# A Generic Framework for Collecting and Mining Client Paradata for Web Applications

Natheer Khasawneh
Department of Software Engineering, Jordan University of Science and Technology, Irbid 22110, Jordan
Email: natheer@just.edu.jo

Rami Al-Salman
SFB/TR 8 Spatial Cognition, University of Bremen, Bremen, Germany
Email: rami@informatik.uni-bremen.de

Ahmad T. Al-Hammouri
Department of Network Engineering and Security, Jordan University of Science and Technology, Irbid 22110, Jordan
Email: hammouri@just.edu.jo

Stefan Conrad
Institute of Computer Science, Heinrich Heine University, Dusseldorf, Germany
Email: conrad@cs.uni-duesseldorf.de

***Abstract*— In this paper, we propose a general framework to track and collect user interactions with dynamic webpages. Using the AJAX, PHP, and MySQL technologies, we implement and realize the client-side-scripting framework to collect client paradata in a seamlessly manner. Being stored in a persistent storage at the server, the data were then structured and analyzed to understand the user behavior. We exploited the framework by applying it to two online systems: E-Survey and E-Commerce web applications. In the E-Survey system, we collected student interactions while filling in an online feedback form. We then used the collected data to model the user behavior. With the resultant model, we can infer whether a student is mindful and conscious while answering the feedback questions. In the E-Commerce system, we collected user interactions with a products page. Using a generated classifier, we can predict a user selection based on his or her navigation pattern on the page.**

***Index Terms*—Client paradata, client-side scripting, online forms, Web usage mining, mindfulness, consciousness, user behavior prediction.**

## I. Introduction

Web Usage Mining (WUM) is concerned with analyzing user interactions with Web applications [1]. The main goal of the analysis is to first understand the Web-client behavior and preferences, and to then serve the client in a better way. Many websites use different WUM techniques to better serve their clients [1]. WUM techniques are used to generate personal recommendations by computing similarities between a given client preferences and other clients'. In general, WUM consists of three steps [1]. First, the preferences of a group of clients are collected, recorded, and clustered into different classes based on similarity metrics. Second, the group of clients whose preferences are similar to a target client is selected. Finally, recommendations to the target client are made based on this chosen group [2]. Major online stores, such as Amazon, use WUM techniques to recommend products to clients based on clients' previous purchases [3]. In general, WUM can be used to enhance web page usability by personalizing the web page toward a specific user experience [1][4], clustering groups of similar users [5][6], or predicting future users' behavior [7]. However, many of WUM techniques rely on the server logs data in the analysis [8]. Unfortunately, these logs are restricted to a limited number of attributes, such as the IP address of the host, the date and the time of the request, and the request type. Most of client interactions done on the client side are not recorded in the server logs; and thus, are not considered during the mining of usage behavior. These interactions include typing in a textbox, selecting or unselecting a radio button, checking or unchecking a checkbox, and hovering over a textbox.

In this paper, we first introduce a framework for collecting client side interactions with Web applications. On one hand, the framework does not interfere with or affect the user experience. On the other hand, user interactions are recorded into a persistent data storage without any intervention from the user. Second, we apply the framework to two systems: an E-survey Web-based form and an E-Commerce online shopping webpage. The E-Survey represents a student survey on the learning outcomes conducted for several courses at the end of a semester. The collected data of the student interactions was then analyzed and mined to evaluate students' mindfulness and consciousness while filling the online survey. The E-Commerce is an online store presenting


doi:10.4304/jetwi.4.4.324-332

products along with their attributes in a tabular form. The data collected from the interaction with the products page was then mined to predict future user's purchases behavior.

The roadmap of this paper is as follows. We survey related work in Section II. In Section III, we present our generic approach of mining client-side data and thoroughly explain its different steps. Section IV presents two case studies to evaluate our proposed approach. We finally conclude the paper in Section V.

## II. Related Work

Several client-side data-loggers and events-capturing systems have been proposed in literature. In this section, we survey the most relevant ones to our approach.

One of the earliest automatic systems for capturing client-side events was WebVIP [9]. WebVIP was mainly used for tracking the number of different event types. The client events were captured and stored in a database "as is" without any preprocessing steps.

Tracking client movements and actions was presented in [10], where the authors proposed a JavaScript framework that recorded the client interactions with webpages. The client events were also combined with server logs to give more information about client behavior. Based on the collected data, the webpage was then modified by injecting adequate JavaScript code via a proxy server. The authors did not mention how and based on what they modified the webpage.

An approach for detecting client browsing patterns was presented in [11]. The detection was performed based on the client data. Client events were presented as a tree that contained each accessed element and the associated accessed time. Then, the Sequence Alignment Method (SAM) was applied to discover the client patterns from the navigated elements. SAM was applied directly without a preprocessing step. Furthermore, it was unclear how the events were reordered.

A framework that catches the client events and the associated AJAX events was proposed in [12]. The key point of the approach was the ability of the framework to catch the executed AJAX events. The objective was to evaluate user willingness to take part in a remote usability test of a website.

A mouse tracking system using client-side scripting was presented in [13]. The system tracked the mouse movements (position and associated timestamps). The data were then sent to a backend server via the AJAX framework. The client's mouse movements were visualized to understand the client behavior. It was mentioned that the collected events were not sufficient to understand the client behavior.

Detection and remote evaluation of the usability were supported by WebRemUSINE tool [14]. The idea was to compare between the paths made by users and the optimum task model previously configured. Additionally, the clients must select the task option. Then, captured events were related to clients' tasks. The capturing of the events was not automatic and required explicit client selection.

Two systems, which used client-side logs for understanding client behaviors over Web search engines, were presented in [15] and [16]. The first system did not catch only search terms and the query sequences, but also the clickstream for each client. By this way, subsequently visited hyperlinked pages could be identified. In addition, a finite-state Markov model was used for extracting clients search patterns. In the second system, client-side logs were used both for searcher-goals' detection and for clicks-over-content prediction.

## III. Mining Client Paradata

In this section, we propose a general system for collecting and mining client paradata. The system collects and mines the data, without the need for the client's intervention. The ultimate objective is to understand users' mindfulness and predicting users' navigation sequence inside a webpage. The proposed system consists of three modules: data collection, data preprocessing, and data classification. In the data collection module, the data is collected from the client side in seamlessly manner. Data outliers are then purged in the data preprocessing step. Finally, different data mining techniques are applied to discover user behavior. In the next subsections, we elaborate on these steps.

### A. Data Collection

As shown in Fig. 1, the data collection consists of four steps: session identification, event identification, event storing, and event exporting.

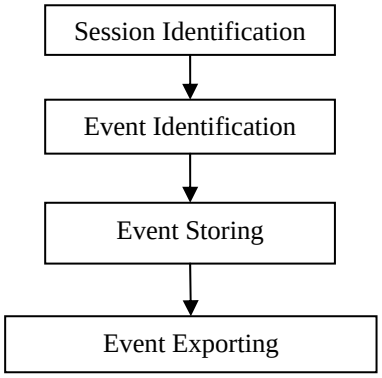


Fig. 1. The data collection steps.

*1) Session identification:* In the session identification phase, each period of activity for every unique user and for a given Web server is identified. `SessionId`, which is a unique number among all identified sessions, is created. `SessionId` is a function of the client's IP address and the current timestamp on the server. This guarantees the uniqueness of the `SessionId` across all sessions.

Once a client requests a webpage, the session identification function is fired by the `OnLoad` event. As shown in Fig. 2, when the `major_in()` function is called, a new Date object is instantiated. Then, the session starting time (`maj_in` variable) value is obtained

via the `getTime()` function. The `user_id()` function is called to obtain the `sessionId` value from the Date object using the `getTime()` function too. The returned value of `getTime()` represents the number of milliseconds since midnight January 1, 1970. By this way, the assigned `sessionId` for each client is unique. The generated `sessionId` is used to identify all recorded events that belong to the same user. Finally, for finishing the current `sessionId` value and/or instantiate a new `sessionId` value, the `major_out()` function is called. The `major_out()` function calculates the session end time (`maj_out` variable). The time difference between the start and the end time is the calculated in seconds (i.e., `maj_out – maj_in/1000`).

```
function user_id(){
    var usertime=new Date();
    usertime=usertime.getTime();
    userid=usertime;
}

function major_in(){
    var usertime=new Date();
    maj_in=usertime.getTime();
    user_id();
}

function major_out(){
    var usertime=new Date();
    maj_out=usertime.getTime();
//create new session
    total_major=
     (maj_out-maj_in)/1000;
    total_mouse=0;
    }
```

Fig. 2. Session identification code.

*2) Event identification:* In the event identification stage, we identify the Web elements and the associated events. The events are classified into two categories: Clickstream based and Time based. Clickstream-based events track the events that are related to the clickable Web elements. For example, a user clicks on either a button or a division Web element. Time-based events, on the other hand, track the events that are related to the time spent over a Web element. For example, a user hovers the mouse over a Web element for a given time and then moves away from that element. Therefore, the time difference between entering and exiting the Web element is recorded.

To track clickstream-based events, once a client clicks on the element, the function inside `OnClick` event is fired, and the target data is transferred along with `userID` via the `XmlHttpRequest` AJAX call. The transferring of data is a lightweight operation due to the use of the AJAX technology. As shown in Fig. 3, the `content` variable—which will be sent to the server—consists of seven other variables, which are concatenated and delimited by ",". The seven variables are `name`, `value`, `Item_time`, `sessionId`, `Date`, `TotalMouses`, and `Personalized`. The `name` variable represents the name of the Web element. The `value` variable represents the value of the Web element. The `Item_time` variable represents the amount of time spent over a specific Web element. That is, `Item_time` variable stores the time difference between `OnMouseOver` and `OnMouseOut` events. The `sessionId` variable represents the current `sessionId` for a specific client. In addition, the final `sessionId` state value will be identified, when the value of `Personalized` variable is not null. The `Date` variable represents the event date and time. We record the date on the client side for two reasons: to determine the date of the element event, and to determine whether the date on the client machine matches that of the server machine. The `TotalMouses` variable is a counter, which is incremented on every click. Moreover, the final counter value is sent when the `Personalized` value is not null. The `Personalized` value indicates if the client session is finished or not. In this framework, the focus is not only on the name and the value of the element, but also on the time that is spent on each element. Thus, we inject two types of events: `OnMouseOver` and `OnMouseOut`. Once the client hovers the mouse over a Web element, a function is invoked to instantiate the client's time object. As shown in Fig. 2, this operation can be accomplished using the JavaScript `Date` Object. Likewise, when the client moves the mouse away from over the Web element, another function, related to `OnMouseOut` event, is invoked. This function calculates the time difference between `OnMouseOver` and `OnMouseOut` events and stores this value in the `Item_time` variable. Next, the `OnMouseOut` event uses the `XmlHttpRequest` protocol to send the `content` variable to the server. All concatenated variables inside the content variable are identical to the Clickstream based approach, except that for the newly added variable `Item_time`.

```
function minor_in(){
    var usertime=new Date();
    min_in=usertime.getTime();
    total_minor_axis=" ";
}

function minor_out(name){
    var usertime=new Date();
    min_out=usertime.getTime();
    total_minor=(min_out-min_in) /
     1000;
    total_mouse++;
    content=
     name+',' + value + ',' +
     Item_time +',' +
     sessionId +',' + Date + ',' +
     TotalMouses+',' +
     Personalized;
     http.open("GET",
     "script_page.php?content=" +
     content, false);
     http.onreadystatechange =
     handleHttpResponse;
     http.send(null);
}
```

Fig. 3. Code to record and send events to the server.

*3) Event storing:* In the event storing phase, the identified events are stored in a relational database on the server side. The data are sent from the client in raw format, and are parsed and stored in a structured format on the database. In the exporting phase, event records are grouped per client session (user id).

The client data, which is sent via the GET method inside `XmlHttpRequest`, is received by the backend server (`script_page.php`). As shown in Fig. 4, because the data was concatenated and delimited with "," on the client side, it must be parsed to extract the variables values. The `explode` function is used to convert the data into array of variables. In addition `ip` and `id` variables are initialized to the server's IP address and current date, respectively. Then the variables are inserted into the `items_events` table. Finally, the `user_id` and `personalized` variables are inserted into `user_select`.

```
<?php
$str = strtolower($_GET['content']);
$str= explode(",",$str);
$name=$str[0];
$item_time=$str[1];
$total_time=$str[2];
$userId=$str[3];
$total_mouse=$str[4];
$total_minor_axis=$str[5];
$pers=$str[6];
$ip=$_SERVER['REMOTE_ADDR'];
$id = uniqid("");
    if(strlen($ff)){
        $sqlquery   =   "INSERT INTO
all_items_mobile VALUES
('$userId','$name','$item_time','$ip','$total_t
ime','$total_minor_axis','$total_mouse','$pers'
,'$id')";
        if($name=="personalized"){
                $sqlquery = "INSERT INTO
        final_select1 VALUES
        ('$userId','$name')";
    }
?>
```

Fig. 4. Code to expand and store data in the database.

Fig. 5 shows the database schema, which consists of two tables: the `Events` table and the `session primary key` table. The `Events` table stores vectors of values for the client transactions (events). On other hand, the `Session primary key` table stores the `sessionId` for each client, which is the primary key, and the final personalized item. The relationship between these tables allows us to easily merge data for client sessions.

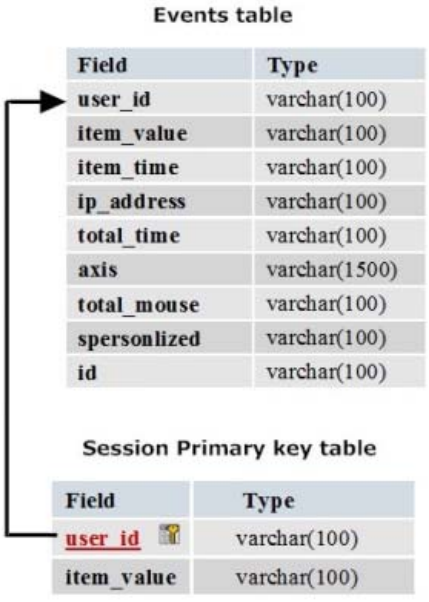


Fig. 5. ER Diagram of the Database.

*4) Event exporting:* The operation of exporting data has two modes: general mode and time-based mode. In the general mode, the exporting is done by a simple aggregation of all records within a specific session. The output of the aggregation operation is a vector data-structure consisting of six fields: all clickstreams, the date for each click stream, the session ID, the total time spent in the session, the total number of clicks, and the IP address. Fig. 6 shows a snapshot of the aggregated records for the general mode after elimnating the IP address and the date columns.

In the time-based mode, the scenario is similar to the general mode except that while the records are aggregated, the time spent over each item is accumulated. In this case, the vector data-structure includes *n* fields to store *n*-accumulated times one for each Web element. The data structure also includes the fields of the general mode but without the clickstream field. Fig. 7 shows a snapshot of the aggregated records for the time-based mode but without both the IP address and the date columns.

| content | offset | pers |
|---|---|---|
| justifyright,insertorderedlist,mcerepaint,mceinse... | 19.88 | mceprint |
| justifyleft,justifycenter,justifyfull,justifyrigh... | 11.967 | mcesave |
| inserthorizontalrule,mceinsertcontent,mceadvlink,... | 1790.948 | mcepreview |
| justifyright,justifycenter,justifyleft,justifyrig... | 609.418 | mcepreview |
| justifyfull,mceattributes,mcedel,mceinsertcontent... | 50.683 | mcesave |
| formatblock,fontname,insertunorderedlist,mceinser... | 33.39 | mceprint |
| justifyfull,justifyright,justifycenter,justifylef... | 11.25 | mcepreview |
| formatblock,fontname,fontsize,justifyfull,mceprev... | 11.678 | mcepreview |
| mceinserttable,mceinsertlayer,mceins,mcedel,remov.. | 26.505 | mceprint |
| justifyfull,justifyright,justifycenter,justifylef... | 13.107 | mcepreview |
| indent,outdent,insertorderedlist,insertunorderedl... | 15.276 | mcesave |
| justifyfull,justifyright,justifycenter,justifylef... | 4.058 | mcepreview |

Fig. 6. Snapshot of the aggregated records in general mode.

| cmp1 | cmp2 | cmp3 | cmp4 | cz1 | | cpr4 | t_time | len | t_onmouse | pers |
|---|---|---|---|---|---|---|---|---|---|---|
| 0.238 | 0.118 | 0.092 | 0.081 | 0 | | 0 | 47.809 | 18 | 17 | camera3 |
| 0.23 | 0.302 | 0.408 | 0.236 | 0.243 | ... | 0 | 11.464 | 39 | 38 | camera3 |
| 0.677 | 0.129 | 0.304 | 0.593 | 0.17 | | 0.742 | 26.663 | 44 | 43 | camera2 |
| 0.104 | 0.421 | 0.715 | 0 | 0.835 | | 0 | 42.29 | 58 | 57 | camera2 |
| 0.238 | 0.118 | 0.092 | 0.081 | 0 | | 0 | 47.809 | 18 | 17 | camera3 |
| 0.433 | 0.071 | 0.015 | 0 | 0.183 | | 0.294 | 56.456 | 32 | 31 | camera4 |
| 0.154 | 0.148 | 0.168 | 0 | 0 | | 0 | 65.296 | 24 | 23 | camera3 |
| 0.104 | 0.421 | 0.715 | 0 | 0.835 | | 0 | 42.29 | 58 | 57 | camera2 |
| 0.23 | 0.302 | 0.408 | 0.236 | 0.243 | ... | 0 | 11.464 | 39 | 38 | camera3 |
| 0.677 | 0.129 | 0.304 | 0.593 | 0.17 | | 0.742 | 26.663 | 44 | 43 | camera2 |
| 0.104 | 0.421 | 0.715 | 0 | 0.835 | | 0 | 42.29 | 58 | 57 | camera2 |
| 0.238 | 0.118 | 0.092 | 0.081 | 0 | | 0 | 47.809 | 18 | 17 | camera3 |
| 0.433 | 0.071 | 0.015 | 0 | 0.183 | | 0.294 | 56.456 | 32 | 31 | camera4 |
| 0.154 | 0.148 | 0.168 | 0 | 0 | ... | 0 | 65.296 | 24 | 23 | camera3 |
| 0 | 0 | 0 | 0 | 0.219 | | 0 | 69.52 | 20 | 19 | camera1 |

Fig. 7. Snapshot of the aggregated records in time based mode.

## B. Data Cleaning

In the data cleaning step, outlier data are identified and removed from the data set.

## C. Classification Techniques

Three classification techniques were used to analyze and predict the client-side data. The classification techniques used in this paper are Naïve Bayes, the C4.5 decision tree, and Support Vector Machines.

Naïve Bayes [17] is a probabilistic model based on Bayesian theorem. Although it is simple, it often outperforms most of the other classifiers especially if it is trained using supervised learning methods. If $C$ represents a class and $F$ represents a feature, then the conditional probability of $C$ given $F$ is given by

$$P_r(C \mid F) = \frac{P_r(F \mid C)P_r(C)}{P_r(F)}. \quad (1)$$

The C4.5 decision tree classifier [18][19] is a supervised machine-learning algorithm. C4.5 generates decision trees from a set of training data based on the information entropy concept. The training data is a set $S = (s_1, s_2, s_3, s_4, ...)$ of already classified samples. Each samples = $(x_1, x_2, x_3, x_4,...)$ is a vector, where each component, $x_1, x_2, x_3, x_4,...$, represents an attributes or a features of the sample. Finally, each vector value is labeled by a class label. C4.5 chooses one attribute at any given level for splitting the data. The attribute that is chosen owns the highest information gain. In addition, C4.5 uses the divide-and-conquer technique to construct a suitable tree from a training set $S$ of cases.

SVM [20] is a supervised machine-learning algorithm. The main idea is to find a separator line, which called a hyper-plane. The hyper-plane separates the *n*-dimensional data completely into two classes. However, since the training data cannot always be separated linearly, the concept of a kernel is introduced in SVM. The main functionality of the kernel is to convert the data into an *n*-dimensional space where the data can be separable. SVM can be expressed mathematically as follows. Assume there are $L$ training examples $\{x_i, y_i\}$, $i = 1, ..., L$, where each training example has $d$ inputs ($x_i \in R^d$) and a class label with one of two values ($y_i \in \{1, -1\}$). Now, all hyper planes in $R^d$ are parameterized by a vector, $w$, and a constant, $b$, expressed as

$$x \cdot w + b = 0. \quad (2)$$

Given such a hyper-plane ($w$, $b$) that separates the data, this gives the function

$$f(x) = sign(w \cdot x + b), \quad (3)$$

which correctly classifies the training data. However, a given hyper-plane represented by ($w$, $b$) is equally expressed by all pairs $\{w, b\}$ for $R^+$. Therefore, we define the canonical hyper-plane to be one that separates the data from the hyper-plane by a distance of at least 1. That is, we consider the data that satisfy

$$x_i \cdot w + b \geq +1 \text{ when } \quad y_i = +1, \text{and} \quad (4)$$

$$x_i \cdot w + b \leq -1 \text{ when } \quad y_i = -1. \quad (5)$$

## D. Accuracy Measures

To evaluate the generated classifiers, we use the three well-known measures [21]: Precision, Recall, and F-measure.

Precision (P) is the fraction of retrieved documents that are relevant. P is computed as

$$P = \frac{\#\text{ retrieved relevant items}}{\#\text{ retrieved items}} \cdot P_r(\text{relevant} \mid \text{retrieved}). \quad (6)$$

Recall (R) is the fraction of relevant documents that are retrieved as expressed in

$$R = \frac{\#\text{ retrieved relevant items}}{\#\text{ relevant items}} \cdot P_r(\text{retrieved} \mid \text{relevant}). \quad (7)$$

F-measure is a single measure that trades off precision versus recall. F-measure is the weighted harmonic mean of precision and recall, and is calculated as follows

$$F = \frac{2 \cdot P \cdot R}{\mathrm{P} + \mathrm{R}}. \quad (8)$$

## IV. Case Studies

As case studies for the proposed system, we implemented and applied the previous steps on two systems: E-survey and E-commerce Web applications.

### A. Online Survey

The main goal of this case study is to understand the client behavior while he or she is responding to an online survey to help us measure the mindfulness or consciousness of the clients. Results obtained from online surveys, in specific, and from surveys, in general, are used to make serious decisions or conclusions. In most cases, respondents are mandated to or given precious incentives to fill in surveys. Therefore, the basic idea is to tell whether a given respondent has answered the survey objectively or he/she has finished it without paying any attention to it, especially when the identity of the respondents is anonymized—which is often a valid assumption in almost all surveys.

*1) Data collection:* The survey we used is a student learning outcomes survey taken toward the end of the semester. The survey consists of 11 multiple-choice questions and 1 essay question, where the students are encouraged to type in their comments about the course other than those asked via the multiple-choice questions. Among the 11 questions, we have a one flag question that is constructed carefully and deliberately to determine whether a student has filled the survey consciously. As shown in Fig. 8, the Web form is designed so as each question has a very low opacity and unreadable unless a user hovers the mouse directly over it. This allows us to accurately measure the time spent on each question (because we are sure the student is reading and focusing on that question not on another one).

*2) Data preprocessing:* Overall, we collected interactions data for 101 students who participated in the survey. We excluded the data for those who did not fully complete the survey or who completed the survey in less than 25 seconds (these data are considered outliers). According to this criterion, we excluded the data for 11 students' transactions.


Fig. 8. Screen capture of the survey.

*3) Data analysis:* To understand student behavior while they answered the questions, a model is generated based on the interval time frequencies in seconds; see Fig. 9. The generated model is an exponential distribution.

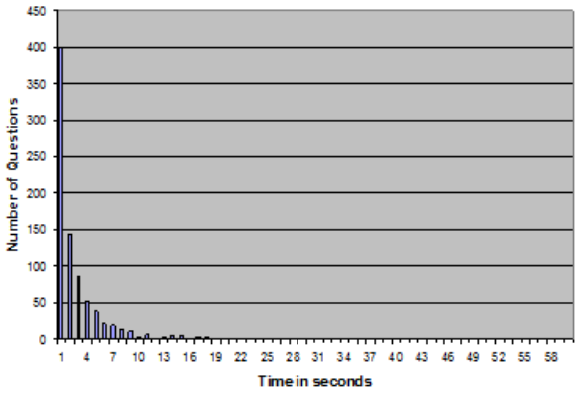


Fig. 9. Student's model behavior while filling the survey.

*4) Data classification:* In any classification method, efficient features lead to a high classification accuracy rate. Thus, Features have been carefully chosen. The times that are spent over the 12 questions are used as the main 12 features. In addition, the total session time and the number of visited questions represent two other features. Therefore, the total number of features is 14.

The output of classification step will be the trained classifiers, which are able to classify any new student of whether he is mindful or not based on his answering behavior to the questions.

For the classification purpose, Naïve Bayes, Support Vector Machine and Decision Tree are used. Table I shows the different accuracy measures using the 5-cross validation technique [22]. All classifiers achieved accuracy over 85% using the 5-fold cross validation technique. The Naïve Bayes classifier gave the best accuracy of 88.7%.

TABLE I.
ACCURACY MEASURE VALUES FOR E-SURVEY USING DIFFERENT CLASSIFIERS.

| | C4.5 Decision Tree | Naïve Bayes | SVM |
|---|---|---|---|
| Precision | 0.852 | 0.887 | 0.843 |
| Recall | 0.923 | 0.495 | 0.901 |
| F-Measure | 0.886 | 0.6 | 0.861 |

## B. E-Commerce Application

The main purpose of this system is to predict which product the user will choose/buy based on his/her navigation sequence on the presented items on a given webpage. To track the user navigation behavior, the E-Commerce system was designed in way that the opacity of the item the user is hovering over is greater than other items (i.e., similar to the questions in the E-survey application). This way we are sure where the user is focusing at certain point of time. Fig. 10 shows a snapshot the E-Commerce system, where a list of available mobile phones along with each item features are presented.


Fig. 10. Snapshot of the E-commerce Web application.

*1) Data collection:* For collecting the data, students from Jordan University of Science and Technology, Jordan; and Heinrich-Heine University of Dusseldorf, Germany were asked to use the website. Beside that, the application URL was posted on different social media websites and people were invited to use it. The database record showed that 58 clients bought cameras and 54 clients bought mobile phones. Then, the data is merged and exported according to the steps described earlier.

*2) Data preprocessing:* The time spent over any cell within a specific user session was aggregated. Users bought items in less than 7 navigation sequences were eliminated. Furthermore, users bought items in less than 20 seconds were eliminated too. Based on these values, 40 clients transactions were pruned from the cameras data; and the remaining sessions were 18. For Mobile phones data, 35 sessions were pruned; and the remaining sessions were 20.

*3) Data classification:* Different classification techniques were applied to the data to predict which item a user will choose/buy after a series of navigation on different page items. The features applied to the classifiers are the time spent on each item in the webpage along with the aggregated time spent over all items in the webpage.

The features were then applied to three different classifiers: Naïve Bayes, Support Vector Machine and Decision Tree. Table II shows the different accuracy measures using the 5-cross validation technique. The system did not perform well as all classifiers gave accuracy below 35%. This is because the training data was not sufficient while we had a large number of features that require a larger number of users to use the system and to train the classifier afterward.

TABLE II.
ACCURACY MEASURE VALUES FOR E-COMMERCE USING DIFFERENT CLASSIFIERS.

| | C4.5 Decision Tree | Naïve Bayes | SVM |
|---|---|---|---|
| Precision | 0.313 | 0.328 | 0.202 |
| Recall | 0.350 | 0.313 | 0.450 |
| F-Measure | 0.325 | 0.315 | 0.279 |

## V. CONCLUSIONS

In this paper, we presented a framework to collect and analyze client paradata. The proposed framework was implemented using JavaScript, PHP, and MySQL. The events on the client-side were collected and sent back to the server. At the server, the data were structured and analyzed to understand the user behavior. The data were then analyzed and mined to help predicting future user behaviors. We applied the proposed system to two case studies: E-survey and E-Commerce Web applications.

For the E-survey application, we applied the proposed system to better understand user mindfulness when

interacting with Web forms. We were able to draw a mathematical model that describes the distribution of the time spent in filling an E-Survey for different users. Also, we applied different classification algorithms to generate classifiers that can predict the user consciousness while filling the E-survey. The prediction depended on the time the user spends on answering each question.

In the E-Commerce application, we applied the proposed system to predict a user selection based on his or her navigation behavior in the page.

## ACKNOWLEDGMENTS

We would like to thank the anonymous reviewers for their insightful and constructive comments that greatly enhanced the clarity and the presentation of this paper.

## REFERENCES

[1] A. G. Büchner, and M. D. Mulvenna, "Discovering Internet marketing intelligence through online analytical web usage mining," ACM SIGMOD Record, vol. 27, no. 4, pp. 54-61, December 1998.

[2] L. Zhuhadar and O. Nasraoui, "A Hybrid Recommender System Guided by Semantic User Profiles for Search in the E-learning Domain," Journal of Emerging Technologies in Web Intelligence, vol. 2, no. 4, pp. 272-281, November 2010.

[3] K.D. Fenstermacher, and M. Ginsburg, "Mining Client-Side Activity for Personalization," in Proceedings of the Fourth IEEE International Workshop on Advanced Issues of E-Commerce and Web-Based Information Systems (WECWIS'02), Newport Beach, California, pp. 205-212, June 2002.

[4] H. R. Al-Lawati, A. Al-Hosni, A. Al-Hamadani, and M. Al-Badawi, "Discovery of Popular Structural Properties in a Website for Personalization and Adaptation," Journal of Emerging Technologies in Web Intelligence, vol. 3, no. 3, pp. 253-260, August 2011.

[5] M. Okabe, S. Yamada, "An Interactive Tool for Human Active Learning in Constrained Clustering," Journal of Emerging Technologies in Web Intelligence, vol. 3, no. 1, pp. 20-27, February2011.

[6] S. Park, N.C. Suresh, and B.-K. Jeong, "Sequence-based clustering for Web usage mining: A new experimental framework and ANN-enhanced K-means algorithm," Data and Knowledge Engineering, vol. 65, no. 3, pp. 512-543, June 2008.

[7] C.-H. Lee, Y.-l. Lo, and Y.-H. Fu, "A novel prediction model based on hierarchical characteristic of web site," Expert Systems with Applications: An International Journal, vol. 38, no. 4, pp. 3422-3430, April 2011.

[8] F.M. Facca, and P.L. Lanzi, "Mining interesting knowledge from weblogs: a survey," Data and Knowledge Engineering, vol. 53, no. 3, pp. 225-241, June 2005.

[9] WebVIP, `http://zing.ncsl.nist.gov/WebTools/WebVIP/overview.html`, last accessed February 2009.

[10] R. Atterer, M. Wnuk, and A. Schmidt, "Knowing the user's every move: user activity tracking for website usability evaluation and implicit interaction," in Proceedings of the 15th international conference on World Wide Web, Edinburgh, Scotland, pp. 203-212, May 2006.

[11] V.F. De Santana, and M.C. Baranauskas, "Summarizing observational client-side data to reveal web usage patterns," in Proceedings of the 2010 ACM Symposium on Applied Computing (SAC 10), Sierre, Switzerland, pp. 1219-1223, March 2010.

[12] R. Atterer, and A. Schmidt, "Tracking the interaction of users with AJAX applications for usability testing," in Proceedings of the SIGCHI conference on Human factors in computing systems, San Jose, California, pp. 1347-1350, April-May 2007.

[13] F. Mueller, and A. Lockerd, "Cheese: tracking mouse movement activity on websites, a tool for user modeling," in CHI '01 extended abstracts on Human factors in computing systems, Seattle, Washington, pp. 279-280, March-April 2001.

[14] P. Mutzel, and P. Eades, "Graphs in Software Visualization–Introduction," Revised Lectures on Software Visualization, International Seminar, pp. 285-294, 2001.

[15] N. Kammenhuber, J. Luxenburger, A. Feldmann, and G. Weikum, "Web search clickstreams," in Proceedings of the 6th ACM SIGCOMM conference on Internet measurement, Rio de Janeriro, Brazil, pp. 245-250, October 2006.

[16] Q. Guo, and E. Agichtein, "Ready to buy or just browsing?: detecting web searcher goals from interaction data," in Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval, Geneva, Switzerland, pp. 130-137, July 2010.

[17] G.H. John, and P. Langley, "Estimating continuous distributions in Bayesian classifiers," in Proceedings of the Eleventh conference on Uncertainty in artificial intelligence, Montréal, Qué, Canada, pp.338-345, August 1995.

[18] J.R. Quinlan, "Induction of Decision Trees," Machine Learning, vol. 1, no. 1, pp. 81-106, March 1986.

[19] J.R. Quinlan, "C4.5: programs for machine learning," Morgan Kaufmann Publishers Inc., San Francisco, CA, 1993.

[20] C. Cortes, and V. Vapnik, "Support-Vector Networks," Machine Learning, vol. 20, no. 3, p. 273-297, September 1995.

[21] C.D. Manning, P. Raghavan, and H. Schtze, "Introduction to Information Retrieval," Cambridge University Press, New York, NY, 2008.

[22] B. Efron, "Estimating the Error Rate of a Prediction Rule: Improvement on Cross-Validation," Journal of the American Statistical Association, vol. 78, no. 382, pp. 316-331, June 1983.

**Natheer Khasawneh** is an Associate Professor in the Department of Software Engineering at Jordan University of Science and Technology. He received his B.S. in Electrical Engineering from Jordan University of Science and Technology in 1999. He received his Master and Ph.D. degrees in Computer Science and Computer Engineering from University Akron, Akron, Ohio, USA in the years 2002 and 2005, respectively. His current research interest is data mining, biomedical signals analysis, software engineering and web engineering.

**Rami Al-Salman** was born in Irbid, Jordan in 1986. He received his M.S. degree in computer engineering JUST in 2011. He is currently working on his Ph.D. in Hybrid qualitative and quantitative spatial reasoning and analysis in the computer science department (Cognitive Systems group) at the University of Bremen, Germany. In addition, he received a three months research scholarship at the Heinrich-Heine University of Düsseldorf, Germany during the summer of 2010. His research interests include security, data mining, information assurance and data reasoning. Al-Salman was a committee member in

International Conference on Informatics, Cybernetics, and Computer Applications (ICICCA2010).

**Ahmad T. Al-Hammouri** is an Assistant Professor in the Department of Network Engineering and Security at Jordan University of Science and Technology. He received the B.S. degree with first-class honors in Electrical Engineering from Jordan University of Science and Technology, Irbid, Jordan, in 2000; and the M.S. and the Ph.D. degrees in Computer Engineering from Case Western Reserve University, Cleveland, Ohio, in 2004 and 2008, respectively. He has held research positions at Case Western Reserve University's Netlab. During the Summer of 2011, he was a Visiting Research Associate with the Department of Industrial Information and Control Systems, KTH—Royal Institute of Technology, Stockholm, Sweden. His Research interests are in the areas of cyber-physical systems, smart power grid, Internet congestion control, and middleware for real-time sense-and-respond systems.

**Stefan Conrad** is a Professor in the Department of Computer Science at Heinrich-Heine-University Duesseldorf, Germany. He was an Associate Professor in the Department of Computer Science at Ludwig-Maximilians- University in Munich, Germany, from 2000 to 2002. From 1994 to 2000, he was an Assistant Professor at the University of Magdeburg where he finished his 'Habilitation' in 1997 with a thesis on federated database systems. From 1998 to 1999, he was also a Visiting Professor at the University of Linz, Austria. He received his PhD in Computer Science in 1994 at Technical University of Braunschweig, Germany. His current research interests include database integration, knowledge discovery in databases, and information retrieval. He is a (co)author of two books (in German) and a large number of research papers.