

©2015 IEEE. This is the authors version of the work. It is posted here for your personal use. Not for redistribution. The definitive Version of Record was published in Proceedings of the 2015 IEEE 21st International Conference on Embedded and Real-Time Computing Systems and Applications, Hong Kong, 2015, pp. 41-50.

Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting / republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

DOI: 10.1109/RTCSA.2015.21

Smart²: Smart charging for smart phones

Alma Pröbstl, Philipp Kindt, Emanuel Regnath and Samarjit Chakraborty

Institute for Real-Time Computer Systems

Technische Universität München, Germany

Email: {alma.proebstl, philipp.kindt, emanuel.regnath, samarjit}@tum.de

Abstract—In this paper, we present Smart², an advanced smartphone charger that mitigates battery’s capacity fading, which until now has usually been ignored. Smart² exploits the fact that many users charge their phones over night. Since the overnight charging duration is unnecessarily long, the battery is subjected to a high average state of charge (SOC), which accelerates battery aging. Therefore, we delay the charging adaptively to be done shortly before the phone is unplugged. With this scheme, clearly when averaged over the duration of the night, the average SOC is lower and hence aging is reduced. Indicators are a set alarm clock and/or statistics of previous usage. Similarly, we lower the maximum target SOC. To enable this, the main challenges are firstly to find a solution that does not negatively influence the usability and secondly to quantify the achieved savings in terms of aging mitigation. Towards this, we propose a novel charging scheme which can be implemented in the smartphone’s firmware. Furthermore, we propose a modified battery charging device that can be used with almost all existing smart phone models. Using our proposed techniques, the average battery cycle life can be nearly doubled from 3.7 to 6.6 years.

Keywords—*smartphone; battery aging; capacity fading; charging*

I. INTRODUCTION

One of the most important limitations of smartphones is their battery runtime. Despite the fact that batteries have evolved during the last years and in spite of the high-capacity lithium-ion cells that smartphones are equipped with nowadays, smartphones usually have to be recharged every day. Whereas the available runtime might be sufficient shortly after the purchase of the device, the capacity fading effect reduces the available capacity over time until after 2-3 years only 80% of the initial capacity might be available and the battery gets empty before the user usually recharges it.

Battery Aging: Battery aging usually means power and capacity fading. In this work, we are interested in the latter and use the terms aging and capacity fading interchangeably. Battery aging is influenced by the battery temperature, average state of charge (SOC), SOC swing and charge-/discharge current. While the battery’s capacity decreases even when not in use, the reduction of remaining available capacity increases whenever the battery is cycled, in other words when it is charged or discharged.

As a result of capacity fading, either larger batteries are built into the phone that will last for a whole day even when aged or the battery needs to be replaced, which imposes a significant cost. Moreover, there are many phones which have batteries that cannot be replaced by the user. If the battery runtime decreases below an acceptable value, the whole device usually is replaced. In addition to the additional expenses, the

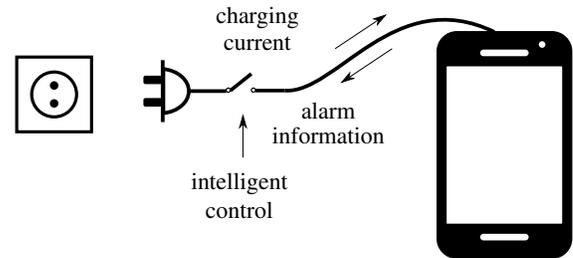


Fig. 1. Intelligent smartphone charger. Charging is adapted to the use pattern in order to reduce battery degradation.

high amount of electrical waste is of concern. Considering a decreasing innovation rate on the smartphone’s hardware and a resulting longer usage time, a fast aging battery remains the bottleneck in smartphone disposal and resulting environmental issues.

Hence, battery aging needs to be mitigated. We propose a novel aging aware charging scheme that mitigates the capacity fading by reducing the average SOC because a high average SOC is one of the factors that increase capacity fading. There already exist intelligent software components in other device classes such as laptop computers that manage the charging process in order to increase the cycle life of the battery. For example, the Lenovo Battery Manager [1] allows to select a maximum state of charge. For smartphones, simply reducing the maximum SOC would reduce the battery aging, but in some cases it would artificially shorten the battery runtime below the acceptable level. However, the detailed personal data available on smartphones, viz., a detailed history of the phone usage and recharging behavior can be exploited to make predictions to optimize the charging procedure towards reducing the battery aging without impairing the usability of the device.

In this paper, we for the first time present a novel context aware scheme for aging aware battery charging that achieves the reduction of the average SOC and hence contributes to mitigation of the battery’s capacity fading. This is done by shifting the start of the charging process in time and by lowering the maximum SOC: Since smartphones are usually charged during the night, batteries usually reach 100% of their maximum SOC a long time before their chargers are unplugged. In this period of time, the SOC remains at its maximum level thereby accelerating the battery aging. By shifting the charging process to be done shortly before the phone is unplugged, the phone remains at a much lower SOC for most hours of the night. Clearly the average SOC is lower without having a negative impact on the user experience. In order to address an unexpected need for phone energy during night, we propose a two stage charging process. When the

charger is plugged in, the battery is charged to a lower SOC first (e.g., 50%) and the remaining charge is provided in a second charging process that finishes shortly before the phone is unplugged from the charger. The unplugging time is estimated based on the alarm clock a user has configured in his mobile phone (see Figure 1) as well as statistical usage data. In addition, the charge which is consumed during the day is predicted to allow for an adaptive reduction of the maximum SOC while still providing enough charge to keep the phone operative during the whole day. Combining both, shifting the charging in time and lowering the maximum target SOC, the battery lifetime can be almost doubled from 3.7 to 6.6 years, as shown in Section V.

Using our scheme allows for building batteries with smaller capacity into phones that due to less aging, will be usable for a longer period of time. Another advantage of these batteries is their smaller size and lower weight. Furthermore, the amount of electronic waste can be reduced.

In this work, we make the following contributions:

- We present a novel context aware charging scheme that uses a combination of shifting the charging process in time and adaptively lowering the maximum charging SOC according to the required amount of charge needed per day. Hereby, the average SOC is reduced, which results in a mitigation of the battery’s capacity fading.
- Towards this, we compare three predictors that exploit the available usage data and the regularity of usage patterns on smartphones to adaptively adjust the charging process. We compare a simple moving average, an exponential moving average and a probabilistic predictor.
- We describe two different implementations of this system: One has been realized as a pure software solution running as an application on an Android smartphone, the other one is an external charger that works independently of the specific charger chip of the smartphone for all Android devices.
- We evaluate our proposed scheme using real-world measurements combined with a battery aging simulation model and estimate the impact of our system on the battery aging.
- Finally, we conduct a user study to support our assumptions on usage patterns and test the usability of our smart charger.

The rest of this paper is organized as follows. In Section II, we present related work including theory on battery aging and previously known aging aware charging systems. In Section III, we present our proposed system and describe multiple implementation possibilities. In Section IV we give details on the three predictors. To demonstrate the smart charger’s effectiveness, we evaluate its impact on the battery aging in Section V. Finally, we conclude our work in Section VI.

II. RELATED WORK AND BACKGROUND

In this section, we present previous work on battery aging and derived models in order to justify the aging mitigation of our approach. In addition, we summarize other aging aware applications for laptops and smartphones. Finally, we give an overview of works that analyze and predict battery usage patterns of smartphone users.

A. Battery Aging

Battery aging of Li-Ion batteries means either loss of power capability due to the increase of the internal resistance, or capacity fading. Capacity fading is the reduction of the available charge stored in the battery within one cycle. Our work aims at the mitigation of the latter. There are two types of aging, calendar aging and cycle aging. A battery is always exposed to calendar aging, independent of whether the battery is actually used or not. Factors that influence calendar aging are the ambient temperature and the (average) SOC of the battery. When the battery is in use, its degradation is additionally increased by cycle aging, which depends on the SOC deviation and the charging/discharging current. The impact of these factors on the aging also depends on the cell chemistry.

Measurements on cycle aging of a 900 mA LiCoO₂ battery at 25°C that are presented in [2] reveal the following results. A higher cut-off voltage (full battery) while charging increases the capacity fading, while there is no dependence on the the cut-off voltage (empty battery) while discharging. Furthermore, lower charging and discharging rates decrease aging. Also a longer duration of float charging, that compensates the self discharge of a fully charged battery and maintains the full charge level as long as the battery is still connected to the charger, increases the capacity fading.

Calendar aging of Li-ion cells is investigated with measurements on a cell with a hard carbon anode and a LiNiMnCoO₂ cathode in [3]. The authors find that capacity fading depends on the temperature, whereat a lower temperature is better. Another factor of influence is the cell voltage which corresponds to the battery SOC. Clearly, a lower voltage (or SOC, respectively) is preferable. The same holds true for a LiCoO₂ cell as shown in [4]. In [5], an increased degradation is found when the cell voltage is above 4V, which means in this case a SOC of approximately 50% or higher.

In this work, an aging model is used for quantifying the amount of battery degradation with and without our proposed scheme. In [6], a comprehensive overview of battery aging models is given. We select the aging model suggested by [7], which uses average SOC, SOC deviation and temperature as inputs and simulates the capacity fading over time. The parameter fitting in this model is done for a lithium iron phosphate cell, while the typical cell chemistry used in portable devices is LiCoO₂. In fact, most available and already fitted battery aging models do use other cell chemistries than LiCoO₂ and instead take data of batteries developed for use in electric vehicles that are optimized for high power and slower aging. According to [8], the cycle life of LiFePO₄ cells is more than two times higher than the one of LiCoO₂. Therefore, our results on battery aging presented in Section V can be seen as an upper bound, while the real cycle life is likely to be even shorter and hence the beneficial impact of our proposed charging technique is expected to be even higher. Also, from a consumer perspective, building batteries with improved degradation profiles into portable devices would be desirable. A sign that battery aging awareness also is of concern to manufacturers is a patent filed by Apple Inc. [9] that describes a tracking procedure for capacity fading in portable devices.

The model [7] which we use in our simulations considers single charge and discharge cycles of an identical pattern. One

cycle consists of the time interval T_m . In order to derive the average state of charge \overline{SOC} , we calculate

$$\overline{SOC} = \left(\int_{T_m} SOC(t) dt \right) / T, \quad (1)$$

where T is the duration of the time interval.

The normalized SOC deviation σ is calculated by

$$\sigma = 2\sqrt{3 \int_{T_m} (SOC(t) - \overline{SOC})^2 dt / T}. \quad (2)$$

Typically a single cycle does not charge from 0% to 100% and then discharge back to 0%. Therefore, the effective number of throughput cycles N needs to be determined:

$$N = \int_{T_m} \frac{|i(t)| dt}{2Q_{nom}}, \quad (3)$$

where $i(t)$ is the charge or discharge current and Q_{nom} is the nominal amount of charge that can be stored in the battery.

Now the prepared data can be fed into the iterative battery aging model [7] that is based on a crack propagation model. A parameter L_1 which is referred to as *incrementing life parameter* is calculated by

$$L_1 = K_{co} N \exp\left((\sigma - 1) \frac{T_{ref} + 273}{K_{ex}(T_B + 273)}\right) + 0.2 \frac{t_{cycle}}{t_{life}}, \quad (4)$$

where K_{co} is a normalization coefficient for N and K_{ex} is a constant exponent for depth of discharge. T_{ref} is the reference battery temperature of 25°C and T_B the battery temperature. t_{cycle} is the duration of one cycle and t_{life} the *shelf life* at 25°C and 50% SOC until 80% of the initial capacity remain (we set this value to 10 years). The life parameter is further adjusted by

$$L_2 = L_1 \exp(4K_{soc}(\overline{SOC} - 0.5))(1 - L), \quad (5)$$

using a constant K_{soc} to account for the average SOC. Finally, the total increase in the life parameter L is given by

$$L(T_m) = L_2 \exp\left(K_t (T_B - T_{ref}) \frac{T_{ref} + 273}{T_B + 273}\right), \quad (6)$$

where K_t accounts for a doubling of the decay rate for each 10°C rise in temperature.

Summing up the damage done by each cycle, one gets the remaining life of the battery. Assuming that the following cycles will follow the same profile, one can simulate the time until the defined end of life is reached.

B. Battery Management Applications

From the studies and models described above, it is clear that lowering the maximum SOC would help to reduce the battery aging. For notebook computers, there already exist programs that stop the charging after a predefined maximum SOC has been reached. This maximum SOC can be chosen manually by the user. Examples for such notebook applications are the Lenovo Power Manager [1] and the Samsung Battery Life Extender [10], which work only on notebooks of the respective manufacturers. Hence for laptops as well as

smartphones, the battery control possibilities also depend on the specific device. Our application uses the concept of limiting the maximum SOC and expands it by adaptively doing so as well as by adaptively setting the start time of the charging process based on the usage pattern. Both concepts reduce the overall average SOC. Lowering the target SOC actually avoids the disadvantageous high voltages, while delaying the charging process only reduces the time spent at the high voltages. The typically more distinct regularity of smartphone usage patterns compared to the ones of laptops facilitates an adaptive implementation on smartphones.

In case of smartphones, the application *Battery Doctor* [11] claims to prolong battery life by a three stage charging method. As the source code of this application is not publicly available, we cannot find out whether the application's developers control the charging current, which however seems unlikely as Android does not provide access to charging control functionalities on unrooted devices. It seems that the application mainly encourages the user to develop a battery friendly charging behavior by notifying the user on, e.g., when to unplug the phone.

C. Smartphone Usage Predictors

As already mentioned, we suggest to adaptively delay the point in time when the SOC is charged to its target value in order to lower the average SOC. Hence, we need to estimate the unplug time of the phone by either evaluating the alarm clock time or predicting the unplug time based on statistics of the previous user behavior in order to adjust the starting time of the charging process. The assumption that most people charge their phone over night has been confirmed by [12] and [13]. Users that mainly charge over night have a longer average plug duration than users that also charge during the day, which means that keeping ones phone plugged over night [12] leads to an unnecessary high average SOC. In [13] it is shown that in 77% of the cases the smartphone is plugged in for longer than 30 min after the full charge level has been reached. Therefore, the charging start time can be delayed in order to decrease the average SOC.

In addition, predicting the SOC drain between two charging sessions helps to adjust the cut-off SOC (or voltage) when charging. Often different power sources, such as USB and AC, go along with different plug durations. The usage frequency of these different sources varies among users [14]. Also the start of charge and end of charge level depends on patterns determined by the individual use [14]. According to [13], the average lowest SOC level before recharging determined among multiple users is at 30%. Accordingly, a great amount of users could discharge their phones to a lower SOC while still having enough energy left in their batteries before the next recharge.

Several predictors of future battery level have been presented in literature [14], [12]. We need to estimate times of unplug events and the energy consumption until the next plug event determining which requires similar methods as the ones on predicting the battery level that are presented in literature.

III. AGING AWARE CHARGING

In the following, we explain our aging aware charging concept and discuss challenges and design options when realizing our approach on different target platforms.

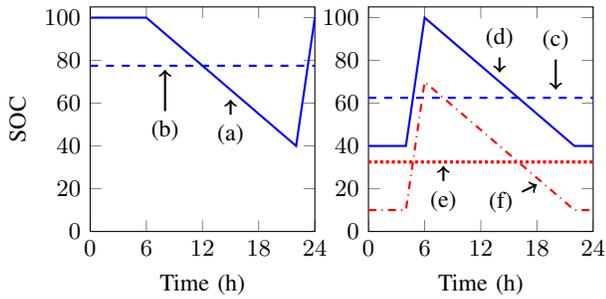


Fig. 2. Comparison of charging schedules with and without smart charging. Curve (a) is the conventional charging scheme, curve (d) shifts the charging in time and curve (f) additionally has a lower maximum target SOC. Curves (b), (c) and (e) are the respective average SOC.

A. Aging Mitigation by Delayed Charging

As stated before, one of the factors that increase the battery aging is a high average SOC. By lowering the average SOC of a smartphone battery, its degradation can be decreased. This is done by shifting the start of the charging process in time and by adaptively reducing the maximum SOC. Figure 2 illustrates this charging scheme. The left graph shows a typical profile (a), where the SOC stays at 100% over night with an average SOC of 77.5% (b). In the right graph our proposed optimizations are applied: The average SOC is lowered to 62.5% (c) by delaying the charging process to be done in the morning (d). In the lower graph on the right, the average SOC is further reduced to 32.5% (e) by lowering the charging target SOC (f).

The delay of the charging process is either set manually by the user, gathered automatically by reading the alarm clock or predicted from the user’s statistical usage pattern. Similarly, the target SOC can be either selected by the user or can be learned by the system based on the average daily usage including a safety margin. As a wrong assumption on the unplug time causes great inconvenience for the user, i.e., the battery is still uncharged when unplugged, we suggest to immediately charge to a medium SOC and delay the charging only after this medium SOC has been reached (see Section III-D), which however obviously results in a less efficient aging mitigation.

An obvious objection towards this scheme is that charging over night with a reduced charging current might also have an improving effect on battery degradation. Hence the questions arises which of the two schemes should be used. First of all, no definitive answer can be provided as this question needs further investigation in future work. There already exists a patent [15] for charge rate reduction which argues that a reduced charging current has a positive impact on the battery lifespan in terms of reducing the capacity fading. Unfortunately no measurement or simulation results are provided. However, one has to keep in mind that the charging the smartphone is likely to have an impact on the temperature of the smartphone, which has a strong effect on the battery aging. Therefore it cannot be said without further discussion, whether a reduced charging current has a better effect than a delayed one.

In the following, we describe two realization options of the proposed system: A software implementation as a smartphone application and a dedicated charging device. We then introduce an algorithm that relies either on values manually entered by

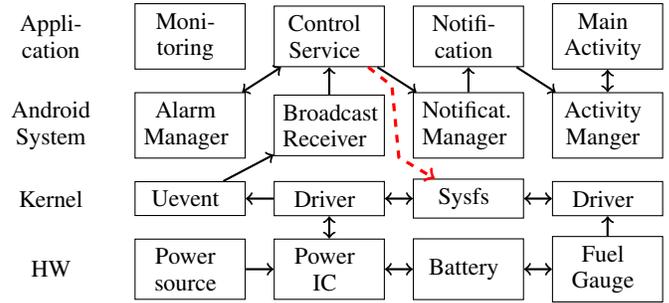


Fig. 3. System Abstraction Layers in Android.

the user or uses the alarm clock and predictors to calculate delay and target SOC.

B. Smart Charger Android Application

The above described solution can be implemented either directly on the smartphone or on an external charging device. In order to later understand the drawbacks of the direct implementation, we previously need to explain the dependencies of the Android operating system and the smartphone hardware with its respective drivers.

Figure 3 shows the abstraction layers of the hardware and the software. On the hardware layer, there are three main components that are involved in charging the smartphone battery. The power IC manages power sources and the power distribution to the system. Among other tasks, it is responsible for voltage regulation, voltage scaling, power source selection and charging functions. The charger controls the charging process of the battery and one of its tasks is to ensure a safe charging process. The fuel gauge monitors the battery and provides state of charge, state of health, state of connection, capacity and voltage readings to the system.

On the software level, the application has a registered broadcast receiver for the power connector state. A broadcast receiver receives a notification if a system or application event occurs, i.e., when a charger is connected or disconnected. If the application is notified about a plug event and the desired charging time is still in the future, the charging has to be disabled. Hereby, the virtual file system *sysfs* is used to exchange information between the power management chip and the user space, i.e., writing to the corresponding *sysfs* file can enable or disable the charging process.

Due to different power ICs with different drivers, not all smartphone models provide the same charging control functionalities. At the time of writing, the *sysfs* charging control varies. I.e., for the Nexus 4 phone, a switch variable enables or disables charging, whereas in the Nexus 5 phone, a parameter is provided to reduce the charging current.

As the *sysfs* is located in the kernel space and no standardized API is provided by the Android system, it is currently not possible to write a universal application that works for all Android smartphones. When writing an application for a specific smartphone, root access is required in order to access the *sysfs* interface. However, a solution which can be used for almost all current smartphones is implementing our smart charging scheme into the charger. Such a smart charger hardware is described next.

C. Smart Charger Hardware

As explained above, no standardized implementation from the smart charger as a pure Android application is currently possible. Therefore, we develop a second solution that is based on hardware and hence works independent of the smartphone model. The attractiveness of this hardware solution arises from its independence of the smartphone power IC and a less profound intervention in the operating system. A switch is inserted in the supply lines of the USB cable used for charging in order to interrupt the current supply if needed. The switch is controlled by a microcontroller which also is used as a USB host controller in order to receive SOC data and the alarm clock information from the smartphone.

D. Alarm Clock Based Charging Delay

The battery health aware charging algorithm can be implemented either on the external microcontroller or directly on the smartphone. In order to reduce the average SOC, we adjust two parameters: We adaptively delay the start time of the charging process and we lower the target SOC soc_{target} . Algorithm 1 determines the reserve amount of charge soc_{medium} , target SOC soc_{target} and the unplug time t_{unplug} . These are then used to manage the charging scheme.

Algorithm 1 Smart Charging

```

1: function SMARTCHARGING(mode)
2:   if mode == manual then
3:      $soc_{medium} = \text{GetReserveChargeFromSettings}()$ 
4:      $soc_{target} = \text{GetTargetSocFromSettings}()$ 
5:      $t_{unplug} = \text{GetUnplugTimeFromSettings}()$ 
6:   else
7:     if mode == alarm then
8:        $t_{unplug} = \text{GetUnplugTimeFromAlarmTime}()$ 
9:     else
10:       $t_{unplug} = \text{GetUnplugTimeFromPredictor}()$ 
11:    end if
12:     $soc_{medium} = \text{SetReserveChargeToConstantValue}()$ 
13:     $soc_{target} = \text{GetTargetSocFromPredictor}()$ 
14:  end if
15:  EnableCharging( $soc_{medium}$ )
16:  DisableCharging( $t_{unplug} - \Delta t_{charge}$ )
17:  EnableCharging( $soc_{target}$ )
18: end function

```

The *mode* variable allows to switch between three operation scenarios: *manual*, *alarm* and *predictor*. If *mode* is set to *manual* the target SOC soc_{target} , reserve amount of charge soc_{medium} and unplug time t_{unplug} are set manually by the user in the preferences settings menu. To retrieve this data, we determine three functions:

- $\text{GetReserveChargeFromSettings}()$: Gets the manually set value for the reserve amount of charge soc_{medium} .
- $\text{GetTargetSocFromSettings}()$: Gets the manually set value for the target SOC soc_{target} .
- $\text{GetUnplugTimeFromSettings}()$: Gets the manually set value for the unplug time t_{unplug} .

Alternatively, the unplug time is set to either the value of the alarm clock if it is set or else it is predicted based on

unplug statistics as indicated by the *mode* variable being set to *alarm* or *predictor*, respectively:

- $\text{GetUnplugTimeFromAlarmTime}()$: In case the alarm clock is set, the unplug time t_{unplug} is determined from its set value. We use the alarm clock application that comes pre-installed with Android and is used by most users.
- $\text{GetUnplugTimeFromPredictor}()$: In case no alarm clock is set, a predictor is used to determine the unplug time t_{unplug} .

The reserve amount of charge and the target SOC are obtained in the same way for both modes *alarm* and *predictor*:

- $\text{GetTargetSocFromPredictor}()$: Another predictor determines the target SOC soc_{target} .
- $\text{SetReserveChargeToConstantValue}()$: The medium SOC soc_{medium} is set to a predefined, relatively low value.

Once the smartphone is plugged in, it charges to the medium SOC soc_{medium} to have a safety margin in case of the smartphone being unplugged earlier than expected. After charging to the reserve SOC soc_{medium} , charging is disabled until the charging time is reached. The charging time is derived from the difference between the determined unplug time and the estimated duration: $(t_{unplug} - \Delta t_{charge})$. The duration Δt_{charge} describes the time it takes to charge the phone to the target SOC soc_{target} . For the sake of simplicity of presentation, in our implementation, we set this duration to a constant value because the variation among phones is small. Nevertheless, this duration could also be estimated from previous charging cycles which might be included in future versions. Two functions control the charging:

- $\text{EnableCharging}(soc)$ charges the battery until a certain SOC value has been reached.
- $\text{DisableCharging}(time)$ disables the charging for a given time.

IV. PLUG DURATION MODEL

Three predictors that model the plug duration have been developed and will be compared in the following: a simple moving average predictor, an exponential moving average predictor and a probabilistic predictor.

As discussed in Section II, the best results in lowering the average battery SOC and hence the battery aging are expected when delaying the night time charging. Previous studies [12] as well as our own sample data in Figures 4 and 5 show that plug durations tend to be longer when the plug event occurs at night time.

As expected, the charging behavior varies among users, see Figure 6. The x-axis shows the charging duration rounded to full hours. The number of events per a certain duration is counted. User 2 has many short charges. Both users have an increased number of samples for 8-9 hours which is likely to be the charging over night. Due to the differences among users, adaptive predictors are of advantage.

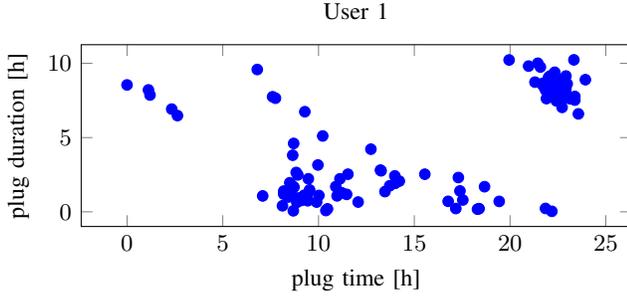


Fig. 4. If plugged at day the charging durations tends to be shorter.

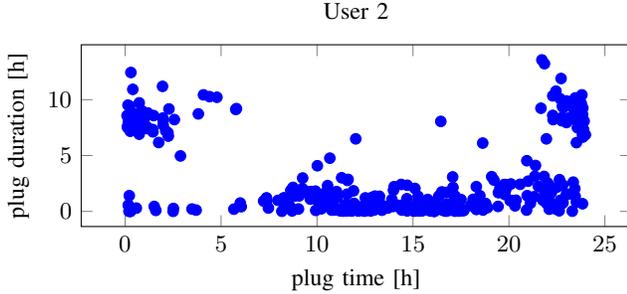


Fig. 5. Longer charging durations occur when plugged at night times.

A. Simple Moving Average Predictor

If the manual mode is not active, two predictors take action. In case the alarm clock is not available, a first predictor estimates the time the phone will be unplugged from the power source (AC, USB). In order to further lower the SOC, a second predictor computes a target SOC soc_{target} below 100%. A moving average can be used for a simple implementation. However there exist more elaborate models in literature.

In order to predict the time the phone usually remains plugged Δt_{plug} , we sort the last plug durations $\Delta t_{m,k}$ into multiple bins k according to the plug time. We use one bin per hour of the day, hence the total number of bins is $K = 24$. Once a plug event is detected, the moving average in bin k over the last $N_{plug} = 5$ events (the number compromises between smoothing over the last numbers and forgetting the past fast enough when the user's behavior changes) is calculated by

$$\Delta t_{plug,avg,k} = \frac{1}{N_{plug}} \sum_{m=M-(N_{plug}-1)}^M \Delta t_{m,k}, \quad (7)$$

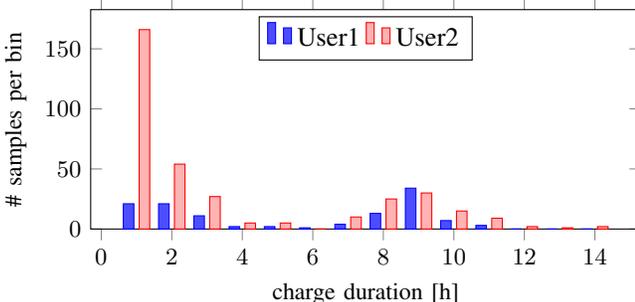


Fig. 6. Number of charging events for given charge duration.

where M is the total number of entries into one specific bin and $t_{m,k}$ are the stored durations in bin k and counter m .

The time when charging should be enabled is determined by

$$t_{startcharging} = t_{now} + \Delta t_{plug,avg} - \Delta t_{charge} - \Delta t_{buffer}, \quad (8)$$

where t_{charge} is the duration it takes to charge the battery to the target SOC soc_{target} and Δt_{buffer} is an additional safety time to account for possible mispredictions.

We need a second predictor that determines the target SOC soc_{target} for our evaluation. It uses the same number of bins $k = 0..23$ and calculates the moving average over the last $N_{soc} = 5$ events in the respective bin k :

$$\Delta soc_k = \frac{1}{N_{soc}} \sum_{m=M-(N_{soc}-1)}^M \Delta soc_{m,k}, \quad (9)$$

where $\Delta soc = soc(t_{lastunplug}) - soc(t_{plugtime})$ with $soc(t_{lastunplug})$ being the SOC when the phone was previously unplugged and $soc(t_{plugtime})$ being the SOC when the phone was plugged. The target SOC soc_{target} is then calculated by

$$soc_{target} = \Delta soc_k + \Delta soc_{buffer}, \quad (10)$$

with Δsoc_{buffer} being a safety margin to account for longer usage. Also note that the bin k_{soc} is determined by the time the phone was last unplugged $t_{lastunplug}$ instead of using the plug time $t_{plugtime}$ as for the plug duration Δt_{plug} . Algorithm 2 shows the sequence of calculations. If a plug event occurs, first the history of Δsoc values is updated in the respective bin (`UpdateSocHistory(k_{soc} , Δsoc_M)`). Next, the unplug time t_{unplug} and the target SOC soc_{target} are predicted (`CalculateUnplugTime(k_{plug} , H_P)` and `CalculateTargetSoc(k_{soc} , H_S)`), based on the current hour. When the phone is unplugged, the plug duration is stored in the respective bin (`UpdatePlugDurationHistory(k_{plug} , Δt_M)`).

Algorithm 2 Predictor

- 1: **if** PlugEvent **then**
 - 2: $H_S \leftarrow$ UpdateSocHistory(k_{soc} , Δsoc_M)
 - 3: CalculateUnplugTime(k_{plug} , H_P)
 - 4: CalculateTargetSoc(k_{soc} , H_S)
 - 5: **end if**
 - 6: **if** UnplugEvent **then**
 - 7: $H_P \leftarrow$ UpdatePlugDurationHistory(k_{plug} , Δt_M)
 - 8: **end if**
-

B. Exponential Smoothing

As a comparison, we use exponential smoothing. We simply replace Equation 7 and calculate the exponential moving average recursively by

$$\begin{aligned} \Delta t_{plug,k,1} &= \Delta t_{1,k}, \text{ for } m = 1 \\ \Delta t_{plug,k,m} &= \alpha \Delta t_{m,k} + (1 - \alpha) \Delta t_{plug,k,m-1}, \quad (11) \\ &\text{for } m > 1, \end{aligned}$$

where the smoothing factor α is set to 0.2 in our example.

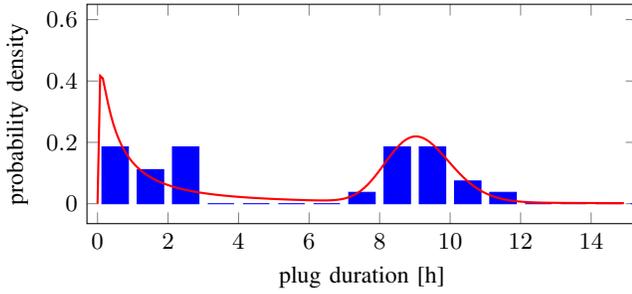


Fig. 7. Fitting a bimodal log-normal probability density function to the statistical data.

C. Probabilistic Predictor

Our third predictor fits the data to log-normal distributions, similar to the one suggested by [16] for stay durations at a given location.

Given a certain plug time x , we can fit the conditional probability density function $f(y|x)$ to the statistical data. We use discrete hourly bins $x = 0..23$. As can be seen from Figure 6, the plug duration is either very short or lies in the range of 8-9 hours. Furthermore, the duration cannot take negative values. Hence, we model the probability density function of the charging delay as a bimodal log-normal distributions. We fit the data to a mixture of log-normal functions:

$$f(y|x) = \frac{p}{y\sigma_1\sqrt{2\pi}} \exp\left(-\frac{(\ln y - \mu_1)^2}{2\sigma_1^2}\right) + \frac{1-p}{y\sigma_2\sqrt{2\pi}} \exp\left(-\frac{(\ln y - \mu_2)^2}{2\sigma_2^2}\right), \quad (12)$$

where y are the durations, σ_1 and σ_2 are the variances of the log-normal functions, μ_1 and μ_2 are the mean values of the two functions and p is a weight applied to the two probability density functions. The data is fitted using a maximum likelihood estimator. Figure 7 gives an example of a resulting curve compared to the measured data.

Apart from the additional information that is provided by the mixture of log-normal distributions, such as variance, expected value and the respective weights, we can also determine a value that gives us a prediction on the duration.

One drawback of this predictor, which fits the statistical data to probability density function, is that in case the sample size is very low, fitting is impossible or very inaccurate. Hence, our implementation deals with the following cases: if no samples are available, the delay is set to zero and charging starts immediately. If there are only one or two data samples the mean value of these samples is calculated. In case of three samples or if the bimodal predictor fails to fit the data and degrades, a unimodal predictor is used:

$$f(y|x) = \frac{1}{y\sigma\sqrt{2\pi}} \exp\left(-\frac{(\ln y - \mu)^2}{2\sigma^2}\right). \quad (13)$$

Figure 8 shows data fitted to a unimodal log-normal function. For four samples and more, the bimodal log-normal predictor is used.

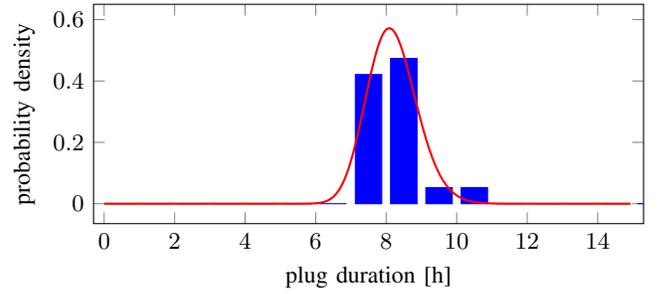


Fig. 8. Statistical data fitted to a unimodal log-normal function.

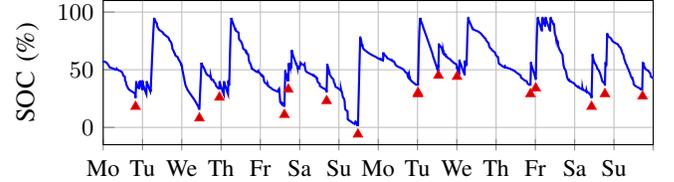


Fig. 9. Battery profile for a user that runs the smart charger application. Charging is delayed according to a set alarm. The red triangles indicate the plug times.

V. EVALUATION

In this section, we show the amount of battery aging that can be mitigated by delaying the charging and lowering the target SOC. We first evaluate our smart charger application, which delays charging based on the alarm clock, in a real user test. We then compare the results of an omniscient simulator to the ones achieved by the predictor that determines its decisions on statistical usage patterns.

A. Smartphone Application Evaluation

Our first experiment shows how charging is delayed merely based on the alarm clock in a real usage scenario. A user runs our smart charger application on his smartphone for two weeks and the charging is delayed according to his alarm clock settings. As can be seen in Figure 9, the target SOC is adjusted according to user preferences on the maximum SOC in the morning and minimum SOC in the evening and a linear interpolation when charged during the day. Therefore, the SOC usually varies between 95% and 50%. The simulated cycle life for this plot is 6.6 years. When comparing Tuesday and Thursday in week 1 to Wednesday in week 2, one can see that the minimum SOC was set from 50% to 40% during the test period. The profile does not have long periods of being at a high SOC, and starts discharging shortly after being unplugged (only exception is the second Friday, when no alarm was set). It can be seen that having reached the target value when charging the smartphone triggers drawing a rather high discharge current, with a gradient that is even higher than the discharge current while in use. We assume that this is due to some background processes remaining active in the assumption that the phone is powered by AC since it is plugged in even though charging is disabled. Currently we counteract by frequently recharging to the target value. However, a better solution would be to schedule the background processes differently.

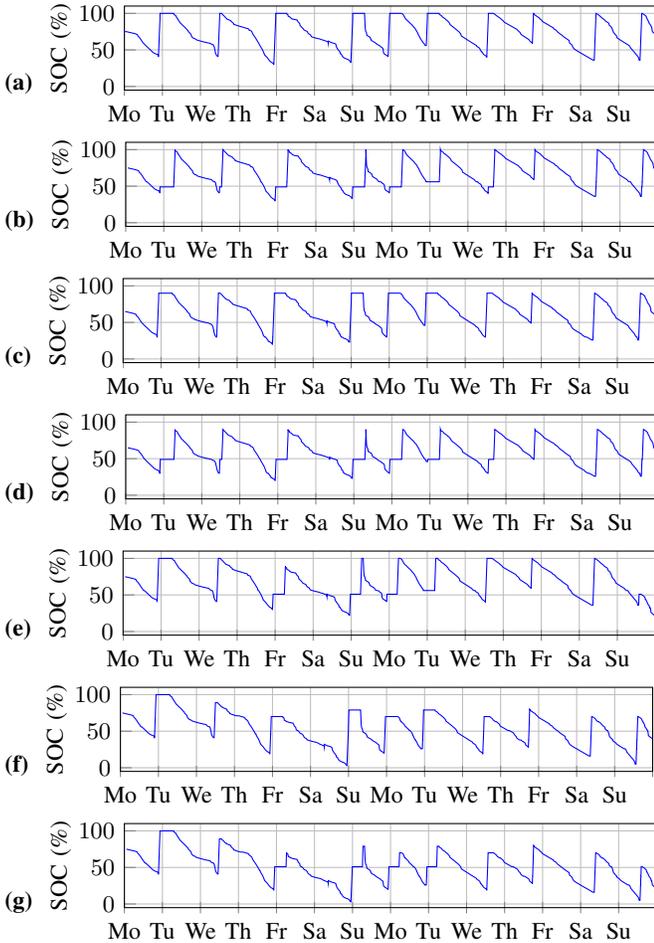


Fig. 10. Charging plot of User 4 with recorded data (a), perfect predictor for delayed charging (b), lowered $soC_{target} = 90\%$ (c) and a combination of both (d). As well as simple moving average prediction of delayed charging (e), predicted lowered soC_{target} (f) and a combination of both (g).

B. Smart Charging with Perfect Predictor

Next, we would like to quantify the separate as well as joint effects on aging of shifting the charging process in time and adjusting the target SOC. Towards this, we derive the battery aging for the unmanaged curve and compare it with three scenarios: Delaying the charging, lowering the target SOC and the combination of both. We first calculate the aging for the three scenarios by optimally delaying the charging with a perfect predictor. In Section V-C, we determine the aging for the same three scenarios but instead of optimal adjustments, we simulate the use of a predictor instead. We collect real user data using the Android application Battery Log [17] for a duration of two weeks. The average SOC per profile vary between 54.4% and 78.5%. The accumulated hours spent at 100% SOC during the recording period are between 16h and 120h.

Using these profiles, we simulate a smart charging behavior and determine the corresponding amount of battery aging using the aging model from [7] implemented in Matlab. The Matlab model takes the SOC profile and temperature data as an input. Remember that the model provides a bound on the cycle life and real aging is likely to be even faster than our numbers

Profiles	1	2	3	4	5	6	7	8	9
Original	2.6	2.5	3.3	4.2	3.7	3.1	5.6	3.3	4.6
Perfect Predictor for									
Delay charge time	3.4	3.6	6.0	5.2	4.3	5.1	6.3	3.8	5.2
$soC_{target} = 90\%$	3.8	3.6	4.8	6.0	5.4	4.6	8.1	4.8	6.7
Delay and soC_{target}	4.8	4.9	7.6	7.2	6.1	6.7	8.9	5.7	7.5
SMA Predictor for									
Delay charge time	3.2	3.0	5.2	4.3	4.7	3.8	6.2	4.0	6.0
soC_{target}	3.3	3.4	3.7	6.7	5.8	4.1	7.1	3.7	5.9
Delay and soC_{target}	4.4	4.4	5.8	7.2	6.7	6.4	7.9	4.5	8.3

TABLE I. SIMULATED CYCLE LIFE IN YEARS FOR RECORDED DATA, PERFECT PREDICTOR AND SIMPLE MOVING AVERAGE (SMA) PREDICTION OF DELAYED CHARGING, LOWERED TARGET SOC AND A COMBINATION OF BOTH. THE GRAPH ON THE RIGHT VISUALIZES THE AVERAGE CYCLE LIFE AND THE RANGE FROM THE LOWEST AND HIGHEST DATA POINT.

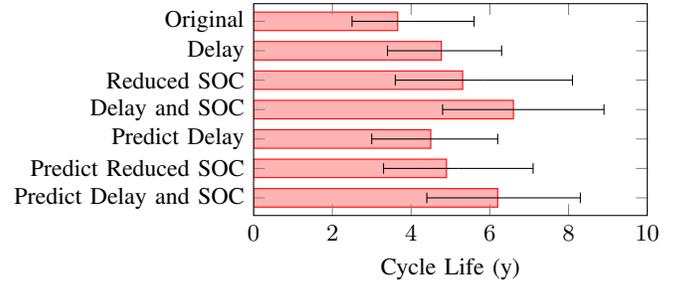


Fig. 11. Lifetime Comparison

suggest as has been explained in Section II. We use a perfect predictor, therefore the battery reaches full charge precisely at the moment when it is unplugged. In order to not leave the user with no charge in case he unexpectedly has to unplug the phone earlier, the battery is charged to a SOC of 50% immediately when plugged. After that charging is delayed.

As can be seen in Table I and Figure 11, considering the original unmanaged profile, most batteries last for about three to four years, with some exception that would last up to almost six years for normal charging behavior. If the batteries would be charged in a smart way, the expected cycle life would have an average above six years. Hence, delaying the charge time increases the cycle life, i.e., the duration until the battery capacity has faded to 80% of its initial value, by factor 1.1 (profiles 7,9) to 1.8 (profile 3). Average over all profiles is 1.3.

Next, we lower the target SOC to a fixed value of 90%. The effect on aging mitigation lies within a similar range as delaying the charging start time. The battery cycle life is increased by a factor of 1.5 on average for our examples.

Note that sometimes simulations of lowering the target SOC result in a full drain of the battery prior to the plug event in the recorded data set. Also lowering the target SOC potentially effects the user behavior, e.g., the user might decide to plug in his phone earlier. This kind of psychological effects can of course not be captured in the simulation. Delaying the charging process usually does not result in similarly negative simulation effects.

A combination of both, delay of the charging process and lower target SOC, yields the most benefit. Here the achieved increase in lifetime has a factor of 1.6 up to 2.2 compared to the battery cycle life without any SOC management. The average is 1.8. Figure 10 (a)-(d) show the recorded and simulated battery profiles for User 4.

C. Smart Charging with Predictor

Now the same comparison as in the previous section is done for the aa simple moving average predictor that estimates the delay time and the target SOC. The predictor shows good results performing only little worse than the results of the perfect predictor, see Table I. On average, the improvement factor, which is the predicted cycle life divided by the original cycle life, is decreased by 0.1 for each of the three prediction combinations (predict delay, predict target SOC, joint prediction). For the joint simple moving average prediction of delayed charging and decreased target SOC, the gain of lifetime over the original cycle life yields factor 1.7, while a factor of 1.8 was achieved with the perfect predictor results.

A problem that occurs when using a predictor is that the battery is sometimes drained down to 0% of SOC due to wrong prediction of the charging delay or wrong prediction of the target SOC and remains at its empty state for some time. This lowers the average SOC and therefore increases the cycle life, however it is not a desired behavior as the smartphone cannot be used during this time. Such happenings can partly be seen from the data provided in Table I in case the predicted cycle life for the delay predictor is larger than the simulated cycle life. Making the user recharge during the day is not a desired behavior of the predictor but infrequent occasions may be considered acceptable in terms of battery aging.

Further notice that the predicted target SOC is not allowed to be below 70% and hence lies within the range of 70% to 100%. Therefore in some cases, the target SOC predictor may perform better than the fixed value simulation because the predicted target SOC may be lower than the one fixed at 90%. In Figure 10(e)-(f), the charge plots for User4 are shown. To sum up, even the relatively simple moving average predictor helps to reduce aging. However, further means of improvement are the optimized choice of bin size and buffer terms. Also the SOC predictor should be replaced by more elaborate suggestions from literature which incorporate further environment data. Alternatives for the plug duration predictor are discussed in the following.

As explained earlier, a mayor drawback of the use of the predictor is that in case of wrong prediction and earlier plug off time the user will be left with an empty battery. We suggest two measures to deal with this drawback: Firstly, when plugged in, the phone should immediately charge to a medium SOC in order to provide a reserve amount of charge in case of an earlier unplug time at the cost of slightly less efficient aging mitigation. Secondly, the predicted unplug time should be displayed in a user interface to provide the user the opportunity to immediately charge the phone or to manually set an earlier unplug time.

D. Comparison of Predictors

In Section IV, we discuss three predictors for estimating the plug duration: A simple moving average predictor, an exponential moving average predictor and a probabilistic predictor. For evaluating these predictors, we use two data sets of two different users that contain data for several weeks. In future work, a further evaluation with bigger data sets containing more users is planned.

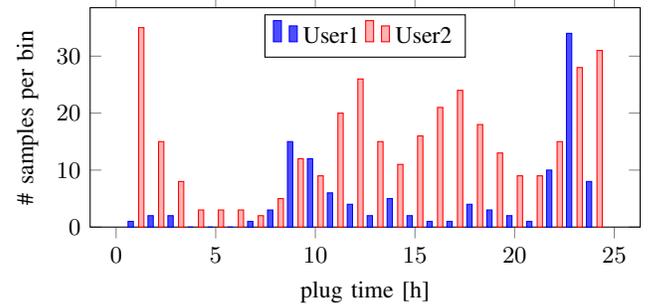


Fig. 12. Sample sizes per plug time bin for the two data sets. The two users favor different plug times.

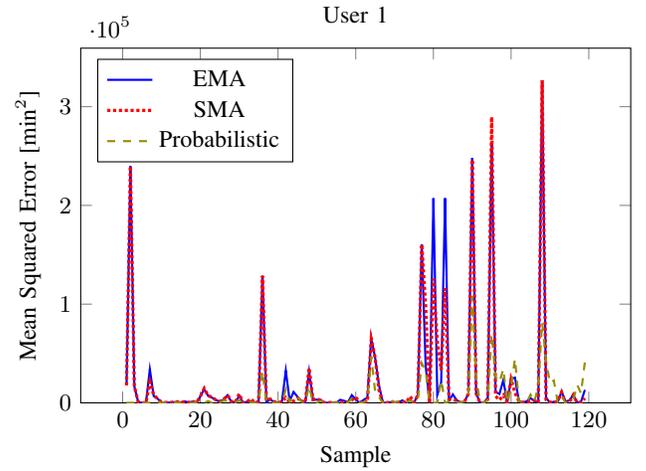


Fig. 13. Mean squared error per sample for User 1. The outliers are likely to represent empty bins. All predictors have similar difficulties with outliers.

As can be seen from Figure 12, the two users show distinct patterns. The sample sizes per bin (plug times rounded to hours of day) vary. All three predictors have difficulties with outliers, independent of the data set (user 1 or user 2), see Figures 13 and 14. The peaks of increased error are either real outliers or represent the first entry within a bin for which no previous knowledge can be used.

A comparison of the predictors is done by calculating the mean squared error per sample and computing the average of all errors. The results are shown in Figure 15. It can be seen that the exponential moving average predictor is slightly better than the one using a simple moving average. The probabilistic predictor is more accurate for both data sets.

Figure 16 shows a comparison of mean squared errors within a single bin. Outliers occur also at later points in time and all three predictors have similar difficulties in dealing with them.

In summary, the advantage of the probabilistic predictor is that it is more accurate. However, the simple moving average predictor and the exponential moving average predictor are easier to implement. Using a probabilistic predictor can bring the efficiency of aging mitigation from the previous experiment even closer to their optimum.

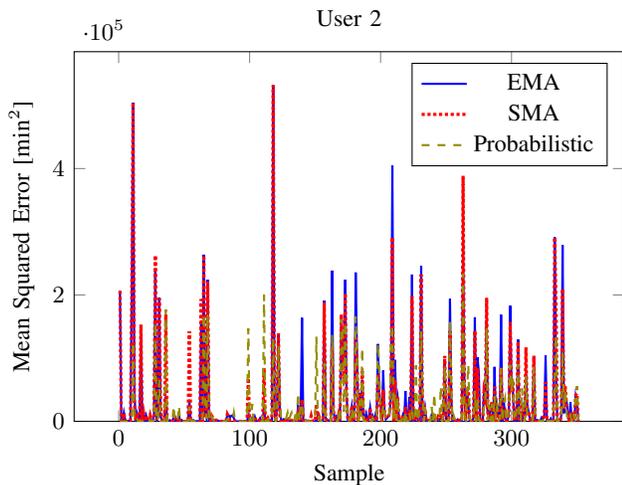


Fig. 14. Mean squared error per sample for User 2. The outliers are likely to represent empty bins. All predictors have similar difficulties with outliers.

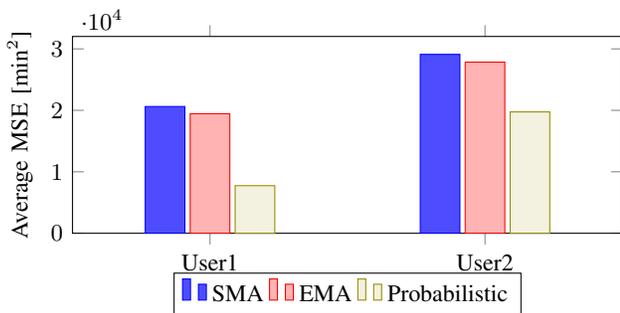


Fig. 15. Comparison of the mean squared error over all samples using the simple moving average predictor, the exponential moving average predictor and the probabilistic predictor.

VI. CONCLUSION

The concept of an intelligent smartphone battery charger has been presented. The charger makes use of typical smartphone usage profiles, i.e., most smartphones are charged over night, delays charging and hence lowers the time spent at a high SOC, which is one of the factors that increase battery degradation. We presented Smart² a context aware charging device that determines the possible charging delay by either manually applied setting, reading an alarm or using a predictor. Simulation of the battery aging showed, that the battery cycle life could be extended by a factor of 1.8 on average when

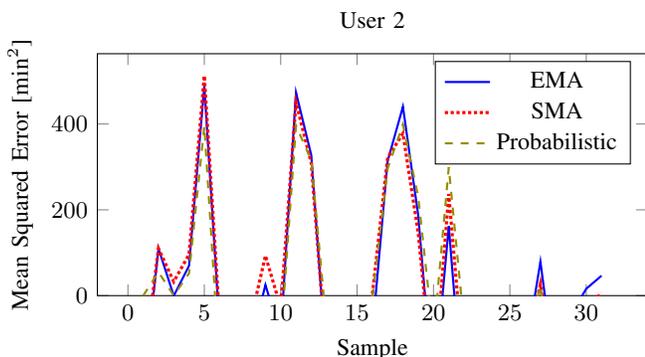


Fig. 16. Comparison of errors within a single bin. Outliers occur also at later points in time and all three predictors have difficulties.

delaying charging and lowering the target SOC. A probabilistic predictor for plug duration estimation shows the most accurate results compared to a simple moving average and exponential moving average predictor.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions that help improve the quality of this paper. This work is partially supported by the Bavarian Ministry of Economic Affairs and Media, Energy and Technology as part of the *EEBatt* project and by the Federal Ministry of Education and Research of Germany (BMBF) as part of the *HE2mT* project.

REFERENCES

- [1] Lenovo, "Power manager," <http://support.lenovo.com>.
- [2] S. S. Choi and H. S. Lim, "Factors that affect cycle-life and possible degradation mechanisms of a li-ion cell based on licoo₂," *Journal of Power Sources*, vol. 111, no. 1, pp. 130–136, 2002.
- [3] M. Ecker, J. B. Gerschler, J. Vogel, S. Käbitz, F. Hust, P. Dechent, and D. U. Sauer, "Development of a lifetime prediction model for lithium-ion batteries based on extended accelerated aging test data," *Journal of Power Sources*, vol. 215, pp. 248–257, 2012.
- [4] K. Lim, A. Laclner, P. Braatz, W. Smith Jr, J. Margerum, and H. Lim, "Accelerated lifetime studies on commercial lithium-ion battery cells," in *Proceedings of the Symposium on Batteries for Portable Applications and Electrical Vehicles*, vol. 18. The Electrochemical Society, 1997, p. 470.
- [5] K. Takei, K. Kumai, Y. Kobayashi, H. Miyashiro, N. Terada, T. Iwahori, and T. Tanaka, "Cycle life estimation of lithium secondary battery by extrapolation method and accelerated aging test," *Journal of Power Sources*, vol. 97, pp. 697–701, 2001.
- [6] A. Barré, B. Deguilhem, S. Grolleau, M. Gérard, F. Suard, and D. Riu, "A review on lithium-ion battery ageing mechanisms and estimations for automotive applications," *Journal of Power Sources*, vol. 241, pp. 680–689, 2013.
- [7] A. Millner, "Modeling lithium ion battery degradation in electric vehicles," in *Innovative Technologies for an Efficient and Reliable Electricity Supply (CITRES), 2010 IEEE Conference on*. IEEE, 2010, pp. 349–356.
- [8] T. Reddy, *Linden's Handbook of Batteries, 4th Edition*. McGraw-Hill Education, 2010.
- [9] N. Mattisson, P. Ungar, T. Greening, and J. Koller, "Tracking fading battery capacity in a plugged-in portable electronic device," Sep. 11 2014, US Patent App. 13/791,328. [Online]. Available: <http://www.google.com/patents/US20140257724>
- [10] Samsung, "Battery life extender," www.samsung.com.
- [11] Cheetah Mobile Inc., "Battery doctor (battery saver) 4.14," <https://play.google.com>.
- [12] E. A. Oliver and S. Keshav, "An empirical approach to smartphone energy level prediction," in *Proceedings of the 13th international conference on Ubiquitous computing*. ACM, 2011, pp. 345–354.
- [13] D. Ferreira, A. K. Dey, and V. Kostakos, "Understanding human-smartphone concerns: a study of battery life," in *Pervasive Computing*. Springer, 2011, pp. 19–33.
- [14] J.-M. Kang, S.-s. Seo, and J.-K. Hong, "Usage pattern analysis of smartphones," in *Network Operations and Management Symposium (APNOMS), 2011 13th Asia-Pacific*. IEEE, 2011, pp. 1–8.
- [15] N. Matsumura, N. Otani, and K. Hamaji, "Intelligent battery charging rate management," Oct. 1 2009, US Patent App. 12/059,967. [Online]. Available: <https://www.google.com/patents/US20090243549>
- [16] T. M. T. Do and D. Gatica-Perez, "Contextual conditional models for smartphone-based human mobility prediction," in *Proceedings of the 2012 ACM conference on ubiquitous computing*. ACM, 2012, pp. 163–172.
- [17] T.-r. Hwang, "Battery log, version 2.0.3," <https://play.google.com>, 2013.