Understanding the Impact of Laptop Power Saving Options on User Satisfaction Using Physiological Sensors

Matthew Schuchhardt
Northwestern University
matt.schuchhardt@
u.northwestern.edu

Ben Scholbrock
Northwestern University
benjamin.scholbrock@
eecs.northwestern.edu

Utku Pamuksuz
Northwestern University
utkupamuksuz2011@
u.northwestern.edu

Gokhan Memik
Northwestern University
memik@eecs.northwestern.edu

Peter Dinda
Northwestern University
pdinda@northwestern.edu

Robert P. Dick
University of Michigan
dickrp@eecs.umich.edu

ABSTRACT

Several techniques are available to save power consumption in laptop computers. However, their effect on user satisfaction has not been well studied. We analyze how user satisfaction is affected by these techniques and show that, within a fixed power budget, some techniques cause more dissatisfaction than others. Second, we study the use of physiological sensors and show that the sensor readings are stable across times when no technique is applied, whereas they show statistically significant changes when power-saving techniques are employed. Finally, we demonstrate a prediction mechanism using these sensors that predicts user satisfaction with over 80% accuracy.

Categories and Subject Descriptors: C.5.3 [**Computer System Implementation**]: Microcomputers—*Personal computers*; H.1.2 [**Models and Principles**]: User/Machine Systems—*Human Factors*; J.4 [**Computer Applications**]: Social and Behavioral Sciences—*Psychology*;

Keywords: Affective computing, human factors, user-aware computing, power saving techniques, empathic systems and architectures

1. INTRODUCTION

The primary goal of a computer system is to satisfy the end-user. Traditionally, computer designers measure the effect of their optimizations using metrics expected to correlate with user satisfaction (e.g., tasks per time, instructions per cycle, packets per time). However, previous work has shown that a) providing higher performance levels does not necessarily improve the user satisfaction and b) the level of performance necessary to satisfy a particular user varies significantly among users (e.g., [7, 16].) With no knowledge of user satisfaction regarding machine performance, optimizations exercising trade-offs between performance and power consumption will be designed to satisfy the nonexistent "typical" user with a "one-size-fits-all" approach. This will often lead to sub-optimal configuration choices, which could be improved by understanding individual user satisfaction.

Computer optimizations are typically performed from the perspective of a single component or system block. For example, Dynamic Voltage and Frequency Scaling (DVFS) [12], which reduces power at the cost of CPU performance for a period of time, is controlled independently from screen brightness, which may be adjusted to conserve power. User satisfaction is a function of both CPU performance and screen brightness, but current control schemes do not consider combined effects. This paper aims to lay the groundwork for a new method of measuring computer performance that would allow the integration of such effects and their derivation from easy-to-acquire user measurements. This new method of evaluating computer systems describes how well a computer performs, not only in traditional terms of raw computational capabilities or power efficiency, but from the perspective of end-user satisfaction.

In this study, we first construct a linear regression-based power model to estimate power consumption of a computer system using software-visible metrics. We leverage this model to estimate the impact of various power-saving techniques on total system power. We then perform user studies, activating combinations of these power-saving techniques during machine use. We request that users in the study verbally report dissatisfaction with the computer system. We also employ several physiological and behavioral sensors to monitor the user. These sensors measure the galvanic skin response (GSR), pupil dilation and movement, the amount of wrist movement, the pressure applied to the keyboard during key presses, and the frequency of key presses. Finally, we analyze the sensor readings based on the reported dissatisfaction.

We next demonstrate the impact of power-saving techniques on user satisfaction. We find that the frequency of user-reported dissatisfaction varies by power-saving technique, even though the same reduction in power consumption is achieved. We also find that changes in user satisfaction

This work was supported by the United States National Science Foundation through grants CCF-0916746, CCF-0747201, CNS-0720691, and DGE-0948017.

ISLPED'12, July 30–August 1, 2012, Redondo Beach, CA, USA.

are reflected in measurable physiological responses. In situations where saving power does not affect user satisfaction, physiological sensor data remains constant. However, when a power-saving technique affects reported user satisfaction, sensor data changes. These findings motivate satisfaction-oriented power/performance optimizations in light of constrained power requirements on personal computers.

We also create a prediction system that predicts whether a power-saving technique changes the user satisfaction. Our system achieves over 80% accuracy. Finally, we perform a supplementary study which demonstrates that our prediction system performs equally well whether users verbally indicate dissatisfaction or not.

The rest of the paper is organized as follows. In Section 2, we present our experimental setup. In Section 3, we present our analysis of the sensor data, as well as the results of our prediction system. In Section 4, we discuss related work in the field, and we conclude in Section 5.

2. EXPERIMENTAL SETUP

We now describe the environment in which we carried out our studies, and the studies themselves.

2.1 Power Model

Before correlating user satisfaction with power savings techniques, we first develop a model for accurately estimating the power saved by each technique. Techniques may then be parameterized to save roughly equivalent levels of system power.

Our target platform is an IBM Thinkpad T61 with a 2.2 GHz Intel Core 2 Duo T7500 processor and 2 GB DDR2 SDRAM running Microsoft Windows XP. This computer is also the platform used during user studies.

The predictors we target in the power model follow: RAM bus transactions, disk accesses, processor utilization percent, screen brightness, screen on/off state, and WiFi on/off state. We also directly measure the power being used by the system; this is our dependent variable.

One of the most common techniques for saving power is reducing CPU frequency. However, changing the CPU frequency substantially alters the amount of power that the other predictor variables would contribute to the power model. Hence, we generate separate power models for the system running at 1.6 GHz and 2.2 GHz.

To create the power model, we run a variety of benchmarks, each which primarily stress one component on the system. We then create a linear regression model from this data.

We also perform four user-driven verification workloads, each which stress a different aspect of the computer. Using these workloads, we compare the actual power readings with the power levels predicted by our model. Averaged across all readings and verification workloads, our mean absolute error is less than 2%.

2.2 Physiological Sensors

In this work, we utilize a GSR sensor, a 3-axis accelerometer, an eyetracker, and piezoresistive force sensors to measure the physiological and behavioral effects of different power saving techniques. Additionally, we also record keyboard button presses via software hooks.

The GSR sensor and the accelerometer are both mounted on a wrist strap, the signals from which are fed into an Arduino-based microcontroller. The force sensors are attached to the arrow keys on our keyboard, and their signals are also fed into an Arduino device. The eyetracker is made by MobileEye [2], and consists of a head-mounted camera which is controlled by a dedicated laptop.

From those sensors and logging software, we extract seven different metrics:

- AccelMag: sum of squares of X, Y, and Z accelerometer axes
- DeltaGSR: change in GSR value since the last sensor reading
- Keypress: time since the last keyboard button press
- MaxForce: largest current value from the force sensors
- NormalMaxForce: same as MaxForce, but normalized to each key's highest force reading
- PupilMovement: change in position of the pupil since the last pupil reading
- PupilRadius: the radius of the pupil.

We tried to match the metric to what we expected to be the most interesting behavior of the sensor. For instance, GSR sensors tend to exhibit a low-frequency change in sensor values on top of the more interesting high-frequency signal, so we only look at the change in GSR since the last sensor reading to reduce the low-frequency effects.

2.3 User Studies

In this section, we describe the techniques that are used to reduce power consumption. To simplify the discussion, we call these techniques 'annoyance events' because they are applied for short durations during our user studies, and they may cause user dissatisfaction/annoyance.

From the results of the power model, we decided to focus on the CPU utilization (CPUUtil), CPU frequency (CPUFreq), and screen brightness (ScreenDim and ScreenGrad). CPUUtil limits the scheduling time of the target application without changing the frequency. CPUFreq reduces the frequency of the system's processor. ScreenDim reduces the brightness of the screen to save power. ScreenGrad does the same, but does so gradually over 10 seconds. The rest of the predictors either would completely disrupt the experiment (e.g., WiFi on/off), or are not easily throttleable (e.g., RAM accesses).

We select a variety of different annoyance events which total up to 4, 8, or 12 Watts of power savings for our target system (Table 1). If multiple annoyances are in effect at the same time, the power savings is split between them (e.g., if 3 annoyances are active for 4 Watts of total savings, each annoyance saves 1.33 Watts). We also include 'pseudo annoyances', where no annoyance is initiated by our system. We initiate each annoyance event twice during user studies. Every annoyance event is active for 30 seconds after which the computer reverts back to the original state for 25-40 seconds. The order of annoyance events are randomized to eliminate any possibility of biasing.

The workload that each user in this study performs is a racing game called Motocross Madness 2. We use a racing game as our workload because it is relatively CPU-bound, and requires consistent attention to play. Future work will likely explore additional workloads. The laptop typically consumes 40-41 Watts of power while running the game without any active annoyance events.

Table 1: Total number of times (out of 40 instances) that the annoyance event combination was indicated as being annoying. If multiple events are active, each contributes an equal amount to the total amount of power saved.

Power Saved	Annoyance Events Active	# Times User Indicated Annoyance
0W	None	1
4W	CPUFreq	1
4W	CPUUtil	15
4W	ScreenDim, CPUFreq, CPUUtil	15
4W	ScreenDim	30
4W	ScreenGrad	31
8W	CPUFreq	0
8W	CPUFreq, CPUUtil	18
8W	ScreenDim, CPUFreq, CPUUtil	27
8W	ScreenDim, CPUUtil	30
8W	ScreenDim, CPUFreq	31
8W	CPUUtil	31
12W	ScreenDim, CPUFreq, CPUUtil	33

As the users are playing the game, we request that they verbally notify us when they become dissatisfied. Other methods of reporting dissatisfaction are possible, such as using a physical button or periodically rating machine performance on a scale of values. Simple verbal reports require minimal user action, which is desirable since gameplay is continuous through annoyance events.

We ran this study on 20 users. The studies are conducted in a well-lit room with overhead fluorescent lighting and drawn shades, to minimize the effect that the time of day and screen brightness have on the user's pupil dilation.

The advertisements for the study were placed around various parts of Northwestern's campus, and the respondents were all university students. Of the 20 users, there were 12 males and 8 females. 10 were undergraduate students, and 10 were graduate students.

3. RESULTS

We now examine the results of our study in order to derive a model for predicting user satisfaction from the physiological sensors.

3.1 Annoyance Event Analysis

In this section, we examine how users react to the annoyance events. As the users are playing the game, we request that they verbally notify us when they are dissatisfied. We then note how many times users indicated annoyance for each annoyance event. In Table 1 we list how frequently the users indicated annoyance.

One interesting piece of information to note from this table is that different users have very different reactions to the CPUFreq and CPUUtil annoyances. Users very rarely express annoyance when only the CPU frequency is reduced, even though the corresponding CPU utilization reduction saves the same amount of power. Reducing the frequency not only reduces the number of cycles the processor can complete in a given amount of time, but also allows the processor to reduce its voltage, which may explain this discrepancy in indicated annoyance events.

It's also interesting how users readily notice screen brightness changes. Users frequently report annoyance with the system when the screen is dimmed. One notable exception is the 4-Watt ScreenDim + CPUFreq + CPUUtil annoyance: even though there is a ScreenDim annoyance active in that set, the screen isn't dimmed for this annoyance as much as other cases. This seems to suggest that there is a particular threshold for when users indicate annoyance with a system component. This result alone shows the importance of understanding user satisfaction with power saving techniques: a naïve optimization would dim the screen (with or without a change in CPU frequency) to save power. However, our results show that for this target application, the best method is to partially reduce the CPU frequency, and partially dim the screen . A system that can understand user satisfaction with different optimizations will choose this option over the rest.

3.2 Statistical Sensor Value Analysis

To analyze the data collected from the biometric sensors, we first note that there are two distinct sets of data that relates to every individual annoyance event: data during the non-annoyance phase preceding the introduction of the annoyance event, and data during the annoyance phase.

Intuitively, we would expect the sensors to either substantially change before and during an annoyance event, or to remain approximately the same. We hypothesize that events which the users either don't notice or don't care about will not substantially change the sensor metrics. Conversely, we expect that annoyance events which the user perceives as annoying would cause a measurable change between the two sets of sensor readings.

Since the data before and during each annoyance event consists of hundreds of sensor readings, we compute the standard deviation, mean, and median of each of the metrics before and during each annoyance event.

We do not know for certain how the users' biometric data will change for each annoyance event. There's a good chance that we would miss some sensor responses if we only looked at one static span of data for each event; some sensors may have a delayed reaction, or the change may happen over a larger or smaller window of time. To improve our likelihood of capturing the timespan that is most indicative of annoyance for each sensor, we consider multiple data windows and offsets to see how the different sensors respond. We look at window sizes of 10, 15, 20, and 25 seconds for multiple offsets from the start of the annoyance event. Please note that the larger windows do not have as much room to move within the 25-second window of non-annoyance, and hence, we do not analyze as many offsets for those windows.

Since we are only concerned with detecting changes in metrics when the user is actually annoyed, we first only consider event instances which are indicated by the user as unsatisfactory. For each event, we have data from seven sensor metrics (Section 2.2). For each metric, we collect data from the various window/offset combinations, as described above. On each of those, we then calculate a standard deviation, mean, and median. Each of these data frames consists of the mean, median and standard deviation from the window before and during one annoyance event instance.

All of those similar data frames are then compared from before and during the annoyance using a paired 2-tailed t-test. 2-tailed t-tests with a relevant p-value ($p<0.1$ in this

study) suggest a statistical difference between two sets of samples.

Due to space constraints, we do not present graphs for every sensor, but the results for the selected set of sensor metrics are shown in Figure 2.

One interesting thing to note in these graphs is that the metrics and windows which give statistically significant t-tests differences vary for each sensor. Some of the sensors respond best with a larger window, while others seem to react best using a smaller one. Additionally, it seems that lower offsets perform better than larger offsets, which would suggest that these sensors reflect external stimuli fairly quickly. Many of the sensors show promise for our future studies, as we can statistically show the difference between the data before and during an annoyance event with a high degree of confidence for many of the windows and offsets.

3.3 Equivalence Analysis

A false positive occurs whenever the user annoyance prediction system indicates that the user is annoyed when they actually are not. A theoretical system could be devised which always detected a difference in some sensor metric. This system would always indicate that the user is annoyed, and so would be 100% effective in detecting annoyance, but would never detect non-annoyance. Thus, we have to prove that our sensor metrics are not only good at detecting differences in sensor data due to annoyance events, but we also must show that they perform reasonably well at detecting non-annoyance. In other words, the system needs both a low false negative rate and a low false positive rate.

'TOST' [15], or 'two one-sided tests', is one way of demonstrating equivalence between two sets of data. TOST works by running two paired one-tailed t-tests with some offset ϵ.

When proving dissimilarity (Section 3.2), we only analyze data sets that the user deems annoying. Conversely, for this equivalence analysis, we only consider data sets in which the user did not indicate annoyance. From the previous t-test analysis, we take the sets of data with the 10 lowest p-values, but at least one and no more than two from each sensor. We then determine the smallest possible ϵ value that is required to achieve a confidence of 90% that the two sets of data are equivalent. In Table 3, we present the results of our equivalence analysis. We represent the relative size of the offset by dividing ϵ by the average of the means.

The required ϵ values for all of the metrics we analyze are relatively small, compared to the rest of the data. However, a few of the t-tests (PupilMove, one of the AccelMag) are statistically significant: this would suggest that those particular sensor/window/offset combinations may not be the best choice for an eventual prediction system. The rest of the metrics behave well, however.

3.4 User Annoyance Prediction

In this section, we determine how accurately we can actually predict user annoyance.

To perform this analysis, we employ standard data mining techniques using labeled instances of data. We employ the Weka [1] data mining suite to assist in handling the data and building the models. In this analysis, we label any event where the user expressed annoyance as 'True', and instances where they did not as 'False'.

We then create a large number of attributes on those labeled instances. To generate our attributes, we take data from 0–2 sec, 2–4 sec, 4–6 sec, 6–8 sec, 0–5 sec, 5–10 sec, 10–15 sec, 15–20 sec, 0–10 sec, and 10–20 sec, from before and from after each annoyance event. For each of those time periods, we find both the means and standard deviations. Additionally, we take the difference from before and after. This gives us a total of 60 attributes. We run this attribute generation method for each of our 7 sensor metrics.

If we were to simply use all of those attributes in a data mining algorithm, the most useful attributes would get lost in the noise of the rest of the attributes. Thus, to improve our eventual model's accuracy, we employ CFS [8], a feature selection algorithm, which reduces the number of attributes we end up using in the actual model.

After feature selection, we then run the actual data mining algorithm. To build our model, we use logistic regression [18], which is a type of generalized linear model frequently used on binary labeled data. For all of our analyses, we use 10-fold cross-validation.

We generate a different model for each user. We also include a 'baseline' set of data. This baseline was generated by using ZeroR, which is a naïve algorithm that always chooses the most common label (True or False). The results are presented in Figure 4.

As the graph shows, all of the individual sensors have medians better than the ZeroR baseline case. However, when all of the sensors are combined in the model, the prediction accuracy is significantly better. We would like to note that some users did not indicate any annoyance, which gives ZeroR a slight advantage (since it is always correct for those users), but we did not discard the data; we assume that they could not tell a difference in performance.

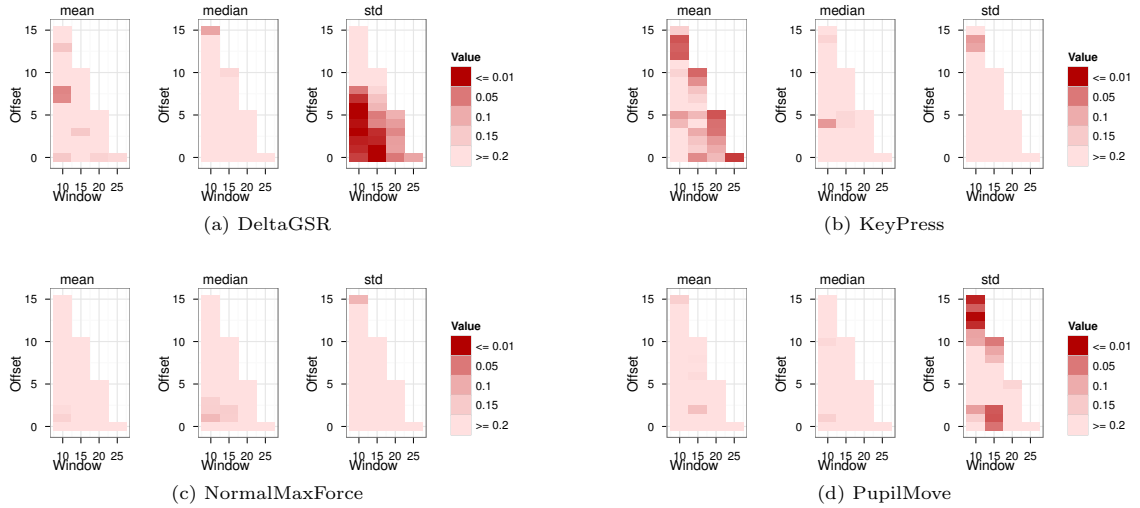
We must also know how accurate the system is at individually predicting 'True' and 'False' instances, as a system which has a very high 'True' prediction accuracy, but very low 'False' prediction (or vice-versa) is not as useful. Using the combined sensors model, our system predicts 'True' accurately 212/263=80.6% of the time, and 'False' accurately 203/257=79.0% of the time (false positive=19.4%, false negative=21%).

3.5 Results Verification

In our original experiment, the users verbally notified us when they were annoyed. However, their verbal notification could alter the readings on our biometric sensors. To address this possibility, we run a small supplementary user study on five users to verify that we can detect when the user is annoyed without a verbal indication. The primary difference in this study is that we do not have the users notify us when they are annoyed; we simply have them play the game as the performance changes.

Since we already have determined which annoyances the user perceives as annoying and non-annoying, we only use the top 5 most annoying events, as reported by users in our original study (see Table 1), as well as 5 'pseudo' annoyance events where there is no change in performance. We run each of those annoyance events two times. We increase the time of non-annoyance to 55–65 seconds, with 30 seconds of annoyance.

We then run a similar analysis to our original study, but instead of using the user-reported annoyances to label the annoyance events as 'True' or 'False', we label all of the intentionally annoying events as 'True', and all of the pseudo annoyance events as 'False'. The graphs in Figure 5 are the



(a) DeltaGSR (b) KeyPress

(c) NormalMaxForce (d) PupilMove

Figure 2: Windowed t-test analysis for selected sensors. Low p-values, represented by dark areas, indicate metric/offset/window combinations where sensor data differs before and during annoyance events. Only user-indicated annoyances are analyzed.

Table 3: Confidence intervals for equivalence test analysis, using data from the non-annoying events. Only includes selected metrics which performed well in the previous t-test analysis.

Sensor	Metric	Window	Offset	Average Before Annoyance	Average During Annoyance	Required ϵ for 90% Confidence	ϵ / avg of means	t-test P-value
AccelMag	std	15	3	3.0123	3.2871	0.7048	0.2238	0.1853
AccelMag	std	20	0	3.0288	3.4610	0.9376	0.2890	0.0949
DeltaGSR	std	10	5	38.9661	39.3882	6.8749	0.1755	0.7906
DeltaGSR	std	10	6	39.0690	39.4917	6.9296	0.1764	0.7889
Keypress	mean	25	0	0.1968	0.1878	0.0249	0.1297	0.2565
MaxForce	median	15	1	142.7626	138.3891	20.1704	0.1435	0.5764
NormalMaxForce	std	10	15	0.1468	0.1431	0.0130	0.0896	0.3681
PupilMove	std	10	13	7.6988	6.7270	1.7515	0.2428	0.0267
PupilMove	std	10	15	7.5278	6.8014	1.4574	0.2034	0.0717
PupilRadius	std	15	2	25.7464	25.2551	2.1452	0.0841	0.5001

result of our prediction analysis. The baseline for this graph is exactly 50%: there were exactly 10 annoying events and 10 non-annoying events for each user, so ZeroR would always be correct 50% of the time.

As the graph shows, all sensors perform much better than the baseline, and again, the combined sensor model performs the best, with an average accuracy of 79%. This shows that our system works, regardless of whether there are any verbal user indications of annoyance or not.

Using the combined sensors model, we are able to predict 'True' instances 38/50=76% of the time, and 'False' instances 41/50=82% of the time (false positive=24%, false negative=18%).

4. RELATED WORK

Mouse and keyboard actions alone are often used to determine user state in human-computer-interaction scenarios. Fox et al. [6] try to determine user satisfaction based on these in the context of web searching. Macaulay [10] uses mouse click frequency as an indicator for user anxiety.

Endo et al. [5] explore using OS event handling latency, rather than computational throughput, as a measure for performance. Video frame rates and application response times are also used as a user-perceivable performance metric for the system, as a proxy for user satisfaction [11]. Gupta et al studied the effects of resource scheduling decisions on directly reported user satisfaction [7]. Even further studies investigate leveraging biometric data to adjust microprocessor power level, to account for user satisfaction [17]. However, none of these studies analyzed the impact of different power saving methods.

Picard et al. [13], proposes a method of using a variety of sensors to reduce the negative impact that interruptive technologies have on users. That same group also proposed a method of predicting user frustration using sensors in a learning environment (Kapoor et al. [9]), but use a different set of sensors, and do not consider power saving methods in this case.

Considerable work has been put into systems which use sensors to acquire affect data from users in order to enable computers to perform certain user-facing tasks. For example, the educational field has looked into using sensors to improve tutoring systems. Cooper et al. [4] describes a tutoring system which uses multiple sensors to improve the relevancy

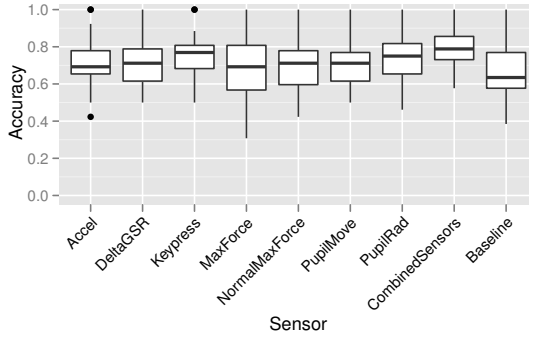


Figure 4: Model prediction accuracy using individualized user models. We include results for 7 individual sensors, a combined model which uses all 7 sensors, and a baseline prediction accuracy. The box's bottom and top are the 25th and 75th percentile, the middle line is the median, and the whiskers go to the highest and lowest points within 1.5 IQR of the higher and lower quartiles. Dots represent outliers beyond 1.5 IQR.

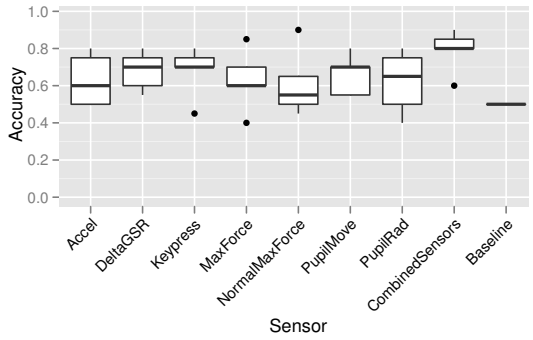


Figure 5: This boxplot is similar to the plot in Figure 4, but the users don't verbally indicate if they are annoyed. Data is labeled based on if the annoyances were typically annoying or not.

of questions on a standardized test studying platform by maximizing user engagement. Burleson describes a system [3] where a virtual agent helps guide the user through the Towers of Hanoi problem; the agent uses sensors to try to empathize with the user's current emotional state. Predinger and Ishizuka describe another agent-based interface that empathizes with the user through the use of physiological sensors [14]. However, these studies do not consider computer system performance or power saving methods.

5. CONCLUSION

We have argued for measuring and understanding end-user satisfaction using physiological sensors, and supported our argument through a user study whose data allow us to correlate voiced satisfaction, biometric information, and intentionally-introduced power-savings events. We found that different power-savings techniques reduce user satisfaction to different degrees, and that saving a given amount of power using a combination of techniques is generally preferable to using a single technique alone.

We have also shown that it is possible to accurately measure changes in user satisfaction via various physiological metrics. When the performance is constant, biometric readings remain constant, while when performance varies, biometric readings also vary. This correlation can be used to predict user satisfaction from the metrics, and we showed how to do so with 80% accuracy through the use of per-user models developed through data-mining.

In a design environment where power and performance trade-offs directly impact user experience, understanding the relationship between user satisfaction and power-saving techniques is critical. This paper highlights some of these effects, and indicates that biometric sensors are a powerful way to acquire feedback about the individual end-user without requiring any effort on his part.

References

[1] Weka 3 - data mining with open source machine learning software in java. http://www.cs.waikato.ac.nz/ml/weka/.

[2] ASL. Applied science laboratories - mobile eye: Lightweight tetherless eye tracking, 2008.

[3] W. Burleson. Advancing a multimodal real-time affective sensing research platform. *New Perspectives on Affect and Learning Technologies*, pages 97–112, 2011.

[4] D. Cooper, I. Arroyo, and B. Woolf. Actionable affective processing for automatic tutor interventions. *New Perspectives on Affect and Learning Technologies*, pages 127–140, 2011.

[5] Y. Endo and M. I. Seltzer. Using latency to evaluate interactive system performance. In *Proc. of the Intl. Conf. on Measurements and Modeling of Computer Systems*, 2000.

[6] S. Fox, K. Karnawat, M. Mydland, S. Dumais, and T. White. Evaluating implicit measures to improve web search. *ACM Trans. Inf. Syst.*, 23:147–168, April 2005.

[7] A. Gupta, B. Lin, and P. A. Dinda. Measuring and understanding user comfort with resource borrowing. In *Proceedings of the 13th IEEE International Symposium on High Performance Distributed Computing (HPDC 2004)*, June 2004.

[8] M. Hall. *Correlation-based feature selection for machine learning*. PhD thesis, The University of Waikato, 1999.

[9] A. Kapoor, W. Burleson, and R. Picard. Automatic prediction of frustration. *International Journal of Human-Computer Studies*, 65(8):724–736, 2007.

[10] M. Macaulay. The speed of mouse-click as a measure of anxiety during human-computer interaction. *Behaviour and information Technology*, 23(6):427–433, 2004.

[11] A. Mallik, J. Cosgrove, R. Dick, G. Memik, and P. Dinda. PICSEL: Measuring user-percieved performance to control dynamic frequency scaling. In *Proceedings of the Intl. Conference on Architectural Support for Programming Languages and Operating Systems*, March 2008.

[12] Microsoft. Windows native processor performance control. In *Windows Platform Design Notes*, November 2002.

[13] R. Picard and K. Liu. Relative subjective count and assessment of interruptive technologies applied to mobile monitoring of stress. *International Journal of Human-Computer Studies*, 65(4):361–375, 2007.

[14] H. Prendinger and M. Ishizuka. The empathic companion: A character-based interface that addresses users'affective states. *Applied Artificial Intelligence*, 19(3-4):267–286, 2005.

[15] D. Schuirmann. A comparison of the two one-sided tests procedure and the power approach for assessing the equivalence of average bioavailability. *Journal of Pharmacokinetics and Pharmacodynamics*, 15(6):657–680, 1987.

[16] A. Shye, B. Ozisikyilmaz, A. Mallik, G. Memik, P. A. Dinda, R. P. Dick, and A. N. Choudhary. Learning and leveraging the relationship between architecture-level measurements and individual user satisfaction. In *Proceedings of the Intl. Symposium on Computer Architecture*, June 2008.

[17] A. Shye, Y. Pan, B. Scholbrock, J. S. Miller, G. Memik, P. A. Dinda, and R. P. Dick. Power to the people: Leveraging human physiological traits to control microprocessor frequency. In *Proceedings of the Intl. Symposium on Microarchitecture*, December 2008.

[18] I. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2005.