

Слайд 1, титульник.

Здравствуйте, уважаемые члены комиссии. Представляю свою выпускную квалификационную работу.

Слайд 2, цели и задачи.

Целью работы является разработка метода первичного анализа поведения пользователей на веб-страницах.

Для достижения цели в работе поставлены задачи, представленные на слайде...

Подождать 5 секунд.

Слайд 3, предметная область.

Контекст работы включает в себя понятие системы электронного обучения, т. е. LMS, с англ. – Learning Management System – комплекс программных приложений, используемых для помощи в учебном процессе при электронном обучении.

LMS используется в том числе в МГТУ им. Н.Э. Баумана. Внедрение подобных систем вызвано цифровизацией образования, которое входит в ключевые направления развития высшего образования, правовой основой для которых, в частности, являются национальный проект “Цифровая экономика” и Федеральный закон №273.

Существует множество определений пользовательского поведения, большинство авторов рассматривают это понятие как совокупность действий, реализуемых в сфере потребления.

В текущей работе с поправкой на контекст используется следующее определение: пользовательское поведение – действия, связанные с принятием решений относительно выбора, потребления или отказа от товаров и услуг.

Пользовательское поведение в контексте веб–ресурса выражается во взаимодействии потребителя с элементами интерфейса, в его способе навигации, временных затрат и повторяющихся паттернах

действий. Карта кликов фиксирует цифровые следы пользователей, что необходимо для:

- визуализации проблемных областей;
- адаптации интерфейса для достижения целевых сценариев использования.

Качество электронного обучения зависит от того, как обучающиеся взаимодействуют с интерфейсом LMS. Карта кликов фиксирует цифровые следы пользователей, а их обработка необходима для:

- визуализации проблемных областей;
- адаптации LMS под целевые сценарии использования.

Слайд 4, формализация задачи.

На этом слайде приведена формализация задачи в нотации IDEF0.

- в качестве входных данных выступают клики пользователей на веб-страницах;
- в качестве выходных – карта кликов для веб-страниц и метаданные;
- “механизмами” обозначены веб-сервер и веб-драйвер;
- к “управлению” или же “контролю” отнесены параметры отображения карты кликов и дерева файлов с выборками кликов и веб-страниц для обработки.

Слайд 5, существующие решения.

Существующие решения в большинстве своём обладают следующими ограничениями:

- требование публичного домена;
- обработка и передача данных осуществляется на инфраструктуре третьих лиц;
- решения зачастую обладают проприетарной формой распространения;
- конечный пользователь не имеет доступа к архитектуре, что затрудняет масштабирование и оптимизацию.

На слайде приведена сравнительная таблица некоторых рассматриваемых решений.

Слайд 6, формализация задачи с учётом модульной структуры.

Решением, учитывающим ограничения, является метод из трёх этапов:

- сбора;
- обработки;
- и отображения.

Так, после этапа сбора данных “сырые” данные о кликах попадают на обработку, в результате которой образуются метаданные и карта кликов, используемая на этапе визуализации.

Слайд 7, схема алгоритма сбора.

На данном слайде представлена схема алгоритма сбора. Этап сбора отвечает за регистрацию и сохранение пользовательских кликов на веб-страницах.

В левой части слайда приведён пример записи клика, который сохраняется в лог-файле.

Слайд 8, схема алгоритма обработки.

Этап обработки выполняет форматирование собранных данных и на их основе составляет карты кликов.

Для каждой веб-страницы, на которой был сделан клик, с помощью вебдрайвера создаётся скриншот. Если хэш кликов на странице не изменился с предыдущего запроса на генерацию карты кликов, она остаётся такой же. Иначе для каждого клика в зависимости от количества кликов в некоторой области определяется его цвет, после чего клик отображается на карте.

Слайд 9, схема алгоритма визуализации.

А этап визуализации данных служит для навигации и просмотра созданных на предыдущем этапе карт кликов.

Слайд 10, схема разработанного ПО.

Схема разрабатываемого алгоритма выглядит следующим образом. Описанным ранее этапам соответствуют модули. Несмотря на локальное развёртывание ПО имеет клиент-серверную архитектуру, где на стороне клиента выполняется только фиксация кликов и отправка данных о них на сервер.

В качестве основных инструментов для разработки использовались языки JavaScript, PHP и Python.

Слайд 11, исследование.

Целью анализа являлось установление зависимости производительности разработанного приложения от объёма обрабатываемых данных.

Так, зависимость от количества обрабатываемых кликов была установлена для:

- времени выполнения;
- средней загрузки ЦПУ;
- среднего и пикового потребления оперативной памяти.

Слайд 12, результаты исследования.

На слайде приведены результаты исследования.

График среднего времени выполнения принимает линейный характер. График средней загрузки ЦПУ тоже, причём с очень малым коэффициентом наклона. А вот графики среднего и пикового потребления оперативной памяти показали логарифмическую зависимость от количества обрабатываемых кликов.

Слайд 13, заключение.

Таким образом, в результате был разработан поставленный метод, выполнены все поставленные задачи, а цель работы – достигнута.

Слайд 14, дальнейшее развитие.

В качестве направлений дальнейшего развития можно выделить:

- увеличение количества собираемых данных о кликах пользователей на веб-страницах;
- добавление на их основе новых данных и метрик в статистику;
- оптимизацию времени обработки кликов;
- и добавление системы рекомендаций.

На этом всё. Готов ответить на ваши вопросы.

—

Визуализация плотности кликов реализована через построение двумерного массива плотности $D(x, y)$, в который добавляются значения плотности с гауссовым весом от каждого клика.

Пусть

- (x_k, y_k) — координаты k -го клика;
- r_k — радиус влияния, определяемый по формуле:

$$r_k = \min(r_0 + 5 \cdot \sqrt{n_k}, r_{\max}), \quad (3.1)$$

где

- r_0 — базовый радиус;
- n_k — количество повторных кликов в точке (x_k, y_k) ;
- r_{\max} — максимально допустимый радиус.

Для каждой точки изображения (i, j) вычисляется евклидово расстояние до точки клика:

$$d = \sqrt{(i - x_k)^2 + (j - y_k)^2}. \quad (3.2)$$

Если $d \leq r_k$, плотность $D(i, j)$ увеличивается на вес w , заданный ядром Гаусса:

$$w = \exp\left(-\frac{d^2}{2 \cdot \sigma^2}\right), \quad \sigma = \frac{r_k}{2}. \quad (3.3)$$

Максимальная плотность D_{\max} используется для нормализации цветовой шкалы.

Цвет пикселя определяется функцией `get_color_for_density`, где используются интервалы по значению плотности:

- если $D(i, j) \leq 1$, цвет синий: $(0, 0, 255, 160)$;
- если $1 < D(i, j) < D_{\max}/2$, цвет изменяется от синего к фиолетовому:

$$\text{ratio} = \frac{D(i, j)}{D_{\max}/2}, \quad (3.4)$$

$$\text{color} = (255 \cdot \text{ratio}, 0, 255 \cdot (1 - \text{ratio}), 180); \quad (3.5)$$

- если $D(i, j) \geq D_{\max}/2$, цвет переходит от фиолетового к красному:

$$\text{ratio} = \frac{D(i, j) - D_{\max}/2}{D_{\max}/2}, \quad (3.6)$$

$$\text{color} = (255, 0, 255 \cdot (1 - \text{ratio}), 200). \quad (3.7)$$