the average battery discharge is around 4% per hour. However, we can also see past records of the battery charge having dropped down to 48%, presumably when the user had been using the mobile device heavily. Towards the other side of the charge axis, we can see recorded charges up to 94%, indicating that the maximum charging speed for this device is 24%/h.

Figure 5 shows the probability distribution $\overline{D}^{u,70}_{C,t}$ for the same user u, for measuring a particular battery charge C, t minutes after a battery charge of 70% was measured.

Using the calculated probability estimations, we can now create a binary classifier, by choosing a threshold probability θ. The classifier will predict a valid user session, when $P_u(C \mid C'_t) \approx \overline{D}^{u,C'}_{C,t} \geq \theta$ and an invalid user session when $P_u(C \mid C'_t) \approx \overline{D}^{u,C'}_{C,t} < \theta$.

3.4. Threat Model and Attack Vectors. The use of battery charge measurements as an additional source of information for multifactor authentication assumes that it is not trivial for

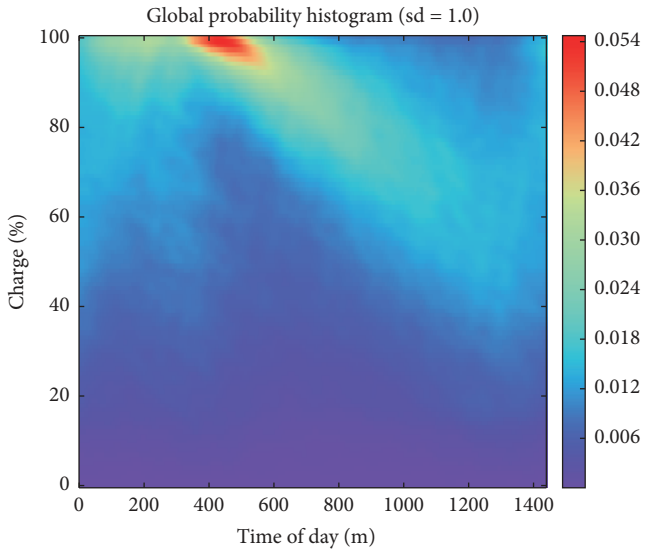

Figure 3: Graphical representation of the normalized global battery histogram (Gaussian smoothing $\sigma = 1$ applied).

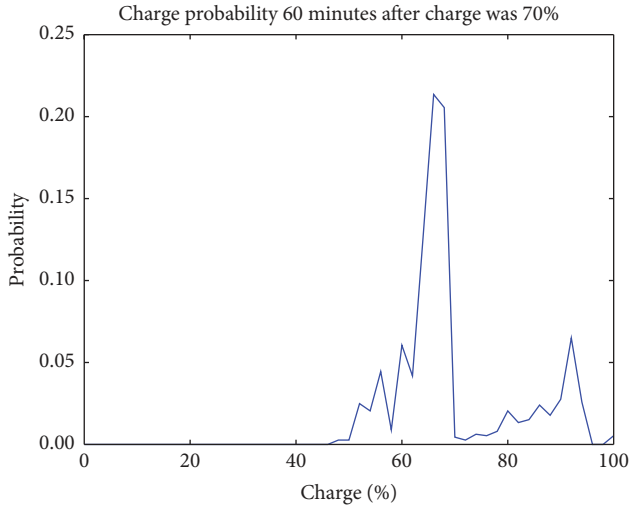

Figure 4: Battery charge probability distribution for one of the devices, 60 minutes after an initial battery charge of 70% was seen.

a malicious adversary to guess and spoof the battery charge for a particular device on a given time of the day.

In this work, we assume that the adversary is able to collect data from different devices to compute likely values throughout the day, possibly by relying on data sets we have also used for our experiments. This means he is able to compute the probability distribution of battery charges for any minute of the day (as in Figure 3) and can try to spoof the authentication system by using the most likely battery charge value. To evaluate the effectiveness of the proposed scheme, we evaluate the impact of two different types of attacks:

(i) *Zero-effort attack*: the adversary is simply another subject in the database that acts as a casual impostor

(ii) *Nonzero effort attack*: the adversary actively masquerades as someone else by spoofing the battery charge of the claimed identity

In the zero-effort attack, we use the data of the other subjects as negative examples for a given user to get insights into the probability of accidentally authenticating on another device. For the nonzero effort attack, we assume the adversary implements a nonpersonalized attack vector that requires minimal effort to spoof the battery measure. In the latter case, we distinguish between two scenarios: (1) the adversary has no information from the target's device, but just a nonpersonal probability distribution of battery charges, and (2) the adversary can exploit previous battery charge information.

4. Evaluation

An important tool for assessing the value of a feature and a corresponding classifier for active authentication is metrics that describe the effectiveness of the classifier:

(i) *False acceptance rate (FAR)*: the ratio of the number of classifications where a nonauthentic user is falsely accepted as authentic by the classifier over the total number of classifications performed.

(ii) *False rejection rate (FRR)*: the ratio of the number of classifications where an authentic user is falsely rejected by the classifier over the total number of classifications performed.

(iii) *Equal error rate (ERR)*: in many cases, a classifier's FAR can be decreased by modifying the classifier to be more selective, at the cost of an increasing FRR and vice versa. This typically allows balancing security (requiring a low FAR) with usability (requiring a low FRR). The point where the classifier is tuned to have a FAR which is equal in value to the FRR is called the Equal Error Rate.

4.1. Evaluation of the User Verification Classifier

Calculating FAR and FRR. We used the described classification method to make predictions on the validity of a user session, purely based on *battery charge* and *time of day*. For calculating the *False Rejection Rate*, we used a 4-fold cross-validation strategy on a per-week basis: One week was excluded from the training data and set aside to be used for testing the built model, therefore ensuring the training data is not used to validate the built model. We then repeated the test three more times, each time excluding another week from the training data and using it to test the model.

Since the (m_C, m_t) samples are time series, excluding random samples in a typical stratified k-fold cross-validation strategy would lead to overly optimistic results, since those testing samples can easily be fitted in the corresponding gaps in the training data. This is why we chose to exclude large contiguous blocks from the training data, opting for 4-fold cross-validation (resulting in blocks of one week), rather than the standard 10-fold cross-validation.

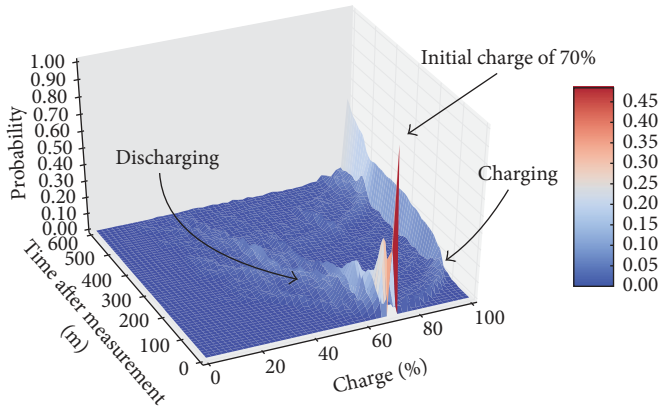

Figure 5: Battery charge probability distribution for one of the tracked devices, in the time after an initial battery charge of 70% was seen.

Table 1: Results of battery histogram classification.

	Mean	stdev
FAR	0.044	0.025
FRR	0.938	0.039

To calculate the *False Acceptance Rate*, all measurements available from all other users were fed to a user's classifier, to verify if they were falsely accepted as authentic. While the user's classifier does include measurements from all other users, due to the use of the of the normalized global battery histogram (as shown in (5)), we believe that this will have little effect on the accuracy of the results, since the global battery histogram is an average over 645 different users.

The results are provided in Table 1.

The results listed in Table 1 look far from useful: The False Acceptance Rate is excellent, but clearly the False Rejection Rate is abominable. Looking closer at Figure 2(a) reveals that since the matrices we are using are very fine-grained and since only 21 days of training data are used, the matrices do not really indicate a probability; rather they merely contain a past record of observed samples. To create useful probability estimations, we can use Kernel Density Estimation with a suitable bandwidth to estimate underlying probabilities, as illustrated in Figure 6, where a Gaussian kernel was used and a bandwidth of 5.0.

However, since the collected samples are already discretized in percentage charge and minute of the day, using a Gaussian smoothing algorithm on the battery histograms will achieve a very similar result, at a performance cost which is orders of magnitude lower. Instead of a Kernel Density Estimation bandwidth, we can fine tune the classifier using the Gaussian blurring standard deviation: for certain applications, one might be interested in decreasing the FRR (thereby increasing the usability), at the cost of an increasing FAR (thereby sacrificing security) or vice versa. Applying Gaussian smoothing with different standard deviations for the Gaussian kernel to the battery histograms can achieve exactly this, where we use a standard deviation in the time dimension, which is 14.4 times larger than the standard deviation

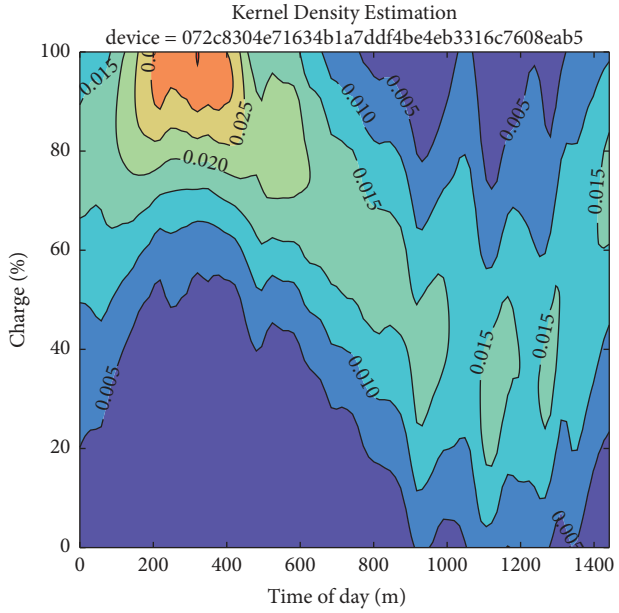

Figure 6: Using Kernel Density Estimation to build a charge/time probability model for the device that was shown in Figure 2.

used in the charge dimension, to compensate for the fact that the probability matrices are 14.4 times more fine-grained in the time dimension than the charge dimension. For simplicity of notation, in the remainder of this work, we refer to the standard deviation of the charge dimension; the standard deviation in the time dimension should be multiplied by 14.4. A graphical representation of the effect of the Gaussian filter on the battery charge histograms can be observed in Figures 2(b) and 2(c), where Gaussian kernel standard deviations of 1.0 and 5.0 were used. The results of this approach on the classification errors are listed in Table 2 and plotted in Figure 7.

Using linear interpolation on the FAR/FRR curve between standard deviation 1.4 and 3 yields an Equal Error Rate of 0.413.

Table 2: Results of battery histogram classification with a given Gaussian Smoothing filter.

Smoothing level	FFR		FAR	
	Mean	stdev	Mean	stdev
10	0.383	0.111	0.475	0.064
8.0	0.370	0.109	0.466	0.069
6.0	0.366	0.105	0.458	0.062
5.0	0.369	0.103	0.448	0.066
4.0	0.376	0.102	0.439	0.063
3.5	0.382	0.101	0.435	0.061
3.0	0.389	0.100	0.427	0.057
2.5	0.400	0.099	0.420	0.059
2.0	0.415	0.099	0.411	0.056
1.75	0.425	0.099	0.405	0.053
1.5	0.438	0.100	0.396	0.054
1.4	0.444	0.100	0.393	0.050
1.2	0.458	0.100	0.384	0.049
1.0	0.477	0.101	0.374	0.050
0.8	0.501	0.102	0.360	0.050

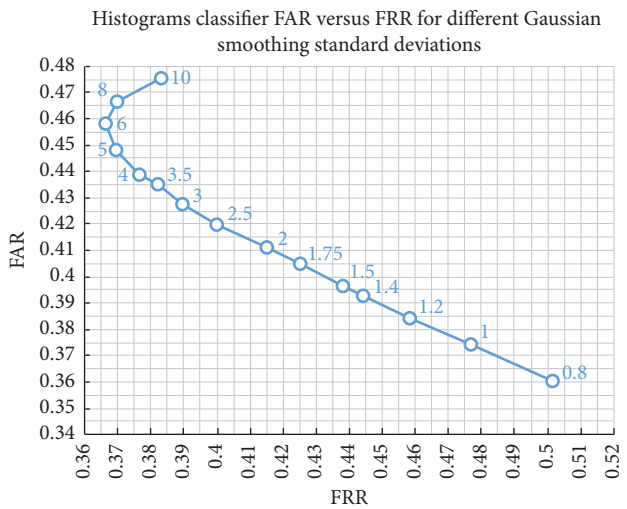

Figure 7: False Acceptance versus False Rejection Rates for different smoothing levels on the user models (indicated in the data labels). FAR was calculated by feeding 5 random (time, charge) samples from every known user to the classifier.

Attacker Model. Earlier in this section, we calculated the FAR by feeding the (charge, time) measurements from all other users into the proposed first classifier and recording the number of times the classifier wrongly predicted that this could be the user we are evaluating. The results of this looked modest but still useful with an EER of 41.26%, taking into account the fact that it was achieved only using a 2-figure battery charge measurement. However, this approach did not assume an attacker who deliberately tries to circumvent the active authentication system. A clever attacker will investigate which is the most likely battery charge to present to the authentication system at any given time. We can assume that the attacker will not have access to the detailed per-user

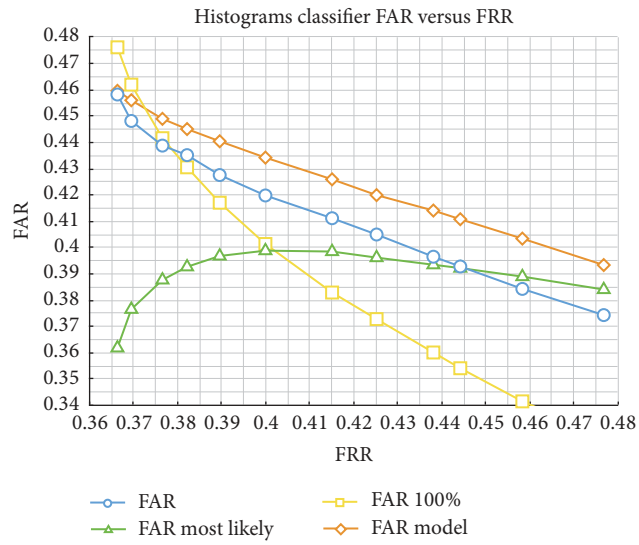

Figure 8: False Acceptance versus False Rejection Rates for different smoothing levels on the user models. FAR was calculated using four different attack models: (i) using 5 random (time, charge) samples from every known user (blue line); (ii) using the global probability estimation to find the most likely charge value at each time of the day (green line); (iii) the adversary always reporting 100% battery charge (yellow line); and (iv) using 2500 random samples of other users having the same mobile device model (orange line).

charge probability histograms; however the Normalized Global Battery Charge Histogram $\overline{B}$ can be regarded as public knowledge. It could be used by an attacker to determine the most likely battery charge for the average user at any given time of the day. Another likely value to be used by an attacker would be the 100% charge value, since this is especially in the morning a very likely battery charge.

Repeating our analysis where instead of feeding measurements from different users into the classifier, we present the most likely battery charge, and repeating one more time, presenting invariably the 100% battery charge results in different FAR/FRR values, as shown in Figure 8.

We observe that in the classifier range with FRR between 0.38 and 0.44 (smoothing standard deviations between 1.4 and 3.5), these targeted attacks on the classifier do not perform better than a random sample choice. Our hypothesis is that since the classifier is based on the difference between the user's past charge history and the average user's past charge history, it works particularly well for distinguishing average charge data from the target user's charge data.

Finally, one last attacker model was interesting to consider: one of the core assumptions of this work is that the battery charge constitutes both a device fingerprint and a user behavior fingerprint. To investigate how large the impact of the device was on the classifier, we calculated the False Acceptance Rate, by picking 2500 random samples from other users, who used an identical model of mobile device (in this FAR calculation, since we compare to other users with the same device model, we dropped the data from devices models that were used only by a single user; therefore, the data

Table 3: Mean and standard deviation of metrics for different classification algorithms in a zero-effort attack scenario.

Algorithm	Accuracy		Precision		Recall		*F*1		FAR		FRR	
	Mean	stdev	Mean	stdev	Mean	stdev	Mean	stdev	Mean	stdev	Mean	stdev
LR	0.569	0.079	0.559	0.073	0.591	0.159	0.570	0.107	0.453	0.087	0.408	0.159
DT	0.616	0.074	0.601	0.064	0.688	0.159	0.634	0.095	0.455	0.121	0.311	0.159
RF	0.625	0.075	0.610	0.064	0.684	0.157	0.638	0.097	0.434	0.103	0.315	0.157
GBT	0.627	0.079	0.616	0.068	0.661	0.150	0.633	0.101	0.405	0.085	0.338	0.150
NB	0.523	0.030	0.522	0.030	0.523	0.052	0.522	0.040	0.477	0.025	0.476	0.052
MLP	0.556	0.060	0.556	0.056	0.587	0.261	0.538	0.150	0.474	0.237	0.412	0.261
KNN	0.611	0.069	0.598	0.056	0.657	0.117	0.624	0.081	0.434	0.053	0.342	0.117

of 533 devices (of the total of 645 devices) was used for this FAR calculation). As can be observed in the orange data plot of Figure 8, the FAR rate becomes slightly higher, with an average increase of 0.013, indicating that although the classifier appears to be measuring mostly a user behavior fingerprint there is indeed a very small effect of the device fingerprint present in the classifier.

4.2. Comparison with Conventional ML Classification Algorithms. In this section, we investigate the feasibility of conventional machine learning algorithms to classify battery charges as either genuine or not. For each of the 645 devices, we create a separate model that we train on genuine data of one device and attacker data. For the latter, we implement the two attack vectors, that is, the zero-effort attack by relying on data from other subjects and the nonzero effort attack by attacking with likely battery values.

In Table 3 we list the aggregated results of the following classification algorithms, that is, logistic regression (LR), decision trees (DT), random forest (RF), gradient boosted trees (GBT), naive Bayes (NB), multilayer perceptron (MLP), and k-nearest neighbors (KNN). The data sets that have been used for testing and training are based on 7200 battery charge measurements of a genuine device and 7200 battery charge measurements of an attacker (random samples of other users). For the training data we take 70% of the samples in the dataset and 30% for the test set. The split is not random but in time, which means that the test samples have been collected after the training samples. The motivation for taking equal amount of genuine device and attacker samples is that imbalanced training and test datasets would severely affect the classification accuracy of some machine learning algorithms (e.g., KNN would favor those classes with many more samples). For each of the 645 devices, we constructed the corresponding datasets and computed the *Accuracy, Precision, Recall, F1, False Acceptance Rate,* and *False Rejection Rate* classification metrics. Table 3 provides an overview of the mean and standard deviation of these metrics, showing that the gradient boosted trees classification algorithm gets the best results in terms of accuracy. The FAR is around 0.405 which means it is about 10% better compared to random choice where 50% of test samples would be falsely accepted.

In Table 4 we show the results of a nonzero free effort attack where the adversary tries to spoof the battery charge by either (1) selecting the most likely battery charge value based on previously collected information from many users or (2) simply using a full battery charge level. Based on the previous histograms one could conclude that using a 100% battery charge level would be a good guess, but as the table shows several classifiers can reject these false attempts.

Table 4: Mean and standard deviation of metrics for different classification algorithms in two nonzero effort attack scenarios: (1) the most likely battery charge and (2) a 100% full battery charge.

Algorithm	FAR (most likely)		FAR (100%)	
	Mean	stdev	Mean	stdev
LR	0.406	0.333	0.488	0.484
DT	0.462	0.269	0.368	0.304
RF	0.440	0.274	0.351	0.293
GBT	0.406	0.244	0.338	0.282
NB	0.482	0.054	0.465	0.182
MLP	0.472	0.285	0.465	0.364
KNN	0.471	0.207	0.397	0.241

Also worth noting is that there is a slight difference in the standard deviation of the above metrics. In case of a large standard deviation, it means that some targets are easier to spoof than others. As such, one could use these estimates as a personal risk indicator whether a battery charge is a good parameter to use for multifactor authentication on an individual basis.

From a performance point of view, some of the above conventional machine learning algorithms might classify better, but at a significant cost of performance and memory consumption (at least an order of magnitude higher compared to our histogram-based approach) which would make these techniques less feasible for implementation and deployment on an identity and access management platform that must handle thousands of users concurrently.

4.3. Evaluation of the User Session Classifier

Calculating FAR and FRR. The second classifier was evaluated by creating probability histograms, per user, per amount of time passed since a previously recorded battery charge (with a resolution of 5 min). The histograms record the probability of recording a battery charge C, given an earlier battery charge reading C', received t minutes earlier.

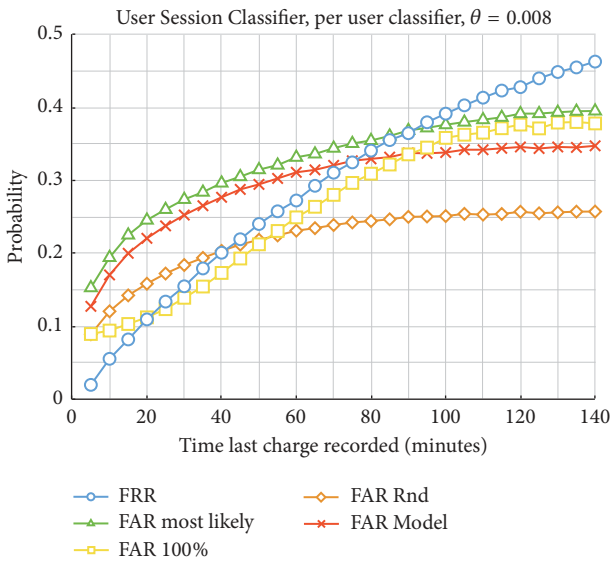

Figure 9: FAR and FRR rates for the User Session, for different elapsed time periods since last known battery charge. Chosen threshold $\theta = 0.008$.

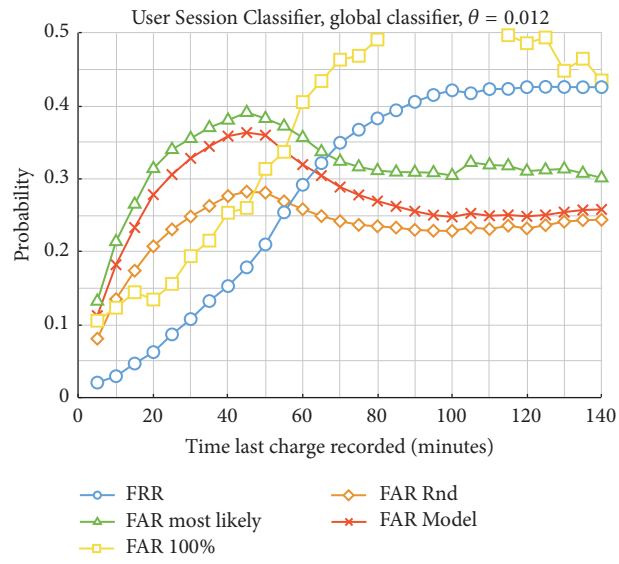

Figure 10: FAR and FRR rates for User Session Classifier, when using the average probability histograms, for different elapsed time periods since last known battery charge. Chosen threshold $\theta = 0.012$.

To calculate the *False Rejection Rate*, we again use a 4-fold cross-validation approach per measurement day, leaving out a week of data when training the model and then making predictions for the week left out of the training data, tallying the false predictions for valid user data and then repeating 3 more times with different week folds and averaging the obtained FRR.

To calculate the *False Acceptance Rate*, we adopted the attacker model introduced in Section 4.1. The FAR is again calculated in four different ways:

(i) Using random samples.

(ii) Using the most likely battery charge value for the time of the day as predicted by the Global Battery Charge Histogram $\overline{B}$.

(iii) Using consistently 100% battery charge.

(iv) Using battery charge measurements taken at the same time of day by a device identical to that of the user we try to impersonate.

The FAR and FRR rates were calculated for different time periods since the user was last seen. The results are shown in Figure 9. Similarly to the fact that σ can be used in the User Verification Classifier to tune the FAR/FRR ratio, the User Session Classifier FAR/FRR ratio can be fine-tuned by selecting different threshold values θ, allowing usability to be balanced by security, or allowing better fine tuning in an Active Authentication's fusion algorithm that combines different classifiers.

As expected, the error rates gradually increase as time goes by. Past battery measurements which are 60 minutes stale can be used for predicting session authenticity with a FRR of 0.273 and a worst-case FAR (the attacker providing the most likely charge for the time of the day) of 0.332, which is significantly better than a random guess and therefore suitable as a component for active authentication. A past battery charge measurement that is 120 minutes old will yield a FRR of 0.428 and a worst-case FAR of 0.392, which is at the limit of its usefulness for active authentication.

To obtain these results, we have collected the training material by monitoring the participants' mobile devices battery charge for 28 days at regular time intervals (at least every 15 minutes). This is a scenario that cannot be used, for instance, by a website where users are not continuously connected. In such an application, training the classifier to learn the charge and discharge behavior for each individual user may not be realistic. Therefore, we investigated how well the classification would work, when using a single classifier, trained on all user data, instead of a unique classifier per user. The results are shown in Figure 10.

The error rates are slightly higher than when training for individual users, but still useful: With a past battery measurement that is 60 minutes old, the FRR is 0.292 and the worst-case FAR (when the attacker reports 100% battery charge) is 0.406. Measurements older than 80 minutes have a FAR above 0.50 and can be considered beyond usefulness for active authentication.

Since no individual user training is required; this option is realistically usable for active authentication, for instance, for websites to detect session hijacking (sessions can be hijacked in several ways, e.g., by predictable session tokens, cross-site scripting attacks, and malicious JavaScript code). As long as the user requests web pages with intervals no longer than 60 minutes, a compromised session could be detected. Many web sites have a policy to expire sessions after a certain timeout period, which is often less than 60 minutes.

Interestingly, for both versions of the User Session Classifier, the FAR when calculated with charge samples from identical devices follows closely but is slightly lower than the

FAR calculated with the most likely battery value for the time of the day. This might be explained by the fact that the most likely battery charge is calculated over many more devices and therefore represents a better probability estimation than the samples from identical devices only.

4.4. Discussion. The concept of active authentication is based on fusing different sensor inputs together, thereby reducing the total error rate to acceptable levels, both from a usability point of view (implying a low FRR) and a security point of view (requiring a low FAR). In this paper, we have studied if battery charge can be used as one of these sensors. Clearly, the $\log_2(100) = 6.6$ bits of entropy provided by a 2-figure battery charge reading can impossibly uniquely identify a user; however, combined with other inputs, battery charge may be a useful component of an active authentication system, especially for building a confidence score for estimating session authenticity.

The first proposed classifier showed an *Equal Error Rate* of 41.3%. This is 8.7% better than a random guess and implies it could make a small contribution to an active authentication system. The fact that the False Acceptance and False Rejection Rates can be tuned is an additional interesting property.

The second classifier is trained on past data, creating a series of battery charge probability histograms. These histograms present the probability of finding a particular charge a certain time after a certain earlier charge was observed. This is a very suitable technique to detect anomalies introduced for instance by user impersonation or session hijacking where the adversary does not know the current battery charge of the user. Stored in binary format, 240,000 bytes would be sufficient to store this data (assuming 1 byte per charge bucket, 100×100 charge buckets per histogram, for 24 time periods). A classification requires a simple table lookup for the three dimensions (C, C', t) and comparison to a predetermined threshold θ. This implies that this system is practically implementable in authentication systems, in terms of both performance and data storage required.

Building the histogram tables with past measurements is straightforward: Although (6) suggests a $O(n^2)$ effort with the size of the measurement data, in fact, for each examined time period (in our case 24), the collection of measurements is iterated once, making it an $O(n)$ effort. It may not be practical, however, to build per-user histogram tables, since many measurements are required to build accurate histogram tables. However, our analysis in Section 4.3 showed that the classifier still performs well when histogram tables are constructed for classes of users. Therefore, these tables need not be constructed by the service providing active authentication services but can be harvested from independent research.

5. Conclusions

In this work, we investigated the use of battery information from mobile devices for multifactor authentication with the objective to offer user-friendly means of continuous or active authentication against an online service or application. Battery charge is an interesting mobile information property, as its value is largely determined by the user's use of the mobile device and therefore constitutes a behavioral fingerprint that characterizes the user in an implicit way. Furthermore, harvesting such battery charge data on web pages is simple through the HTML5 draft Battery Status API.

While other behaviometrics have been proposed in the literature for authentication purposes, the novelty of this work is the use of battery charge and discharge behavior as a new metric. The objective was to quantify the added value of battery information on its own and to evaluate the resilience of this behaviometric against zero-effort and nonzero effort attack vectors.

We proposed a first binary classifier for determining a user's authenticity, solely based on battery charge and time of day. This classifier characterizes the likelihood of a given battery charge for that particular user (or device). With an *Equal Error Rate* of 41.3% (compared to 50% for random guessing whether the user is authentic or not), we believe the classifier can bring a small contribution to active authentication. Indeed, as a two-digit battery charge measurement does not carry much information, in practice it would be combined with other parameters to further strengthen continuous multifactor authentication. Compared to conventional machine learning algorithms, our classification method performs better or has comparable results but at a computationally lower cost that makes the proposed technique more feasible for deployment on a large scale where the behavior of multiple users must be analyzed concurrently. An additional advantage is that the classifier's False Rejection Rate and False Acceptance Rate can be tuned, to balance security with usability or to meet the requirements of the decision fusion system typically implemented in an active authentication system.

We also proposed a complementary second binary classifier, used to determine session authenticity when a recent past battery measurement is available. We investigated the evolution of the *False Acceptance Rate* and *False Rejection Rate* with the age of the previous measurement and concluded that previous battery measurements of up to 2 hours old can contribute to active authentication. We also evaluated the classifier when trained on all user data, eliminating the need to train classifiers for each individual user, concluding that measurements of up to 1 hour old can still contribute to active authentication.

The complementary nature of these binary classifiers means that they each are able to find different types of anomalies in battery charge measurements depending on the data that is available on the subject. While both can detect spoofing attacks in different ways, the second technique performs better when an attacker is not able to directly collect battery charge measurements from the mobile device of the targeted subject and when the authentication system has recently collected genuine measurements.

In future work we will combine battery charge with additional authentication features to further validate the feasibility of active authentication on other battery-powered devices.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Disclosure

Please note that the University of Cambridge Computer Laboratory does not bear any responsibility for this analysis or the interpretation of the Device Analyzer Dataset or data therein.

Acknowledgments

This research is partially funded by the Research Fund KU Leuven. The authors wish to thank the Device Analyzer team of the University of Cambridge and Alastair Beresford in particular for making the Device Analyzer Dataset available.

References

[1] D. Dasgupta, A. Roy, and A. Nag, "Toward the design of adaptive selection strategies for multi-factor authentication," *Computers & Security*, vol. 63, pp. 85–116, 2016.

[2] C. Militello, V. Conti, F. Sorbello, and S. Vitabile, "A fast fusion technique for finger-print and iris spatial descriptors in multimodal biometric systems," *Computer Systems Science and Engineering*, vol. 29, no. 3, pp. 205–217, 2014.

[3] T. Heyman, D. Preuveneers, and W. Joosen, "Scalability analysis of the OpenAM access control system with the universal scalability law," in *Proceedings of the 2nd International Conference on Future Internet of Things and Cloud (FiCloud '14)*, pp. 505–512, Barcelona, Spain, August 2014.

[4] E. Hayashi, S. Das, S. Amini, J. Hong, and I. Oakley, "CASA: context-aware scalable authentication," in *Proceedings of the 9th Symposium on Usable Privacy and Security (SOUPS '13)*, pp. 3:1–3:10, ACM, Newcastle, UK, July 2013.

[5] S. J. Shepherd, "Continuous authentication by analysis of keyboard typing characteristics," in *Proceedings of the European Convention on Security and Detection*, pp. 111–114, May 1995.

[6] E. Shi, Y. Niu, M. Jakobsson, and R. Chow, "Implicit authentication through learning user behavior," in *Proceedings of the 13th International Conference on Information Security (ISC '10)*, pp. 99–113, Springer, Berlin, Germany, 2011, http://dl.acm.org/citation.cfm?id=1949317.1949329.

[7] R. Crossler, A. Johnston, P. Lowry, Q. Hu, M. Warkentin, and R. Baskerville, "Future directions for behavioral information security research," *Computers and Security*, vol. 32, pp. 90–101, 2013.

[8] H. G. Kayacik, M. Just, L. Baillie, D. Aspinall, and N. Micallef, "Data driven authentication: on the effectiveness of user behaviour modelling with mobile device sensors," https://arxiv.org/abs/1410.7743.

[9] Y. Lindell and B. Pinkas, "Secure multiparty computation for privacy-preserving data mining," IACR Cryptology ePrint Archive 2008/197, 2008, http://eprint.iacr.org/2008/197.

[10] D. Mittal, D. Kaur, and A. Aggarwal, "Secure data mining in cloud using homomorphic encryption," in *Proceedings of the IEEE International Conference on Cloud Computing in Emerging Markets (CCEM '14)*, pp. 1–7, IEEE, Bangalore, India, October 2014.

[11] M. Jakobsson, E. Shi, P. Golle, and R. Chow, "Implicit authentication for mobile devices," in *Proceedings of the 4th USENIX Conference on Hot Topics in Security (HotSec '09)*, p. 9, USENIX Association, Berkeley, Calif, USA, 2009, http://dl.acm.org/citation.cfm?id=1855628.1855637.

[12] O. Riva, C. Qin, K. Strauss, and D. Lymberopoulos, "Progressive authentication: deciding when to authenticate on mobile phones," in *Proceedings of the 21st USENIX Security Symposium (USENIX Security '12)*, pp. 301–316, USENIX, Bellevue, Wash, USA, 2012, https://www.usenix.org/conference/usenixsecurity12/technical-sessions/presentation/riva.

[13] L. Fridman, S. Weber, R. Greenstadt, and M. Kam, "Active authentication on mobile devices via stylometry, application usage, web browsing, and GPS location," *IEEE Systems Journal*, 2016.

[14] F. Li, N. Clarke, M. Papadaki, and P. Dowland, "Active authentication for mobile devices utilising behaviour profiling," *International Journal of Information Security*, vol. 13, no. 3, pp. 229–244, 2014.

[15] C. Imbert, "Beyond the cookie: using network traffic characteristics to enhance confidence in user identity," 2014, http://software-security.sans.org/resources/paper/reading-room/cookie-network-traffic-characteristics-enhance-confidence-user-identity/.

[16] M. Antal, L. Z. Szabo, and I. Laszlo, "Keystroke dynamics on android platform," in *Proceedings of the 8th International Conference Interdisciplinarity in Engineering (INTER-ENG '14)*, vol. 19, pp. 820–826, Tirgu Mures, Romania, October 2014, http://www.sciencedirect.com/science/article/pii/S221201731500119X.

[17] Y. Deng and Y. Zhong, "Keystroke dynamics user authentication based on Gaussian mixture model and deep belief nets," *ISRN Signal Processing*, vol. 2013, Article ID 565183, 7 pages, 2013.

[18] J. Wu and Z. Chen, "An implicit identity authentication system considering changes of gesture based on keystroke behaviors," *International Journal of Distributed Sensor Networks*, vol. 11, no. 5, 2015, http://journals.sagepub.com/doi/abs/10.1155/2015/470274.

[19] P. S. Teh, A. B. J. Teoh, and S. Yue, "A survey of keystroke dynamics biometrics," *The Scientific World Journal*, vol. 2013, Article ID 408280, 24 pages, 2013.

[20] N. Karapanos, C. Marforio, C. Soriente, and S. Capkun, "Soundproof: usable two-factor authentication based on ambient sound," in *Proceedings of the 24th USENIX Security Symposium (USENIX Security '15)*, pp. 483–498, USENIX Association, Washington, DC, USA, August 2015, https://www.usenix.org/conference/usenixsecurity15/technical-sessions/presentation/karapanos.

[21] A. Kale, N. Cuntoor, B. Yegnanarayana, A. N. Rajagopalan, and R. Chellappa, "Gait analysis for human identification," in *Audio- and Video-Based Biometric Person Authentication: 4th International Conference, AVBPA 2003 Guildford, UK, June 9–11, 2003 Proceedings*, vol. 2688 of *Lecture Notes in Computer Science*, pp. 706–714, Springer, Berlin, Germany, 2003.

[22] C. Ntantogian, S. Malliaros, and C. Xenakis, "Gaithashing: a two-factor authentication scheme based on gait features," *Computers and Security*, vol. 52, pp. 17–32, 2015.

[23] A. J. Oliner, A. P. Iyer, I. Stoica, E. Lagerspetz, and S. Tarkoma, "Carat: collaborative energy diagnosis for mobile devices," in *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems (SenSys '13)*, pp. 10:1–10:14, Roma, Italy, November 2013.

[24] E. Peltonen, E. Lagerspetz, P. Nurmi, and S. Tarkoma, "Energy modeling of system settings: a crowdsourced approach," in *Proceedings of the 13th IEEE International Conference on Pervasive*

Computing and Communications (PerCom '15), pp. 37–45, IEEE, St. Louis, Mo, USA, March 2015.

[25] D. T. Wagner, A. Rice, and A. R. Beresford, "Device analyzer: large-scale mobile data collection," *ACM SIGMETRICS Performance Evaluation Review*, vol. 41, no. 4, pp. 53–56, 2014.

[26] J. Spooren, D. Preuveneers, and W. Joosen, "Mobile device fingerprinting considered harmful for risk-based authentication," in *Proceedings of the 8th European Workshop on System Security (EuroSec '15)*, pp. 6:1–6:6, Bordeaux, France, April 2015.

[27] I. Traore, I. Woungang, M. S. Obaidat, Y. Nakkabi, and I. Lai, "Online risk-based authentication using behavioral biometrics," *Multimedia Tools and Applications*, vol. 71, no. 2, pp. 575–605, 2014.

[28] S. P. Banerjee and D. Woodard, "Biometric authentication and identification using keystroke dynamics: a survey," *Journal of Pattern Recognition Research*, vol. 7, no. 1, pp. 116–139, 2012.

[29] C.-L. Liu, C.-J. Tsai, T.-Y. Chang, W.-J. Tsai, and P.-K. Zhong, "Implementing multiple biometric features for a recall-based graphical keystroke dynamics authentication system on a smart phone," *Journal of Network and Computer Applications*, vol. 53, pp. 128–139, 2015.

[30] M. Frank, R. Biedert, E. Ma, I. Martinovic, and D. Song, "Touchalytics: on the applicability of touchscreen input as a behavioral biometric for continuous authentication," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 1, pp. 136–148, 2013.

[31] P. S. Teh, N. Zhang, A. B. J. Teoh, and K. Chen, "A survey on touch dynamics authentication in mobile devices," *Computers and Security*, vol. 59, pp. 210–235, 2016.

[32] R. V. Yampolskiy and V. Govindaraju, "Behavioural biometrics: a survey and classification," *International Journal of Biometrics*, vol. 1, no. 1, pp. 81–113, 2008.

[33] H. Witte, C. Rathgeb, and C. Busch, "Context-aware mobile biometric authentication based on support vector machines," in *Proceedings of the 4th International Conference on Emerging Security Technologies (EST '13)*, pp. 29–32, September 2013.

[34] Z. Chair and P. K. Varshney, "Optimal data fusion in multiple sensor detection systems," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 22, no. 1, pp. 98–101, 1986.

[35] A. Fridman, A. Stolerman, S. Acharya et al., "Decision fusion for multimodal active authentication," *IT Professional*, vol. 15, no. 4, pp. 29–33, 2013.

[36] K. O. Bailey, J. S. Okolica, and G. L. Peterson, "User identification and authentication using multi-modal behavioral biometrics," *Computers and Security*, vol. 43, pp. 77–89, 2014.

[37] D. Preuveneers and W. Joosen, "SmartAuth: dynamic context fingerprinting for continuous user authentication," in *Proceedings of the 30th Annual ACM Symposium on Applied Computing (SAC '15)*, pp. 2185–2191, Salamanca, Spain, April 2015.

[38] L. Olejnik, G. Acar, C. Castelluccia, and C. Diaz, "The leaking battery: a privacy analysis of the html5 battery status api," Report 2015/616, Cryptology ePrint Archive, 2015, http://eprint.iacr.org/.

[39] P. Eckersley, "How unique is your web browser?" in *Privacy Enhancing Technologies: 10th International Symposium, PETS 2010, Berlin, Germany, July 21–23, 2010. Proceedings*, vol. 6205 of *Lecture Notes in Computer Science*, pp. 1–18, Springer, Berlin, Germany, 2010.