

Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií
FIIT-182905-73915

Bc. Michal Hucko

Identifikácia zmätenia používateľa vo webovej aplikácii

Diplomová práca

Študijný program: Inteligentné softvérové systémy
Študijný odbor: 9.2.5 Softvérové inžinierstvo (hlavný),
9.2.8 Umelá inteligencia (vedľajší)
Miesto vypracovania: Ústav informatiky, informačných systémov a softvérového
inžinierstva
Vedúci práce: prof. Ing. Mária Bieliková, PhD
apríl 2019

Zadanie diplomovej práce

Meno študenta: **Bc. Michal Hucko**
Študijný program: Inteligentné softvérové systémy
Študijný odbor: Softvérové inžinierstvo – hlavný študijný odbor
Umelá inteligencia – vedľajší študijný odbor

Názov práce: **Identifikácia zmätenia používateľa vo webovej aplikácii**

Samostatnou výskumnou a vývojovou činnosťou v rámci predmetov Diplomový projekt I, II, III vypracujte diplomovú prácu na tému, vyjadrenú vyššie uvedeným názvom tak, aby ste dosiahli tieto ciele:

Všeobecný cieľ:

Vypracovaním diplomovej práce preukážete, ako ste si osvojili metódy a postupy riešenia relatívne rozsiahlych projektov, schopnosť samostatne a tvorivo riešiť zložité úlohy aj výskumného charakteru v súlade so súčasnými metódami a postupmi študovaného odboru využívanými v príslušnej oblasti a schopnosť samostatne, tvorivo a kriticky pristupovať k analýze možných riešení a k tvorbe modelov.

Špecifický cieľ:

Vytvorte riešenie zodpovedajúce návrhu textu zadania, ktorý je prílohou tohto zadania. Návrh bližšie opisuje tému vyjadrenú názvom. Tento opis je záväzný, má však rámcový charakter, aby vznikol dostatočný priestor pre Vašu tvorivosť.

Riadte sa pokynmi Vášho vedúceho.

Pokiaľ v priebehu riešenia, opierajúc sa o hlbšie poznanie súčasného stavu v príslušnej oblasti, alebo o priebežné výsledky Vášho riešenia, alebo o iné závažné skutočnosti, dospejete spoločne s Vaším vedúcim k presvedčeniu, že niečo v texte zadania a/alebo v názve by sa malo zmeniť, navrhnete zmenu. Zmena je spravidla možná len pri dosiahnutí kontrolného bodu.

Miesto vypracovania: Ústav informatiky, informačných systémov a softvérového inžinierstva, FIIT STU v Bratislave

Vedúci práce: **prof. Ing. Mária Bieliková, PhD.**

Termíny odovzdania:

Podľa harmonogramu štúdia platného pre semester, v ktorom máte príslušný predmet (Diplomový projekt I, II, III) absolvovať podľa Vášho študijného plánu

Predmety odovzdania:

V každom predmete dokument podľa pokynov na www.fiit.stuba.sk v časti:
home > Informácie o > štúdiu > harmonogram štúdia > diplomový projekt.

V Bratislave dňa 12. 2. 2018

**SLOVENSKÁ TECHNICKÁ UNIVERZITA
V BRATISLAVE**

Fakulta informatiky a informačných technológií
Ilkovičova 2, 842 16 Bratislava 4

prof. Ing. Pavol Návrat, PhD.
riaditeľ Ústavu informatiky, informačných systémov
a softvérového inžinierstva

Návrh zadania diplomovej práce

Finálna verzia do diplomovej práce ¹

Študent:

Meno, priezvisko, tituly: Michal Hucko, Bc.
Študijný program: Inteligentné softvérové systémy
Kontakt: hucko.miso@gmail.com

Výskumník:

Meno, priezvisko, tituly: Mária Bieliková, prof. Ing. PhD.

Projekt:

Názov: Identifikácia zmätenia používateľa vo webovej aplikácii
Názov v angličtine: Identification of user confusion in a web application
Miesto vypracovania: Ústav informatiky, informačných systémov a softvérového inžinierstva, FIIT STU, Bratislava
Oblasť problematiky: strojové učenie, modelovanie používateľa, vzory správania, interakcia človek počítač

Text návrhu zadania²

Webové aplikácie sú dnes už bežnou súčasťou našich životov. Pri ich používaní častokrát narazíme na situáciu, kedy máme problém sa ďalej pohnúť. Táto situácia je spravidla sprevádzaná stavom zmätenia používateľa. Tento stav je veľmi špecifický pre každého z nás. Jeho identifikáciou a predvídaním vieme uľahčiť prácu s webovými aplikáciami, napríklad prezentovaním nápovedy v správnom okamihu, čo nám umožní zefektívniť činnosť.

Analyzujte oblasť modelovania správania používateľa z implicitnej spätnej väzby pri interakcii s webovými aplikáciami. Sústreďte sa pri tom na signály, ktoré môžu indikovať rôzne stavy používateľa, špeciálne jeho zmätenie. Preštudujte metódy strojového učenia vhodné pre prácu s dátami z implicitnej spätnej väzby používateľa.

Navrhnite metódu odhaľovania zmätenia používateľa webovej aplikácii, ktorá umožní v reálnom čase tento stav odhaliť. Pokúste sa tento stav aj predikovať. Navrhnite pre riešenie využitie vhodných metód strojového učenia a ich prípadnú kombináciu. Navrhnutú metódu overte na reálnych dátach z používania vybratej webovej aplikácie. Vychádzajte z dát implicitnej spätnej väzby vybranej aplikácie. Kľúčovými v tomto prípade sú pohyby myšou, skrolovanie, kliky. Experimentujte a zvážte pri tom využitie dát zo sledovania pohľadu používateľa. Navrhnuté riešenie diskutujte aj v kontexte iných dostupných riešení pre daný problém.

¹ Vytlačiť obojstranne na jeden list papiera


² 150-200 slov (1200-1700 znakov), ktoré opisujú výskumný problém v kontexte súčasného stavu vrátane motivácie a smerov riešenia

Literatúra³


- THOMAS, P.. Using interaction data to explain difficulty navigating online. ACM Transactions on the Web (TWEB), vol. 8, no. 4, 2014, Article no. 24, 41 pages.
- TIKHONOV, A., PROKHORENKOVA, L., CEHLNOKOV, A., BOGATYY, I., GUSEV, G. . What can be Found on the Web and How: A Characterization of Web Browsing Patterns. In: Proceedings of the ACM Web Science Conference. ACM Press, 2015, Article no,14, 10 pages.

Vyššie je uvedený návrh diplomového projektu, ktorý vypracoval(a) Bc. Michal Hucko, konzultoval(a) a osvojil(a) si ho prof. Ing. Mária Bieliková, PhD. a súhlasí, že bude takýto projekt viesť v prípade, že bude pridelený tomuto študentovi.

V Bratislave dňa 10.1.2018



Podpis študenta



Podpis výskumníka

Vyjadrenie garanta predmetov Diplomový projekt I, II, III

Návrh zadania schválený: áno / nie⁴

Dňa: 12.2.2018



Podpis garanta predmetov

³ 2 vedecké zdroje, každý v samostatnej rubrike a s údajmi zodpovedajúcimi bibliografickým odkazom podľa normy STN ISO 690, ktoré sa viažu k téme zadania a preukazujú výskumnú povahu problému a jeho aktuálnosť (uveďte všetky potrebné údaje na identifikáciu zdroja, pričom uprednostnite vedecké príspevky v časopisoch a medzinárodných konferenciách)

⁴ Nehodiace sa prečiarknite

Čestné prehlásenie

Čestne vyhlasujem, že som túto diplomovú prácu vypracoval samostatne, iba s použitím uvedenej literatúry a podľa usmernení vedúceho diplomovej práce.

V Bratislave, apríl 2019

.....

Bc. Michal Hucko

Podakovanie

V prvom rade sa chcem poďakovať pani profesorke Bielikovej, ktorá svojimi nápadmi, časom a najmä vynikajúcim prístupom významne dopomohla k vypracovaniu tejto diplomovej práce. Taktiež ďakujem členom skupiny Pewe, ktorí si pravidelne našli čas a konzultovali so mnou môj postup a tak svojimi nápadmi posunuli kvalitu mojej práce vždy o kus ďalej. Ďakujem všetkým vo firme Seges, kde som mohol pracovať a vyvíjať svoju diplomovú prácu v produkčnom prostredí. Chcem tiež poďakovať spolužiakovi Matejovi Váľkymu, s ktorým som pracoval na používateľskej štúdii a na všetkých implementačných ťažkostiach, ktoré nastali. Na záver ďakujem svojim priateľom, rodine a hlavne mojej priateľke Katke za ich podporu počas vypracovania tejto práce.

Bc. Michal Hucko

Anotácia

Slovenská technická univerzita v Bratislave

FAKULTA INFORMATIKY A INFORMAČNÝCH TECHNOLOGIÍ

Študijný program: Inteligentné softvérové systémy

Autor: Bc. Michal Hucko

Diplomová práca: Identifikácia zmätenia používateľa vo webovej aplikácii

Vedúci diplomovej práce: prof. Ing. Mária Bieliková, PhD

apríl 2019

Webové aplikácie sú denne navštevované každým z nás. Zastúpenie ich používateľov je pestré, od expertov až po začínajúcich. Mnohokrát sa stane, že pri napĺňaní svojej potreby je používateľ zmätený. Príkladom môže byť osoba, ktorá aplikáciu navštíví po prvýkrát. Zmätenie sa však môže týkať aj ľudí, ktorí aplikáciu poznajú. Napríklad v prípade aktualizácie alebo len skúšania inej ako bežnej funkcionality. V tejto práci sa venujeme predikcii zmätenia používateľa webovej aplikácie s využitím dát z implicitnej spätnej väzby. Konkrétne nami navrhnutá metóda pracuje s interakčnými dátami používateľa z myši a okulografu. V práci sa tiež zameriavame na odhalenie momentu, kedy je vhodné používateľovi poskytnúť nápovedu, ktorá by mu pomohla pri naplnení svojho cieľa v aplikácii. Na natrénovanie modelu strojového učenia sme zostavili používateľskú štúdiu. Navrhnutú metódu sme na získaných dátach aj overili. Súčasťou práce je porovnanie modelu s využitím myši s modelom z dát z okulografu.

Annotation

Slovak University of Technology Bratislava
FACULTY OF INFORMATICS AND INFORMATION TECHNOLOGIES
Degree course: Intelligent Information Systems

Author: Bc. Michal Hucko
Master Thesis: Identification of user confusion in a web application
Supervisor: prof. Ing. Mária Bieliková, PhD
2019, Apríl

Web applications are daily visited by everyone of us. There is a variety of users from the novice ones to the experts. While fulfilling our needs we might be confused, many times. For example, it might be in the situation where we face the application for the very first time. But confusion might also occur dealing with application we are experienced with, while trying some new feature or after the update. These situations might be problematic. In this work we deal with the automatic identification of the user confusion in a web application using data from the user implicit feedback. Proposed method works with the interaction data from the mouse and the eyetracker. We focus on identification of the right moment to display a guide to the user. This guide may consist of hints which will explain how to work with the certain web application. We conduct a user study to train our machine learning model. We evaluate our method on data gathered at the study. In this work we also provide comparison between our approach and the model using only the eyetracking data.

Obsah

1	Úvod	1
2	Prístupy k detekcii stavov používateľa v aplikáciách na webe	3
2.1	Explicitná spätná väzba	3
2.2	Implicitná spätná väzba	4
2.3	Detekcia zmätenia používateľa	6
2.4	Zhrnutie a diskusia	8
3	Metriky správania používateľa na webe	11
3.1	Metriky myši	11
3.1.1	Metriky myši založené na pohybe	12
3.1.2	Metriky myši založené na klikoch a váhavosť	13
3.1.3	Metriky myši založené na pauzách	14
3.2	Metriky klávesnice	15
3.3	Histogramová reprezentácia črt	15
3.4	Sledovanie pohľadu	15
3.4.1	Okulograf	16
3.4.2	Zrenička	16
3.4.3	Oblasti záujmu	17
3.4.4	Metriky sledovania očí	17
3.5	Metriky HTTP serveru	17
3.6	Zhrnutie a diskusia	19
4	Predikčný model	21
4.1	Klasifikácia v reálnom čase	21
4.2	Logistická regresia	21
4.3	Náhodný les	22
4.4	Vyhodnotenie modelov	24
4.4.1	Presnosť a úplnosť	25
4.4.2	Plocha pod krivkou charakteristiky prijímajúceho operátora	25
4.4.3	Akeikeho informačné kritérium	26
4.5	Zhrnutie a diskusia	27
5	Návrh metódy predikcie zmätenia	29
5.1	Predspracovanie dát logeru myši a extrakcia črt	29
5.2	Predspracovanie dát okulografu	31
5.3	Trénovanie modelov a vyhodnotenie	32
6	Používateľská štúdia a overenie	33
6.1	Pilotná štúdia	33
6.1.1	Vyhodnotenie implicitnej spätnej väzby	34

6.1.2	Vyhodnotenie infraštruktúry logovania	34
6.2	Hlavná štúdia	35
6.2.1	Získavanie spätnej väzby	36
6.2.2	Mapovanie identít účastníka	37
6.3	Exploratívna analýza surových dát	38
6.3.1	Získané dáta	38
6.3.2	Tlačidlo zmätenia	38
6.4	Exploratívna analýza metrík	41
6.4.1	Metriky myši	42
6.4.2	Metriky okulografu	43
6.5	Výsledky predikcie	44
6.5.1	Výsledky nad celou úlohou	45
6.5.2	Výsledky nad časovými oknami	46
7	Zhodnotenie	49
	Literatúra	51
	Príloha A - Protokol pilotnej používateľskej štúdie	A-1
	Príloha B - Protokol používateľskej štúdie	B-1
	Príloha C - Loger myši a predikcia zmätenia v aplikácii YesElf	C-1
	Loger	C-2
	Predikcia v reálnom čase v aplikácii YesElf	C-3
	Príloha D - Opis digitálnej časti práce	D-1
	Príloha E - IITSRC článok a plagát	E-1
	Príloha F - UMAP článok (ACM Press, prijatý do sekcie Late Breaking Results)	F-1
	Príloha G - Návrh článku na konferenciu	G-1
	Príloha H - Zhodnotenie plánov práce	H-1
	Príloha I - Technická dokumentácia	I-1

1 Úvod

Webové aplikácie sú neodmysliteľnou súčasťou každého používateľa počítačov a iných výpočtových zariadení. Mnohé z nich denne navštevujú milióny ľudí. Môžeme cez ne nakupovať, pracovať, vykonávať bankové transakcie, študovať, registrovať sa alebo inak naplňať svoje potreby. Viaceré webové aplikácie poskytujú mnoho z týchto funkcií naraz. Používateľ po príchode na sídlo aplikácie mnohokrát netuší, čo všetko mu je poskytnuté. Jeho cieľom v danom čase je naplniť svoje potreby a to čo najrýchlejšie. Vynaloženie príliš veľkého úsilia zo strany používateľa môže viesť k strate záujmu o aplikáciu.

Jedným z hlavných cieľov webových aplikácií je vo všeobecnosti naplnenie potrieb návštevníka. Na to, aby sa tento cieľ podarilo uskutočniť musí byť vytvorené prostredie, ktoré je zrozumiteľné. Problémovou skupinou v tomto smere sú používatelia, ktorí navštevujú sídlo po prvýkrát. Takýto používatelia mnohokrát nevedia čo všetko im sídlo ponúka, a tak sa často nedokážu pohnúť ďalej alebo dokonca začať. Avšak spomínaný stav môže nastať aj v prípade skúsenejších používateľov. Príkladom môže byť objavovanie novej funkcionality alebo len interakcia s aplikáciou. Tu opäť zohráva dôležitú rolu čas potrebný na získanie požadovanej informácie. Ak je prekročená znesiteľná hranica zákazník mnohokrát aplikáciu opustí, čo môže viesť k strate z pohľadu vlastníka (používateľ si napríklad zakúpi inú službu). V práci [42] sa tento stav nazýva dezorientácia (angl. *disorientation*) alebo zmätenie (angl. *confusion*).

Zmätenie je spojené so zážitkom používateľa a je zjavne dôležité pracovať na jeho minimalizovaní. Z pohľadu používateľského zážitku (angl. *user experience*) existuje mnoho ciest ako dosiahnuť tento cieľ. Sú to napríklad dotazníky, rozhovory, ankety, kvalitatívne experimenty a iné metódy založené na individuálnom prístupe a zbieraní explicitnej spätnej väzby. Počas nich je správanie sledované pri práci s prostredím s cieľom identifikovať nedostatky aplikácie. Tieto metódy sa však veľmi ťažko škálujú. Ich analýzou potom vieme odstrániť problémové časti aplikácie. Treba si však uvedomiť, že častokrát nežiadúci stav (dezorientácia, zmätenie) ani nemusí byť dôsledkom nejakého problému v návrhu, ale komplexnosti aplikácie. Stav môže byť spôsobený slabšou webovou gramotnosťou používateľa alebo fyziologickými faktormi akým je napríklad únava, či len nešťastnou náhodou.

V našej práci sa venujeme identifikácii zmätenia používateľa webovej aplikácie. Prezentujeme metódu, ktorá je založená na automatizovanej predikcii tohto stavu. Metóda vychádza zo správania, ktoré je možné zaznamenať pri interakcii používateľa s aplikáciou. Pracujeme s udalosťami získanými z používania myši a okulografu. Jedná sa teda o implicitnú spätnú väzbu. Dáta z myši vieme technicky jednoducho získať, ak predpokladáme prístup ku zdrojovému kódu samotnej aplikácie. Pri navrhovaní metódy sme vychádzali z prác, ktoré sa tejto alebo podobnej problematike venujú [29, 28, 42, 36, 35].

Nami navrhnutá metóda môže byť použitá pri online pomocníkoch (angl. *guides*). Ich úlohou je vysvetliť používateľovi kroky s cieľom naplnenia jeho potrieb. Problémom pomocníkov je neznalosť stavu používateľa. Ich pravidlá sa dajú generalizovať na odporúčania tých istých krokov pre všetkých používateľov, prípadne nadviazať na spúšťač (časový, udalostný). Kvôli špecifickým

potrebám zákazníkov je však tento prístup nepostačujúci. Príkladom takéhoto pomocníka môže byť aplikácia *YeseEf*¹ od firmy *Brainware*, ktorá svojím jednoduchým rozhraním umožňuje používateľom navrhnúť týchto pomocníkov a jednoducho ich integrovať s už existujúcou webovou aplikáciou. V rámci tejto práce navrhujeme metódu predikcie zmätenia v reálnom čase, ktorú je možné priamo integrovať s nástrojom *YesElf*.

V kapitole 2 našej práce sumarizujeme prístupy v získavaní explicitnej a implicitnej spätnej väzby používateľa webovej aplikácie. V tejto kapitole tiež predstavujeme existujúce prístupy v identifikácii zmätenia používateľa. Ďalej sa v tejto kapitole zaoberáme sprievodcami ako nástrojmi, ktoré riešia situáciu zmätenia poskytnutím rady. V kapitole 3 poskytujeme prehľad metrík k zaznamenávaniu správania používateľa pomocou klávesnice, myši a okulografu. V kapitole 4 prezentujeme prístupy strojového učenia na riešenie klasifikačných problémov. Sumarizujeme v nej aj metriky vyhodnotenia takýchto modelov. Kapitola 5 predstavuje nami navrhnutú metódu predikcie zmätenia. V kapitole 6 predstavujeme používateľskú štúdiu, ktorú sme uskutočnili s cieľom overenia výsledkov. V tejto kapitole tiež dokumentujeme výsledky.

¹www.yeself.com/

2 Prístupy k detekcii stavov používateľa v aplikáciách na webe

Úspech tej ktorej webovej aplikácie závisí aj od toho ako sa podarí naplniť potreby používateľa. Na to, aby sme mohli identifikovať problematické miesta je potrebné získať od používateľa spätnú väzbu. V tejto kapitole sa venujeme získavaniu spätnej väzby s cieľom identifikovať stav zmätenia. Sumarizujeme niekoľko postupov dokumentovaných v prácach autorov, ktorí sa venujú práve problému s používateľským zážitkom a zmätením.

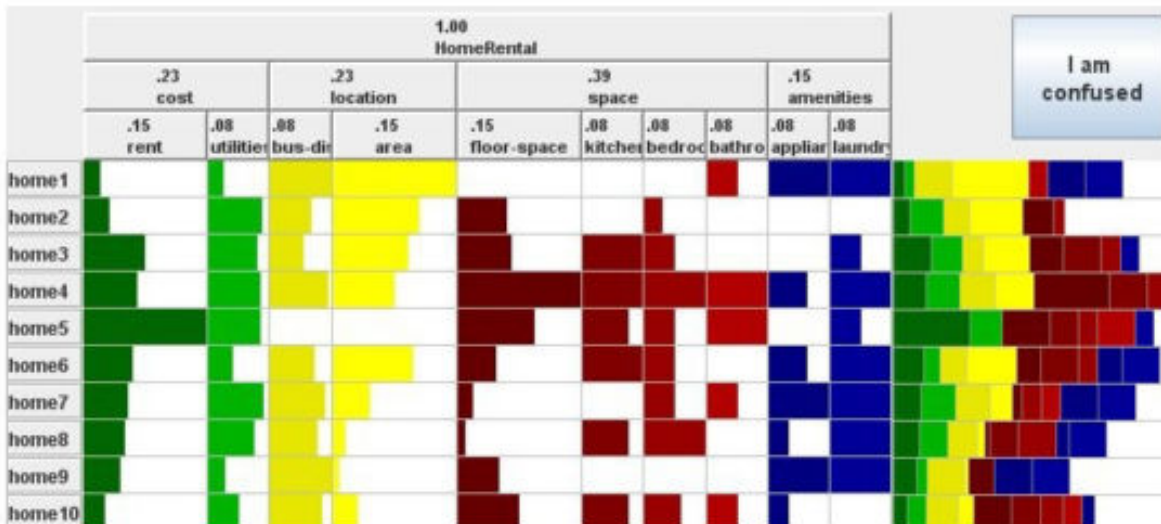
Detekcia stavov používateľa sa realizuje spracovaním jeho spätnej väzby. Tu rozlišujeme explicitnú spätnú väzbu (podkapitola 2.1) a implicitnú (podkapitola 2.2). Z možných stavov používateľa sa sústreďíme na stav zmätenia (podkapitola 2.3).

2.1 Explicitná spätná väzba

Získanie spätnej väzby od používateľa je predpokladom k odhaleniu chýb, ktoré v aplikácii vznikajú. Jedným z jej druhov je explicitná spätná väzba. V tomto prípade ide o priame vyjadrenie názoru na danú problematiku. Získanie veľkého množstva tejto spätnej väzby od jednotlivca môže byť veľmi náročné kvôli nadmernej kognitívnej záťaži, ktorá z tejto úlohy vyplýva. Taktiež, v prípade veľkého množstva používateľov môže byť jej získanie veľmi nákladné kvôli problematickej škálovateľnosti.

Výstupom explicitnej spätnej väzby sú zväčša odpovede na dotazníky, ktoré sú tvorené niekoľkými otázkami. Autori otázok sa zameriavajú primárne na rýchlosť pri vyplňaní otázok. Príkladmi druhov otázok na získavanie explicitnej spätnej väzby sú:

- *Likertova škála.* [4] Ide o škálu na vyjadrenie intenzity názoru. Samotná škála vznikne spočítaním odpovedí pre jednotlivé prvky škály.
- *Hviezdičkový systém.* V tomto prípade sa jedná o iný druh vizualizácie škály. Hviezdičky majú široké využitie pri systémoch na odporúčanie produktov [9]. Používateľ vyjadrí svoju spokojnosť priradením určitého množstva hviezdíčiek.
- *Binárne hodnotenie.* V tomto prípade ide o vyjadrenie súhlasu alebo nesúhlasu na otázku. Obrázok 1 ukazuje použitie tlačidla v pravom hornom rohu v práci [28] na vyjadrenie zmätenia (túto prácu analyzujeme v nasledujúcich kapitolách).



Obr. 1: Príklad binárneho hodnotenia na vyjadrenie zmätenia [28].

Jednou z hlavných výhod explicitnej spätnej väzby v porovnaní s implicitnou spätnou väzbou je presnosť. Používateľ totiž jasne definuje svoj názor. V práci [21] sa autori venujú jej charakteristikám v prípade odporúčacieho systému pre rádiové služby. Pracujú so systémom, ktorý používateľom dovolí označiť skladbu ako obľúbenú (angl. *loved*) alebo neželanú (angl. *banned*). V experimente zozbierali dáta od viac ako 10 000 používateľov so záznamami o viac ako milióno skladieb. Výsledky ich experimentu ukazujú, že použitie explicitnej spätnej väzby klesá s časom využitia aplikácie. Autori sa domnievajú, že tento jav môže byť zapríčinený tým, že používatelia nevidia zmysel v jej odovzdávaní.

V práci [43] sa autori venujú porovnaniu explicitnej a implicitnej spätnej väzby v prípade vyhľadávania na webe. Ich cieľom bolo zistiť či implicitná spätná väzba môže nahrádzať explicitnú. Zostrojili experiment so 16 účastníkmi, kde úlohou bolo ohodnotiť výsledky vyhľadávania pre daný dopyt. Systém vždy vrátil prvých 30 dokumentov vyhľadávачa podľa relevantnosti. V prípade explicitnej spätnej väzby, používateľ len zvolil či je daný výsledok relevantný. Pri implicitnej variante sledovali správanie používateľa pri interakcii s výsledkami. Za relevantný sa v tomto prípade považoval dopyt, pri ktorom si účastník vyžiadal detailnejšie zobrazenie. Výsledky ich experimentu hovoria o tom, že dáta explicitnej a implicitnej spätnej väzby neboli rozdielne. Hypotézu overili štatistickým testom.

2.2 Implicitná spätná väzba

Pri implicitnej spätnej väzbe sa dáta získavajú zo správania používateľa bez toho aby používateľ priamo poskytoval nejakú informáciu. Získanie interpretácie implicitnej spätnej väzby, t.j. jej vyhodnotenie však v mnohých prípadoch nie je triviálna záležitosť. Na jej úspešné vyhodnotenie potrebujeme mnoho dát o správaní používateľa. Dôležité je označiť a teda vyhodnotiť správa-

nie používateľa, ktoré tieto dáta opisujú. Mnohokrát sa na označenie správania používa práve explicitná spätná väzba [42, 29].

V práci [22] sa autori venujú vyhodnocovaniu dát z prechádzania webom a reformulácie dopytov vo vyhľadávaní na webe. Okrem toho sa venujú generovaniu trénovacej sady na učenie s použitím implicitnej spätnej väzby. Na základe používateľskej štúdie, kde účastníkom prezentovali zoznam položiek pre zadaný dopyt, zistili ako jednotlivci interagujú s výsledkami vyhľadávača. Na zaznamenanie správania účastníka používali okulograf. Výsledky potom prispeli k vytvoreniu stratégie získavania spätnej väzby z klikov. Podľa nej zhotovili používateľskú štúdiu, kde na overenie použili explicitnú spätnú väzbu z dotazníka na doménovú znalosť vyhľadávača. V prvej časti položili účastníkom 10 otázok. 5 bolo navigačných a 5 bolo informatívnych. Štúdia prebehla v dvoch fázach. V prvej ju absolvovalo 34 študentov (vek 18-23), kde všetci boli motivovaní extra bodmi. Druhá časť štúdie prebehla s tými istými otázkami, kde študenti boli vybraní tým istým spôsobom, len prezentácia výsledkov bola v troch rôznych poradiach:

- *Normálne poradie.* Výsledky vyhľadávača neboli upravené.
- *Zamenené poradie.* Prvé dva výsledky vyhľadávača podľa priority boli vymenené.
- *Opačné poradie.* 10 výsledkov vyhľadávača bolo v opačnom poradí.

Hlavné zistenie práce poukazuje na to, že spätná väzba z klikov je vychýlená najmenej v dvoch smeroch:

- *Vychýlenie na základe dôveryhodnosti* (angl. *trust bias*). Ľudia klikajú viac na odkazy, ktoré vyhľadávač umiestni na vyššiu pozíciu aj napriek tomu, že ich abstrakt nie je relevantný oproti ostatným prezretým odkazom.
- *Vychýlenie na základe kvality kontextu* (angl. *quality-of-context bias*). Autori ukazujú, že klikanie na odkazy nie je ovplyvnené len relevantnosťou abstraktu odkazu, ale aj odkazmi, ktoré ho obklopujú a ich abstraktami.

V článku [32] sa autori venujú spracovaniu logov zo serverov s cieľom identifikácie vzorov, ktoré sa v nich často vyskytujú. Na ich následnú analýzu používajú metódy strojového učenia. Nástroj, s ktorým pracujú sa nazýva *WebTool*. Cieľom tejto práce je na základe týchto vzorov správania reštruktúrovať stránku za účelom zväčšenia efektivity používateľov pri vyhľadávaní informácií. Ukazujú postup, v ktorom využívajú profily používateľov. Na automatické odhaľovanie profilov používajú asociačné pravidlá (angl. *association rules*) a sekvenčné vzory (angl. *sequential patterns*). Príklad asociačného pravidla: '50% návštevníkov, ktorí prišli na linky A a B navštívili aj C'. Príkladom sekvenčného vzoru môže byť: '60% klientov, ktorí navštívili A a B v jednej transakcii, taktiež navštívili C za nasledujúci mesiac'. Ich metóda ukazuje, že použitím analýzy sekvencií na zozbieraných logoch vedia odhaliť vzory v návštevách používateľa. Autori taktiež ukazujú využitie tejto informácie pri štruktúrovaní informácií na stránke.

2.3 Detekcia zmätenia používateľa

V tejto práci sa venujeme zmäteniu používateľa. Pred samotnou analýzou existujúcich prístupov detekcie zmätenia definujeme pojem zmätenie. Výkladové slovníky charakterizujú slovo zmätenie ako neistotu v tom, čo sa má diať alebo v tom, čo je požadované. Slovník *Webster*² charakterizuje zmätenie ako situáciu, v ktorej sú ľudia neistí v tom, čo robiť alebo v tom, ako situáciu pochopiť. Dôsledkom zmätenia na webe je neschopnosť používateľa plynule pokračovať v napĺňaní svojho cieľa. Práve toto vedie k znechuteniu používateľa.

V práci [42] označujú tento stav slovom obtiažny (angl. *struggling*). Vo svojej práci prezentujú nástroj *LATTE* na automatické odhalovanie obtiažnych sedení používateľa. Cieľom nástroja je zlepšenie používateľského zážitku aplikácie, do ktorej je nasadený skrz odporúčania na vylepšenie. Na vytváranie modelu používajú celé používateľské sedenia, v ktorých identifikujú signály založené na správaní používateľov indikujúce obtiažne sedenia. Výstupom ich nástroja sú rady pre vlastníkov aplikácií, v ktorých identifikujú problémové miesta, prípadne mäťúce linky aj iné nedostatky. Pri hľadaní signálov autori vychádzali z kontrolovanej štúdie, kde zaznamenávali výstupy myši, klávesnice a adresy stránok, ktoré účastníci navštívili. Dokopy mali 22 účastníkov. Počas štúdie sa sústredili primárne na jednoduché metriky, ktoré dokážeme na stránke zaznamenať bez použitia rozšírenia. Práve tieto metriky sa v ich výsledkoch javia ako najlepšie ukazovatele obtiažnych sedení. Menovite ide o celkový čas sedenia, celkový počet stránok sedenia a čas na stránku (všetkých presnosť lepšia ako 0,75 pri predikovaní obtiažneho sedenia). Ich kombináciou pomocou logistickej regresie dosiahli presnosť až 0,86 pri predikcii. Okrem toho skúmali aj predikciu s využitím metrick, ktoré vieme zaznamenať rozšírením aplikácie pomocou jazyka *javascript*. Tu sa najlepšie ukázala metrika počtu skrolov (presnosť až 0,84) a metrika pohybu myši (presnosť 0,74). Vo výsledkoch autori ukazujú, že je možné pre každú metriku vypočítať hraničnú hodnotu, ktorej presiahnutie by indikovalo obtiažne sedenie. Príkladom je, že pri presiahnutí 12 stránok na úlohu sa sedenie klasifikuje ako zmätené.

V práci [28] sa autori zameriavajú práve na identifikáciu zmätenia používateľa s využitím okulografu na stránke s vizualizáciami grafov. V ich prípade sa jednalo o grafy domov a ich atribútov. Pre každý atribút bolo zobrazené ako veľmi ovplyvňuje cenu domu. Autori zostavili štúdiu, ktorá pozostávala z úloh nad touto stránkou. Podarilo sa zozbierať záznamy zo 136 sedení, kde účastníci vypracovali sadu úloh. Jednalo sa o úlohy vyhľadávania vo vizualizácii (nájdite najlacnejší dom alebo nájdite najlepší dom podľa pomeru rozloha a cena). Dokopy každý účastník prešiel 40 úloh. Počas celého sedenia bol pohľad účastníka zaznamenávaný okulografom *Tobii T120*, v miestnosti bez okien (kvôli rovnakej veľkosti zreničky). Na označenie mäťúcich úloh nechali používateľa stlačiť tlačidlo v prípade, že sa cítil zmätený, pričom pri jeho návrhu vychádzali z práce [7]. Pred experimentom inštruovali účastníkov o použití tohto tlačidla. Práca dokumentuje presnosť modelu 0,61.

V práci [36] autori predikujú zmätenie s využitím dát myši v počítačovej hre. Hra bola inšpirovaná adventným kalendárom a pozostávala z hracej plochy s 24 očíslovanými dlaždicami rôznej veľkosti a rôzneho umiestnenia. Cieľom hry bolo postupne odklikať dlaždice v číselnom poradí.

²www.merriam-webster.com/

Autori si proces riešenia hry prešli so štyridsiatimi štyrmi vybranými účastníkmi. Prechod od jednej dlaždice k druhej označili ako jeden pohyb. Po absolvovaní hry účastníci priradili každému z 24 pohybov hodnotu likertovej škály (od 1 do 7), ktorá reprezentovala stav zmätenia (7 pre maximálne zmätenie). Spolu zozbierali 1056 pozorovaní. Už pri úvodných pozorovaniach našli vysokú mieru korelácie (0,86 pearsonov korelačný koeficient) medzi štandardizovanými hodnotami času pohybu a priradenej hodnoty zmätenia. Keďže účastníci spätne neboli schopní označiť začiatok a koniec zmätenia, na výber časového úseku vzorky pre zmätenie použili nasledujúce pravidlá: pre nezmätené pozorovanie zobrali druhú polovicu pohybu a pre zmätené pozorovanie prvú polovicu (vychádzajúc z faktu, že pohyb vždy skončil v cieľi). Následne dáta myši použili na výpočet 33 črt založených na zmenách smerov pri pohybe. Na predikciu natrénovali 4 klasifikátory, kde najlepšiu metriku F1 skóre 0.946 dosiahol algoritmus podporných vektorov strojov (angl. *support vector machines algorithm*).

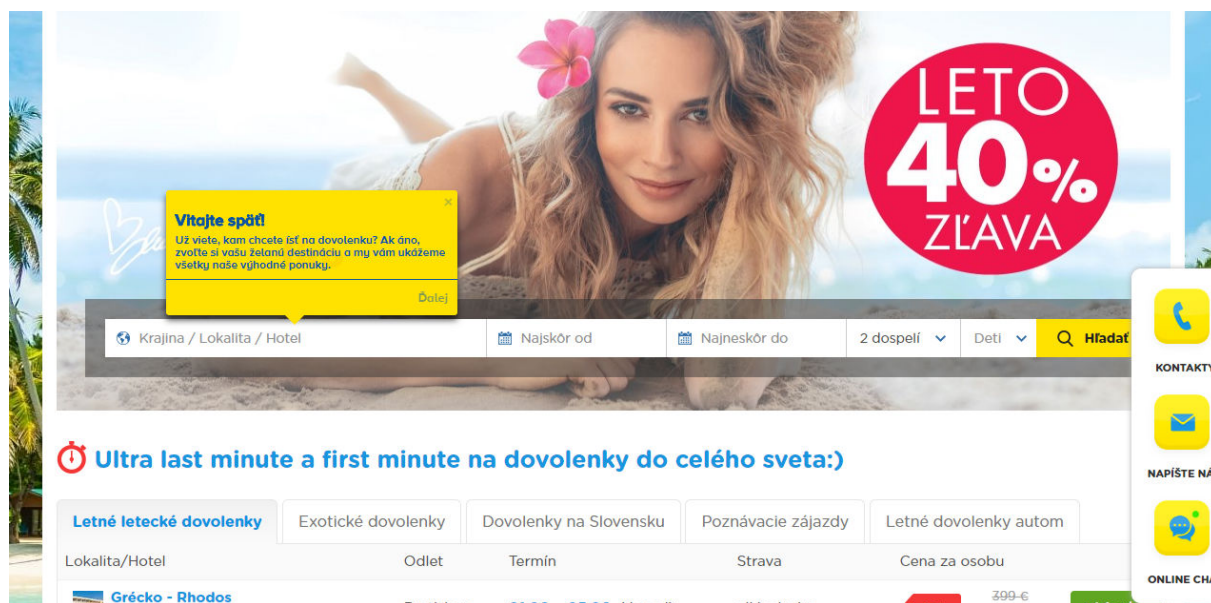
Tí istí autori analyzujú využitie predchádzajúceho prístupu založenom na zmenách smerov v správaní myši pri webovej aplikácii v práci [35] s využitím tej istej hry. V tomto článku využívajú oveľa väčšiu vzorku na overenie svojej hypotézy (916 hier od 282 používateľov dokopy 21984 záznamov). Na zaznamenanie skutočnej hodnoty zmätenia aplikujú protokol hlasného rozprávania (angl. *think aloud protocol*). V rámci protokolu nahrávajú hlas účastníka, ktorý opisuje postup riešenia úlohy a emócie s tým spojené. Z týchto nahrávok potom označovali úseky zmätenia. V tomto prípade sa im podarilo dosiahnuť najvyššiu hodnotu metriky F1 skóre 0,938 pri použití algoritmu náhodných lesov (angl. *Random forest algorithm*). V závere práce autori poukazujú na možnosť využitia tejto metódy na predikciu zmätenia pri prechádzaní jednotlivými elementami stránky, avšak upozorňujú na nefunkčnosť pri úlohách s čítaním rozsiahlejších textov. Tvrdia, že v takom prípade niektorí ľudia zvyknú nasledovať text kurzorom, čo môže vytvárať vzory rozpoznávané ako zmätenie. Ďalšou z limitácií ich riešenia bola implementácia logeru udalostí, ktorý zaznamenal pohyb len pri prekonaní vzdialenosti myšou viac ako 10 pixelov. Z toho vypláva, že nie všetky pohyby boli zaznamenané.

Identifikácia stavu zmätenia používateľa nám môže pomôcť odhaliť problém, ktorý má používateľ a tak buď vylepšiť webovú aplikáciu analýzou tohto problému, alebo vylepšiť používateľský zážitok v konkrétnom kontexte. Tento kontext však nie vždy vieme vopred identifikovať. V prípade práce [42] autori vytvorili nástroj, ktorý označí miesta, ktoré tento problém spôsobujú. Vlastník stránky potom môže tieto miesta vhodne opraviť. V spomínanej práci [32] zase autori poskytnú prehľad o ceste používateľa v stránke, čo tiež môže prispieť k reštrukturalizácii obsahu webu.

Iné riešenie môže byť poskytnutie nápovedy s inštrukciami pre daný problém. To by však znamenalo identifikovanie samotného problému, čo je mimo rozsah tejto práce. Tieto nápovedy však môžu byť všeobecného charakteru týkajúce sa domény stránky. Tomuto postupu sa venujú sprievodcovia (angl. *guides*). Uvedením správnej nápovedy v čase zmätenia používateľa by sme vedeli zabrániť negatívnej skúsenosti. Príkladom softvéru na vytváranie nápovedí je aplikácia *Yeself*³. Aplikácia umožňuje vloženie nápovedy do stránky pomocou jazyka *javascript*. Obrá-

³www.yeself.com

zok 2 poskytuje ukážku využitia produktu na stránke *Firotour*⁴ (stránka s dovolenkami). Bublina s obsahom, ktorý radí ako využiť vyhľadávacie pole na stránku je navrhnutá priamo doménovým expertom.



Obr. 2: Ukážka použitia nápovedy na webovej stránke.

2.4 Zhrnutie a diskusia

V tejto kapitole sme poskytli prehľad možností získavania explicitnej a implicitnej spätnej väzby používateľa s cieľom odhalenia zmätenia vo webovej aplikácii. Explicitná spätná väzba sa javí presnejšia. Ukázali sme však problém s jej získavaním a škálovaním. Naopak pri implicitnej spätnej väzbe nemusíme dosiahnuť očakávanú prenosnosť výsledkov. Taktiež pri nej čelíme problémom spojenými s jej interpretáciou. Ďalším z problémov pri explicitnej spätnej väzbe je jej získavanie. Ak je realizované formou nesprávne konštruovaných otázok môže to veľmi narušiť jej presnosť. Opýtaní si môžu mylne interpretovať znenie týchto otázok a preto nesprávne odpovedajú.

Zaujímavý prístup pri získavaní explicitnej spätnej väzby zmätenia používateľa poskytla práca [28]. Tlačidlo "som zmätený" však nemusí vrátiť očakávané výsledky v prípade, že používateľ nevie presne čo to zmätenie znamená. Autori identifikovali tento problém a riešili ho vhodnými inštrukciami. Taktiež po každom sedení si s účastníkmi prešli momenty, kedy bolo tlačidlo stlačené. Týmto postupom predišli nevhodnej interpretácii zaznamenaných dát. Ďalšie problémy s tlačidlom sa týkajú umiestnenia veľkosti a dizajnu samotného tlačidla, ktoré môžu ovplyvniť používateľa. Táto práca ukazuje využitie explicitnej spätnej väzby na natrénovanie predikčného modelu nad implicitnou z okulografu.

⁴www.firotour.sk/

Pri implicitnej spätnej väzbe v práci [22] môžeme vidieť, že autori poukázali na vychýlenosť výsledkov založených na klikoch myši. Môžeme teda predpokladať, že na vhodnú analýzu zmätenia používateľa potrebujeme viac ako len kliknutia, ktoré počas sedenia nastali. Na druhej strane však práce [42, 32] ukazujú, že na odhalenie stavu používateľa často-krát postačujú udalosti, ktoré vieme zaznamenať jednoduchým HTTP serverom.

V tejto kapitole sme taktiež analyzovali zmätenie ako také. Z existujúcich prác vidieť, že autori neposkytujú definíciu tohto pojmu alebo ho definujú každý po svojom. Taktiež sme si všimli, že autori označujú tento pojem ako obtiažnosť alebo neistotu.

3 Metriky správania používateľa na webe

Pri interakcii používateľa s aplikáciou vieme vyhodnotiť správanie pomocou záznamov aplikácie (angl. *logs*). Samotné záznamy však môžu byť uchovávané v rôznej forme. Preto sa pre popis správania používajú metriky alebo hodnoty, ktoré sú odvodené od nich. V tejto kapitole sa venujeme metrikám. Poskytujeme prehľad prístupov pri vytváraní metrick zo záznamov okulografu a logeru.

V podkapitole 3.1 sa venujeme myši, ako zdroju metrick. Umiestnením vhodného skriptu do stránky vieme jednoducho zbierať záznamy. Z tohto dôvodu vieme použiť tieto dáta aj pri reálne nasadenej aplikácii mimo kontrolovaného prostredia. Podobne sú na tom metriky uvedené v podkapitole 3.2 spojené s klávesnicou. Diskutujeme ich reprezentáciu pomocou histogramu, ktorá sa využíva v oblasti autentifikácie používateľa. Keďže pohľad používateľa je dôležitý zdroj implicitnej spätnej väzby a v našom laboratóriu máme prístup k zariadeniam na sledovanie pohľadu (okulograf), poskytujeme tiež prehľad metrick odvodených z pohľadu používateľa (3.4). Dobrým zdrojom dát pre webové aplikácie sú aj metriky *HTTP* serveru, ktoré krátko diskutujeme v podkapitole 3.5.

3.1 Metriky myši

Počítačová myš je jedným z hlavných zariadení pri interakcii človeka s webovou aplikáciou. Pomáha pri jednoduchom prechádzaní vrámci jednej alebo viacerých stránok sídla aplikácie. Medzi hlavné akcie myši patria:

- Pohyb myši (angl. *mouse movement*). Prejavuje sa presúvaním kurzora po obrazovke v rovnakom smere ako sa pohybuje ruka s myšou.
- Kliknutie myši (angl. *mouse click*). Akcia sa vykoná po interakcii používateľa s jedným z tlačidiel myši. V zásade môže ísť o ktorékoľvek z tlačidiel. Pri bežných myšiach rozlišujeme:
 - Lavý klik (angl. *left click*).
 - Pravý klik (angl. *right click*).
 - Dvojklik (angl. *double click*). Akcia, ktorá je vyvolaná dvomi kliknutiami za sebou (zväčša sa jedná o lavý klik), medzi ktorými je čas menší ako hraničná doba δ .
- Skrol myši (angl. *mouse scroll*). Akcia sa vykoná po interakcii používateľa so skrolovacím kolieskom myši.
- Ťahaj a pusť (angl. *drag and drop*). Akcia medzi stlačením tlačidla myši a pustením tlačidla myši (zväčša lavý klik), spojená s pohybom myši nad hraničnú dobu δ .

Keďže počas interakcie používateľ generuje veľké množstvo týchto záznamov (v prípade pohybov myši, môže ísť až o niekoľko desiatok za sekundu), v praxi sa používajú metriky, ktoré sa získajú pomocou výpočtov z týchto "hrubých dát"(angl. *raw data*).

3.1.1 Metriky myši založené na pohybe

Väčšina akcií myši pri interakcii používateľa s webovou aplikáciou je tvorená práve záznamami o pohybe myši. Je preto dôležité opísať tieto udalosti vhodnými metrikami, ktoré potom charakterizujú správanie používateľa v určitých časových úsekoch.

V prácach [15, 16] sa autori snažia autentifikovať používateľa na základe interakcie v počítačovej hre s dlaždicami. Počas hrania boli akcie myši zaznamenané na serveri. Následne na to boli zo záznamov extrahované pohyby myši používateľa. Pod pojmom pohyb autori rozumejú sadu koordinátov myši v časovom úseku kde rozdiel medzi jednotlivými časovými pečiatkami je menší ako hraničná hodnota γ (autori použili $\gamma = 120ms$). Každý z pohybov musí obsahovať viac ako 4 záznamy myši ináč sa vyradí. Následne bol pohyb charakterizovaný vektormi:

- $\mathbf{t} = \{t_i\}_{i=1}^n$ - Vektor časov jednotlivých vzoriek.
- $\mathbf{x} = \{x_i\}_{i=1}^n$ - Vektor x-ových súradníc jednotlivých vzoriek.
- $\mathbf{y} = \{y_i\}_{i=1}^n$ - Vektor y-ových súradníc jednotlivých vzoriek.

Kde $\delta x_k = x_{k+1} - x_k$ a $\delta y_k = y_{k+1} - y_k$ sú zmeny v jednotlivých súradniciach. Neskôr boli vypočítané metriky. V tabuľkách 1 a 2 môžeme vidieť metriky, ktoré autori používali.

Tabuľka 1: Základné metriky odvodené z pohybu myši [15, 16].

	Názov metriky	Popis	Výpočet
1	Horizontálna rýchlosť	Rýchlosť vzhľadom na x-ovú os	$v_x = \frac{\delta x}{\delta t}$
2	Vertikálna rýchlosť	Rýchlosť vzhľadom na y-ovú os	$v_y = \frac{\delta y}{\delta t}$
3	Rýchlosť	Prvá derivácia dráhy podľa času	$v = \sqrt{v_x^2 + v_y^2}$
4	Zrýchlenie	Druhá derivácia dráhy podľa času	$\dot{v} = \frac{\delta v}{\delta t}$
5	Trhnutie	Tretia derivácia dráhy podľa času	$\ddot{v} = \frac{\delta \dot{v}}{\delta t}$

Tabuľka 2: Iné črty odvodené z pohybov myši [15, 16].

	Názov metriky	Popis	Výpočet
1	Dĺžka	Celková suma dĺžok pohybov	$s = \sum_{i=1}^n \sqrt{(x_i^2 + y_i^2)}$
2	Trvanie	Celková suma trvania pohybov	$t = \sum_{i=1}^n \delta t_i$
3	Rovnosť	Rovnosť pohybu (angl. <i>straightness</i>)	$S = \frac{\sqrt{(x_1 - x_n)^2 (y_1 - y_n)^2}}{s}$
4	Chvenie		$J = s'/s$

Okrem toho, sa na namerané pohyby aplikovalo kubické vyhladzovanie (angl *cubic spline smoothing*) čím sa vypočítali vyhladené hodnoty pre koordináty x', y' spolu s vyhladenou dráhou s' . Následne sa pre každú z nameraných metrick $x', y', v_x, v_y, v, \dot{v}, \ddot{v}$ vypočítali štatistické hodnoty pre každý pohyb:

- Maximum - Výber najvyššej hodnoty pre daný atribút.
- Minimum - Výber najnižšej hodnoty pre daný atribút.

- Priemer - Výber priemernej hodnoty pre daný atribút.
- Štandardná odchýlka - Štatistika charakterizujúca rozptyl hodnôt pre daný atribút.
- (Minimum - maximum) - Štatistika charakterizujúca rozdelenie hodnôt.

V práci [13] sa autori snažia odhaliť neautorizovaného používateľa aplikácie pomocou jeho správania sa prostredníctvom myši. Okrem vyššie spomenutých metrík použili aj ďalšie, ktorých výpočet a popis dokumentuje tabuľka 3. V práci [27] sú tieto metriky označené ako holistické (angl. *holistic*) a taktiež boli použité pri autentifikácii používateľa aplikácie.

Tabuľka 3: Holistické metriky odvodené z pohybu myši [13].

	Názov metriky	Popis	Výpočet
1	Centrum trajektórie masy	Suma čiastkových vzdialeností násobená prislúchajúcou dobou zmeny a predelená dĺžkou pohybu (angl. <i>Trajectory center of mass TCM</i>).	$TCM = \frac{1}{s_n} \sum_{i=1}^{n-1} t_{i+1} \sqrt{\delta x^2 + \delta y^2}$
2	Rozptylový koeficient	Suma čiastkových vzdialeností na druhú, násobená prislúchajúcou dobou zmeny a predelená dĺžkou pohybu na druhú (angl. <i>Scattering coefficient SC</i>).	$SC = \frac{1}{s_n} \sum_{i=1}^{n-1} t_{i+1}^2 \sqrt{\delta x^2 + \delta y^2} - TCM^2$
3	Tretí moment	TCM s tretou mocninou trvania zmeny (angl. <i>Third moment</i>).	$M_3 = \frac{1}{s_n} \sum_{i=1}^{n-1} t_{i+1}^3 \sqrt{\delta x^2 + \delta y^2}, k = 3$
4	Štvrtý moment	TCM so štvrtou mocninou trvania zmeny (angl. <i>Fourth moment</i>).	$M_4 = \frac{1}{s_n} \sum_{i=1}^{n-1} t_{i+1}^4 \sqrt{\delta x^2 + \delta y^2}, k = 4$
5	Zakrivenie zrýchlenia	Priemer z inverznej hodnoty zrýchlenia.	$VCrv = \frac{\ddot{v}}{(1+\ddot{v}^2)^{\frac{3}{2}}}$

3.1.2 Metriky myši založené na klikoch a váhavosť

Okrem pohybu myši je jednou z akcií, ktoré používateľ vykonáva klikanie. Tento jav nastane po stlačení tlačidla myši. Pozostáva z dvoch častí:

- *Stlačenia tlačidla*. Charakteristikou tohto javu sú hodnoty spojené s pozíciou kurzora x_{down}, y_{down} a časom udalosti t_{down} .
- *Uvoľnenie tlačidla*. Charakteristikou tohto javu sú hodnoty spojené s pozíciou kurzora x_{up}, y_{up} a časom udalosti t_{up} .

V práci [13] autori používajú tieto charakteristiky klikov na výpočet metrík, ktoré sú sumarizované v tabuľke 4. Opäť aj v tomto prípade sa vypočítajú hodnoty štatistík pre každú z metrík.

Ďalším zaujímavým pohľadom na spracovanie dát o klikoch je analýza váhavosti používateľa. V práci [38] autori identifikujú váhavosť používateľa webovej aplikácie na základe dĺžky páuz v správaní pred klikmi. Ich hypotéza je založená na tvrdení, že čím je dĺžka pauzy pred klikom

Tabuľka 4: Metriky založené na klikoch [13].

	Názov metriky	Popis	Výpočet
1	Trvanie	Čas medzi stlačením a uvoľnením tlačidla myši (angl. Duration CD).	$t_{up} - t_{down}$
2	Dĺžka	Dĺžka prejdená medzi stlačením a uvoľnením tlačidla (angl. Length CL).	$\sqrt{(x_{up} - x_{down}) + (y_{up} - y_{down})}$
3	Pauza pred kliknutím	Pauza medzi uvoľnením tlačidla a ďalším kliknutím.	

Tabuľka 5: Metriky založené na nečinnosti [15].

Názov metriky	Popis	Výpočet
Počet páuz	Celkový počet páuz u používateľa	$NP = \sum_{i=1}^n p_i, p_i = \begin{cases} 1, & t_i > \beta \\ 0 & \end{cases}$
Čas strávený nečinnosťou	Súčet dĺžok páuz u používateľa	$PT = \sum_{i=1}^n p_i t_i, p_i = \begin{cases} 1, & t_i > \beta \\ 0 & \end{cases}$
Pomer času nečinnosti	Podiel času nečinnosti s celkovým časom.	$PTR = \frac{PT}{t_n}$

väčšia tým je váhavosť používateľa vyššia. Pod pojmom pauza rozumejú žiadnu činnosť s myšou ani s klávesnicou. Na základe informácie o váhavosti autori identifikujú nedostatky v používateľskom rozhraní. Nazbierané hodnoty váhavosti boli porovnané s analýzou sedení používateľským expertom, ktorý označil body, kde nastala váhavosť účastníka. Pri vyhodnocovaní výsledkov autori experimentovali s normalizáciou hodnoty na základe priemernej pauzy používateľa. Ich výsledky ukazujú ako posúvanie tejto hodnoty ovplyvňuje celkovú presnosť pri vyhodnocovaní používateľských problémov rozhrania.

Podobný prístup na predikciu používateľských problémov v aplikácii pomocou váhavosti z páuz používateľa bol navrhnutý v práci [33] už v roku 1995, kde autori merali váhavosť na základe dĺžky páuz medzi zadáním úlohy a prvého úderu klávesnice. V tomto prípade šlo o aplikáciu knižničného katalógu založenú na príkazovom riadku (angl. *command line based application*). Na overenie hypotézy zostrojili experiment s 12 účastníkmi a pri vyhodnocovaní boli odhalené všetky problémy použiteľnosti.

3.1.3 Metriky myši založené na pauzách

Jednou z ďalších zaujímavých vlastností používateľa, spojenou s myšou, je nečinnosť alebo pauza. V spomínanej práci [15] autori využili aj metriky odvodené od tejto vlastnosti. Tabuľka 5 poskytuje prehľad s opisom spolu so vzorcami.

Pri týchto metrikách je dôležité spomenúť definíciu pauzy. V prácach [15, 13, 27] je tento jav definovaný ako nečinnosť po dobu dlhšiu ako β .

3.2 Metriky klávesnice

Okrem počítačovej myši používateľ pri interakcii s webovou aplikáciou používa klávesnicu. Typickými príkladmi interakcie sú textové polia vo formulároch, vyhľadávacie polia v katalógoch alebo klávesové skratky. Údaje o interakcii používateľa s klávesnicou môžu byť jednou z jeho charakteristík. Svedčí o tom práca [27], kde autori využívajú takéto dáta k autentifikácii používateľa. Jednou z týchto charakteristík je dynamika úderov klávesov (angl. *keystroke dynamics*). Je definovaná časom od stlačenia klávesy až po jej uvoľnenie. Takéto údaje súvisia s jedným úderom. V prípade páru (angl. *digraf*) môžeme pracovať so štyrmi hodnotami. Časom od stlačenia po stlačenie, od uvoľnenia po uvoľnenie, od stlačenia po uvoľnenie a od uvoľnenia po stlačenie.

Autori práce [14] sa venovali autentifikácii používateľa pomocou dynamiky úderov už v roku 1980. Pracujú so záznamami o písaní šiestich pisárov, ktorých úlohou bolo prepísať tri texty. Následne na to pre každý pár klávesov vypočítali štatistické hodnoty pre jednotlivé časy. Ich metóda bola vyhodnotená na texte s 300 slovami a dosiahla 4% chybovosť. To znamená, že len v 4 percentách prípadov by systém autentifikoval neoprávneného používateľa (angl. *false acceptance rate FAR*).

3.3 Histogramová reprezentácia črt

Črty ako také môžu veľa prezrádzať o správaní používateľa. Svedčí o tom mnoho prác spomínaných v predchádzajúcich kapitolách. Ich získanie je však mnohokrát spojené s agregovaním iných hodnôt. Vezmime si napríklad prácu [15], kde autori najprv rozdelili získané dáta na pohyby, následne extrahovali črty myši pre každý pohyb zvlášť a vo výsledku vypočítali agregované hodnoty. Takto vytvorený model používateľa však stráca informácie o pohyboch zvlášť. Problém je načrtnutý aj v práci [27], kde sa autori venujú biometrii.

Ako riešenie poskytujú histogramovú reprezentáciu črt pohybov myši. Podrobnejšie je riešenie rozobrané v práci [2], kde sa autori venujú biometrii správania používateľa založenej na pohyboch myši s použitím neurónovej siete. Spomínaný problém má však niekoľko obmedzení. Jedným z nich je určenie počtu stĺpcov histogramu. Riešenie tohto problému môže byť spojené s natrénovaním niekoľkých modelov s rôznym počtom stĺpcov a následným výberom najlepšieho modelu.

Druhý problém je spojený s hraničnými hodnotami pre stĺpce histogramu. V práci [27] autori zvolili postup, kde priestor medzi minimálnou a maximálnou hodnotou danej črty lineárne rozdelia. Výsledky však môžu byť ohrozené vybočujúcimi hodnotami (angl. *outlayers*). Iné riešenie môže byť spojené s lineárnym rozdelením priestoru v medzikvartilovom rozpätí hodnôt.

3.4 Sledovanie pohľadu

Každý používateľ pri interakcii s webovou aplikáciou spracováva informácie primárne pomocou zrakového vnemu. Údaje o pohybe očí človeka počas interakcie s webovou aplikáciou môžu ho-

vorit o jeho zmätení. Svedčia o tom spomínané práce [29, 28]. V našej práci sa snažíme porovnať prístupy predikcie zmätenia založené na dátach z okulografu s našou metódou založenou na dátach z myši. Z toho dôvodu uvádzame existujúce metriky okulografu popisujúce správanie používateľa. Podľa práce [10] ľudské oko pri svojich pohyboch používa šesť svalov. V práci sa uvádza, že rozlišujeme niekoľko základných pohybov:

- *Sakády*. Sú zároveň dobrovoľné a reflexné pohyby oka. Ich trvanie je v intervale od 10 do 100ms čo je príliš krátke na zachytenie obrazu.
- *Hladké nasledovanie* (angl. *Smooth pursuit*). Ide o pohyb oka, kedy je prenasledovaný objekt pohľadom.
- *Fixácie*. Je jav, počas ktorého je stabilizovaná sietnica. Počas neho je pozornosť upriamená na objekt. V práci [10] sa uvádza, že fixácie sú zväčša spojené s miniatúrnymi pohybmi očí.

3.4.1 Okulograf

Na to, aby sme mohli zaznamenávať pohyby očí potrebujeme okulograf (angl. *eyetracker*). Ide o zariadenie, ktoré mapuje tieto pohyby do súradnicovej sústavy. Podľa práce [10] poznáme dva základné druhy takýchto zariadení. Prvý je hardvér pripevnený na hlavu (angl. *head-mounted hardware*). Ide o zariadenie podobné okuliarom, ktoré priamo zachytáva pohyby oka z bezprostrednej blízkosti. Druhou skupinou je hardvér pripevnený na stole (angl. *table-mounted hardware*). V tomto prípade ide o zariadenia, s ktorými sa počas experimentu nemanipuluje. Príkladom sú produkty firmy *Tobii* [41].

Jednou z kľúčových vecí pri nahrávaní pohľadu človeka pomocou okulografu je frekvencia zariadenia. Ide o počet záznamov za sekundu, ktoré je nástroj schopný zaznamenať. Štandardne sa pracuje s frekvenciou 60 až 300Hz [10].

Podľa práce [10] je pred samotným nahrávaním pohľadu s okulografom dôležitá kalibrácia. V tomto kroku je účastníkovi experimentu nastavované zariadenie na sledovanie pohľadu. V prípade modelov pripevnených na stole sa pracuje s výškou stoličky a vzdialenosťou okulografu od hlavy. Následne sa prechádza kalibračná sekvencia. Ide o obrazovku, na ktorej účastník pohľadom nasleduje guľôčku postupne po všetkých rohoch obrazovky. V prípade zariadení pripevnených na hlavu ide o veľmi podobné sekvencie nasledovania objektov pohľadom.

3.4.2 Zrenička

Súčasťou ľudského oka je aj zrenička. Okulografy okrem dát o bode pohľadu dokážu poskytnúť aj informácie o jej veľkosti. Ich spracovaním vieme získať informácie spojené s mentálnou činnosťou používateľa. V práci [19] sa autori zaoberajú dilatáciou zreničky. Dilatácia zreničky je zmena veľkosti. Zostavili experiment s dvanástimi účastníkmi, kde sa riešili úlohy rôznej mentálnej obtiažnosti. Ich výsledky ukazujú, že obtiažnosti týchto úloh boli spojené so zmenou veľkosti

zreničky. Taktiež v práci [5] autori ukazujú vzťah medzi kognitívnou záťažou človeka a jeho zreničkou.

Pri týchto dátach však treba byť opatrný. Dilatácia zreničky je ovplyvnená svetelnými podmienkami prostredia. To znamená, že jej veľkosť môže byť odlišná v závislosti od svetelných podmienok alebo farieb monitoru. Vplyvu efektu svietivosti displeja na zreničku sa venujú autori práce [23], kde pre tento účel zostavili model. Jeho úspešnosť overili na experimente so 73 účastníkmi.

3.4.3 Oblasti záujmu

Oblasti záujmu (angl. *areas of interest* AOI) sú jednou z dôležitých vecí v experimente so sledovaním pohľadu. V prípade experimentu s webovou aplikáciou sa jedná o časti rozhrania, na ktoré sa chceme zamerať. Môže ísť napríklad o časti html kódu stránky. Oblasti záujmu sa však dajú vymedziť aj pri experimentoch s desktopovou aplikáciou. Vždy je však potrebné vyznačiť časti, na ktoré sa zameriavame.

Sledovanie pohľadu používateľa v oblastiach záujmu môže veľa prezrádzať o jeho správaní. V knihe [10] sa metriky odvodené od AOI označujú ako bežné. V práci [29] autori pomocou takýchto metrick odhaľujú problémové správanie používateľov pri práci s vizualizáciami. V ich experimente sa snažia predikovať učiacu krivku (angl. *learning curve*). Taktiež v práci [17] využívajú tieto metriky pri vyhodnocovaní nástrojov na kreslenie.

3.4.4 Metriky sledovania očí

V tejto časti chceme zhrnúť metriky pohľadu používané v spomínaných dielach. Prehľad sumarizujeme tabuľkou 6. Okrem metrick prác z predchádzajúcich podkapitol poskytujeme aj prehľad základných metrick odvodených z fixácií a sakád. V prácach [24, 25] ich autori využívajú pri generovaní modelu na podporu študentov pri práci s interaktívnymi simuláciami. Vo väčšine prác sa však pracuje so štatistikami týchto metrick. Vypočítajú sa hodnoty priemeru, minima, maxima a štandardnej odchýlky a následne sú použité v modeli.

3.5 Metriky HTTP serveru

Jednou z najbežnejších softvérových architektúr v prípade webových aplikácií je architektúra klient server. V rámci nej klientská časť kódu vo forme *html* a *css* je poslaná zo serveru priamo do prehliadača cieľového používateľa pri dopyte na cieľovú url. Pri prechode používateľa medzi jednotlivými pod-stránkami aplikácie sa na server posielajú dopyty. Server následne odpovedá opäť formou *html* a *css* kódu pre odosielateľa. Na väčšine moderných serverov sa celá komunikácia loguje do súborov. Tieto logy vedia poskytnúť cenné informácie o správaní používateľa a zároveň ich získavanie nie je nákladné.

Tabuľka 6: Prehľad metrík sledovania pohľadu z literatúry.

Názov	Anglický názov	Popis	Zdroje
Odvođené od sakád			
Dĺžka sakády	<i>Saccade duration</i>	Vzdialenosť prejdená medzi dvomi po sebe idúcimi fixáciami.	[24, 25]
Uhol sakády (relatívny)	<i>Relative saccade angle</i>	Uhol sakády k predchádzajúcej sakáde.	[24, 25, 17]
Uhol sakády (absolútne)	<i>Absolute saccade angle</i>	Uhol sakády k x-ovej osi.	[24, 25, 17]
Počet sakád	<i>Number of saccades</i>	Počet sakád pre pozorovanie.	[12]
Trvanie sakády	<i>Saccade duration</i>	Trvanie sakády v milisakundách	[17]
Pomer sakád	<i>Saccade ratio</i>	Pomer počtu sakád k celkovému počtu časovej doby.	[17]
Odvođené od fixácii			
Počet fixácii	<i>Number of fixations</i>	Počet fixácii pre pozorovanie	[24, 25, 17]
Frekvencia fixácii	<i>Fixation rate</i>	Je to počet fixácii za jednotku času	[24, 25, 17]
Dĺžka trvania fixácie	<i>Fixation duration</i>	Je to časová doba fixácie	[24, 25, 17]
Priestorová hustota fixácii	<i>Spatial density</i>	Hustota fixácii v priestore	[17]
Čas do prvej fixácie	<i>Time to first fixation</i>	Je to čas, ktorý uplynie pred prvou fixáciou	[8, 37]
Odvođené od AOI			
Počet fixácii pre oblasť	Fixation rate on AOI	Je to množstvo fixácii na konkrétnu oblasť.	[29, 17]
Dĺžka fixácie pre oblasť	Fixation length on AOI	Dĺžka fixácie pre oblasť.	[29, 17]
Počet prechodov z danej oblasti inam	Number of transitions	Počet jednotlivých prechodov z vybranej oblasti do inej oblasti.	[29, 17]
Odvođené od zreničky			
Šírka zreničky	Pupill width	Šírka zreničky pre záznam.	[19]

Tabuľka 7: Prehľad metrík odvodených z HTTP serveru z práce [42].

Metrika	Popis
Celkový čas	Doba od prvého záznamu sedenia až po posledný záznam.
Počet stránok	Počet unikátnych url navštívených počas sedenia.
Čas na stránku	Priemerný čas pre navštívenú stránku v rámci sedenia.
Počet cyklov	Cyklom sa označuje taká cesta stránok, pri ktorej sa užívateľ vráti k už raz navštíveneju stránke.
Počet zámen (angl. swap)	Zámenou sa označuje taká postupnosť stránok, v rámci ktorej používateľ prejde na stránku inej domény a následne sa vráti (napr. využitie vyhľadávачa).

V už spomínanej práci [42] autori využívajú metriky odvodené z týchto logov na predikciu obtiažnych sedení. Prehľad metrík poskytuje tabuľka 7. O širšom využití dát z web logov svedčí aj práca [31], kde autori poskytujú prehľad možností ich sekvenčného spracovania. Autori v článku prešli široké využitie týchto algoritmov a výstupom práce je taxonómia, ktorá delí algoritmy spracovania sekvencií z logov do troch skupín.

3.6 Zhrnutie a diskusia

V tejto kapitole sme opísali metriky myši, klávesnice a pohľadu vhodné na modelovanie správania používateľa. Na to aby sme vedeli tieto metriky extrahovať a ďalej s nimi pracovať, je potrebné aby sme mali dáta v požadovanej forme. Napríklad, v prípade metrík myši potrebujeme aby sme mali dáta vo forme pohybov (skupina záznamov súradníc myši v čase, kde časový rozdiel medzi jednotlivými záznamami je menší ako hraničná hodnota). Tu sa nám otvára priestor pre experimentovanie s hraničnou hodnotou, jej posúvaním totiž vieme získať iné rozdelenie súradníc do pohybov a teda aj iné hodnoty metrík.

V prípade okulografu potrebujeme záznamy koordinátov rozdeliť do fixácií a sakád. Na toto rozdelenie vieme použiť jeden z už existujúcich implementácií fixačných filtrov. Keďže týchto implementácií je viacero opäť sa tu otvára priestor na experimentovanie. V našom prípade použijeme fixačný filter, ktorý je súčasťou infraštruktúry, ktorú použijeme pri používateľskej štúdií.

Je vhodné spomenúť, že pri metrikách myši sme sa venovali najmä tým, ktoré sa využívajú pri autentifikácii používateľa na webe. Dôvod je slabé využitie metrík myši pri predikcii zmätenia. Avšak ako sme spomínali tieto dva problémy sú veľmi podobné, keďže pri oboch sa snažíme namodelovať daného používateľa na základe jeho správania. Pri autentifikácii ide o klasifikáciu do viacerých tried, zatiaľ čo pri predikcii zmätenia nás zaujímajú hlavne dve triedy a to zmätený a nezmätený. Poskytli sme zaujímavý pohľad na spracovanie metrík formou histogramov. Tu chceme pripomenúť, že na to aby bolo toto použitie efektívne je potrebné, aby pre každý histogram používateľa bolo dostatok dát. Nemá totiž zmysel využiť takúto reprezentáciu, keď chceme rozdeliť len pár dát do histogramov používateľa (napr. menej ako 10).

Problémom metrík pohľadu je dostupnosť okulografov. Bežný používateľ nedisponuje takýmto zariadením v domácnosti. Aplikácia modelu založenom na takýchto metrikách môže byť proble-

matická. Avšak ich použitie na kontrolovanom experimente je ideálne. Využitím dát z pohľadu môžeme lepšie pochopiť akcie používateľa. Využijeme ich na porovnanie modelu založenom len na správaní z myši a klávesnice.

Stále sa tu však otvára diskusia spojená s ochranou osobných údajov. Pri prípadnom produkčnom využití nášho riešenia je potrebné na to pamätať. V takomto prípade môže byť problém pri zaznamenávaní správania klávesnice, ktoré so sebou nesie citlivé informácie o používateľovi. Z tohto dôvodu sa budeme ďalej sústreďiť na dáta z myši a okulografu.

4 Predikčný model

Po získaní dát a extrahovaní črt z pohľadu a myši je potrebné natrénovať model predikcie zmätenia. V tomto prípade sa bude jednať o binárnu klasifikáciu a učenie s učiteľom. Je preto vhodné zvoliť model a následne vybrať metriky na porovnávanie medzi netrénovanými modelmi.

Učenie s učiteľom (angl. *supervised learning*) je metóda strojového učenia (angl. *machine learning*), pri ktorej sa model učí z vopred označených dát. Následne vo fáze predikcie model využije nadobudnuté informácie [26]. Okrem toho poznáme ešte učenie bez učiteľa (angl. *unsupervised learning*), kde dáta nie sú vopred označené a učenie s odmenou a trestom (angl. *reinforcement learning*).

V našom prípade sa bude jednať o klasifikáciu dát do dvoch skupín zmätený, nezmätený. Klasifikačný problém je taký, kde predikujeme kategorickú premennú a nevieme povedať vzdialenosť medzi jej jednotlivými hodnotami. Opakom je regresný problém, kde predikovaná premenná je taká pri ktorej vieme určiť vzdialenosť medzi dvomi jej hodnotami. Príkladom regresného problému môže byť predikcia ceny domu [11].

4.1 Klasifikácia v reálnom čase

Jedným zo zámerov našej práce je využitie detekcie zmätenia, t.j. výsledkov klasifikácie v reálnom čase. Snahou tejto práce je vedieť odhaliť moment, kedy stav zmätenia u používateľa nastal tak, aby bolo možné čo najskôr na identifikovaný stav reagovať. V prístupe práce [42] autori používajú pri predikcii obtiažnych sedení dáta z celého sedenia. Takýto prístup klasifikácie by sa dal využiť v našom prípade vtedy, keď by sme chceli označiť sedenia, kde nastalo zmätenie po ich uplynutí. Čo však prípady, kde chceme predikovať zmätenie už počas sedenia?

V už spomínaných prácach [29, 28] autori predikujú zmätenie v reálnom čase na základe časových okien. Pod pojmom časové okno (angl. *time window*) autori rozumejú časový interval rovnakej veľkosti (napríklad niekoľko sekúnd), pomocou ktorého sa rozdelia záznamy o sedení. Na samotné delenie autori využili algoritmus posuvného okna (angl. *sliding window algorithm*), ktorý na začiatku extrahuje okno od časovej pečiatky prvého záznamu. Následne sa toto okno posúva o n sekúnd a extrakcia pokračuje až kým sa nedosiahne posledný záznam. Každé okno predstavuje jeden záznam datasetu. Predmetom výskumu v tomto prístupe je určenie veľkosti okna. Autori práce pracovali hneď s niekoľkými nastaveniami (5 a 10 sekúnd).

4.2 Logistická regresia

Ako jedným zo základných klasifikačných algoritmov je logistická regresia. V našom prípade, predikcie len dvoch hodnôt sa javí ako jedna z najvhodnejších možností. Výstupom predikcie logistickej regresie v prípade dvoch hodnôt je pravdepodobnosť s akou pozorovanie o patrí do jednej z tried. Označením $P(\text{zmätený} = \text{áno} \mid \text{črta})$ skrátene $P(\text{črta})$ budeme rozumieť pravdepodobnosť s akou bude pozorovanie klasifikované ako zmätené vzhľadom k črte. Štandardne sa

pozorovanie klasifikuje vždy vtedy, keď je táto hodnota väčšia ako 0.5, avšak hraničná hodnota sa dá upraviť v závislosti od problému.

Model logistickej regresie vychádza z lineárnej regresie, ktorá slúži na riešenie regresných problémov. Pri nich je predikovaná hodnota kvantitatívna, a teda vieme určiť vzdialenosť medzi predikovanými hodnotami. Na výpočet modelu lineárnej regresie sa používa vzorec [39]

$$\varnothing(x_1, x_2, \dots, x_n) = \beta_0 + \beta_1 x_1 + \dots + \beta_n x_n, \quad (1)$$

kde označením x_1 až x_n rozumieme zložky vektoru pozorovania (v našom prípade hodnoty vybraných metrik pre konkrétne pozorovanie) a pod označením β_1 až β_n rozumieme parametre modelu.

Ak by sme model aplikovali na predikciu kategorických dát je jasné, že výstupom funkcie by boli hodnoty mimo intervalu $[0;1]$. Aby sme sa vyhli tomuto problému musíme použiť funkciu, ktorá mapuje tieto výsledky do intervalu $(0;1)$. V prípade logistickej regresie sa používa *logistická funkcia* [34]

$$\varnothing(x_1, x_2, \dots, x_n) = \frac{e^{\beta_0 + \beta_1 x_1 + \dots + \beta_n x_n}}{1 + e^{\beta_0 + \beta_1 x_1 + \dots + \beta_n x_n}}. \quad (2)$$

Získaná funkcia má tvar písmena S a vždy nadobudne hodnoty z intervalu $(0;1)$. Jej úpravou vieme získať [34] vzorec

$$\log\left(\frac{\varnothing(x_1, x_2, \dots, x_n)}{1 - \varnothing(x_1, x_2, \dots, x_n)}\right) = \beta_0 + \beta_1 x_1 + \dots + \beta_n x_n, \quad (3)$$

kde ľavá časť sa nazýva *logit*.

Súčasťou tréningu modelu Logistickej regresie je učenie sa parametrov $\beta_0 \dots \beta_n$. Ako je spomenuté v knihe [34] na ich učenie sa používajú tréningové dáta a maximalizovanie funkcie (angl. *maximum likelihood*)

$$l(\beta_0, \beta_1) = \prod_{i:y_i=1} p(x_i) \prod_{i':y_{i'}=0} (1 - p(x_{i'})). \quad (4)$$

Kde parameter β_0 sa špeciálne označuje slovom zachytenie (angl. *intercept*).

4.3 Náhodný les

Algoritmus náhodný les (angl. *Random forest*) je ďalšou možnosťou v prípade klasifikácie. V práci [29] bol použitý pri binárnej predikcii zmatenia nad dátami z okulografu. Jednou z jeho výhod je nelineárny prístup k riešeniu problémov. Algoritmus má aj svoju modifikáciu na riešenie regresných problémov. Náhodný les je jedným z algoritmov súborového učenia (angl. *ensemble learning*) [44]. Klasifikátory súborového učenia sú založené na aplikácii viacerých klasifikátorov súčasne. V prípade náhodného lesu ide o kombináciu viacerých rozhodovacích stromov (angl. *Decision tree*).

Rozhodovací strom je algoritmus založený na sérii pravidiel začínajúcich v koreni stromu. Listy,

alebo terminálne uzly stromu, tvoria predikované hodnoty algoritmu. Miesta vo vnútri stromu, kde sa vetvy delia sa nazývajú interné uzly a vytvárajú ďalšie vetvy. Každá z vetiev je zakončená hodnotou (listom) alebo ďalším interným uzlom. Podľa knihy [20] tréning algoritmu regresie pomocou rozhodovacieho stromu pozostáva z dvoch krokov:

1. Rozdelenie priestoru pozorovaní X_1 až X_N do J neprekrývajúcich sa regiónov R_1 až R_J .
2. Hodnota regiónu je reprezentovaná priemernou cieľovou hodnotou všetkých pozorovaní v regione.

Delenie na regióny je s cieľom minimalizovať sumu štvorcových chýb (pri regresie). Avšak vo väčšine prípadov existuje veľké množstvo kombinácií delení. Z tohto dôvodu sa používa chamtivý prístup zhora na dol (angl. *top-down greedy approach*) označovaný ako rekurzívne binárne delenie (angl. *recursive binary splitting*). Začína sa v koreni stromu, kedy všetky pozorovania patria do jedného regiónu. V každom kroku delenia sa vytvoria vždy dva nové regióny. Chamtivým sa označuje tento prístup preto, že výber rozdelenia na danej úrovni je ovplyvnený minimalizáciou chyby len pre danú úroveň stromu bez ohľadu na možné zlepšenia pri ďalších úrovniach. Takýto algoritmus však vyprodukuje veľké stromy. V praxi sa potom tieto stromy orezávajú (angl. *pruning*) na menšie stromčeky. Výber menších stromčekov sa realizuje na základe chyby na krížovej validácii.

Pri klasifikácii pomocou rozhodovacieho stromu sa pre daný región vyberie najčastejšie vyskytovaná trieda (angl. *most commonly occurring class*), následne je však potrebné región ohodnotiť. V knihe [20] sa uvádza niekoľko spôsobov ako vyhodnocovať výber regiónov:

1. *Klasifikačná chyba* (angl. *classification error*)

$$err = 1 - \max_k(p_{mk}), \quad (5)$$

Kde p_{mk} vyjadruje proporciu počtu triedy k v m -tom regione. Avšak táto chyba nie je citlivá na rast stromu, preto sa vo väčšine problémov nevyužíva.

2. *Gini index*

$$gini = \sum_{k=1}^K p_{mk}(1 - p_{mk}), \quad (6)$$

ktorý nadobúda nízke hodnoty v prípade, že všetky hodnoty p_{mk} sú blízke k 0 alebo 1. Malé hodnoty tohto indexu hovoria o tom, že v regione dominuje jedna z tried.

3. *Entropia*

$$entr = - \sum_{k=1}^K p_{mk} \log(p_{mk}), \quad (7)$$

ktorá nadobúda rovnaké hodnoty ako Gini index.

Pri algoritme náhodného lesu sa trénuje n stromov súčasne. Na vytvorenie odlišných tréningových sad pre každý zo stromov sa využíva metóda vzorkovania (angl. *bootstrapping*), kedy sa celková tréningová sada rozdelí náhodným výberom vzoriek.

4.4 Vyhodnotenie modelov

Po natrénovaní modelov nad dátami je potrebné ich vyhodnotiť. Pred samotnou aplikáciou metrík vyhodnotenia je vhodné si dataset rozdeliť na tréningovú a testovaciu sadu. Štandardne je tréningová sada niekoľkonásobne väčšia ako testovacia. Používajú sa pomery ako 80 ku 20 percentám, alebo 70 ku 30 percentám. Pri klasifikácii je však potrebné, aby sa zachoval aj pomer v početnosti jednotlivých tried pre tréningovú a testovaciu sadu.

Pri mnoho problémoch je však dataset vysoko nevyvážený [29, 42]. V takomto prípade je vhodné ho vyvážiť pomocou techniky vzorkovania. Poznáme dve techniky vzorkovania:

1. Pod-vzorkovanie (angl. *undersampling*). Je technika, kedy sa z majoritnej triedy vzoriek vyjme len určitá menšia reprezentatívna vzorka. V práci [30] autori ukazujú priaznivý vplyv pod-vzorkovania na metriku plocha pod priamkou (opísanú v ďalšej podkapitole).
2. Nad-vzorkovanie (angl. *oversampling*). Je opakom pod-vzorkovania. Kedy sa snažíme zväčšiť početnosť minoritnej triedy v datasete. Tento postup je mnohokrát spojený s generovaním umelých dát odvodených od existujúcich.

V práci [1] autori predstavujú symetrickú techniku nad-vzorkovania minoritnej triedy (*Smote*). Technika je založená na kombinácii pod-vzorkovania a nad-vzorkovania vzhľadom k minoritnej triede datasetu. Aj tu autori opisujú priaznivé výsledky pri klasifikácii vzhľadom na metriku plochy pod priamkou s použitím klasifikátora rozhodovacích stromov. Počas vzorkovania sa vytvárajú syntetické reprezentácie minoritnej triedy na základe k -najbližších susedov a náhody. Zároveň počas toho je pod-vzorkovaná majoritná trieda v rovnakom pomere s použitím náhodného výberu. Táto technika je tiež použitá v práci [29] pri predikcii zmätenia s použitím okulografu.

Po vyvážení datasetu a natrénovaní modelu na tréningovej sade vzoriek pristupujeme k časti predikcia. Vezmeme testovacie vzorky $x_{1..n}$ a necháme model predikovať ich triedy $y_{1..n}$. Ich skutočné triedy budeme označovať $d_{1..n}$. V prípade binárnej klasifikácie vieme, že $y_i, d_i \in \{0, 1\}$. Špecifickejšie pre náš prípad 1 znamená zmätený a 0 nezmätený. Po predikcii vieme identifikovať 4 možné množiny [40]:

1. Správne pozitívne pozorovania (angl. *true positive* TP). Sú také pozorovania z testovacej množiny, kde $y_i = 1$ a zároveň $d_i = 1$.
2. Správne negatívne pozorovania (angl. *true negative* TN). Sú také pozorovania z testovacej množiny, kde $y_i = 0$ a zároveň $d_i = 0$.
3. Nesprávne pozitívne pozorovania (angl. *false positive* FP). Sú také pozorovania z testovacej množiny, kde $y_i = 1$ a zároveň $d_i = 0$.
4. Nesprávne negatívne pozorovania (angl. *false negative* FN). Sú také pozorovania z testovacej množiny, kde $y_i = 0$ a zároveň $d_i = 1$.

Tieto množiny vieme lepšie zobrazit do matice zmätenia (angl. *confusion matrix*), zobrazenej v tabuľke 8.

Tabuľka 8: Všeobecná matica zmätenia, y_i je predikovaná hodnota a d_i je očakávaná hodnota [40].

	$d_i = 1$	$d_i = 0$
$y_i = 1$	TP	FP
$y_i = 0$	FN	TN

V ďalších častiach opíšeme metriky vyhodnocovania modelov klasifikácie odvodené pomocou matice zmätenia.

4.4.1 Presnosť a úplnosť

Podľa práce [40] označujú autori metriky presnosť (angl. *precision*) a úplnosť (angl. *recall*) ako jedny z najpopulárnejších metrík na vyhodnotenie klasifikačných modelov. Tieto metriky boli taktiež použité v práci [42] pri porovnaní modelov na predikovanie obtiažnych sedení. Úplnosť bola použitá v práci [29] na porovnanie modelov na predikciu zmätenia používateľa z jeho pohľadu.

Presnosť označuje úspešnosť klasifikátora pri označení, že testovacia vzorka patrí do danej triedy. Jej vzorec je [40]

$$precision = \frac{TP}{TP + FP}. \quad (8)$$

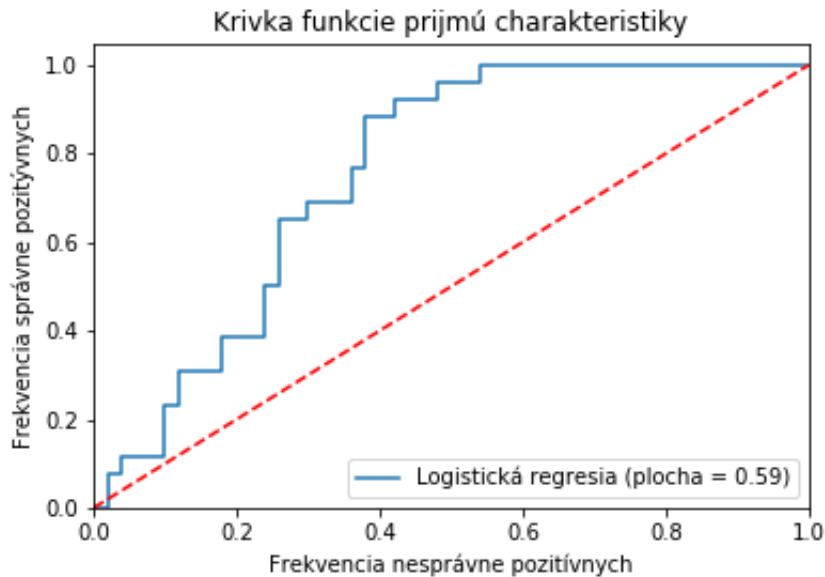
Na druhej strane tu máme úplnosť, ktorá hovorí o tom, ako veľmi vieme pomocou natrénovaného modelu predikovať všetky výskyty danej triedy. Jej vzorec je [40]

$$recall = \frac{TP}{TP + FN}. \quad (9)$$

4.4.2 Plocha pod krivkou charakteristiky prijímajúceho operátora

Plocha pod krivkou (angl. *area under curve* AUC) je metrika, ktorá sa počíta vyjadrením obsahu pod krivkou funkcie prijímú charakteristiky (angl. *Reception Operating Characteristic* ROC). ROC krivka nám zase hovorí o schopnosti modelu rozlišovať triedy na testovacej sade [18]. Je založená na rastúcom vzťahu medzi TP a FP meraniami. Hodnota AUC je následne len výpočtom plochy pod touto krivkou. Čím je hodnota AUC vyššia tým je model lepší.

Obrázok 3 poskytuje ukážku ROC krivky s prislúchajúcim výpočtom AUC. Na osách grafu môžeme vidieť hodnoty relatívnych počtov pre TP a FP . Taktiež je na grafe vyznačený prerušovanou čiarou najhorší možný model.



Obr. 3: Ukážka *ROC* krivky (plná čiara). Prerušovanou čiarou je označená najhoršia možná situácia. Zdroj grafu sú prvotné experimenty na dátach z myši.

V práci [40] *AUC* označujú ako jednu z najpopulárnejších metrík na vyhodnotenie klasifikačných problémov. Metrika je použitá na porovnanie modelov aj v spomínaných prácach [28, 42]. Taktiež autori uvádzajú vzorec na výpočet

$$AUC = \frac{1}{2} \left(\frac{TP}{TP + FN} + \frac{TN}{TN + FP} \right). \quad (10)$$

4.4.3 Akeikeho informačné kritérium

Akeikeho informačné kritérium (angl. *Akaike information criterion* AIC). Metrika pomenovaná podľa slávneho štatistika Hirotugu Akaikeho, ktorá určuje, ktorý z dvoch modelov je lepší. Na vyhodnotenie okrem chyby na testovacej sade (angl. *sum of squares error* SSE), berie do úvahy aj zložitosť modelu, ktorá negatívne vplyva na hodnotu tejto metriky. Vzorec na výpočet *AIC* potom vyzerá nasledovne

$$AIC = N * \ln\left(\frac{SSE}{N}\right) + 2 * k, \quad (11)$$

kde N označuje počet meraní, SSE označuje sumu chýb štvorcov a k označuje počet parametrov modelu [3]. Čím nižšia je hodnota *AIC*, tým je model považovaný za lepší.

Okrem toho poznáme aj Bayesovo informačné kritérium (angl. *Bayesian information criterion* BIC), ktoré ešte viac penalizuje počet parametrov modelu

$$BIC = N * \ln\left(\frac{SSE}{N}\right) + k * \ln(N). \quad (12)$$

4.5 Zhrnutie a diskusia

V tejto kapitole sme sa venovali predikčným modelom pre problém binárnej klasifikácie. Pred samotným predstavením možných modelov sme prešli možnosti predikcie v reálnom čase. V produkcii totiž nemôžeme predikovať na celej úlohe, keďže nevieme určiť začiatok a koniec. Využitie spomínaných okien rovnakej veľkosti sa javí ako možné riešenie. Otvára sa tu však priestor na experimentovanie s veľkosťou okna. Výber príliš malých okien vedie k slabej úspešnosti. Svedčí o tom najmä práca [29]. Na druhej strane výber príliš veľkých okien robí problém v produkcii, kde mnohokrát potrebujeme predikovať zmätenie okamžite po príchode používateľa, nemôžeme čakať, kým sa okno naplní.

Ďalej sme predstavili klasifikátor logistickej regresie a klasifikátor náhodného lesu. Prihliadnuc k literatúre [29, 28, 42, 20, 44] sa tieto algoritmy ukazujú ako ideálne na riešenie problémov binárnej klasifikácie. Algoritmus logistickej ukazuje lepšie výsledky pri lineárne separovateľných problémoch zatiaľ čo náhodný les naopak. Pri ich tréningu sa otvára priestor na experimentovanie s hyper parametrami. Napríklad v prípade náhodného lesu vieme pracovať s množstvom a hĺbkou stromov a v prípade logistickej regresie zase s váhami predikovaných tried.

Opísali sme aj metriky, ktoré je možné využiť pri vyhodnotení modelov. Hlavná myšlienka našej práce je, že plánujeme nasadiť naše riešenie priamo do produkčného prostredia. Tu je veľmi dôležitá presnosť pri predikcii triedy zmätenia a teda počty jej správne pozitívnych pozorovaní a nesprávne pozitívnych pozorovaní. Z tohto dôvodu sa sústredíme práve na metriku presnosti a na samotnú maticu zmätenia pri modeloch. Identifikácia zmätenia v produkcii bude sprevádzaná ponúknutím rady zmätenému používateľovi. Chceme teda predísť zbytočnému zaťažovaniu nezmätených používateľov.

5 Návrh metódy predikcie zmätenia

V našej práci sa venujeme identifikácii stavu zmätenia používateľa vo webovej aplikácii. Táto kapitola predstaví návrh metódy klasifikácie. Predpokladáme dva zdroje dát na predikciu. Prvým zdrojom je myš, a jej údaje získané pri interakcii s webovou aplikáciou. Druhým zdrojom dát je okulograf a údaje získané zo sledovania pohľadu používateľa. Dáta z okulografu poslúžili na porovnanie výsledkov klasifikácie.

Našu metódu sme rozdelili do dvoch scenárov. V prvom sa snažíme detekovať zmätenie na základe dát z celého sedenia. Podobný postup zvolili autori v práci [42]. V druhom scenári sa snažíme odhaliť presný bod zmätenia v reálnom čase interakcie užívateľa s webovou aplikáciou. Pod pojmom presný bod rozumieme moment, kedy používateľ nevie pokračovať v napĺňaní svojich potrieb v aplikácii vo vlastnej réžii. Tradične je tento bod spojený s využitím nápovedy, zákazníckej podpory alebo externej služby (vyhľadávača). Taktiež sa môže stať, že je tento moment spojený s rezignáciou, kedy používateľ aplikáciu opustí. Keďže v našom prípade využijeme metódu s nápovedami sústredíme sa na taký moment, kedy je používateľ ochotný priznať si, že takúto nápovedu naozaj potrebuje. Môže sa teda jednať aj o interval, ktorý takémuto bodu predchádza.

Našu metódu predikcie zmätenia sme rozdelili do niekoľkých krokov:

1. *Predspracovanie surových dát.* V tomto kroku čistíme surové dáta (angl. *raw data*). To znamená, že odstraňujeme duplikáty a neúplné záznamy. Okrem toho pri dátach z myši extrahujeme pohyby a pri okulografe fixácie a sakády. V prípade predikcie v reálnom čase taktiež extrahujeme okná rovnakej časovej dĺžky. Experimentovali sme s oknami dĺžky 5 a 10 sekúnd podobne ako v prácach [28, 29].
2. *Extrakcia črt.* V tomto kroku pristúpime k extrakcii črt z oboch zdrojov dát. V prípade vysoko nevyváženého datasetu je potrebné vyvážiť minoritnú triedu. Jednou z techník v tomto prípade je vzorkovanie (angl. *oversampling*).
3. *Natréňovanie klasifikátora.* V tomto kroku pristúpime k trénovaniu vybraných klasifikátorov. V prípade viacerých sa vyberie ten najúspešnejší na základe zvolených metrick: presnosť a úplnosť pri predikcii zmätenia, plocha pod krivkou charakteristiky prijímajúceho operátora a akaikeho informačné kritérium v prípade porovnaní viacerých nastavení rovnakého klasifikátora.
4. *Klasifikácia.* Po natréňovaní modelu prechádzame do predikcie. V prvom prípade pôjde o predikciu nad dátami z celého sedenia, zatiaľ čo v druhom predikujeme nad oknami rovnakej veľkosti.

5.1 Predspracovanie dát logeru myši a extrakcia črt

Z analýzy existujúcich riešení vyplýva, že forma surových dát získaných pri zaznamenávaní správania nie je vhodná pre tréňovanie predikčných modelov. Je preto potrebné dáta upraviť.

Tabuľka 9 poskytuje prehľad záznamu, ktorý bol zaznamenaný naším logerom. Opis jednotlivých polí:

1. **Šírka okna prehliadača** (*displayWidth*) - nastavená šírka displeja pri logovaní. Celočíselná hodnota.
2. **Výška okna prehliadača** (*displayHeight*) - nastavená výška displeja pri logovaní. Celočíselná hodnota.
3. **Zachytená udalosť** (*event*) - udalosť zachytená pri logovaní. Refazec môže obsahovať jednu z nasledujúcich hodnôt:
 - (a) Pohyb myši (*mousemove*).
 - (b) Skrolovanie myši (*scroll*),
 - (c) Stlačenie myši (*mousedown*).
 - (d) Uvoľnenie myši (*mouseup*).
 - (e) Stlačenie klávesy (*keydown*).
 - (f) Uvoľnenie klávesy (*keyup*).
 - (g) Klik (*click*).
4. **Identifikátor používateľa** (*userId*) - jedinečný identifikátor.
5. **X-ová súradnica myši** (*xpos*) - súradnica myši v rámci dokumentu. Celočíselná hodnota.
6. **Y-ová súradnica myši** (*ypos*) - súradnica myši v rámci dokumentu. Celočíselná hodnota.
7. **Xpath elementu pod kurzorom** (*xpath*) - xpath elementu vo formáte refazca.

Tabuľka 9: Príklad surových dát z logera.

display Height	display Width	event	userId	timestamp	xpos	ypos	xpath
4624	1418	mousemove	3c36dbbe-0e2e-4356-a618-8f5077f2b145	2018-07-20 16:26:41.481	1142	321	<xpath>

V nasledujúcom kroku rozdelíme záznamy na časové okná (dĺžky 5 a 10 sekúnd). Pre každé okno extrahujeme pohyby. Vychádzame pritom z prác [15, 16, 27], kde je pohyb definovaný ako sekvencia po sebe idúcich aspoň 4 koordinátov myši, kde časová zmena medzi dvomi po sebe idúcimi koordinátami je menšia ako 120 ms. Je vhodné spomenúť, že situácia kedy hranica okna delí pohyb je vyhodnotená ako dva pohyby, pre každé z okien jeden. Takáto situácia totiž nastala v zanedbateľnom množstve prípadov.

Následne pre každý pohyb vypočítame črty:

1. *Čas*. Celkový čas pohybu myši v milisekundách.

2. *Vzdialenosť.* Vzďialenosť prejdená pohybom v pixeloch.
3. *Rýchlosť.* Priemerná rýchlosť pohybu.
4. *Zrýchlenie.* Priemerné zrýchlenie pohybu.
5. *Hybný moment.* Priemerný hybný moment pohybu.
6. *Počet záznamov.* Počet záznamov o pohybe myši pre daný pohyb.
7. *Zmena času.* Priemerná zmena času medzi záznamami.
8. *Horizontálna rýchlosť.* Rýchlosť voči x-ovej osi.
9. *Vertikálna rýchlosť.* Rýchlosť voči y-ovej osi.

Keďže časové okno môže a nemusí pozostávať z pohybov pridali sme aj nasledujúce črty:

1. *Skroly.* Počet udalostí typu skrol pre dané okno.
2. *Kliky.* Počet udalostí typu klik pre dané okno.
3. *Pohyby.* Počet udalostí pohybov pre dané okno.
4. *Stlačenie klávesy.* Počet udalostí typu stlačenie klávesy pre dané okno.
5. *Uvoľnenie klávesy.* Počet udalostí typu uvoľnenie klávesy pre dané okno.

V prípade klasifikácie nad celým sedením extrahujeme tie isté črty, pričom za okno považujeme celé sedenie. Taktiež na celé sedenie aplikujeme predspracovanie pomocou histogramovej reprezentácie používateľa opísanej v podkapitole 3.3. Ako počet stĺpcov sme zvolili pevnú dĺžku 4. Na výpočet hraničných hodnôt stĺpcov sme použili lineárne rozdelenie rozsahu minima a maxima z rozpätia črt všetkých používateľov. Histogramovú reprezentáciu nemá zmysel aplikovať v prípade predikcie reálnom čase, kde okno správania obsahuje v priemere menej ako 3 pohyby.

5.2 Predspracovanie dát okulografu

V prípade okulografu sú dáta o pohľade veľmi podobné tým z myši. Ide o polohy očí doplnené o údaje o veľkosti zreničky a vzdialenosti od monitora v čase. Hlavný rozdiel oproti myši je v tom, že ide o dvojicu koordinátov pre každé oko zvlášť. Aj v tomto prípade sa jedná o veľké množstvo dát, ktorých objem je podmienený frekvenciou zariadenia.

V prvom kroku predspracovania odfiltrujeme nevalidné dáta [10]. Mnohokrát sa totiž stáva, že okulograf nesníma oči a generuje šum. Celý proces filtrovania pozostáva z identifikácie nevalidných záznamov pomocou validačného kódu zariadenia, ktorý je generovaný pre každý záznam oka zvlášť.

Ďalej je potrebné z dát získať informácie o fixáciách a sakádach. Samotné surové dáta sú totiž len polohy očí v čase. Ako sme uviedli v kapitole 3, pri rýchlych pohyboch oka náš mozog nezaznamenáva obraz. Preto je dôležité získať informáciu o fixáciách a sakádach. Na toto slúžia

fixačné filtre. Existuje viacero druhov fixačných filtrov v závislosti od použitého okulografu. Je preto veľmi dôležité zjednotiť parametre filtra. Následne vypočítame metriky:

1. *Počet fixácií.* Hodnota vyjadruje počet fixácií v rámci časového okna.
2. *Počet sakád.* Hodnota vyjadruje počet sakád v rámci časového okna.
3. *Počet fixácií.* Hodnota vyjadruje počet fixácií v rámci časového okna.
4. *Štatistika dĺžky fixácií.* Hodnoty minima, maxima, priemeru, kvantilov a štandardnej odchýlky dĺžok fixácií pre dané okno.
5. *Štatistika zmeny veľkosti zreničky.* Hodnoty minima, maxima, priemeru, kvantilov a štandardnej odchýlky zmeny veľkosti zreničky oproti priemeru účastníka pre dané okno.
6. *Štatistika vzdialenosti očí od monitora.* Hodnoty minima, maxima, priemeru, kvantilov a štandardnej odchýlky zmeny vzdialenosti očí od monitora pre dané okno.

5.3 Trénovanie modelov a vyhodnotenie

Po extrahovaní metrík natrénujeme model. Pre predikciu porovnáme klasifikátor logistickej regresie a náhodného lesu. Modely otestujeme v dvoch scenároch. V prvom budeme predikovať zmätenie nad dátami z celej úlohy. V tomto kroku porovnáme predikciu s využitím dát z okulografu a z myši. Cieľom je overiť úspešnosť predikcie s využitím jednotlivých metrík a histogramovej reprezentácie.

V druhom kroku pôjde o predikciu v reálnom čase s využitím okien rovnakej veľkosti. Na porovnanie použijeme okná veľkosti 5 a 10 sekúnd podobne ako v práci [29]. V tomto kroku bude potrebné dáta vyvážiť. Je totiž jasné, že ak označíme za zmätené len okná pred klikom na tlačidlo zmätenia a ako nezmätené okná v úlohách kde kliknutie nenastalo, počet zmätených okien bude podstatne nižší. Z tohto dôvodu aplikujeme algoritmus *SMOTE*.

Na vyhodnotenie modelov použijeme metriku presnosti a pokrytia nad predikovanými triedami. Identifikovanie stavu zmätenia bude v produkcii spojené s poskytnutím sprievodcov s cieľom pomôcť používateľovi. Z toho dôvodu je veľmi dôležité predikovať zmätenie s čo najväčšou presnosťou. Na základe týchto metrík na testovacej sade vyberieme najúspešnejší model.

6 Používateľská štúdia a overenie

Pri návrhu štúdie sme riešili hneď niekoľko otázok spojených so získavaním spätnej väzby o zmätení používateľa. Týkali sa hlavne toho ako odhaliť presný čas kedy zmätenie nastalo. Okrem týchto otázok sme riešili aj mnoho otázok spojených s funkčnosťou nami implementovanej infraštruktúry na zber dát myši používateľa. Na zodpovedanie týchto otázok a overenie návrhu štúdie sme zostavili pilotnú štúdiu počas prvej časti nášho diplomového projektu. Túto štúdiu opisujeme v prvej podkapitole.

Pilotná štúdia odhalila drobné nedostatky infraštruktúry, ktoré sme okamžite opravili. Všetky získané poznatky sme potom aplikovali pri hlavnej štúdii, ktorá sa uskutočnila v druhej časti nášho diplomového projektu a sumarizujeme ju v druhej podkapitole.

6.1 Pilotná štúdia

Pred návrhom štúdie na overenie našich hypotéz sme zostavili pilotnú používateľskú štúdiu, v rámci ktorej sme mali hneď niekoľko cieľov:

- *Nájdenie vhodného spôsobu získavania implicitnej spätnej väzby.* Naším cieľom bolo navrhnúť metódu, ktorou by sme vedeli identifikovať čas zmätenia používateľa. V tomto prípade sme sa sústredili na odhalenie presného času na základe explicitnej spätnej väzby.
- *Overenie infraštruktúry logovania aktivity myši a klávesnice.* Cieľom tohto kroku bolo overenie implementácie logera priamo na dátach zo správania používateľa. Keďže sme v rámci štúdie využili aj okulograf, dáta z neho poslúžili pri validovaní dát myši.

Celá štúdia prebiehala na fakulte informatiky a informačných technológií Slovenskej technickej univerzity v Bratislave v Centre používateľského zážitku a interakcie (*UXI Centrum*) [6]. Na nahrávanie pohľadu sme použili okulograf *Tobii TX300* s frekvenciou 300Hz a softvér *Tobii Studio*⁵. Celú štúdiu sme realizovali v spolupráci so študentom inžinierskeho štúdia fakulty Bc. Matejom Váľkym. Spolu s ním sme navrhli a realizovali celú štúdiu. Jeho práca sa zameriavala na interakciu s pomocníkmi vo webovej aplikácii z toho dôvodu boli niektoré úlohy prispôbené jeho výskumným otázkam. Na zaznamenávanie pohybu myši sme využili skript jazyka *javascript* vložený priamo do prehliadača *Mozilla Firefox*⁶. Ten generoval udalosti, ktoré posielal na serverovú časť napísanú nad rámcom *Django Rest* jazyka *Python*, ktorý ich ukladal do databázy. V miestnosti boli závesy a osvetlenie, ktoré bolo nastavené na konštantnú hodnotu počas celej štúdie.

Štúdie sa zúčastnili šiesti účastníci z personálu fakulty a študentov, ktorí boli náhodne vybraní. Každý absolvoval sedenie v plnom rozsahu 20 minút, počas ktorého boli v miestnosti moderátor a zapisovateľ. Pred samotným začiatkom experimentu bol každý z účastníkov oboznámený s

⁵www.tobii.com

⁶www.mozilla.org

celým postupom štúdie. Moderátor sedel celú dobu pri účastníkovi. Pred plnením úloh prebehla kalibrácia pohľadu.

Celý experiment prebiehal nad elektronickým obchodom Alza⁷ a pozostával z piatich úloh, ktorých obsah sa týkal vyhľadania ceny špecifického produktu, alebo zodpovedania otázok doménového typu. Kompletné znenie úloh a protokol štúdie nájdete v prílohe A. Po každej úlohe nasledovala jedno-výberová otázka na obtiažnosť úlohy s týmito možnosťami:

- Veľmi ľahká.
- Pomerne ľahká, ale myslím si, že sa dá vyriešiť efektívnejšie.
- Dost ľahká, ale mal/a som drobný problém, čo ma zdržalo.
- Pomerne náročná.
- Veľmi náročná.

Ďalej nasledovala otvorená otázka so znením: Opíšte priebeh úlohy pričom sa zamerajte na problémové časti. Po prejdení všetkých úloh a zodpovedaní poslednej otázky moderátor spolu s účastníkom spoločne prešli záznam z *Tobii Studia*. Účastník mal za úlohu opisovať kroky, ktoré vykonával a zamerať sa na zmätenie. Všetko bolo dokumentované zapisovateľom.

6.1.1 Vyhodnotenie implicitnej spätnej väzby

Pri prechádzaní videí s účastníkmi sme odhalili, že neboli schopní určiť bod, kedy zmätenie začalo. Každý vedel opísať miesto, kde zmätenie bolo, ale nevedeli určiť začiatok. Tieto určenia boli v zhode s otázkou ohľadom obtiažnosti úlohy, to znamená, že vždy keď účastník spätne slovné komentoval prítomnosť zmätenia počas vyhodnotenia videa tak tú istú úlohu označil ako náročnú/pomerne náročnú. *Pearsonov korelačný koeficient* v tomto prípade dosiahol hodnotu 0,6 na 30 vzorkách.

Záverom je zistenie, že používatelia nie sú schopní spätne anotovať začiatok stavu zmätenia. Avšak sú schopní spätne označiť úlohu, kde zmätenie nastalo.

6.1.2 Vyhodnotenie infraštruktúry logovania

V tejto časti sme prechádzali záznamy zozbierané pomocou našej implementácie logera. Sústredili sme sa na úspešnosť zaznamenávania našej infraštruktúry. Spätne sme modelovali pozície kurzora myši v čase pričom sme zmeny sledovali spoločne s video záznamom účastníkov. V tomto prípade šlo len o vizuálnu kontrolu a žiadne formálne výpočty presnosti neprebehli.

Záverom tejto časti bolo zistenie, že pomocou našej infraštruktúry sme schopní zaznamenávať reálne správanie myši používateľa. Odhalili sme problém spojený s nedoposielaním údajov o správaní pri tvrdej navigácii prehliadača (angl. *hard navigation*). Tento nedostatok sme napravili.

⁷www.alza.sk

6.2 Hlavná štúdia

Štúdia sa uskutočnila na Fakulte informatiky a informačných technológií Slovenskej technickej univerzity v Bratislave v laboratóriu používateľského zážitku pre skupinovú štúdiu UXI sentra (UX Class) [6]. Zmena laboratória bola podmienená typom štúdie, na rozdiel od pilotnej štúdie teraz išlo o kvantitatívnu štúdiu a vybraná miestnosť disponuje dvadsiatimi počítačmi vybavenými okulografom. Štúdie sa zúčastnilo 59 študentov fakulty, ktorí boli rozdelení do štyroch skupín, kde dve skupiny mali sedenie ráno od deviatej a dve vo večerných hodinách od siedmej. Študenti prišli na štúdiu dobrovoľne a každý z nich podpísal informovaný súhlas o spracovaní osobných údajov v zhode so zákonmi Slovenskej republiky.

Štúdia trvala dve hodiny a pozostávala zo šiestich úloh na dovolenkovom portáli *FiroTour*⁸. Zmena portálu štúdie bola spôsobená tým, že v dobe štúdie sme úzko spolupracovali s firmou, ktorá portál *FiroTour* vlastní. Tým pádom sme mali priamo prístup k zdrojovému kódu aplikácie, kde sme mohli náš logger jednoducho integrovať. Celý priebeh dokumentuje protokol štúdie v prílohe A. Na zaznamenávanie pohľadu sme použili okulograf *Tobii X2-60*, s frekvenciou 60Hz. Na zaznamenávanie pohybov myši a interakciu s klávesnicou sme použili náš logger. Účastníci po celú dobu interagovali s produkčnou verzou dovolenkového portálu (žiadne reklamy neboli odstránené).

Priebeh štúdie bol vytvorený v softvéri *iMotions*⁹. Ide o softvér na vytváranie používateľských štúdií s využitím okulografu. Dôvod zmeny softvéru na štúdiu bol v návrhu štúdie, ktorého súčasťou boli otvorené otázky po každej z úloh. Pôvodný softvér *Tobii Studio* neposkytoval možnosť jednoducho zakomponovať takýto druh otázky. Celá štúdia v *iMotions* bola tvorená dvomi časťami. V prvej prebehla kalibrácia okulografu, ktorú tvorila štandardná kalibračná sekvencia okulografu a kalibračná sekvencia zreničky účastníka. Druhá časť pozostávala z plnenia šiestich úloh. V *iMotions* sme vytvorili 6 blokov, ktoré tvorili hlavnú sekvenciu. Každý z blokov pozostával z troch stimulov:

1. *Stimul s inštrukciami*. Štandardný text so znením úlohy. Na konci textu bolo vždy spomenuté tlačidlo f10, ktorým sa dalo medzi jednotlivými časťami štúdie prechádzať. Zvolili sme biely text na sivom pozadí.
2. *Stimul s url*. Po prechode na tento stimul sa účastníkovi sám otvoril prehliadač a bol presmerovaný na url, kde začínala aktuálna úloha. Ako prehliadač sme zvolili *Google Chrome*¹⁰. Dôvod zmeny prehliadača bol spojený s implementáciou logera, ktorá pracuje spoľahlivejšie práve na prehliadači *Google Chrome*. Používateľ mal po dokončení stlačiť tlačidlo f10 na prechod do tretej časti bloku.
3. *Stimul s dotazníkom*. Posledná časť bloku bola tvorená stimulom s tromi otázkami. Prvá bola určená na odpoveď na úlohu. Otázka plnila validačnú funkciu pri neskoršom vyhodnotení štúdie. Druhá bola jedno-výberová otázka o stlačení tlačidla počas úlohy. Opäť úloha

⁸www.frotour.sk/

⁹www.imotions.com

¹⁰www.google.com/chrome/

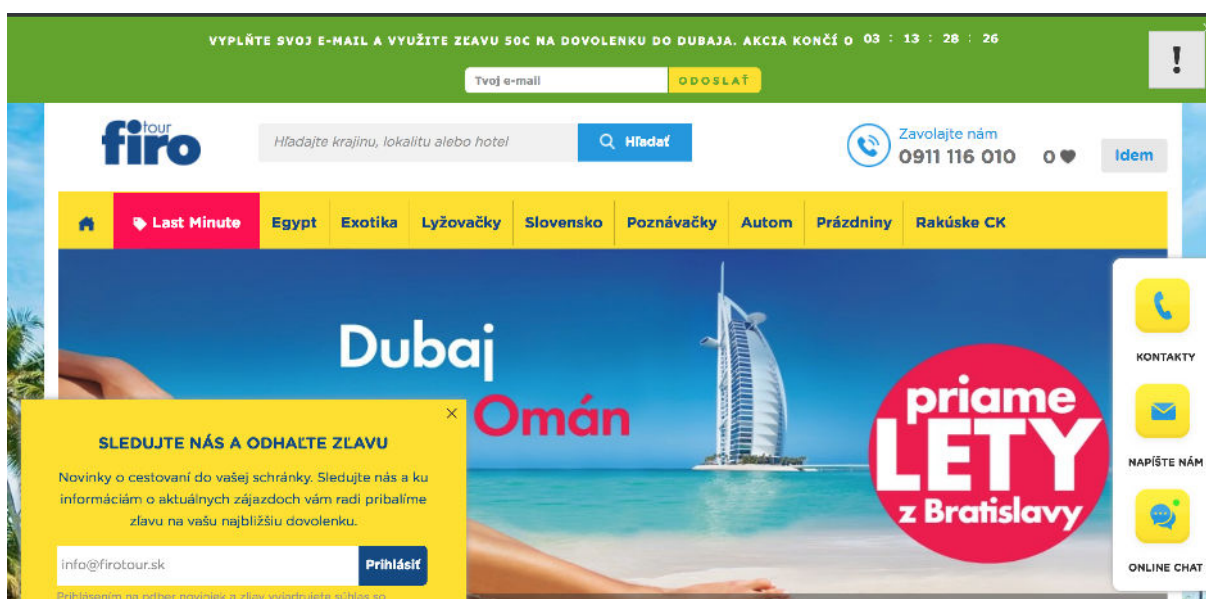
s validačnou funkciou, či používateľ naozaj stlačil tlačidlo. Posledná otvorená otázka slúžila na opis úlohy a opis problémov, ktoré počas riešenia nastali. Po dokončení bolo opäť potrebné stlačiť f10.

Softvér *iMotions* umožňuje náhodné prerozdelenie blokov v rámci spustenej sekvencie. Vďaka tomu bolo zaručené, že každý účastník mal úlohy v náhodnom poradí. Výber úloh sme riešili priamo s cestovnou kanceláriou *FiroTour*. Stretli sme sa so zamestnancami zákazníckeho servisu, ktorí nás informovali o najčastejších problémov aplikácii.

6.2.1 Získavanie spätnej väzby

Kľúčovou časťou používateľskej štúdie bolo zachytávanie spätnej väzby používateľa. Bolo potrebné vystihnúť správny moment, kedy bol používateľ zmätený. Počas pilotnej štúdie sme na určovanie použili dotazník po každej úlohe, kde sme sa pýtali na problémy s úlohou. Tento postup sa však ukázal ako neefektívny. Účastníci mnohokrát zabudli na problémy, ktoré počas úlohy vznikli, a teda ich nespomenuli v dotazníku po úlohe. Navyše sme zistili, že účastníci nie sú schopní spätne odhadnúť začiatok zmätenia.

V hlavnej štúdii sme navrhli a implementovali softvérové tlačidlo do pravého horného rohu stránky. Vychádzali sme pritom z práce [28], ktorú sme analyzovali v kapitole 2.3. Namiesto textu autorov "som zmätený" (angl. *I am confused*) sme však použili symbol "!". Dôvodom bolo nevyvolávať pocit chyby na strane používateľa. Tlačidlo bolo súčasťou logera. Po kliknutí naň podobne ako v citovanej práci nenastala žiadna spätná väzba pre používateľa. Namiesto toho sa odoslala udalosť na server o stlačení, ktorá bola uložená do databázy. Obrázok 4 poskytuje ukážku umiestnenia a dizajnu tlačidla.



Obr. 4: Ukážka použitia tlačidla na získavanie spätnej väzby zmätenia. Tlačidlo sa nachádza v pravom hornom rohu a je označené symbolom "!".

Tlačidlo sme opísali nasledujúcim textom: "Tlačidlo "!": V pravej hornej časti obrazovky prehliadača uvidíte tlačidlo s textom "!". Prosíme vás o to, aby ste na toto tlačidlo klikli vždy vtedy, keď budete potrebovať pomoc s úlohou. Tlačidlo má simulovať stav, kedy by ste chceli stránku opustiť z dôvodu, že by ste nevedeli kde hľadať potrebnú informáciu na splnenie úlohy. Taktiež je na toto tlačidlo vhodné kliknúť vtedy, ak by ste chceli na splnenie úlohy použiť inú aplikáciu ako <doména> (napríklad google). Po stlačení tlačidla nenastane žiadna udalosť zo strany systému. Následne môžete v úlohe pokračovať alebo prejsť ďalej tlačidlom F10."

6.2.2 Mapovanie identít účastníka

Pri experimente sme pracovali s dvomi hlavnými zdrojmi dát o správaní používateľa. Prvým z nich bol okulograf. Identita pri dátach účastníka v tomto prípade bola riešená cez *iMotions*, kde pred začatím štúdie bolo potrebné uviesť identifikátor. Následne uvedený identifikátor bol priradený každému záznamu okulografu. Problém však bol v prípade loggeru myši na stránke FiroTour. Logger nemohol vedieť o identite používateľa, ktorý na ňom pracoval. Pri štúdiu sme totiž použili nami vyvinutú produkčnú verziu skriptu na logovanie dát, ktorá logovala všetkých používateľov na stránke dovolenkového portálu bez nutnosti identity, len na základe *visitorId*. Tento identifikátor bol vygenerovaný pri prvej návšteve a uložený do lokálnej pamäte prehliadača (angl. *local storage*).

Na vyriešenie situácie sme použili mapovanie ip adresy počítačov v učebni, v ktorej štúdia prebiehala, na identifikátor v štúdiu. Počítač si teda vždy pri prvom navštívení url vypýtal identifikátor z externej služby. Následne tento identifikátor pridal do každého záznamu o používaní myši. Ako externú službu sme využili *Bridge*¹¹. Ako identifikátor sme zvolili 6-členný reťazec písmen a čísel. Jeho aplikáciou sme prispeli k anonymizovaniu dát.

¹¹www.bridge.buddyweb.fr/en

6.3 Exploratívna analýza surových dát

6.3.1 Získané dáta

Každý z 59 účastníkov úspešne absolvoval všetkých 6 úloh štúdie. Menšie technické komplikácie sa prejavili pri práci so softvérom, určeným na priebeh štúdie. Z dôvodu týchto komplikácií sme prišli o odpovede po úlohách u troch účastníkov. To znamená, že celkovo sme zozbierali 336 odpovedí na dotazníky po úlohe.

Pri zaznamenávaní správania myši používateľa sme odhalili menšie technické problémy. Tie sa prejavili pri rozlišovaní úlohy, na ktorej používateľ práve bol. Rozlišovanie úloh sme riešili skrytým identifikátorom na úvodnej url úlohy. Pri 7 úlohách sa však stalo, že sa účastník preklikol na podstránku bez identifikátora skôr ako mohla byť informácia so skrytým identifikátorom zaznamenaná na serveri. Z tohto dôvodu sme pri týchto účastníkoch neboli schopní zrekonštruovať poradie úloh a s istotou priradiť záznamy myši k jednotlivým úlohám. Vyriešili sme to vylúčením týchto účastníkov zo vzorky.

Pri dotazníku na konci každej úlohy sme sa účastníkov pýtali na to, či počas riešenia stlačili tlačidlo '!'. Táto otázka plnila úlohu spätnej kontroly tlačidla. Porovnaním prijatej udalosti na strane serveru a odpoveďou na túto otázku sme odhalili technické problémy pri 22 úlohách datasetu. Vždy sa jednalo o problém, kedy účastník v dotazníku označil, že tlačidlo stlačil avšak na server táto informácia nedošla. Nenašli sme žiadny prípad, kedy by to bolo naopak. Následne sme sa pozreli na príčinu týchto nedostatkov. Jedno z možných vysvetlení je omyl na strane účastníka, keďže niektoré úlohy boli veľmi podobné. Iný dôvod problému mohol byť na strane serveru, ktorý túto informáciu zle spracoval. Pre korektnosť výsledkov sme tieto úlohy vyradili z datasetu.

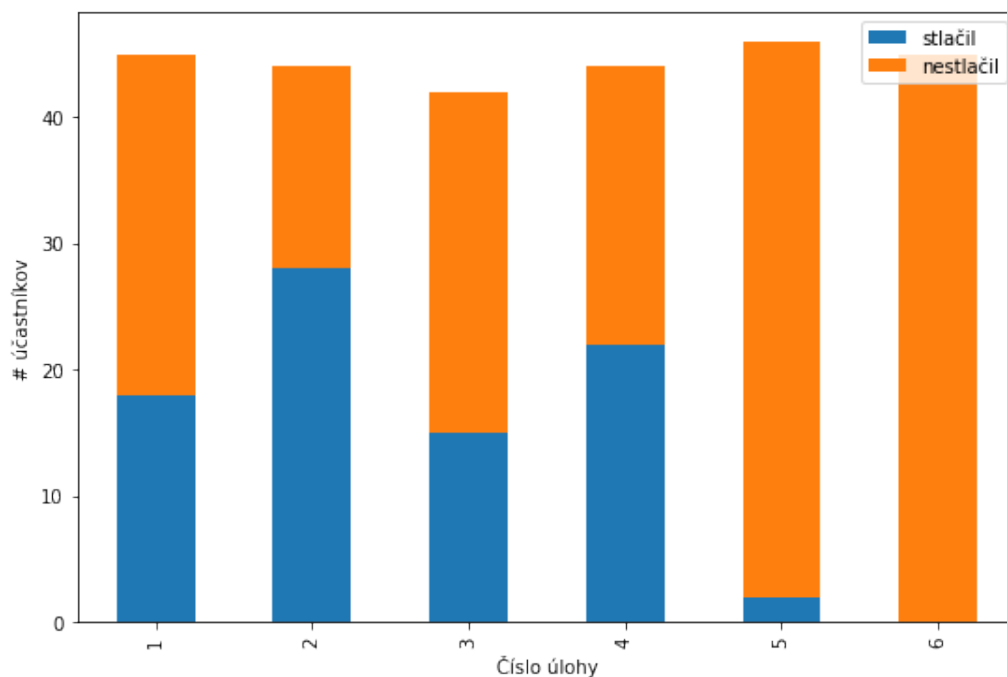
Po aplikácii spomenutých filtrov sme získali dataset o veľkosti 266 záznamov o riešení. Rozdelenie počtov záznamov medzi úlohy reprezentuje tabuľka 10.

Tabuľka 10: Prehľad počtov záznamov jednotlivých úloh v očistenom datasete.

Číslo úlohy	1	2	3	4	5	6
Počet záznamov	45	44	42	44	46	45

6.3.2 Tlačidlo zmätenia

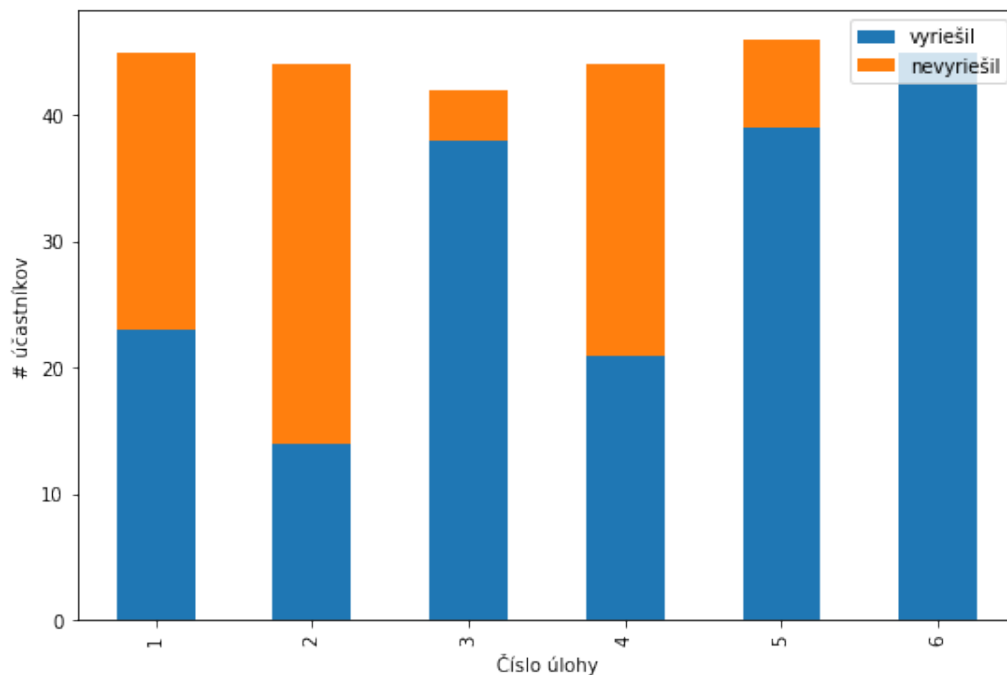
Po dôkladnom čistení datasetu sme analyzovali zmätenie používateľa. Pri návrhu úloh sme dbali na to, aby nazbierané dáta preukázali výskyt zmätenia v úlohách. Obrázok 5 ukazuje na počty stlačenia tlačidla '!'. pri jednotlivých úlohách.



Obr. 5: Graf rozdelenia stlačenia tlačidla '?' pre jednotlivé úlohy.

Na grafe si môžeme všimnúť, že s úlohami 1 až 4 mali účastníci značné problémy. Úloha 2 bola dokonca tak náročná, že viac ako polovica účastníkov stlačila tlačidlo '?' počas jej riešenia. Na druhej strane tu máme úlohy 5 a 6, kde účastníci nemali skoro žiadne problémy pri riešení. Išlo o úlohu jednoduchého vyhľadania informácie, kde stačilo prejsť na inú podstránku a informáciu vyhľadať. Tieto úlohy boli vybrané s cieľom kontrolnej vzorky správania, kedy používateľ nemá žiaden problém.

V ďalšom kroku sme analyzovali odpovede na úlohy. Po každej úlohe nasledoval dotazník, kde účastníci mohli napísať odpoveď na úlohu. Ručne sme anotovali tieto odpovede. Pri anotácii sme sa sústredili na to, či účastník úlohu vyriešil. Teda nešlo nám o správnosť odpovede na úlohu. Za nevyriešené sme brali iba prípady, kedy pole na odpoveď zostalo nevyplnené alebo iné prípady, kedy účastník explicitne vyjadril, že úlohu nevie splniť alebo informácia, ktorú tam napísal nesúvisela s úlohou. Obrázok 6 ukazuje na počty vyriešených záznamov pri jednotlivých úlohách.



Obr. 6: Graf rozdelenia vyriešených záznamov pri jednotlivých úlohách.

Na grafe si môžeme všimnúť očakávané výsledky v prípade úlohy 6, kedy všetci tí čo nestlačili tlačidlo '!' zároveň úlohu vyriešili. Na ostatných grafoch mohol nastať jeden z nasledujúcich javov:

1. Účastník stlačil tlačidlo '!' a úlohu nevyriešil. Toto je očakávaný jav.
2. Účastník nestlačil tlačidlo '!' a úlohu vyriešil. Toto je opäť očakávaný jav.
3. Účastník stlačil tlačidlo '!' a úlohu vyriešil. Toto je jav kedy používateľ, potom čo stlačil tlačidlo "!" úlohu nevzdal a pokračoval v nej.
4. Účastník nestlačil tlačidlo '!' a úlohu nevyriešil. Toto je neočakávaný stav v správaní. Jedným z vysvetlení je, že účastník zabudol kliknúť na tlačidlo pred tým ako sa prepol na inú úlohu. Prípadne, že zmätenie nenastalo, účastník jednoducho úlohu nevedel a neriešil to.

Tabuľka 11 poskytuje početnosti jednotlivých stavov v datasete. Zamerali sme sa na problematické prípady, kedy účastník nestlačil tlačidlo a zároveň nenašiel informáciu k úlohe. Keďže po každej úlohe bol dotazník, kde účastníci mohli opísať priebeh riešenia úlohy, prešli sme tieto odpovede a zistili sme, že vo väčšine prípadoch šlo o to, že používateľ si pomýlil úlohu alebo len nič nenapísal.

Tabuľka 11: Prehľad stavov účastníkov vzhľadom k stlačeniu tlačidla a vyriešeniu úlohy.

Stav	Stlačil a nevyriešil	Nestlačil a vyriešil	Stlačil a vyriešil	Nestlačil a nevyriešil
Počet záznamov	69	154	26	27

6.4 Exploratívna analýza metrík

Po preskúmaní získanej explicitnej spätnej väzby používateľa sme generovali metríky myši pre predikciu. V prvom kroku sme pripravovali dáta na predikciu nad celou úlohou. Tu sme zobrali dáta z úlohy a v prípade, že počas riešenia nastalo stlačenie tlačidla označili sme ich ako zmätené, ak nenastalo označili sme ich ako nezmätené. V tomto kroku sme aj napárovali dáta z okulografu pre jednotlivé úlohy. Pre zaujímavosť spomenieme, že v dátach sme našli aj 4 prípady, kedy bolo tlačidlo zmätenia stlačené viac ako jeden krát počas úlohy (časová vzdialenosť medzi kliknutiami bola väčšia ako 3 sekundy, teda nešlo o dvojklik).

V druhom kroku sme dáta rozdelili na časové okná rovnakej veľkosti s sekúnd. Pri generovaní okien sme ako zmätené označili len okná, ktoré predchádzali prvej udalosti stlačenia tlačidla v úlohe. Na generovanie nezmätených okien sme použili len úlohy, kde stlačenie tlačidla nenastalo. Takéto okná sme vytvorili pomocou už spomínaného algoritmu posúvajúceho sa okna, pričom okno sa vždy posúvalo o jednu sekundu. Pri overovaní predikcie v reálnom čase sme zvolili okná veľkosti 5 sekúnd a 10 sekúnd podobne ako v spomínaných prácach [29, 28].

Pokračovali sme extrakciou pohybov myši z dát. V oboch prípadoch (pri predikcii nad celou úlohou aj pri predikcii v reálnom čase) sme koordináty zoskupili do pohybov. Pohyb je definovaný ako súbor minimálne 4 súradníc, kde rozdiel časov medzi súradnicami je menší ako 120 ms [15]. Pri predikcii v reálnom čase sme vypočítali metríky uvedené v návrhu metódy 5.1. Pri predikcii nad celým sedením sme pre každú z črt myši vypočítali histogram jej hodnôt pre pohyby. Histogramy sme opisovali v podkapitole 3.3. Stanovili sme pevný počet štyroch stĺpcov pre každú črtu a hranice sme určili na základe lineárneho rozdelenia minima a maxima hodnôt danej premennej z celého datasetu. To znamená, že ak by sa časy pohybov pre všetky pozorovania pohybovali od 0 do 100, hraničné hodnoty histogramových stĺpcov by boli nasledovné $\langle 0;25 \rangle$, $\langle 25;50 \rangle$, $\langle 50;75 \rangle$, $\langle 75;100 \rangle$. V prípade 5 a 10 sekundových okien takéto spracovanie nemá zmysel, keďže okná obsahujú priemerne menej ako 3 pohyby.

V práci porovnávame našu metódu predikcie nad celou úlohou s už spomínanou prácou [42], kde autori predikovali obtiažnosť sedenia na základe logov *HTTP* serveru. Na to aby sme porovnanie zrealizovali bolo potrebné vypočítať črty zhrnuté v podkapitole 3.5. Keďže sme po štúdiu nemali prístup k *HTTP* logom vypočítali sme tieto metríky priamo z dát myši. Logy myši totiž v sebe niesli informáciu o url, na ktorej daná udalosť nastala. Vďaka tomu sa nám podarilo vypočítať metríky celkový čas, počet stránok, čas na stránku a počet cyklov. Naša infraštruktúra však neposkytovala informácie o návšteve iných stránok z toho dôvodu sme nevedeli vypočítať metriku počet zámen. Tieto metríky boli vypočítané len pre predikciu nad celou úlohou, keďže nemalo zmysel ich rátať pre okná malých veľkostí.

Pokračovali sme spracovaním metrík okulografu. Väčšiu časť predspravovania z metódy za nás vyriešila práve zvolená infraštruktúra *iMotions*. Vygenerovaný export obsahoval informácie o sakádach a fixáciách pre každý záznam. Softvér za nás aj odfiltroval nevalidné záznamy. V rámci generácie črt bolo potrebné dopočítať črty zreničky a vzdialenosti od monitora. Tieto metriky boli vypočítané len pre predikciu nad celou úlohou, kvôli technickým komplikáciám pri mapovaní okien okulografu na okná myši.

6.4.1 Metriky myši

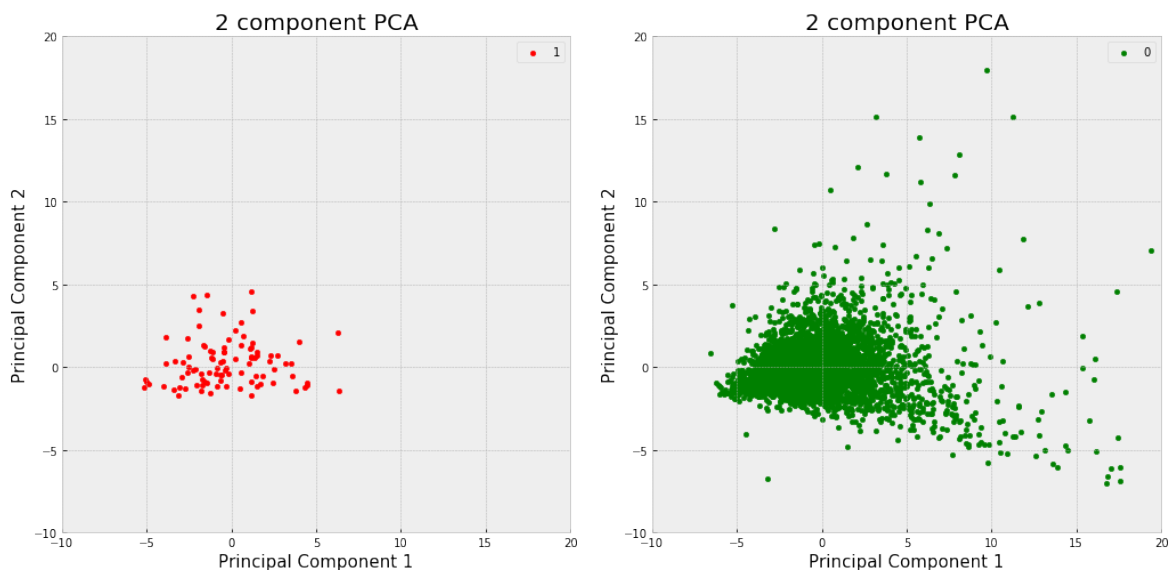
V prvom kroku sme sledovali koreláciu každej z vypočítaných metrík voči zmäteniu v danom okne (voči celému sedeniu v prípade datasetu určenom na predikciu nad celým sedením). V prípade okien veľkosti 5 a 10 sekúnd sme nenašli žiadnu značnú koreláciu s využitím *Pearsonovho koerelačného testu*. Všetky korelácie boli menšie ako 0,1 najvyššiu koreláciu dosahovali metriky rýchlosti a štandardizovanej rýchlosti (hodnoty okolo 0,04). Z týchto výsledkov sa dalo usúdiť, že v dátach nie je žiadna značná lineárna závislosť voči predikovanému stĺpcu. Toto tvrdenie sme sa snažili potvrdiť aj pomocou analýzy principiálnych komponentov (angl. *principal component analysis*), ďalej iba *PCA*. Pomocou *PCA* sme redukovali dimenzionalitu datasetu len na dva principiálne komponenty. Následne sme dáta vizualizovali pomocou grafov v dvojrozmernom priestore.

Obrázok 7 poskytuje ukážku vizualizácie pre okná veľkosti 10 sekúnd. Môžeme si na ňom všimnúť, že zmätené pozorovania (červenou) sú centrované okolo bodu $[0,0]$ podobne ako nezmätené. Jediný rozdiel je v rozptyle pozorovaní, ktorý je pri nezmätených vzorkách väčší. Taktiež z grafu vidieť, že ani po redukcii nejde o lineárne separovateľný problém. Podobnú vizualizáciu sme dostali aj v prípade okien veľkosti 5 sekúnd a pri predikcii nad celou úlohou. Je vhodné spomenúť, že samotnému kroku redukcii predchádzala analýza normálneho rozdelenia jednotlivých metrík. Tá zahŕňala vizualizáciu histogramov jednotlivých metrík. Každý zo stĺpcov spĺňal predpoklady normálneho rozdelenia. Každú z metrík sme normalizovali s využitím štandardného škálovania.

Pri predikcii nad celou úlohou s využitím histogramovej reprezentácie črt sme zaznamenali vyššie korelácie a to v prípade zrýchlenia, kde *Pearsonov koerelačný koeficient* dosiahol hodnotu viac ako 0,18. Zaujímavý výsledok korelácie sme dosiahli v prípade HTTP metriky počet cyklov nad celou úlohou, kde táto metrika dosahovala hodnotu koeficientu až 0,46. Taktiež metrika počtu stránok dosiahla hodnotu 0,41. Na druhej strane metrika času na úlohe dosiahla hodnotu len 0,09 z čoho vyplýva, že zmätenie nekorelovalo s dĺžkou riešenia úlohy.

Pokračovali sme zhotovením jednoduchých prediktorov nad jednotlivými HTTP metrikami. Šlo o klasifikátor logistickej regresie, ktorý predikoval zmätenie vždy len na základe jednej z metrík. Podobný prístup zvolili autori práce [42]. Cieľom týchto prediktorov bolo odsledovať hraničné hodnoty, ktoré sa klasifikátory na dátach naučia.

V tabuľke 12 môžeme vidieť výsledky. Zaujímavý je posledný stĺpec, ktorý slovne interpretuje naučené hranice. Je vidieť, že v prípade počtu cyklov, ak nastali viac ako 2, prediktor sa naučil, že ide o stav zmätenia. Podobný jav nastal aj v prípade počtu stránok.



Obr. 7: Vizualizácia datasetu vzoriek nad 10 sekundovými oknami pomocou redukcie dimenzionality s využitím PCA nad črtami myši. Zmätené pozorovania sú označené červenou a nezmätené pozorovania zelenou

Tabuľka 12: Jednoduché prediktory s využitím individuálnych črt HTTP serveru pri predikcii nad dátami z celej úlohy.

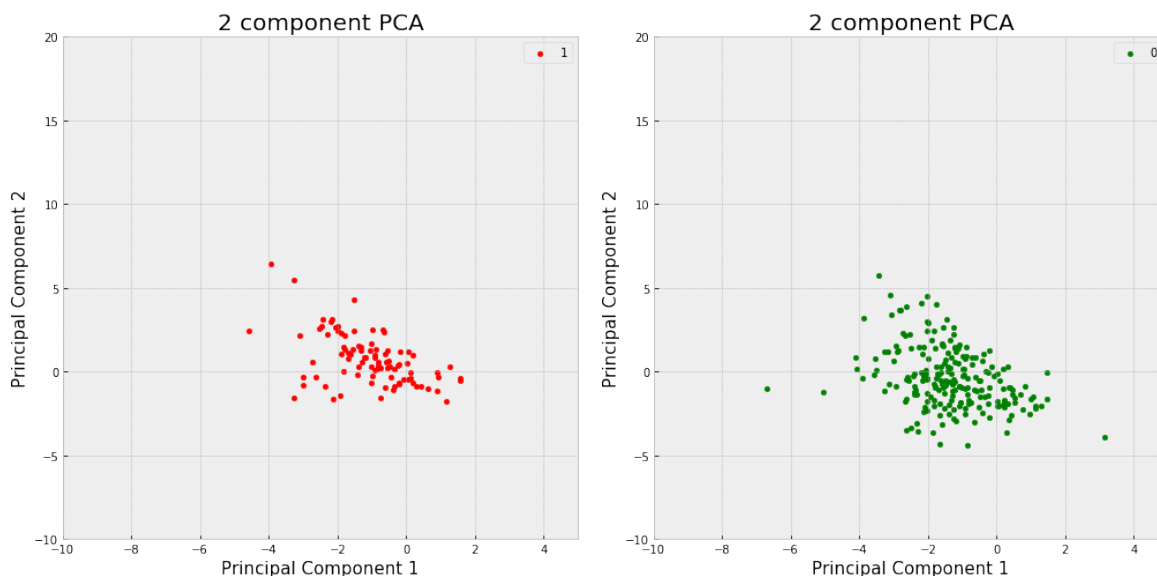
Črta	Presnosť	Úplnosť	AUC	AIC	Naučená hranica
Celkový čas	N/A	0	0.5	258.82	Nikdy nepredikuje zmätenie
Počet stránok	0.71	0.46	0.68	258.74	>9 stránok
Čas na stránku	N/A	0	0.5	258.82	Nikdy nepredikuje zmätenie
Počet cyklov	0.77	0.38	0.66	258.45	>2 cykly
Najlepšia kombinácia	0.72	0.50	0.7	226.43	

6.4.2 Metriky okulografu

Kvôli problémom s mapovaním dát z okulografu na okná veľkosti 5 a 10 sekúnd sme tieto metriky využili iba pri predikcii nad dátami z celej úlohy. Navyše cieľom tejto práce je navrhnúť metódu na predikciu v reálnom čase s využitím v produkčnom prostredí. Keďže okulograf nie je dnes ešte bežnou súčasťou hardvérového vybavenia návštevníkov webových aplikácií, využitie len dát z myši v produkčnom prostredí, t.j. pri predikcii zmätenia v reálnom čase, je naším úmyslom.

V prípade metrík okulografu sme postupovali rovnako ako pri metrikách myši. V prvom kroku sme sa pozreli na normálne rozdelenie metrík. Opäť aj v tomto prípade už samotný vizuálny test toto rozdelenie potvrdzoval. V nasledujúcom kroku sme sledovali koreláciu voči zmäteniu. Najvyššiu hodnotu 0,33 dosiahla metrika počtu fixácií. Všetky ostatné metriky mali hodnoty podstatne nižšie. Pre zaujímavosť sme skúsili opäť vizualizáciu v dvojrozmernom priestore pomocou redukcie dimenzionality algoritmom *PCA*. Obrázok 8 poskytuje prehľad, na ktorom si

môžeme všimnúť, že po aplikovaní algoritmu nejde o lineárne separovateľný problém.



Obr. 8: Vizualizácia datasetu vzoriek nad celou úlohou pomocou redukcie dimenzi-
onalita s využitím PCA nad črtami okulografu. Zmätené pozorovania sú označené
červenou a nezmätené pozorovania zelenou.

6.5 Výsledky predikcie

Všetky výpočty a vizualizácie sme realizovali pomocou knižníc jazyka *python* v prostredí *jupyter lab*¹². Na prácu s dátami sme využili knižnice *pandas*¹³ a *numpy*¹⁴. Na vizualizáciu dát sme použili knižnice *matplotlib*¹⁵ a *seaborn*¹⁶. Na trénovanie predikciu a výpočet metrík pri overovaní sme použili knižnicu *scikit-learn*¹⁷ a na balancovanie datasetu sme použili knižnicu *imblearn*¹⁸. Dáta z logeru sú uchovávané v databáze *elasticsearch*¹⁹, zatiaľ čo dáta z okulografu sme uchovali vo formáte súborov hodnôt oddelených čiarkou (angl. *comma separated values*).

Pri predikcii sme v prvom kroku dáta rozdelili s využitím vrstviacej troj-skupinovej krížovkej validácie (angl. *stratified 3-fold cross validation*). V rámci každej skupiny boli dáta rozdelené v pomere 80 percent objemu pre trénovaciu sadu ku 20 percentám pre testovaciu. Následne pre každú trénovaciu sadu sme natrénovali klasifikátor s využitím prehľadávania mriežky (angl. *grid search*), ktorým sme sa snažili nájsť optimálne parametre pri maximalizovaní metriky presnosť. V rámci prehľadávania mriežky sme opäť použili vrstviacu troj-skupinovú krížovú validáciu. Vybrali sme vždy optimálnu možnosť mriežky, ktorej výsledky predikcie, nad testovacou sadou,

¹²www.jupyterlab.readthedocs.io/en/stable/index.html

¹³www.pandas.pydata.org/

¹⁴www.numpy.org/

¹⁵www.matplotlib.org/

¹⁶www.seaborn.pydata.org/

¹⁷www.scikit-learn.org/stable/

¹⁸www.imbalanced-learn.readthedocs.io/en/stable/index.html

¹⁹www.elastic.co/

sme zapísali do spoločného pola. Toto sme opakovali pre každú skupinu. Na konci sme vypočítali metriky nad týmto polom predikcií.

6.5.1 Výsledky nad celou úlohou

Ako prvé poskytujeme výsledky predikcie nad dátami z celej úlohy. Dôvod výberu tohto scenáru bolo porovnanie s existujúcim prístupom uvedeným v práci [42] na predikciu obtiažnych sedení na stránke. Pri predikcii sme využili histogramovú reprezentáciu základných metrick myši, metriky *HTTP* serveru odvodené z metrick myši a metriky okulografu.

Tabuľka 13 poskytuje výsledky rôznych experimentov. Môžeme si všimnúť pomerne dobré výsledky v prípade predikcie s využitím metrick *HTTP* serveru. Vysoká presnosť spolu s vysokou hodnotou *AUC* svedčia v prospech tohto výsledku. Potvrdila sa vysoká korelácia týchto metrick z predchádzajúcej podkapitoly. Ďalej je zaujímavé porovnanie výsledkov metrick myši a okulografu. Ako si môžeme všimnúť metriky myši majú len o niečo horšie výsledky.

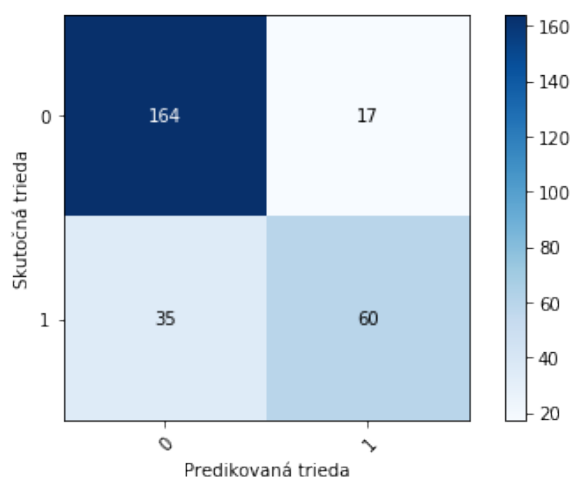
V experimentoch sme pokračovali ďalej a to s využitím metrick myši a *HTTP* serveru. Výber týchto metrick bol podmienený využitím v reálnej produkcii, keď okulograf ako taký momentálne nie je štandardným hardvérovým zariadením počítačov. Tu sme dosiahli najlepšie výsledky s presnosťou až 0,82. Taktiež je vhodné vyzdvihnúť vysokú úplnosť s hodnotou až 0,83.

V tabuľke 13 taktiež poskytujeme výsledky v prípade predikcie s využitím vyvažovania datasetu pomocou *SMOTE* algoritmu [1]. Šlo o nastavenie, kedy sa počet zmätených pozorovaní vyrovnal počtu nezamätených. Môžeme sledovať mierne zlepšenie v prípade predikcie nad dátami *HTTP* serveru. V ostatných prípadoch zlepšenie nenastalo alebo nastalo zhoršenie. Pre úplnosť uvádzame maticu zmätenia pre najlepší z klasifikátorov na obrázku 10. Najväčším problémom je 29 mylne klasifikovaných pozorovaní, kedy boli zamätené pozorovania označené ako nezamätené. Pri reálnom využití prediktoru toto môže spôsobovať problém, kedy sa nepodari odhaliť používateľov s problémami.

Záver tohto scenára je taký, že s využitím dát z myši a *HTTP* serveru sme schopní predikovať zmätenie na celej úlohe s vysokou presnosťou a úplnosťou. Takto natrénovaný klasifikátor sa dá využiť pri spätnej analýze sedení s cieľom odhaliť problémové sedenia, prípadne problémy použiteľnosti v aplikácii. Rovnaký postup predikcie obtiažnych sedení bol použitý v spomínanom článku [42], kde autori dosiahli úspešnosť 0.86 na dátach z *HTTP* serveru.

Tabuľka 13: Porovnanie predikcie pomocou rôznych kombinácií metrík s využitím algoritmu Logistickej regresie. Stĺpce označené slovom *SMOTE* prezentujú výsledky klasifikácie pri použití vyvažovania algoritmom SMOTE [1]. Výsledky boli získané aplikáciou vrstviatej troj-skupinovej krížovej validácie.

Črta	Presnosť	Presnosť (SMOTE)	Úplnosť	Úplnosť (SMOTE)	AUC	AIC
Najlepšia kombinácia metrík HTTP serveru	0.72	0.76	0.71	0.74	0.7	226.43
Metriky myši	0.59	0.59	0.63	0.58	0.58	268.86
Metriky okulografu	0.69	0.74	0.70	0.73	0.59	174.03
Najlepšia kombinácia bez okulografu	0.82	0.82	0.83	0.82	0.86	207.36
Všetky metriky	0.82	0.81	0.83	0.81	0.88	240.41



Obr. 9: Matica zmätenia pri klasifikátore Logistickej regresie pri predikcii zmätenia nad dátami z celej úlohy s využitím vrstviatej troj-skupinovej krížovej validácie (angl. *stratified 3-fold cross validation*). Pri tréningu a predikcii sa použili metriky odvodené z myši a z HTTP serveru. Zmätenia má označenie "1".

6.5.2 Výsledky nad časovými oknami

V nasledujúcom scenári sme predikovali v reálnom čase s využitím časových okien rôznej veľkosti. Kvôli časovej zložitosti výpočtu prezentujeme výsledky okien veľkosti 10 a 5 sekúnd. Pri predikcii sme využili črty dokumentované v kapitole 5.1.

Počas predikcie sme riešili problém s nevyváženým datasetom. Zmätených pozorovaní bolo podstatne menej ako tých nezmätených. V prípade okien veľkosti 10 sekúnd bolo nezmätených pozorovaní viac ako 36 krát viac a v prípade okien veľkosti 5 to bolo viac ako 50 krát viac. Ako riešenie tohto problému sme využili algoritmus *SMOTE* [1], určený na vyvažovanie datasetu. Použili sme implementáciu vyvažovania na základe piatich najbližších susedov. Pričom sme zvolili hneď niekoľko nastavení pri navyšovaní minoritnej triedy a to o 200, 500 a 2000 percent.

Podobné experimenty využili autori v práci [29].

V tabulke 14 prezentujeme výsledky predikcie na oknách veľkosti 10 sekúnd a v tabulke 15 výsledky pre okná veľkosti 5 sekúnd. Ako klasifikátor sme použili logistickú regresiu (LR), avšak pre porovnanie poskytujeme aj prehľad výsledkov pre klasifikátor náhodného lesu (angl. *Random forest*, *RF*). V jednotlivých riadkoch si môžeme tiež všimnúť výsledky pre rôzne nastavenia vyvažovania (označenie SMOTE). Pri tréňovaní prediktorov pomocou prehľadávania mriežky sme sa sústredili na maximalizovanie presnosti. Dôvod bol taký, že v produkcii sa sústredíme na to, aby sme si boli čo najistejší, keď označíme niekoho za zmäteného. Chceme sa vyvarovať zbytočnému obťažovaniu ľudí, ktorí reálne zmätení nie sú. Z toho dôvodu sa ako najlepší preukázal klasifikátor logistickej regresie pri vyvažovaní minoritnej triedy o 200 percent.

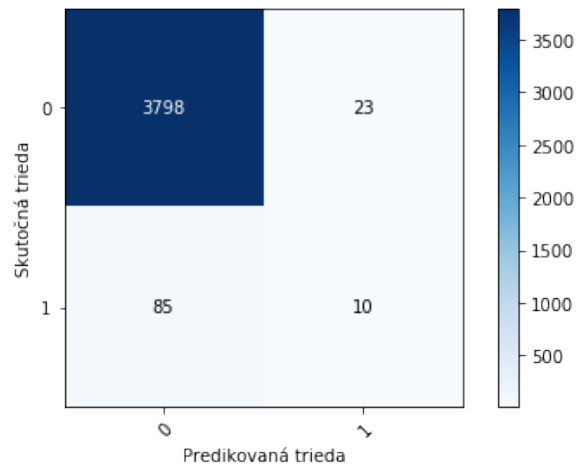
Tabuľka 14: Porovnanie predikcie pomocou algoritmu logistickej regresie *LR* a náhodného lesa *RF* na oknách veľkosti 10 sekúnd. Riadky označené slovom *SMOTE* prezentujú výsledky klasifikácie pri použití vyvažovania algoritmom SMOTE [1]. Výsledky boli získané aplikáciou vrstviatej troj-skupinovej krížovej validácie.

	Presnosť(0)	Presnosť(1)	Úplnosť(0)	Úplnosť(1)
LR	0.98	0.30	0.99	0.11
LR SMOTE 200%	0.98	0.36	0.09	0.09
LR SMOTE 500%	0.98	0.24	0.99	0.11
LR SMOTE 2000%	0.98	0.15	0.96	0.28
RF	0.98	0	1	0
RF SMOTE 200%	0.98	0.25	1.00	0.02
RF SMOTE 500%	0.98	0.27	1	0.07
RF SMOTE 2000%	0.98	0.08	0.95	0.25

Tabuľka 15: Porovnanie predikcie pomocou algoritmu logistickej regresie *LR* a náhodného lesa *RF* na oknách veľkosti 5 sekúnd. Riadky označené slovom *SMOTE* prezentujú výsledky klasifikácie pri použití vyvažovania algoritmom SMOTE [1]. Výsledky boli získané aplikáciou vrstviatej troj-skupinovej krížovej validácie.

	Presnosť(0)	Presnosť(1)	Úplnosť(0)	Úplnosť(1)
LR	0.97	0	0.98	0
LR SMOTE 200%	0.97	0.03	0.97	0.03
LR SMOTE 500%	0.97	0.02	0.95	0.03
LR SMOTE 2000%	0.97	0.03	0.88	0.13
RF	0.97	0	1	0
RF SMOTE 200%	0.97	0	1	0
RF SMOTE 500%	0.97	0	1	0
RF SMOTE 2000%	0.97	0.02	0.97	0.02

Porovnaním výsledkov predikcie s využitím okien 10 a 5 sekúnd môžeme povedať, že zväčšovaním okna získavame presnejšie výsledky. Svedčia o tom aj výsledky na celej úlohe, kde je presnosť predikcie najvyššia. Pri predikcii s využitím okien sa najlepšie javí klasifikátor logistickej regresie s 200 percentným vzorkovaním, kde sme dosiahli presnosť na predikcii zmätenia až 0,98. Pre úplnosť poskytujeme maticu zmätenia s využitím tohto klasifikátoru.



Obr. 10: Matica zmätenia pri klasifikátore logistickej regresie a 200 percentného vzorkovania minoritnej triedy pri predikcii zmätenia nad dátami o veľkosti okien 10 sekúnd, s využitím vrstviacej troj-skupinovej krížovej validácie (angl. *stratified 3-fold cross validation*). Pri trénovaní a predikcii sa použili metriky odvodené z myši. Predikcia zmätenia má označenie "1".

7 Zhodnotenie

V tejto práci sme predstavili metódu predikcie zmätenia používateľa vo webovej aplikácii. Navrhli sme ako natrénovať predikčný model s využitím histogramovej reprezentácie črt správania myši, štatistického vyhodnotenia dát z okulografu a iných metrik opisujúcich správanie používateľa vo webovej aplikácii. Na zhotovenie modelu sme potrebovali zozbierať označený dataset s udalosťami, kedy reálne nastalo zmätenie používateľa. Pre tento prípad sme pripravili používateľskú štúdiu so šesťdesiatimi účastníkmi, počas ktorej účastníci plnili úlohy na dovolenkovom portáli. Celý priebeh pozostával zo šiestich úloh, kde sme zaznamenali pohľad a zároveň myš. Na zaznamenanie myši a klávesnice sme použili nami implementovanú infraštruktúru v rámci projektu *YesElf* firmy *Brainware*.

Na predikciu zmätenia sme porovnali klasifikátory logistickej regresie a náhodného lesu. Je zaujímavé, že logistická regresia preukázala lepšie výsledky pri predikcii aj napriek nelinearite riešeného problému. Ukázali sme, že použitie základných črt správania používateľa akými sú napríklad dĺžka sedenia, počet navštívených stránok a počet cyklov v sekvencii stránok v rámci sedenia, môže byť užitočné pri predikcii zmätenia nad celou úlohou. Takéto metriky vieme jednoducho zaznamenať pomocou štandardného HTTP serveru bez nutnosti rozšírenia stránky. Rovnaké zistenie bolo prezentované aj v práci [42], kde autori využili spomínané metriky na klasifikáciu obtiažnych sedení.

Úspešnosť nášho klasifikátora však porástla s využitím histogramovej reprezentácie črt myši. Pri predikcii na celej úlohe sme využili aj dáta z okulografu. Výsledky predikcie na metrikách z okulografu sa ukázali porovnateľné s výsledkami predikcie len na metrikách myši. Tento výsledok je veľmi cenný keďže naznačuje, že na predikciu zmätenia nám bohato postačujú metriky myši. V súčasnej dobe totiž bežný počítač nie je vybavený zariadením ako je okulograf.

Významnou časťou našej práce a príspevkom k poznaniu je predikcia zmätenia v reálnom čase s využitím dát z myši. Naším cieľom bolo vyvinúť klasifikátor predikcie v reálnom čase, ktorý by vedel odhaliť presný bod zmätenia. Ako sme spomínali pod pojmom presný bod rozumieme moment, kedy používateľ nevie pokračovať v napĺňaní svojich potrieb v aplikácii vo vlastnej réžii. Tradične je tento bod spojený s využitím nápovedy, zákazníckej podpory alebo externej služby (vyhľadávača). Ak by sme vedeli identifikovať zmätenie v reálnom čase, mohla by byť používateľovi poskytnutá efektívna pomoc, napr. vo forme sprievodcov (prostredníctvom aplikácie *YesElf*). V tomto prípade sme využili okná rovnakej dĺžky (5 a 10 sekúnd). Dôvodom voľby časových okien pred celou úlohou bolo produkčné prostredie, kde nemáme možnosť získať dáta z celej úlohy používateľa, kvôli nejasnosti v jej začiatku a konci. Taktiež v takomto prípade potrebujeme zmätenie predikovať čo najskôr po príchode používateľa do aplikácie a z toho dôvodu sa veľkosti 5 a 10 sekúnd javia ako ideálne. Aj v tomto prípade sme využili algoritmy logistickej regresie a náhodného lesu. Podstatne vyššiu presnosť pri predikcii zmätených sedení dosiahla predikcia s využitím logistickej regresie pri oknách veľkosti 10 sekúnd. Metrika presnosti predikcie nad zmätenými pozorovaniami je v prípade produkcie kľúčová, keďže sa chceme vyvarovať prílišnému poskytovaniu sprievodcov používateľom, ktorý ich reálne nepotrebujú. Takéto zobra-

zovanie môže viesť k frustrácii. Výsledky oboch častí overenia potvrdzujú našu hypotézu, že s využitím dát z myši sme schopný predikovať zmätenia používateľa.

V rámci overenia našej metódy sme spolupracovali s firmou *Brainware*, ktorá vyvíja produkt *YesElf*. Spolu s kolegom Matejom Váľkym sme implementovali logger udalostí myši (pozri prílohu C). Veľký dôraz sme kládli na kvalitu softvérovej realizácie logera tak, aby sa dal použiť v produkcii a tak umožnil získanie rozsiahlych dát z používania. Logger bol vyvinutý s dôrazom na jednoduché škálovanie v produkcii a je v súčasnosti využívaný firmou *Brainware* na zaznamenávanie dát o správaní používateľa. Autor tejto práce pre nástroj *YesElf* implementoval aj infraštruktúru na predikciu zmätenia používateľa v reálnom čase. Aj táto infraštruktúra je jednoducho škálovateľná. Umožní spolu s loggerom overenie výsledkov tejto práce na rozsiahlejších produkčných dátach.

V rámci práce sme napísali článok na univerzitnú konferenciu *IIT.SRC* (článok s plagátom v prílohe E). Okrem toho sme publikovali článok na medzinárodnej vedeckej konferencii *UMAP* (článok v prílohe F).

V budúcnosti plánujeme našu metódu overiť produkčne na systéme *Aegon*. V tomto smere už pripravujeme používateľskú štúdiu so systémom s cieľom natrénovať predikčný model. Natrénovaný model nasadíme priamo do systému *YesElf* určeného na vytváranie interaktívnych sprievodcov, ktorý je integrovaný s aplikáciou firmy *Aegon*. Vždy keď u používateľa predikujeme zmätenie, systém *YesElf* mu ponúkne sprievodcov. Detaily aplikácie *YesElf* a predikcie v reálnom čase uvádzame v prílohe C. Okrem testovania v produkcii sa otvára možnosť využitia dát z klávesnice. V našej práci sme len ponúkli prehľad extrakcie metrík z tohto zdroju. V metóde sme však tieto dáta nezahrnuli z dôvodu krátkosti času a taktiež z dôvodu výberu domény *FiroTour*, kde sa klávesnica zväčša nepoužíva. V týchto dátach sa však môžu nachádzať informácie indikujúce zmätenie používateľa.

Literatúra

- [1] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall a W Philip Kegelmeyer. “SMOTE: synthetic minority over-sampling technique”. In: *Journal of artificial intelligence research* Vol. 16 (2002), s. 321–357.
- [2] Ahmed Awad E Ahmed a Issa Traore. “A new biometric technology based on mouse dynamics”. In: *IEEE Transactions on dependable and secure computing* Vol. 4. No. 3 (2007), s. 165–179.
- [3] Hirotugu Akaike. “A new look at the statistical model identification”. In: *IEEE transactions on automatic control* Vol. 19. No. 6 (1974), s. 716–723.
- [4] Gerald Albaum. “The Likert scale revisited”. In: *Market Research Society. Journal.* Vol. 39. No. 2 (1997), s. 1–21.
- [5] Jackson Beatty. “Task-evoked pupillary responses, processing load, and the structure of processing resources.” In: *Psychological bulletin* Vol. 91. No. 2 (1982), s. 276.
- [6] Maria Bielikova, Martin Konopka, Jakub Simko, Robert Moro, Jozef Tvarozek, Patrik Hlavac a Eduard Kuric. “Eye-tracking en masse: Group user studies, lab infrastructure, and practices”. In: *Journal of Eye Movement Research* Vol. 11. No. 3 (2018), s. 6.
- [7] Cristina Conati, Enamul Hoque, Dereck Toker a Ben Steichen. “When to Adapt: Detecting User’s Confusion During Visualization Processing.” In: *UMAP Workshops*. Citeseer. 2013.
- [8] Laura Cowen, Linden Js Ball a Judy Delin. “An eye movement analysis of web page usability”. In: *People and Computers XVI-Memorable Yet Invisible*. Springer, 2002, s. 317–335.
- [9] Simon Doods, Toon De Pessemier a Luc Martens. “An online evaluation of explicit feedback mechanisms for recommender systems”. In: *7th International Conference on Web Information Systems and Technologies (WEBIST-2011)*. Ghent University, Department of Information technology. 2011, s. 391–394.
- [10] Andrew T Duchowski. “Eye tracking methodology”. In: *Theory and practice* Vol. 328 (2007).
- [11] Robin A Dubin. “Predicting house prices using multiple listings data”. In: *The Journal of Real Estate Finance and Economics* Vol. 17. No. 1 (1998), s. 35–59.
- [12] Claudia Ehmke a Stephanie Wilson. “Identifying web usability problems from eye-tracking data”. In: *Proceedings of the 21st British HCI Group Annual Conference on People and Computers: HCI... but not as we know it- Volume 1*. British Computer Society. 2007, s. 119–128.
- [13] Clint Feher, Yuval Elovici, Robert Moskovitch, Lior Rokach a Alon Schclar. “User identity verification via mouse dynamics”. In: *Information Sciences* Vol. 201 (2012), s. 19–36.
- [14] R Stockton Gaines, William Lisowski, S James Press a Norman Shapiro. *Authentication by keystroke timing: Some preliminary results*. Tech. spr. Rand Corp Santa Monica CA, 1980.
- [15] Hugo Gamboa a Ana Fred. “A behavioral biometric system based on human-computer interaction”. In: *Biometric Technology for Human Identification*. Zv. 5404. International Society for Optics a Photonics. 2004, s. 381–393.

- [16] Hugo Gamboa a Ana Fred. “A User Authentication Technic Using a Web Interaction Monitoring System”. In: *Iberian Conference on Pattern Recognition and Image Analysis*. Springer. 2003, s. 246–254.
- [17] Joseph H Goldberg a Xerxes P Kotval. “Computer interface evaluation using eye movements: methods and constructs”. In: *International Journal of Industrial Ergonomics* Vol. 24. No. 6 (1999), s. 631–645.
- [18] Mithat Gönen et al. “Receiver operating characteristic (ROC) curves”. In: *SAS Users Group International (SUGI)* Vol. 31 (2006), s. 210–231.
- [19] Shamsi T Iqbal, Piotr D Adamczyk, Xianjun Sam Zheng a Brian P Bailey. “Towards an index of opportunity: understanding changes in mental workload during task execution”. In: *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM. 2005, s. 311–320.
- [20] Gareth James, Daniela Witten, Trevor Hastie a Robert Tibshirani. *An introduction to statistical learning*. Zv. 112. Springer, 2013.
- [21] Gawesh Jawaheer, Martin Szomszor a Patty Kostkova. “Characterisation of explicit feedback in an online music recommendation service”. In: *Proceedings of the fourth ACM conference on Recommender systems*. ACM. 2010, s. 317–320.
- [22] Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke a Geri Gay. “Accurately interpreting clickthrough data as implicit feedback”. In: *ACM SIGIR Forum*. Zv. 51. 1. Acm. 2017, s. 4–11.
- [23] Tomas Juhaniak, Patrik Hlavac, Robert Moro, Jakub Simko a Maria Bielikova. “Pupillary Response: Removing Screen Luminosity Effects for Clearer Implicit Feedback.” In: *UMAP (Extended Proceedings)*. 2016.
- [24] Samad Kardan a Cristina Conati. “Comparing and combining eye gaze and interface actions for determining user learning with an interactive simulation”. In: *International Conference on User Modeling, Adaptation, and Personalization*. Springer. 2013, s. 215–227.
- [25] Samad Kardan a Cristina Conati. “Exploring gaze data for determining user learning with an interactive simulation”. In: *International Conference on User Modeling, Adaptation, and Personalization*. Springer. 2012, s. 126–138.
- [26] Sotiris B Kotsiantis, I Zaharakis a P Pintelas. “Supervised machine learning: A review of classification techniques”. In: *Emerging artificial intelligence applications in computer engineering* Vol. 160 (2007), s. 3–24.
- [27] Peter Krátky. “Biometric User Model for Recognition on the Web”. In: *Information Sciences and Technologies* Vol. 9. No. 2 (2017), s. 19–24.
- [28] Sébastien Lallé, Cristina Conati a Giuseppe Carenini. “Predicting Confusion in Information Visualization from Eye Tracking and Interaction Data.” In: *IJCAI*. 2016, s. 2529–2535.
- [29] Sébastien Lallé, Dereck Toker, Cristina Conati a Giuseppe Carenini. “Prediction of users’ learning curves for adaptation while using an information visualization”. In: *Proceedings of the 20th International Conference on Intelligent User Interfaces*. ACM. 2015, s. 357–368.

- [30] Xu-Ying Liu, Jianxin Wu a Zhi-Hua Zhou. “Exploratory undersampling for class-imbalance learning”. In: *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* Vol. 39. No. 2 (2009), s. 539–550.
- [31] Nizar R Mabroukeh a Christie I Ezeife. “A taxonomy of sequential pattern mining algorithms”. In: *ACM Computing Surveys (CSUR)* Vol. 43. No. 1 (2010), s. 3.
- [32] Florent Masseglia, Pascal Poncelet a Maguelonne Tisseire. “Using data mining techniques on web access logs to dynamically improve hypertext structure”. In: *ACM sigweb Newsletter* Vol. 8. No. 3 (1999), s. 13–19.
- [33] Roy A Macion et al. “Dependability at the user interface”. In: *Twenty-Fifth International Symposium on Fault-Tolerant Computing. Digest of Papers*. IEEE. 1995, s. 528–535.
- [34] John Neter, Michael H Kutner, Christopher J Nachtsheim a William Wasserman. *Applied linear statistical models*. Zv. 4. Irwin Chicago, 1996.
- [35] Avar Pentel. “Employing think-aloud protocol to connect user emotions and mouse movements”. In: *2015 6th International Conference on Information, Intelligence, Systems and Applications (IISA)*. IEEE. 2015, s. 1–5.
- [36] Avar Pentel. “Patterns of Confusion: Using Mouse Logs to Predict User’s Emotional State.” In: *UMAP Workshops*. 2015.
- [37] Alex Poole a Linden J Ball. “Eye tracking in HCI and usability research”. In: *Encyclopedia of human computer interaction* Vol. 1 (2006), s. 211–219.
- [38] Robert W Reeder a Roy A Macion. “User interface defect detection by hesitation analysis”. In: *International Conference on Dependable Systems and Networks (DSN’06)*. IEEE. 2006, s. 61–72.
- [39] George AF Seber a Alan J Lee. *Linear regression analysis*. Zv. 329. John Wiley & Sons, 2012.
- [40] Marina Sokolova a Guy Lapalme. “A systematic analysis of performance measures for classification tasks”. In: *Information Processing & Management* Vol. 45. No. 4 (2009), s. 427–437.
- [41] Tobii Studio. “1.2 User Manual”. In: *Tobii Technology AB* (2008).
- [42] Paul Thomas. “Using interaction data to explain difficulty navigating online”. In: *ACM Transactions on the Web (TWEB)* Vol. 8. No. 4 (2014), s. 24.
- [43] Ryen W White, Joemon M Jose a Ian Ruthven. “Comparing explicit and implicit feedback techniques for web retrieval: Trec-10 interactive track report”. In: *Proceedings of the Tenth Text Retrieval Conference (TREC-10)*. 2002, s. 534–538.
- [44] Cha Zhang a Yunqian Ma. *Ensemble machine learning: methods and applications*. Springer, 2012.

Príloha A - Protokol pilotnej používateľskej štúdie

Protokol experimentu

Kontext experimentu

Meno a priezvisko Michal Hucko, Matej Valky

Vedúci prof. Ing. Mária Bieliková, PhD.

Názov projektu Identifikácia zmätenia používateľa vo webovej aplikácii
Skúmanie interakcie človeka s aplikáciou obohatenou automatickým poradcom

Názov projektu v AJ:

Identification of user confusion in a web application
Exploration of human-application interaction enriched by an automatic guide

Kľúčové slová Zmätenie, sledovanie pohľadu, udalosti myši

Stručný opis projektu

Zmätenie používateľa je problém spojený s webovou aplikáciou. V niektorých prípadoch aplikácia nemusí byť ani komplexná na to aby bola pre používateľa mäťúca. Príkladom môžu byť napríklad používatelia, navštevujúci aplikáciu po prvý raz. Schopnosť identifikácie stavu zmätenia je v tomto prípade kľúčová. Po jej odhalení vieme používateľovi poskytnúť nápovedy, ktoré mu môžu pomôcť k naplneniu cieľu v aplikácii. Takéto nápovedy vieme konštruovať v špecializovaných nástrojoch (angl. web guides).

V našej práci sa zaoberáme identifikáciou a predikciou zmätenia. Pracujeme na metóde, ktorá bude využívať dáta z udalostí myši zaznamenané interakciou používateľa a webovej aplikácie. Konkrétnejšie sa zameriavame na pohyby, kliky, skrolovanie. V rámci projektu zvažujeme aj použitie okulografu na overenie našej metódy.

Stručný opis projektu v AJ

Confusion on the web is a problem connected with web applications. Sometimes apps do not have to be complex to confuse users. Common problems may have newcomers, when accessing the app for the very first time. Ability to identify confusion can help us to give hints to users. These hints can help user to fulfil their needs. Hints can be combined to web application guides.

In our work we identify and predict confusion on web application. We will make a method for this purpose, using logs from the server. Except them we concentrate on user mouse movements, clicks, scrolling and browsing patterns which describe behaviour. We consider usage of eye-tracker data. Next we apply machine learning methods on gathered data. To confirm our method we will establish an experiment, where we will collect feedback from users. At the moment we are working on selection of appropriate way to collect these feedback. These will be used as labeled data. Our solution can be combined with production systems providing web application guides.

Príprava experimentu

Cieľ experimentu

Cieľom experimentu je oboznámiť sa so správaním používateľa pri interakcii s webovou aplikáciou. V tomto smere sa zameriavame na jeho pohyby očí a udalosti myši. V rámci sledovania myši je cieľom

taktiež otestovať infraštruktúru určenú na zaznamenávanie a zber týchto dát. Okrem toho chceme pozorovať správanie, ktoré môže nastať pri zmätení používateľa.

Hypotézy

1. Pri interakcii užívateľov s webovou aplikáciou je možné identifikovať vzory v správaní, na základe metrik odvodených zo sledovania pohľadu a zaznamenávania udalostí myši.
2. V správaní používateľa webovej aplikácie je možné identifikovať stav zmätenia pomocou okulografu a loggeru myši.

Účastníci

5 dobrovoľníkov v rámci stretnutia Ontožúr. Pôjde buď o študentov, alebo vyučujúcich na FIIT STU.

Sledované metriky - okulograf

V prípade dát zo sledovania pohľadu sa sústreďíme na metriky odvodené z fixácií a sakád:

- atribúty odvodené od fixácií (dĺžka trvania fixácie, čas do prvej fixácie, počet fixácií)
- atribúty odvodené od sakád (relatívny/absolútny uhol medzi sakádami, dĺžka sakády).
- atribúty odvodené od veľkosti zreničky (šírka zreničky, rýchlosť dilatácie zreničky)
- atribúty odvodené od vzdialenosti hlavy od monitoru (vzdialenosť hlavy od monitoru - na začiatku a konci úlohy)

Sledované metriky - myš

V prípade dát z myši sa sústreďíme na metriky:

- Vzdialenosť a uhol medzi dvoma bodmi
- Pohyb myši zakončený kliknutím, alebo pauzou (štatistika pre jednotlivé pozorovania)
- Rýchlosť, zrýchlenie, zakrivenie pre jednotlivé pozorovanie-
- Vzdialenosť, trvanie, smer pohybu myši.

Scenár experimentu

Prvá fáza

Na začiatku experimentu je účastníkovi vysvetlený priebeh experimentu. Je informovaný o tom, že všetko čo spraví bude nahrávané a je taktiež vyžiadaný jeho súhlas so spracovaním týchto údajov.

Kalibrácia

Účastník je informovaný o tom ako kalibrácia prebieha, upozornený, aby po kalibrácii už nemenil svoju polohu. Sú mu vysvetlené základné inštrukcie týkajúce sa stránky, ktorá sa mu zobrazí, ako označiť úlohu za splnenú, ako spustiť nasledujúcu úlohu a že na vyriešenie úloh nie je potrebné ani žiadané aby bol prihlásený. Účastník je tiež požiadaný, aby sa počas vykonávania úloh na nič nepýtal a vykonával úlohy podľa vlastného uváženia. Vykoná sa kalibrácia a zapne sa inštrukcia k prvej úlohe.

Úlohy

Účastník rieši všetky úlohy v rámci Tobii Studia. Pred vykonaním každej úlohy sa mu na jej začiatku zobrazia pokyny, ktorých potvrdením prejde k vykonávaniu danej úlohy. Účastník rieši úlohu až kým ju považuje za ukončenú.

Účet na alza.sk:

- Meno: fakefake@centrum.sk

- Heslo: fakefake

Jednotlivé úlohy:

1)

(<https://www.alza.sk/sony-cybershot-dsc-w800?dq=601733>)

Začiatok: Stránka s detailom konkrétneho fotoaparátu

Úloha: Nachádzate sa na podstránke s detailom konkrétneho fotoaparátu. Zistíte do kedy by bol produkt doručený na predajňu Bratislava - Petržalka, keby si ho kúpíte teraz. Pre začatie úlohy stlačte F10. V prípade, že ste úlohu splnili alebo ju neviete vyriešiť, stlačte F10 pre pokračovanie.

Produkt bude

doručený:.....

2)

Začiatok: hlavná stránka

Úloha: Nachádzate sa na hlavnej stránke portálu Alza. Nájdite najlacnejšiu ponuku rozbalených produktov (produkty vo výpredaji). Pre začatie úlohy stlačte F10. V prípade, že ste úlohu splnili alebo ju neviete vyriešiť, stlačte F10 pre pokračovanie.

Názov najlacnejšej ponuky

je:.....

3)

Začiatok: Stránka so zoznamom fotoaparátov (<https://www.alza.sk/digitalne-foto/18843129.htm>)

Úloha: Nájdite fotoaparát nad 200 eur s najnižšou cenou. Pre začatie úlohy stlačte F10. V prípade, že ste úlohu splnili alebo ju neviete vyriešiť, stlačte F10 pre pokračovanie.

Názov fotoaparátu

je:.....

4)

Začiatok: Stránka s detailom konkrétneho fotoaparátu

(<https://www.alza.sk/sony-cybershot-dsc-w800?dq=601733>)

Úloha: Nachádzate sa na podstránke s detailom konkrétneho fotoaparátu. Zistíte či daný fotoaparát poskytuje funkciu detekcie tváre. Pre začatie úlohy stlačte F10. V prípade, že ste úlohu splnili alebo ju neviete vyriešiť, stlačte F10 pre pokračovanie.

Fotoaparát funkciu podporuje: áno / nie / neviem

5)

Začiatok: Úvodná stránka (<https://www.alza.sk>)

Úloha: Nachádzate sa na stránke alza. Zistíte akú zľavu si viete uplatniť pri 10 000 alzákoch. Pre začatie úlohy stlačte F10. V prípade, že ste úlohu splnili alebo ju neviete vyriešiť, stlačte F10 pre pokračovanie.

Príloha B - Protokol používateľskej štúdie

Protokol experimentu

Kontext experimentu

Meno a priezvisko Michal Hucko

Superviso: prof. Ing. Mária Bieliková, PhD.

Názov projektu: Identifikácia zmätenia používateľa vo webovej aplikácii

Názov projektu v AJ: Identification of user confusion in a web application

Kľúčové slová: Zmätenie, eyetracking, udalosti myši

Stručný opis projektu

Zmätenie používateľa je problém spojený s webovou aplikáciou. V niektorých prípadoch aplikácia nemusí byť ani komplexná na to, aby bola pre používateľa mätúca. Príkladom môžu byť napríklad používatelia, navštevujúci aplikáciu po prvý raz. Schopnosť identifikácie stavu zmätenia je v tomto prípade kľúčová. Po jej odhalení, vieme používateľovi poskytnúť nápovedy, ktoré mu môžu pomôcť k naplneniu cieľa v aplikácii. Takéto nápovedy vieme konštruovať v špecializovaných nástrojoch (angl. web guides).

V našej práci sa zaoberáme identifikáciou a predikciou zmätenia. Pracujeme na metóde, ktorá bude využívať dáta z udalostí myši zaznamenané interakciou používateľa a webovej aplikácie. Konkrétnejšie sa zameriavame na pohyby, kliky, skrolovanie. V rámci projektu zvažujeme aj použitie eyetrackeru na overenie našej metódy.

Stručný opis projektu v AJ

Confusion on the web is a problem connected with web applications. Sometimes apps do not have to be complex to confuse users. Common problems may have newcomers, when accessing the app for the very first time. Ability to identify confusion can help us to give hints to users. These hints can help user to fulfil their needs. Hints can be combined to web application guides.

In our work we identify and predict confusion on web application. We will make a method for this purpose, using logs from the server. Except them we concentrate on user mouse movements, clicks, scrolling and browsing patterns which describe behaviour. We consider usage of eye-tracker data. Next we apply machine learning methods on gathered data. To confirm our method we will establish an experiment, where we will collect feedback from users. At the moment we are working on selection of appropriate way to collect these feedback. These will be used as labeled data. Our solution can be combined with production systems providing web application guides.

Príprava experimentu

Cieľ experimentu

Cieľom je zozbierať dataset správania používateľa vo webovej aplikácii s označeným stavom zmätenia. Ďalší cieľ experimentu je získať dataset správania používateľa vo webovej aplikácii pri interakcii so sprievodcom. Dataset má obsahovať údaje o pohybe očí a pohybe myši. Na zbieranie pohybov očí použijeme eye-tracker a na myš vlastné logovacie zariadenie umiestnenie priamo vo webovej aplikácii. Pred experimentom a počas experimentu položíme účastníkovi dotazníky a testy, ktorým si priblížime osobnú charakteristiku,.

Hypotézy

1. Pri interakcii používateľov s webovou aplikáciou je možné identifikovať vzory v správaní, na základe metrik odvodených zo sledovania pohľadu a zaznamenávania udalostí myši.
2. V správaní používateľa webovej aplikácie je možné identifikovať stav zmätenia pomocou okulografu a loggera myši.
3. Každý používateľ so sprievodcom interaguje iným spôsobom a tieto rozdiely súvisia s charakteristikou používateľa.

Parametre experimentu

A/B kvalitatívny experiment, A skupina - bez sprievodcov, B skupina - so sprievodcami. Skupiny by mali byť rozdelené v pomere 1:1.

Účastníci

Študenti 3. ročníka bakalárskeho štúdia STU FIIT z predmetu Interakcia človeka s počítačom.

Sledované metriky - okulograf

V prípade dát zo sledovania pohľadu sa sústreďíme na metriky odvodené z fixácií a sakád:

- Atribúty odvodené od fixácií (dĺžka trvania fixácie, čas do prvej fixácie, počet fixácií).
- Atribúty odvodené od sakád (relatívny/absolútny uhol medzi sakádami, dĺžka sakády).
- Atribúty odvodené od veľkosti zreničky (šírka zreničky, rýchlosť dilatácie zreničky).
- Atribúty odvodené od vzdialenosti hlavy od monitora (vzdialenosť hlavy od monitora - na začiatku a konci úlohy).

Sledované metriky - myš

V prípade dát z myši sa sústreďíme na metriky:

- Vzdialenosť a uhol medzi dvoma bodmi.
- Pohyb myši zakončený kliknutím alebo pauzou (štatistika pre jednotlivé pozorovania).
- Rýchlosť, zrýchlenie, zakrivenie pre jednotlivé pozorovanie.
- Vzdialenosť, trvanie, smer pohybu myši.

Sledované metriky - afekt¹

- Radosť.
- Hnev.
- Prekvapenie..
- Strach.
- Opovrhovanie.
- Smútok.

¹ https://imotions.com/wp-content/uploads/Guides/iMotions_Guide_FacialExpressions_2016.pdf

- Znechutenie.

Osobnostné dotazníky/osobnostné testy charakteristiky používateľa

- **Na doma** (realizované ako moduly v crowdex.fiit.stuba.sk)
 - Odkaz (treba byť prihlásený a mať práva)
https://crowdex.fiit.stuba.sk/experiment_management/82-osobnostny-dotaznik-pre-experiment-huckovalky
 - **Demografia**
 - Váš vek
 - Vaše povolanie (ak ste študentom, napíšte študijný odbor)
 - Pohlavie
 - **Big five dotazník**
BFI-2, 44 položková verzia 5 Big-Five of Personality dimenzií
<https://www.ncbi.nlm.nih.gov/pubmed/27055049>
 - **Výroky o osobnosti**
 - Vyroky 1 (B cast excelu)
 - Vyroky 2 (C cast excelu)
 - <https://docs.google.com/spreadsheets/d/1nlxzT5RkPuV-R0Aa17afAoT1y5LBx3V1QsqPhCOfZzg/edit#gid=1021683669>
- **Na mieste**
 - **Situačný dotazník**
https://docs.google.com/document/d/1W9T86NvJnF1qsJ9P9yp1sNI9FV_vgBZfzRcevJgTPJo/edit?usp=sharing
 - **Curiosity and Exploration Inventory (CEI - II)**
<https://drive.google.com/file/d/1cgx2DDYiZfF5AM1W-wfMovsJ8YA2mveq/view?usp=sharing>
 - **Testy**
 - <http://www.cs.ubc.ca/cs-research/lci/research-groups/intelligent-user-interfaces/userchar.html>
 - **Perceptual Speed**
<https://www.cs.ubc.ca/~dtoker/phd/Perceptual%20Speed%20Test%20-%20P-3.pdf>
 - **Visual Working Memory**
MATLAB test
 - **Dotazník spätnej väzby po každej úlohe:**
 - Odpoveď na úlohu: <pole pre otvorenú odpoveď>.
 - Stlačil/a som tlačidlo {áno / nie single choice}.
 - Prosím opíšte priebeh riešenia úlohy a uveďte čo sa vám zdalo na úlohe ťažké. Ak ste stlačili tlačidlo, opíšte dôvod stlačenia. <pole pre otvorenú odpoveď>.

Scenár experimentu

Harmonogram

Krok	Odhadovaný čas (minúty)
Uvítanie účastníkov a ich usadenie, úvod prezentácie	5
Podpísanie informovaného súhlasu	2
Situačný dotazník	3
Test 1: Perceptual Speed test	1.5
Test 2: Visual Working memory	6
Pokr. prezentácie: Vysvetlenie úloh a tlačidla	5
Kalibrácia	5
Hlavné úlohy (inštrukcie + 6 úloh + dot. spätnej väzby)	40

1 Dotazníky na doma pred štúdiou

Akcia: pridajte účastníka na dohodnutý beh do Google hárku:

https://docs.google.com/spreadsheets/d/1ymudwrM72Dz_xVgtSZPVG24YOEDhOFWwWgMMo3ZEEUc/edit?ts=5bd04f13#gid=0

- Poznač skupinu (A - bez sprievodcu, B - so sprievodcom)
- Vyber po rade PID z vygenerovaných PID a prirad' účastníkovi
 - <https://gist.github.com/enderstorm/bd28dff154cf4dab9408c6d19b2e98fe>

Akcia: pošli každému účastníkovi jednotlivo pár dní pred experimentom e-mail s nasledovným telom:

Mail účastníkom

Dobrý deň {MENO},

Ďakujeme za ochotu sa zúčastniť štúdie.

Pripomíname, že štúdia sa bude konať {DATUM_BEH} v UX Class (1.26) na FIIT STU.

Pred štúdiou, prosím, vyplňte nasledovný dotazník na odkaze:

https://crowdex.fiit.stuba.sk/ej/hm_oSnMQ2w?t={PID}

Je potrebné zodpovedať všetky časti a otázky dotazníka.

Váš identifikátor účastníka (participant ID, skr. PID) je: **{PID}**

Ďalej prikladáme odkaz na formulár s informovaným súhlasom, kde sa môžete dozvedieť viac o štúdiu a sú tam aj informácie o spracúvaní dát v súlade s platnou legislatívou. Je to štandardná súčasť každej štúdie.

<https://docs.google.com/document/d/1QRpLocZ09DxFGTJH0aG5gcOaha52bAPXGXZo2C9NqpI>

Vyplnenie uvedeného dotazníka je nutná podmienka pre zúčastnenie sa štúdie.

Ďakujeme.

S pozdravom,
Matej Války a Michal Hucko

Akcia: deň pred behom štúdie skontroluj vyplnenosť testov na doma

- Získaj z crowdexu PID, ktoré úspešne vyplnili dotazník (kontaktuj zodpovednú osobu)
- Pošli upomienku účastníkom, ktorí nevyplnili test s motiváciou, že test netrvá dlho

2 Úvodné kroky

Akcia: Vytvor iMotions projekt a pre každý počítač v UX Classe:

- Nakopíruj "ValkyHucko_study.zip".
- Nakopíruj Matlab test.

Akcia: Pred behom aktualizuj IP mapovanie o PID účastníkov.

- Pre každé PID v Participant List priradiť Computer Name.
- Computer Name je v tvare "uxlab-XX" kde XX je číslo 01 do 20 a toto číslo zodpovedá koncu IP (bez nuly).
- Na redundantné počítače vyber záložné PID z Gist zoznamu.
- Vytvor mapovacie CSV v tvare ip,participantid.
- Nahraj mapovacie CSV na.
 - <https://bridge.buddyweb.fr/en/dashboard/apis/628/tables/606/edit>
 - Over nahranosť API volaním v prehliadač.

Akcia: Vytlač a priprav si dokumenty pre každého účastníka daného behu. Do pravého horného rohu každého listu papiera poznač ID účastníka.

Akcia A: Na každom počítači otvor iMotions a projekt "ValkyHucko study". Skontroluj, či je zapnutý eye-tracker a či je korektne zdetekovaný v iMotions.

Akcia B: Na každom počítači otvor iMotions a projekt "ValkyHucko study **Guides**". Skontroluj, či je zapnutý eye-tracker a či je korektne zdetekovaný v iMotions.

Akcia: Otvor Matlab. V Matlabe prepni aktuálny adresár na nasledovný:

"C:\Users\ux\Documents\valky_hucko\FVMatlab"

Akcia: Nastav konštantné svetelné podmienky v laboratóriu.

3 Usadenie účastníkov

Akcia: Účastníkov posad' podľa ich PID - Computer Name páru.

- Otvor si Participant List.
- Vyvolaj Meno Priezvisko.
- Posad' na miesto.

4 Privítanie účastníkov:

4.1 Úvodné slovo a prezentácia:

Vítajte na našej štúdií. V prvom rade by sme vám chceli poďakovať za vašu účasť. Celá štúdia bude prebiehať asi 60 minút. Počas celej štúdie budeme zaznamenávať váš pohľad pomocou sledovača pohľadu a taktiež vaše pohyby myšou a interakciu s klávesnicou. Celá štúdia bude pozostávať z troch častí. V prvej časti vás prosíme o vyplnenie krátkeho dotazníku. V prvej časti vás prosíme o vyplnenie informovaného súhlasu a vykonáme zopár testov. Druhá časť bude kalibrácia zariadení na snímanie pohľadu. Posledná tretia časť bude pozostávať z riešenia úloh na portáli Firotour. **Prosíme si dať vaše telefóny do režimu lietadlo, aj vibrácia naruší výsledky experimentu.**

Ak máte akékoľvek otázky môžete ich položiť práve teraz.

4.2 Odkaz na prezentáciu

https://docs.google.com/presentation/d/1OO9rixPBiaKNKKbRWYIM7PvOe_sITukxEkcRysF-Bmk/edit?ts=5bd585b0#slide=id.g4431ee701a_0_54

5 Testy pred úlohami

5.1 Visual working memory test (VisWM):

Povedzte: "Teraz vykonáme krátky test na počítači, ktorý bude merať vašu vizuálnu pozornosť."

Akcia: Povedzte participantom, aby sa prepli do Matlabu a spustili "WMTaskPractice" (pravý klik, "Run"). Povedzte participantom, že ide len o skúšobné kolo. Povedzte im, že pre tento test znamená ľavá šípka na klávesnici „áno“ a pravá „nie“.

Akcia: Povedzte participantom, aby spustili "WMTask" (pravý klik, "Run"). Mali by zadať nasledovné číslo participanta: "PID".

Povedzte participantom: "Test by mal trvať len okolo 4 minút, a hoci sa vám môže zdať, že to trvá celú večnosť, nebojte sa, má koniec! Veľa šťastia!"

6 Hlavné úlohy

6.2 Úvodné slovo: Pred samotnou kalibráciou zariadení si prejdeme postup pri riešení úloh.

6.3 Popis: Každá úloha bude prebiehať v troch častiach. V prvej sa vám zobrazí okno so znením úlohy. Prosíme vás o to aby ste si inštrukcie dôkladne prečítali. Každé znenie bude označené názvom. Tento názov slúži na to aby ste si inštrukcie vedeli nájsť na priloženom papieri v prípade, že by ste ich zabudli. Po prečítaní úlohy stlačte tlačidlo F10.

6.4 Úloha: V druhej časti sa vám zobrazí okno prehliadača s dovolenkovým portálom firetour a príslušnou podstránkou. Vašou úlohou je vždy nájsť potrebnú informáciu z popisu v prvej časti. Po nájdení potrebnej informácie stlačte tlačidlo F10

6.4.1 Tlačidlo “!”: V pravej hornej časti obrazovky prehliadača uvidíte tlačidlo s textom “!”.

- Prosíme vás o to, aby ste na toto tlačidlo klikli vždy vtedy keď budete potrebovať pomoc s úlohou.
- Tlačidlo má simulovať stav, kedy by ste chceli stránku opustiť z dôvodu, že by ste nevedeli kde hľadať potrebnú informáciu na splnenie úlohy.
- Taktiež je na toto tlačidlo vhodné kliknúť vtedy ak by ste chceli na splnenie úlohy použiť inú aplikáciu ako <domena> (napríklad google). Po stlačení tlačidla nenastane žiadna udalosť zo strany systému. Následne môžete v úlohe pokračovať, alebo prejsť ďalej tlačidlom F10:

Príkladný scenár 1: som na stránke alzy a chcem vedieť koľko stojí najlacnejší monitor s uhlopriečkou 24 palcov. Viem ako sa z úvodnej stránky dostať do elektroniky. Prejdem teda do elektroniky. Následne viem ako vyhľadať monitory. Vyhľadám teda monitory. Zrazu však neviem ako mám vyhľadať monitory podľa uhlopriečky a tak kliknem na tlačidlo potrebujem pomoc chvíľu počkám a úlohu ukončím.

6.5 Formulár: Po stlačení tlačidla F10 sa vám zobrazí formulár pre zadanie získanej informácie. Do poľa pre text prosím zadajte informáciu, ktorú ste získali. Neexistuje nesprávna odpoveď ani nesprávne správanie. Ak odpoveď neviete nechajte prosím pole nevyplnené. Niektoré úlohy majú dokonca viacero správnych odpovedí. Vo formulári budú aj otázky ohľadom tlačidla “potrebujem pomoc” prosíme vás o ich vyplnenie. Taktiež tam bude otvorená otázka, kde môžete opísať priebeh úlohy. Po vyplnení formuláru pre pokračovanie na ďalšiu úlohu stlačte tlačidlo F10.

6.6 Záverečné slovo: Pred samotným riešením úloh bude nasledovať kalibrácia zariadení na sledovanie pohľadu.

7 Kalibrácia pohľadu

Softvér iMotions poskytuje kalibráciu zraku a zreničiek. V tejto fáze je zaznamenaný pohľad účastníka, čo sa môže využiť pri validácii kalibrácii.

Poznámka: Ak sa vyskytne problém s kalibráciou, je potrebné posadiť účastníka za redundantný počítač a POZNAČIŤ zmenu PID. Táto zmena PID sa bude týkať IBA nahraných údajov z yeself-analytics a yeself-logger indexov.

7.1 Úvodné slovo: Prejdem do kroku kalibrácie zariadení na sledovanie pohľadu. Prosím zatvorte okno prehliadača s vyplnením dotazníkom. Po zatvorení uvidíte okno softvéru na nastavenie

zariadenia. V dolnej časti okna vidíte čierny štvorec s červenými okrajmi. Zobrazuje umiestnenie vašich očí pomocou dvoch bielych bodiek. Prosím nastavte sklon monitoru tak, aby vaše oči boli v strede zvýrazneného poľa. Nastavte polohu monitora tak aby vás neobmedzovala pri sedení. Ak potrebujete pomoc neváhajte nás osloviť.

7.2 Priebeh kalibrácie: Ak sú vaše oči v strede poľa. Prejdeme na kalibráciu. Po jej spustení bude monitor meniť farby. Uvidíte striedanie svetlých a tmavých farieb. Po tomto kroku sa na obrazovke budú striedať biele bodky v rôznych miestach. Prosím upriamte na ne zrak, prípadne ich nasledujte po dráhe. Po dokončení tohto kroku nás prosím zavolajte, aby sme skontrolovali kvalitu nastavenia. Po úspešnej kalibrácii prejdete na riešenie úloh.

8 Znenie úloh - Firotour

8.1 Nastavenie úloh v iMotions

Každá úloha je reprezentovaná v iMotions ako blok so stimulmi (fixované poradie):

- 1. Textový stimul: Inštrukcie ku splneniu úlohy (časť Úloha).
- 2. Webový stimul s URL.
- 3. Dotazníky spätnej väzby:
 - Otázka na výsledok úlohy.
 - Otázka na vnímanie sprievodcov .
 - Likertova stupnica: Ako vám bol sprievodca nápomocný pri plnení úlohy?
 - Likertova stupnica:

Poznámka: Poradie úloh je randomizované pomocou softvéru iMotions

1)

(<https://www.firotour.sk/>)

Začiatok: Úvodná stránka Firotour

Úloha: Nachádzate sa na úvodnej stránke portálu Firotour. Zistíte či sa na portáli Firotour dá platiť kartou. Pre začatie úlohy stlačte F10. V prípade, že ste úlohu splnili, alebo ju neviete vyriešiť, stlačte F10 pre pokračovanie.

2)

(<https://www.firotour.sk/>)

Začiatok: Úvodná stránka Firotour

Úloha: Nachádzate sa na úvodnej stránke portálu Firotour. Zistíte cenu a názov ponuky, ktorá je v rámci programu Firáčik najlacnejšia. Pre začatie úlohy stlačte F10. V prípade, že ste úlohu splnili alebo ju neviete vyriešiť stlačte F10 pre pokračovanie.

3)

(<https://www.firotour.sk/>)

Začiatok: Úvodná stránka Firotour

Úloha: Nachádzate sa na úvodnej stránke portálu Firotour. Zistíte názov a autora najnovšieho príspevku v cestovateľskom blogu. Pre začatie úlohy stlačte F10. V prípade, že ste úlohu splnili alebo ju neviete vyriešiť, stlačte F10 pre pokračovanie.

4)

(<https://www.firotour.sk/>)

Začiatok: Úvodná stránka Firotour

Úloha: Nachádzate sa na úvodnej stránke portálu F1rotour. Zistite cenu najlacnejšej ponuky do turecka s odletom z Viedne. Pre začatie úlohy stlačte F10. V prípade, že ste úlohu splnili alebo ju nevieťe vyriešiť, stlačte F10 pre pokračovanie.

5)

(<https://www.f1rotour.sk/>)

Začiatok: Úvodná stránka F1rotour

Úloha: Nachádzate sa na úvodnej stránke portálu F1rotour. Zistite ako ďaleko je Port Ghalib Resort od pláže . Pre začatie úlohy stlačte F10. V prípade, že ste úlohu splnili alebo ju nevieťe vyriešiť, stlačte F10 pre pokračovanie.

6)

(<https://www.f1rotour.sk/>)

Začiatok: Úvodná stránka F1rotour

Úloha: Nachádzate sa na úvodnej stránke portálu F1rotour. Zistite, v ktorom roku bola založená cestovná kancelária F1rotour. Pre začatie úlohy stlačte F10. V prípade, že ste úlohu splnili alebo ju nevieťe vyriešiť, stlačte F10 pre pokračovanie.

9 Koniec behu experimentu

Akcie:

- Debriefing
- Aktualizuj Google Sheet "Participant list - ValkyHucko"

10 Po všetkých behoch

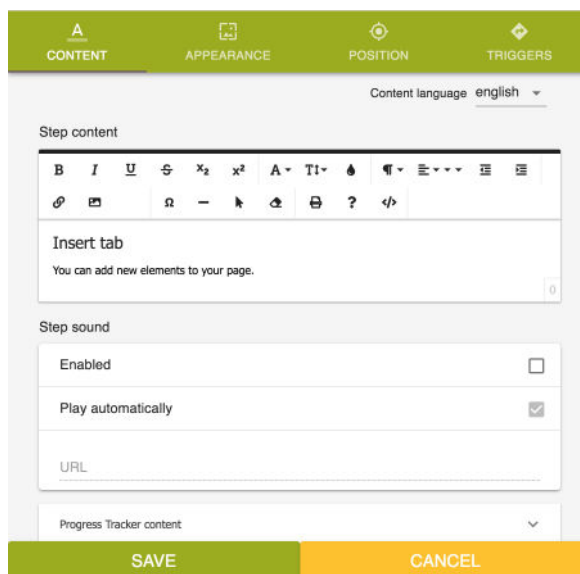
Akcia:

- Zozbieraj údaje z každého počítača
 - Matlabový dotazník v priečinku.
- Získaj odpovede z Big Five dotazníka (Crowdex).
- Zálohuj získaný dataset podľa inštrukcií na PEWE.
- Vypracuj priebeh experimentu, viď. spodok tohto dokumentu.

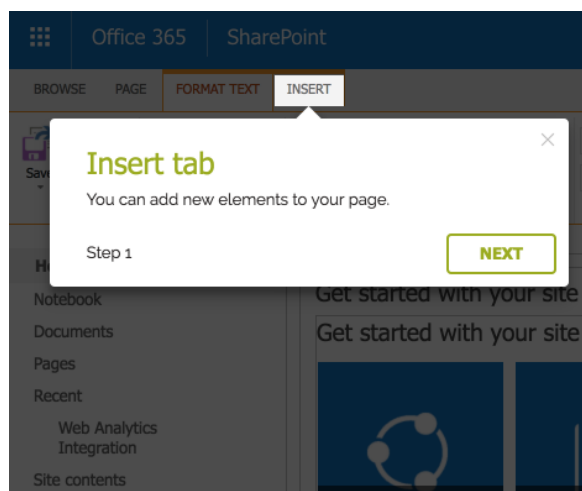
Príloha C - Logger myši a predikcia zmätenia v aplikácii YesElf

YesElf je webová aplikácia určená na vytváranie sprievodcov pre klientské aplikácie. Jej integrácia s klientskou aplikáciou je založená na integrovaní kusu kódu so zdrojovým kódom klientskej aplikácie. *YesElf* potom poskytuje jednoduchý editor, ktorý umožňuje vytváranie sprievodcov. Sprievodca je tvorený krokmi. Každý krok je možné naviazať na *HTML* element v klientskej aplikácii priamo v editore (pozri ukážku editoru na obrázku 11). Naviazanie je realizované pomocou 4 selektorov, čím sa predchádza strate naviazania v prípade modifikácie cieľovej aplikácie.

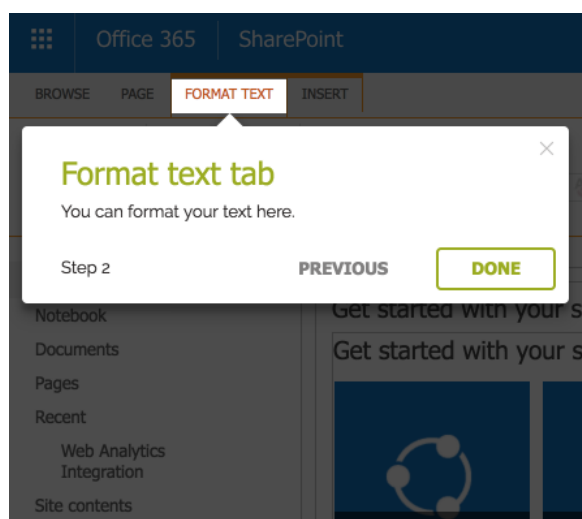
Pre každý krok je možné definovať spúšťač, ktorý je zodpovedný za zobrazenie daného kroku. Príkladom môže byť časový spúšťač, ktorý spustí krok sprievodcu po určitej dobe na stránke, alebo udalostný spúšťač, ktorý zobrazí krok po prechode kurzoru nad elementom kroku. *YesElf* však neposkytuje žiadnu možnosť personalizácie sprievodcov. Odhliadnuc od pravidiel sa kroky zobrazujú všetkým používateľom webovej aplikácie. Ukážku krokov sprievodcov ponúkame na obrázku 12 a 13. V prílohe F ponúkame prijatý článok na konferenciu *UMAP*, kde poskytujeme detailnejší pohľad na aplikáciu *YesElf* a logger.



Obr. 11: Ukážka editoru aplikácie YesElf.



Obr. 12: Ukážka jednoduchého kroku sprievodcu aplikácie *YesElf*, ktorý je pripevnený na *HTML* element.



Obr. 13: Ukážka nasledujúceho kroku, ktorý sa zobrazí po kliknutí na tlačidlo "Next" na obrázku 12.

Logger

Na logovanie udalostí myši používame vlastnú implementáciu logera. Pod pojmom logger budeme označovať softvér zodpovedný za zaznamenávanie dát o interakcii používateľa s webovou aplikáciou. Celá implementácia je výsledkom spolupráce medzi našou fakultou a firmou *Brainware* a vznikla nad rámec tejto diplomovej práce. Logger je súčasťou aplikácie *YesElf* a celá architektúra je implementovaná v spolupráci so študentom Matejom Váľkym. Pri implementácii vychádzame z analyzovaných prác. Architektúra logera je tvorená 3 základnými časťami: klient, server a databáza:

1. *Klient*. Klientská časť logera je zodpovedná za zachytávanie udalostí pomocou jazyka *java* skript. Udalosti sú zozbierané a následne odosielané na serverovú časť formou *HTTP POST* požiadavky. Aby sme predišli nadmernému posielaniu správ a tým zatažovaniu prehliadača klienta implementovali sme dávkové odosielanie. V ňom sa udalosti odošlú až po prekročení limitu. Hodnota limitu je konfigurovateľná. Okrem dávkového spracovania dát, bolo potrebné riešiť problém spojený s presmerovaním po interakcii v rámci stránky. Presmerovanie spôsobovalo, že zásobník v ktorom sa hromadili udalosti myši bol vždy premazaný. Kvôli tomuto problému dochádzalo k značnej strate informácií o správani. Podobný problém nastal aj po zatvorení okna alebo len karty prehliadača s logerom. Situáciu sme vyriešili vhodnou knižnicou, ktorá sa postarala o doposielanie dávky pri predčasnom uzavretí stránky.
2. *Server*. Na strane serveru ide o štandardnú implementáciu *REST* serveru v rámci *django* jazyka *python*. Server počúva na požiadavky klienta, ktorý mu posiela dávky správania. Následne sú tieto dávky spracované a uložené do databázy. Pri implementácii serveru sme vzali do úvahy možné použitie na produkčné účely. V rámci neho je potrebné obslúžiť až stovky klientov naraz. V produkcii je nasadených viacero vlákien s pracovníkmi (angl. *workermi*) cez proxy *uwsgi*. Celé nasadzovanie a integrácia serverovej časti logera sú plne automatizované cez kontajnerovú službu *Docker* ²⁰.
3. *Databáza*. Kľúčovým prvkom implementácie logera bol výber databázy. Zvolili sme dokumentovú databázu *elasticsearch*. Udalosti správania používateľa boli organizované do indexov pre každú aplikáciu zvlášť. Okrem toho sa indexy členili podľa mesiacov v roku. Týmto krokom sme predišli nadmernému hromadeniu záznamov v jednom indexe.

Ako sme spomínali loger je integrovaný do služby *YesElf* ²¹ firmy *brainware* a je umiestnený v klientských aplikáciách.

Predikcia v reálnom čase v aplikácii YesElf

V tejto časti predstavujeme integráciu klasifikátoru zmätenia do aplikácie *YesElf*. Sekvenčný diagram na obrázku 14 predstavuje spoluprácu jednotlivých častí aplikácie *YesElf* pri predikcii zmätenia v reálnom čase.

Je dôležité si uvedomiť, že pod objektom *Nats* rozumieme službu na serverové správy (angl. *server messaging*) od produktu *NATS*²². Táto služba umožňuje asynchrónne posielanie správ medzi jednotlivými časťami infraštruktúry s využitím prístupu publikujúcich a odberateľov (angl. *publishers and subscribers*). Klient teda zaznamenáva správanie a *LoggerServer* zapisuje do databázy, pričom sa v databáze *Redis*²³ počítajú zaznamenané akcie pre dané *userId*. Treba si uvedomiť, že v diagrame abstrahujeme od toho, že toto sa vykonáva pre viacero aplikácii naraz (aplikácii, kde je nasadený produkt). Tieto počty majú v databáze určitú dobu prežitia

²⁰www.docker.com/

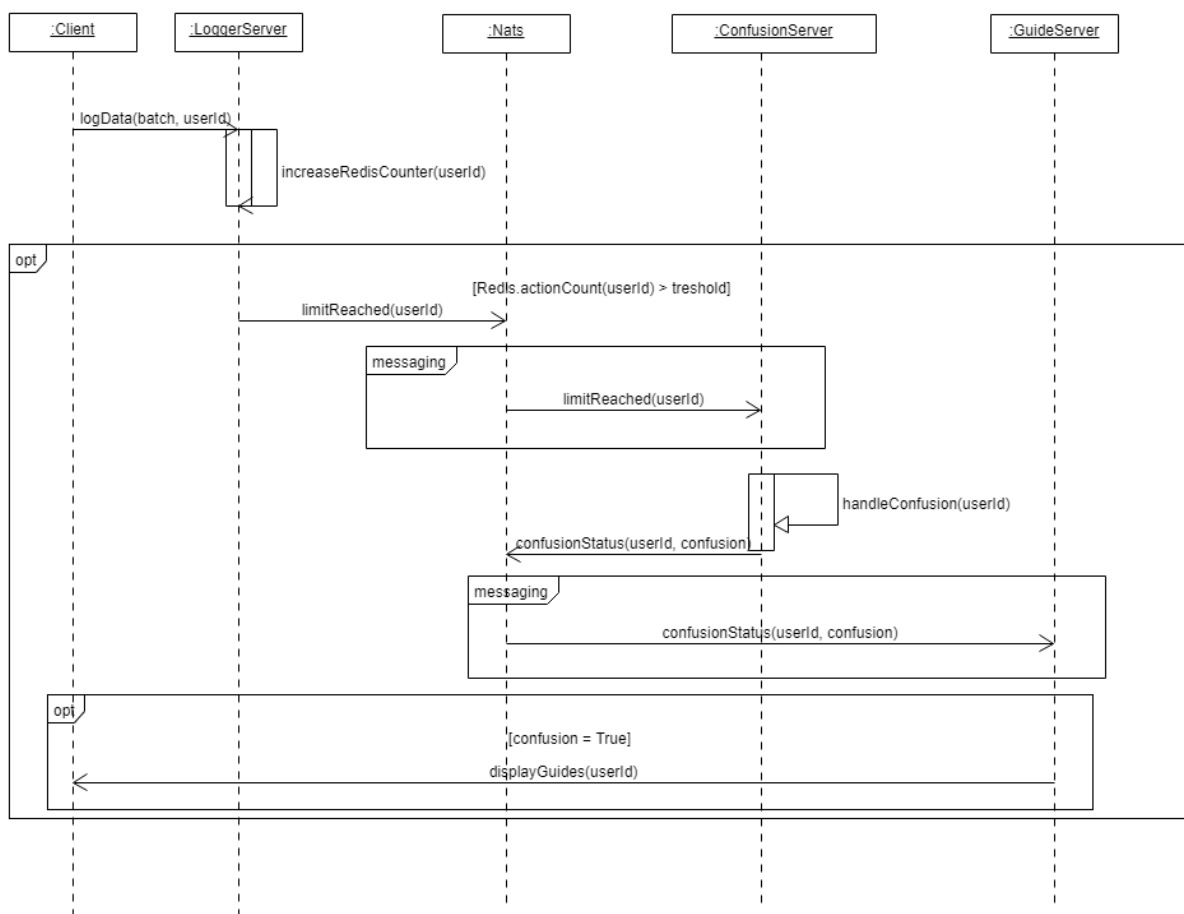
²¹www.yeself.com/

²²www.nats.io/

²³www.redis.io/

(angl. *time to live*), po ktorej uplynutí sa vynulujú. Týmto prístupom riešime situáciu, kedy sa používateľ vráti do aplikácie po určitej dobe.

Po tom čo sa pre dané *userId* zaznamená dostatočne veľa počtov záznamov, do *nats* sa zapíše správa *limitReached*, na ktorú počúva *ConfusionServer*. Ten zistí, či pre dané *userId* nastalo zmätenie. Prakticky vyberie z databázy časové okno údajov, vypočíta metriky a vloží ich na predikciu klasifikátora zmätenia. Výsledok predikcie je opäť zapísaný do *nats* ako správa na ktorú reaguje *GuideServer*. Ten musí pre dané *userId* nájsť aktívneho klienta, na ktorom otvorí sprievodcov, ak nastalo zmätenie.



Obr. 14: Diagram sekvencii implementácie predikcie zmätenia na produkcii.

Príloha D - Opis digitálnej časti práce

FIIT-182905-73915

Obsah digitálnej časti práce (archív ZIP):

/MouseData/*	Archívované údaje z myši rozdelená do štyroch priečinkov podľa dátumov behov používateľskej štúdie.
/WholeSession	Priečinkov pre časť predikcie nad celými úlohami.
/WholeSession/Analysis.ipynb	Exploratívna analýza dát nad celou úlohou.
/WholeSession/Prediction.ipynb	Predikčné modely nad celou úlohou.
/WholeSession/*.csv	Metriky určené na predikciu nad celou úlohou.
/Windows10secs	Priečinkov pre časť predikcie nad oknami veľkosti 10 sekúnd.
/Windows10secs/Analysis.ipynb	Exploratívna analýza dát nad oknami veľkosti 10 sekúnd.
/Windows10secs/Prediction.ipynb	Predikčné modely nad oknami veľkosti 10 sekúnd.
/Windows10secs/*.csv	Metriky určené na predikciu nad oknami veľkosti 10 sekúnd.
Windows10secs/EvaluationTools.py	Pomocné skripty vyhodnotenia modelov.
/Windows5secs	Priečinkov pre časť predikcie nad oknami veľkosti 5 sekúnd.
/Windows5secs/Analysis.ipynb	Exploratívna analýza dát nad oknami veľkosti 5 sekúnd.
/Windows5secs/Prediction.ipynb	Predikčné modely nad oknami veľkosti 5 sekúnd.
/Windows5secs/*.csv	Metriky určené na predikciu nad oknami veľkosti 5 sekúnd.
Windows5secs/EvaluationTools.py	Pomocné skripty vyhodnotenia modelov.
ExperimentExtraction.ipynb	Skripty určené na extrakciu dát z databázy Elasticsearch.
FeatureExtractionSession.ipynb	Skripty určené na extrakciu čít myši z dát pre celú úlohu.
FeatureExtractionWindows.ipynb	Skripty určené na extrakciu čít myši z dát pre časové okná.
EyetrackerData	Priečinkov určený na analýzu dát okulografu.
EyetrackerData/[0-2][0-9]/	Priečinky s dátami z okulografu rozdelené podľa rozmiestnenia do 20 počítačov.
EyetrackerData/EyetrackerAnalysis.ipynb	Skripty určené na spracovanie dát z okulografu.
SurveyData	Priečinkov určený na analýzu dotazníkov.
SurveyData/Surveys/*	Priečinkov s odpoveďami na dotazníky.
SurveyData/SurveyAnalysis.ipynb	Skripty určené na spracovanie dotazníkov.
praca.pdf	Pdf so záverečnou prácou.

Príloha E - IITSRC článok a plagát

Identification of User Confusion in a Web Application

Michal HUCKO*

*Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
hucko.miso@gmail.com*

Abstract. Web applications are daily visited by everyone of us. Confusion is a big issue hindering our user experience with the web application. By predicting the exact moment of the user confusion we would be able to mitigate its impacts, for example by providing helpful guides. In this paper we discuss approaches to confusion prediction using various types of implicit feedback data from web application. Namely we compare mouse interaction, eyetracker and server log data. We also proposed a method for automated confusion detection in the real time using only the mouse data. We report promising results on data collected through a user study with a travel agency's website. For the classification of confusion we used the logistic regression.

1 Introduction

Web applications are closely coupled with our everyday life. We use them at work, for shopping or during our leisure time. One of the main goals of the web applications is to fulfill the needs of the users. To reach this goal a user friendly environment has to be established. In this environment every user should be able to find the information he or she needs.

The most problematic group of users are novice ones. The first experience with a new application may lead to the user confusion. Webster dictionary¹ describes confusion as "a situation in which people are uncertain about what to do or are unable to understand something clearly". When the user visiting the application reaches the tolerability limit of the time invested to find the desired information, he or she may leave the application and never come again. In [8], Thomas et al. describe this user state as a *disorientation* or that the user is *struggling*.

In this paper we leverage mouse interaction data for the prediction of user confusion in the web application. For the purpose of evaluation we conduct a

user study with 60 participants on a travel agency's website, described next. We compare our results with predictors using eyetracking data. We discuss the possibility of real-time confusion prediction in production environment using machine learning models with just the mouse interaction data. We stand the hypothesis that from the mouse interaction data we are able to predict the confusion.

The paper is structured into six sections following the conclusion. In the second section we summarise related work in the field of user confusion detection. Section 3 describes possible features derived from sources of user implicit feedback we are able to collect from the web applications. In the section 4 we describe our method for predicting the user confusion. Section 5 gives overview of the user study conducted to gather the data. Following section provides results on these data.

2 Related work

There have been just few works that deal with the automatic user confusion detection and none of them use mouse interaction data. In [7], Lallé et al. used eye tracker for the detection of user confusion. They conducted a user study with 136 participants, during which they were asked to notify about their confusion using a software button. Participants interacted with an interactive visualization to support multicriteria preferential choice. The resulting classification had a promising precision of 0.60; however, the limitation is the current low penetration of eye tracking technology. Authors proposed a method for prediction in real time by splitting the recorded data into time windows (setups with 5 and 10 seconds) where the results were similar.

In [8], Thomas et al. used an approach where features derived from standard web server logs were used for the detection of struggling sessions (e.g., session

* Master study programme in field: Intelligent Software Systems

Supervisor: Professor Mária Bielíková, Faculty of Informatics and Information Technologies STU in Bratislava

¹ www.merriam-webster.com

duration, number of unique page visits, etc.). Although they reached the precision of 0.84, the selection of the input features makes the method difficult to use in real time, which is crucial for a timely intervention in case of detected user confusion.

3 Implicit feedback in web application

This section summarises existing approaches for extracting features from implicit feedback data. Section is divided into three parts. In first we talk about mouse interaction data, second deals with eyetracking data and the last one is about the data from HTTP servers.

3.1 Mouse implicit feedback

Working with the mouse interaction data we face the logs in the raw format. It is a sequence of mouse coordinates with the timestamps. To apply machine learning algorithms proper preprocessing has to be applied. One possible approach mentioned in the works [3,4] is to split raw data into mouse moves. Gamboa et al. defined a move as a sequence of mouse coordinates where difference between each timestamp is less than 120ms. The move must consist of at least 4 records. After splitting the raw data into moves authors extract the mouse features. This approach was used in authentication of user, based on mouse interaction. This problem is very similar to ours, because in both cases we try to model user based on application independent data.

For the confusion prediction we use these mouse interaction features derived from raw data mentioned in works [3,4].

1. *Horizontal velocity*. Value based on deviation of x coordinate through time $v_x = \frac{\delta x}{\delta t}$.
2. *Vertical velocity*. Value based on deviation of y coordinate through time $v_y = \frac{\delta y}{\delta t}$.
3. *Velocity*. Value based on deviation of both coordinates through time $v = \sqrt{v_x^2 + v_y^2}$.
4. *Acceleration*. Velocity derivation by time deviation $\dot{v} = \frac{\delta v}{\delta t}$.
5. *Jerk*. Acceleration deviation by time deviation $\ddot{v} = \frac{\delta \dot{v}}{\delta t}$.
6. *Distance*. Sum of distances through each coordinate $s = \sum_{i=1}^n \sqrt{(x_i^2 + y_i^2)}$.
7. *Duration*. Sum of moves duration $t = \sum_{i=1}^n \delta t_i$.

Features 1 to 5 were computed for each move within user data. To represent user behavior as whole means of given features were computed. Features 6 and 7 were computed as a sum to represent user behavior.

3.2 Eyetracker implicit feedback

Eyetracker interaction data are very similar to mouse interaction data. In both cases we are working with timeseries of coordinates. According to work [2], T Duchowski says that before actually computing the features we have to pre-process the raw coordinates into fixations and saccades. A saccade is reflexive move of the eye, where human is unable to read. Its duration is between 10 to 100ms. The fixation is a stable moment of eye. According to work of Duchowski, this moment is connect with some micro moves around the center of the fixation. Except these moves there are other characteristics connected with the eyetracker data such as head distance and pupil dilatation. According to work [7], Lallé et al. showed that head distance features were the most significant features in confusion prediction from eyetracker data.

For the confusion prediction we use following features derived from computed fixations and saccades.

1. *Number of fixations*. Fixation count for given time interval.
2. *Number of saccades*. Saccade count for given time interval.
3. *Fixation ratio*. Fixation ratio to time interval.
4. *Saccade ratio*. Saccade ratio to time interval.
5. *Head distance*. Head distance from the eyetracker.
6. *Pupil deviation*. Pupil size deviation from average pupil size.

Each of the features has been mentioned in works [5,6], where Kardan and Conati used it to model the user to determine his or her learning with interactive simulations.

3.3 Server logs

Http server is one of the standard solutions in modern web applications. Requests from client part of the applications are sent to it and processed afterwards. Most of the HTTPS servers store these information in logs. Mentioned in related work Thomas et al. used this data in work [8] to classify struggling sessions. We can use the data to classify confused sessions as well. Advantage of this approach is that collecting of this information is costless thanks to the server logs.

For the prediction we will use following features.

1. *Time spent*. Total time spent within prediction window.
2. *Total pages*. Number of unique pages visited for given window.
3. *Time per page*. Time spent divided by total pages.
4. *Cycles*. Number of cycles for given window. Cycle is an event when a user returns to url he or she

already visited in the past.

4 Automated confusion detection

To identify confused users in a web application we designed a method for automated confusion detection. We divided our problem into two scenarios. In the first one we predict the confusion based on the whole session data and in the second we try to predict confusion in real time. Considering labeled training data, the method consists of following the steps.

1. *Raw data preprocessing.* In this step we clean up the data. In case of the mouse interaction data we compute the mouse moves. In case of the eye-tracker data we compute fixations and saccades, or we use proper filter. In case of real time prediction the data can be split into the windows. This approach has been used in [7]. Windows are collection of interaction data which are equally sized in terms of the time duration.
2. *Feature extraction.* In this step we compute features, summarised in section 3 from the preprocessed data. In case of highly imbalanced dataset, the oversampling of minority class should forego.
3. *Model training.* Having all the features ready, machine learning model is being trained.
4. *Prediction.* After the model being trained, the prediction is being applied on equally sized windows, in case of real time prediction, and on the whole session in the first scenario (while predicting on the session as whole). Again the same data pre-processing and feature extraction must be applied to the test data.

5 User study

To evaluate our method we conduct a study with 60 participants in the UXICenter² at the Faculty of Informatics and Information Technologies at the Slovak University of Technology in Bratislava. Participants solved 6 tasks at the travel agency portal Firotour³. We used 20 computers equipped with the Tobii TX300 eyetracker to record eye moves, each one with frequency of 300Hz. For the recording of the mouse interaction data we use the mouse logger added directly to agency's application with the frequency of 60Hz.

All the tasks of the study were designed with cooperation of the travel agency Firotour helpdesk. Each of the tasks represents common behavior of the appli-

cation visitors. For example the participants should find the cheapest vacancy in Turkey with the departure from Bratislava, or find how far is given hotel from the beach etc.

The challenging part of the study was to record the exact moment of the participant's confusion. We use this data as ground truth labels later in the results of this paper. In our case we modified approach used [7], where Lallé et al. used button with the text "I am confused" situated in the right top corner of the screen. Authors did the pilot studies with this button reported in work of Conati, mentioned earlier. To explain the usage of the button authors provide a descriptive text to the participants. In our case we use the same setup with the minor button text modification to just the exclamation mark. We did not want our participants to feel bad that they have some problems with the tasks. In our opinion the original text can evoke that feelings.

After each task participants had to answer three questions. The first was about the answer of the task they accomplished. Second was a boolean question whether they pressed the confusion button. The last question should be answered only if they pressed the confusion button. In this question we asked about the reasons of pressing. For the whole study setup we used iMotions⁴ software.

6 Results

At the study we had 60 participants with 6 tasks each. After collecting the data minor technical issues were revealed, because of this only 336 valid records were used. The confusion button was pressed 112 times.

Before presenting results it is important to mention that we used python's *scikitlearn* for all the computations. In each case of training of the model we used the grid-search (*GridSearchCV*) for fine-tuning the hyper-parameters of the model. We used stratified k-fold cross validation in the grid-search with 5 folds. For the logistic regression we fine tuned parameters *C* (inverse of regularization strength) and *penalty* (norm in penalization). In the seature setup we used polynomial features where we fine tuned the polynomial degree used. We also used *StandardScaler* for feature normalization and *SelectFromModel* with *LinearSVC* for feature selection.

In the first step we tried to predict the confusion on the whole session. If the given session was confused, we used only the data before the button was pressed, otherwise the whole session data was used. In this case no oversampling was not required, because the train-

² <https://www.pewe.sk/uxi/>

³ www.firotour.sk

⁴ www.imotions.com

ing dataset was balanced. As the classifier we used logistic regression algorithm, because of its results on general binary classification problems. Table 1 shows the results. Here we can see, that the results using only the mouse data are comparable with methods using the eyetracker data and the http server data.

Table 1. Whole session classification results using eyetracker data, mouse data and http server data.

	Prec(0)	Prec(1)	Rec(0)	Rec(1)
Eyetracker	0.70	0.50	0.80	0.38
Mouse	0.80	0.60	0.86	0.40
HTTP	0.75	0.72	0.85	0.45
ALL	0.94	0.77	0.87	0.88

The second step was the real time prediction. In this case we split the data into equally sized windows. We tried various lengths, but the setup with size of 10 seconds seemed to reach the best results. Only seconds before pressing the confusion button were labeled as confused. Because of the sliding window usage on non-confused records, we ended up with highly imbalanced dataset. To overcome this problem we used *SMOTE* oversampling based on *k nearest neighbour* presented in work [1]. We used the parameter *k* of 5 for algorithm setup. We chose to over sample minority class by 200%, 500% and equal size to non-confused windows. Logistic regression has been used. Results are presented in the table 2. As we can see by increasing minority class size we are increasing recall on confusion prediction.

Table 2. Real time classification results using only mouse interaction data and logistic regression with time window of 10s. 200% oversampling (S200), 500% oversampling (S500) and oversampling to majority class (1to1) with *SMOTE* algorithm used.

	Prec(0)	Prec(1)	Rec(0)	Rec(1)
S200	0.99	0.40	0.97	0.38
S500	0.99	0.63	0.97	0.40
1to1	0.99	0.25	0.97	0.61

7 Conclusion

In this paper we analysed confusion prediction while interaction with a travel agency's web application *Firotour*. We compared various approaches in confusion prediction on whole session using mouse interaction, eyetracker and http server data. We also proposed a method for real time confusion prediction

using only mouse interaction data. Our method uses logistic regression classifier for the prediction.

Our results show that we are able to predict the confusion in the real time using only the mouse interaction data. 63% of samples predicted as confused actually were confused and 99% of not confused actually are not confused. Considering the fact that this model predicts on time windows this solution is easy to use in real production environment.

Estimating the exact moment of user confusion is a very useful tool. We can use this information to help the user to overcome troubles. Ideal tool in this issue are user guides with steps navigating the confused user through the web application. *Yeself*⁵ is a software tool providing these guides. Nowadays, we are working on the production module for *Yeself* solution for real time confusion prediction.

Acknowledgement: Here you should mention the names of grants which funded the work on this article. As defined by the particular grant.

References

- [1] Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: *SMOTE: synthetic minority over-sampling technique*. *Journal of artificial intelligence research*, 2002, vol. 16, pp. 321–357.
- [2] Duchowski, A.T.: *Eye tracking methodology. Theory and practice*, 2007, vol. 328.
- [3] Gamboa, H., Fred, A.: *A User Authentication Technique Using a Web Interaction Monitoring System*. In: *Iberian Conference on Pattern Recognition and Image Analysis*, Springer, 2003, pp. 246–254.
- [4] Gamboa, H., Fred, A.: *A behavioral biometric system based on human-computer interaction*. In: *Biometric Technology for Human Identification*. Volume 5404., International Society for Optics and Photonics, 2004, pp. 381–393.
- [5] Kardan, S., Conati, C.: *Exploring gaze data for determining user learning with an interactive simulation*. In: *International Conference on User Modeling, Adaptation, and Personalization*, Springer, 2012, pp. 126–138.
- [6] Kardan, S., Conati, C.: *Comparing and combining eye gaze and interface actions for determining user learning with an interactive simulation*. In: *International Conference on User Modeling, Adaptation, and Personalization*, Springer, 2013, pp. 215–227.
- [7] Lallé, S., Conati, C., Carenini, G.: *Predicting Confusion in Information Visualization from Eye Tracking and Interaction Data*. In: *IJCAI*, 2016, pp. 2529–2535.
- [8] Thomas, P.: *Using interaction data to explain difficulty navigating online*. *ACM Transactions on the Web (TWEB)*, 2014, vol. 8, no. 4, p. 24.

⁵ www.yeself.com

Identification of User Confusion in a Web Application

Author: Michal Hucko

Supervisor: Mária Bielíková

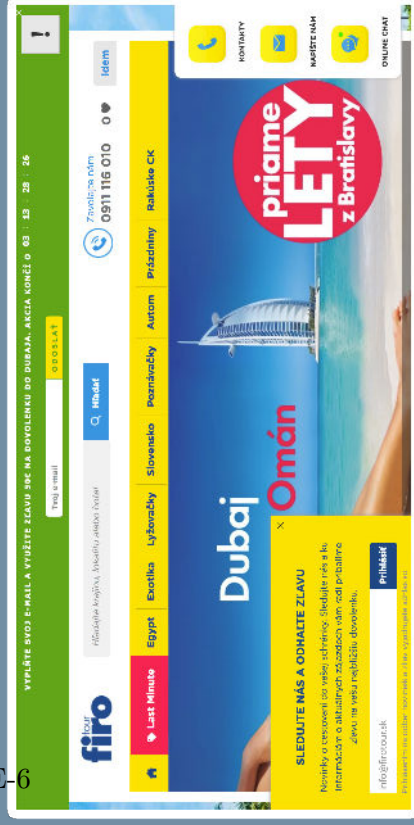
Mouse Logger



Scalable
Realtime

User Study

Confusion button
60 participants
FiroTour



Dataset

68
fixations
saccades
pupil
clicks
scrolls
mousemoves

Whole session
prediction

Mouse features
Eyetracker features
HTTP server features
0.94 precision on not confused data
0.77 precision on confused data

Histogram
representation



Real time
prediction

Mouse features
Time windows
0.99 precision on not confused data
0.40 precision on confused data

Logistic
regression



**Príloha F - UMAP článok (ACM Press, prijatý do sekcie Late
Breaking Results)**

YesElf: Personalized Onboarding for Web Applications

Michal Hucko

Slovak University of Technology in
Bratislava, UXI Research Center,
Slovakia
xhuckom@stuba.sk

Ladislav Gazo

Peter Simun
Seges, Bratislava, Slovakia
gazo,simun@seges.sk

Matej Valky

Slovak University of Technology in
Bratislava, UXI Research Center,
Slovakia
xvalky@stuba.sk

Robert Moro

Slovak University of Technology in
Bratislava, UXI Research Center,
Slovakia
robert.moro@stuba.sk

Jakub Simko

Slovak University of Technology in
Bratislava, UXI Research Center,
Slovakia
jakub.simko@stuba.sk

Maria Bielikova

Slovak University of Technology in
Bratislava, UXI Research Center,
Slovakia
maria.bielikova@stuba.sk

ABSTRACT

Onboarding users to a complex application or a new functionality can be a serious issue, especially for organizations that need to train their new employees. Using a complex application without proper training or guidance can lead to users' confusion and frustration. In this paper, we introduce the onboarding platform *YesElf* intended for web applications. Its approach to onboarding is to use embedded guides within the application; its novelty lies in the robustness, ease of setup and integration of the *YesElf* guides into any web-based application. Most importantly, *YesElf* supports personalized adaptation of user guidance. This, we demonstrate by a novel method for automated recognition of user's confusion in real time that we integrated into *YesElf*. The information on user's confusion serves as a basis for adaptive display of the guides, when they are needed the most. We evaluated the proposed method on the data collected in a user study with 60 participants and achieved the 63% precision which outperforms the state-of-the-art classifier based on the eye tracking data (although, in our case, we used the more readily available mouse movement data).

CCS CONCEPTS

• **Human-centered computing** → **Interaction paradigms**; *Web-based interaction*.

KEYWORDS

Onboarding; confusion detection; guides; personalization

ACM Reference Format:

Michal Hucko, Ladislav Gazo, Peter Simun, Matej Valky, Robert Moro, Jakub Simko, and Maria Bielikova. 2019. *YesElf: Personalized Onboarding for Web Applications*. In *27th Conference on User Modeling, Adaptation and Personalization Adjunct (UMAP'19 Adjunct)*, June 9–12, 2019, Larnaca, Cyprus. ACM, New York, NY, USA, 6 pages. <https://doi.org/10.1145/3314183.3324978>

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

UMAP'19 Adjunct, June 9–12, 2019, Larnaca, Cyprus

© 2019 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 978-1-4503-6711-0/19/06...\$15.00

<https://doi.org/10.1145/3314183.3324978>

1 INTRODUCTION

Onboarding of users (be it customers or employees) to an application is of crucial interest to businesses. If the customers do not know how to use the application, they will look for an alternative. In case of untrained employees, it can directly harm the company by lower productivity, increased costs and ultimately, the frustrated employees may leave and new ones will face the same challenges. Whether the application complexity stems from low usability of its user interface (in particular, its learnability or memorability aspects [13]) or the complexity of the underlying business process, it often confuses its users. By *user confusion*, we mean a short-term state, during which the user is not sure how to use the application to fluently proceed with his/her task. Even with a solid application usability, the confusion still happens: users come with varying prior experience and knowledge (both general and specific) about the user interfaces. Therefore, there are always some users who struggle with the application.

The occurrences of the user confusion hinder the business of organizations whose employees use a digital solution. Confusion slows down the work, frustrates the user and may lead to mistakes. It typically threatens the novice users (e.g., steep learning curves), but happens also to the experienced ones. Sometimes, the user needs to use some rarely-used feature or re-learn to use the application after a major update. In fact, the employees may be so entrenched in the ways they use a particular software, that its major update also means a major software adoption issue for the organization. This means that focusing solely on the novices onboarding is not enough; the applications have to provide the users with help and assistance when needed (but not bother them with any prompts in moments when these would be perceived as disturbing).

In this work, we present *YesElf*¹ – an onboarding platform for web applications that enables adaptive and personalized user guiding. We demonstrate the adaptation capabilities of *YesElf* with an automated user's confusion detection scenario, for which we propose a novel real time detection approach.

Detecting user confusion state in real time is not an easy task. The existing works (e.g., [10]) have about 60% precision when using eye tracking data. In our work, we employ mouse movement data instead, since they are more readily available and hypothesize that we can obtain a comparable or even higher precision.

¹<https://www.yeself.com/>

The rest of the paper is structured as follows. In section 2, we discuss user onboarding systems used nowadays. Next, we present onboarding tool YesElf with its adaptive guides. In section 4, we demonstrate adaptive capabilities of the YesElf by proposing a method for user confusion detection based on mouse movements that we integrated into YesElf and deployed into the customer environment. We evaluated the method by means of a user study conducted with 60 participants; we present the results also in section 4. In the last section, we summarize the main contributions and discuss possible future work.

2 EXISTING SOLUTIONS TO ONBOARDING

Organizations have been mitigating the software usability drawbacks as well as novices' lack of experience with *onboarding* methods. The classical approaches range from demo courses, guided hands-on sessions, video-tutorials, text manuals and FAQs to senior colleague advising. The more recent ones include approaches like microlearning and gamification [7], blended and flipped learning [8, 12, 16], or the use of virtual or augmented reality [4].

There exist several onboarding tools. Depura and Garg [3] shared their experience with the online social tools and the gamification for training new employees. In [15], Renz et al. discussed onboarding of hundreds of daily newcomers in Massive Open Online Courses, where it is impossible to give each of them a consultation. In such cases, *in-situ* guides, which are built into the applications themselves are often used to provide guidance to the user as he/she uses the application. More recently, new onboarding solutions independent of a single application, such as *WalkMe*², *Pendo*³, *Appcues*⁴ or *Nickelled*⁵ have been developed. The range of different recently developed solutions and tools only signifies that user onboarding is recognized as an important problem.

Although some of the solutions (e.g., *WalkMe*) try to adapt the guidance to the users' needs, there are none incorporating adaptation based on the actual state of the users, namely on the identification of their confusion.

3 YESELF GUIDES

Similar to the existing solutions, YesElf guides the user using hints (i.e., guide steps) displayed as pop-up bubbles near relevant UI elements (see Figures 1 and 2). However, YesElf has several distinctive features; namely,

- (1) it can be easily integrated to an existing web application,
- (2) it can be quickly set up (hints be created and assigned), even by a non-technical person, thanks to the provided guide editor,
- (3) it implements fine-grained logging of user activity in real time, and most importantly,
- (4) YesElf uses the detailed logs of user activity (e.g., mouse movement, clicks) and machine learning to *identify the confusion state of the user* and thus display the guides at the right moment.

²<https://www.walkme.com/>

³<https://www.pendo.io/>

⁴<https://www.appcues.com>

⁵<https://www.nickelled.com/>

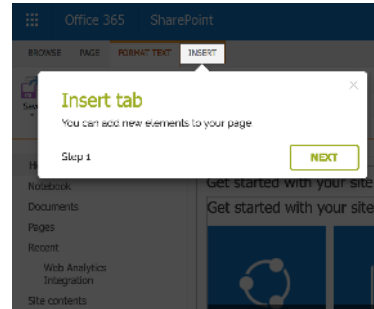


Figure 1: A sample hint anchored to an HTML tab in navigation bar.

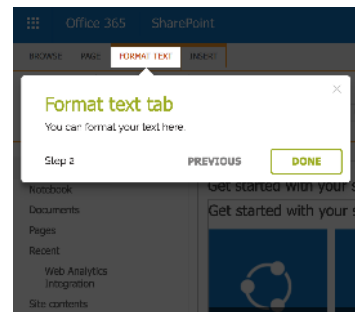


Figure 2: A hint displayed after clicking “Next” in the figure 1.

If an application designer or a company using off-the-shelf software wishes to use YesElf to ease the process of user onboarding in their application, they need to create the *guides* in the YesElf *web editor* (see Figure 3), which is an easy-to-use responsive web application dedicated for that purpose. Each guide consists of a series of hints and actions.

While constructing the guides, the designer has to specify the UI elements to anchor the hints to. YesElf implements four HTML element selectors to ease this process. This makes the UI element bindings robust to the refactoring of the HTML source code of the web application and guides will also persist modifications of CSS code. Created guides are still usable even if the anchored element disappears. Additionally, YesElf can be used on mobile platforms: displayed hints adapt to screen size and changing position of the elements.

Alternately, the guides do not have to be defined entirely manually, but the process of their creation can be semi-automatized by monitoring the work of a domain expert.

The YesElf editor provides tools for the *hint customization*. Besides the hint visuals (e.g., theme, border, position, etc.), the designer can define custom *triggers* (rules), when a given hint should (dis)appear. An example of a trigger is that a specified anchored UI element gets into the middle of the viewport. Here the user can specify even more complex rules depending on the website structure using custom *Javascript* code.

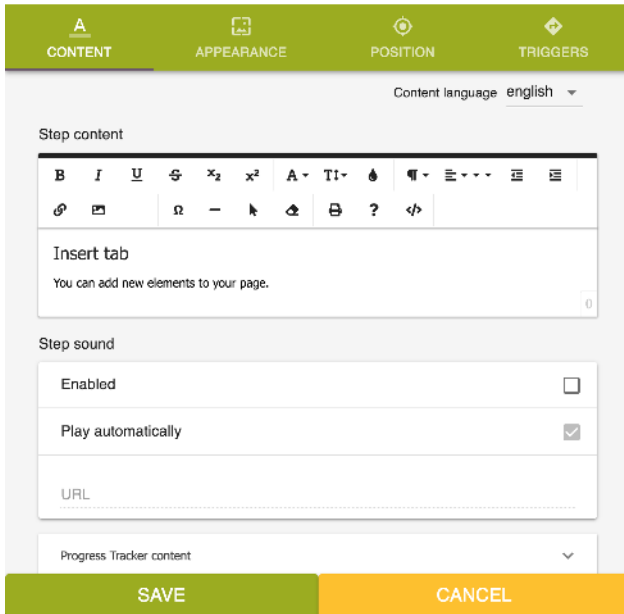


Figure 3: Simple interface for hint editing. Besides editing the hint content, it is possible to customize its appearance, where it will be shown (i.e., its position), and the triggers (rules) when the hint should be presented to the users.

In their basic form, these rules are not personalized and, thus, the guides would appear for all users even if they might not need them. In fact, they can be useless or even annoying to the experienced users (who tend to close them immediately after they appear). Therefore, we proposed and implemented personalization mechanism for user guides display, which is based on detection of user confusion (see next section).

After guides definition, the designer can easily plug them into his/her web application by extending its source code by a small snippet of *Javascript* code. If the designer defined multiple alternative guides, he/she can test them using AB testing functionality provided by YesElf and evaluate the usefulness of the guides (e.g., how many times users closed a guide, or how many of them finished the entire guide). For this purpose, YesElf provides its own analytic tool. In this tool the designer can see interactions with the guides on the steps granularity. He/she can inspect how many users got stuck in the middle of the guide during interaction with the website. More importantly, the analytic tool also allows to identify the problematic parts of the digital solution (i.e., the application) and its bottlenecks and monitor errors caused by users, which can be used to educate the users in future to reduce the error rate.

The Yeself has been used by a dozen of customers so far, e.g., by Imos Systemair (with users from 30 countries in Europe) or by Aegon where it has already been used to onboard 5000 employees. Another one was the travel agency FIRO-tour, where we deployed our guides on agency’s website (which has more than 200,000 unique monthly users) to help the visitors to better navigate in the online store. As a result, by using just a few simple guides, the click through rate increased by more than 9%.

Another customer was the ALITER technologies where Yeself helps with the onboarding of the new employees with the company’s software. In this case, YesElf’s guides reduced the two phases of training cycle to just one. During the initial deployment it led to a 50% reduction of the planned costs in the company.

One another example of fluent adaptation of application users to the YesElf guides is the COOP company. Here the employees work with the YesElf independently without any supervision needed.

4 MAKING THE YESELF GUIDES PERSONALIZED: CONFUSION DETECTION

In the context of guides provided by YesElf, it makes sense to personalize *when* the guides should appear and *what guides* should appear to *what users*. This requires user modelling and intent prediction. The relevant user characteristics to model for this task are, e.g., the level of expertise or knowledge of the user which can be modeled by using learning curves of the users [11]. Besides this, the user state (such as his/her confusion) plays an important role. As to the user intent prediction, existing works often employ eye tracking (see, e.g., [1]), but its prevalence is still limited.

In our current work, to make the YesElf guides personalized, we propose a confusion detection method which aims to identify moments of user interaction when he/she is unable to continue fluently with the work; we consider these moments ideal for showing the user a pre-defined guide. The challenging part of this solution is to make this prediction in real time.

There are only few works that deal with automatic user confusion detection. In [10], the authors used eye tracker for the detection. They conducted a user study with 136 participants, during which they were asked to notify about their confusion using a software button in right top corner of the screen. This method considers the real time prediction also. For the training of the classification model authors split the gathered data into time windows. For each window they compute the gaze features and with them they trained the model. The resulting classification had a promising precision of 0.61; however, the limitation is the current low penetration of eye tracking technology.

In [17], features derived from standard web server logs were used for the detection (e.g., session duration, number of unique page visits, etc.). In this case, no real time prediction has been done. Authors predicted the user state on the whole session data. Although they reached the precision of 0.84, the selection of the input features makes the method difficult to use in real time, which is crucial for a timely intervention in case of detected user confusion.

The design of our method is similar to the existing approach proposed in [10]. However, a major difference is that in our case, we use implicit feedback from mouse interaction data instead of eye tracking data. The method is also designed to be able to predict (classify) the user confusion in real time.

4.1 Mouse features

As mouse device is essential part of almost every computer, we use mouse interaction data to make the confusion prediction. To record the mouse behavior of each user, YesElf provides a logger to the client browser. This logger tracks mouse interaction in the target web application and sends it to the dedicated server. Instead of single

mouse events we send batches of data to reduce the network traffic. These are recorded in the database for future analysis at the server side. Logger can be plugged into the application by incorporating the same *Javascript* snippet (mentioned in the earlier section) to the existing *HTML* document structure. It can be easily configured in the YesElf editor, where the designer can specify frequency and types of events used for mouse logging.

In our method, we extract features from mouse movements similarly to the works [5, 6], where the authors used mouse interaction data for user authentication. Based on their work, we define a mouse movement as a sequence of consecutive mouse events where time difference between each event is less than 120ms. From the mouse movements, we extract the following features that we then use for the user confusion prediction:

- (1) *Total distance*. It is a sum of distances between each mouse event within the mouse movement.
- (2) *Horizontal velocity*. Value based on deviation of *x* coordinate through time.
- (3) *Vertical velocity*. Value based on deviation of *y* coordinate through time.
- (4) *Velocity*. Value based on deviation of both coordinates through time.
- (5) *Acceleration*. Velocity derivation by time deviation.
- (6) *Jerk*. Acceleration derivation by time deviation.
- (7) *Duration*. Sum of moves duration within the time window.

To better model the user, we represent his/her activity with the histograms. For a given time window we compute how many moves belong to each histogram bin of given features. The thresholds of bins and number of bins are computed based on the training data. This approach was inspired by work of [9] for web user authentication based on mouse interaction.

4.2 Confusion detection in real time

We use user activity consisting of *mouse events* to predict moments of users' confusion. The raw activity data is first split into time windows, processed and the key features are extracted from user's mouse movements within windows.

We model a user by using a histogram representation of the features (see Figure 4 for a schema of the proposed confusion prediction method). Before the actual prediction, the module has to be trained on labelled user behavioral data. If there are no labelled data for the user, heuristics based on the statistics of mouse movement features computed from the training data (e.g., mouse velocity features averaged over all (non)-confused users in the training data) are used for detection of his/her confusion.

Our proposed solution is designed to work in real time. To speed up the performance, we use web sockets for communication between the confusion detection module and the client in the web browser.

4.3 Results

To evaluate the proposed confusion prediction method, we conducted a study on FIRO-tour travel agency's web site⁶ (in Slovak) which has more than 200,000 unique monthly users and already a

⁶<https://www.firotour.sk/>

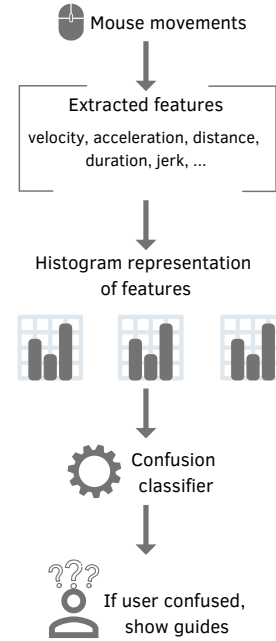


Figure 4: Schema of confusion prediction method.

quite good level of usability. There were 60 participants who took part in the study which consisted of 6 tasks. Each task was designed with the cooperation of travel agency to reflect common needs of websites visitors (e.g., find the cheapest vacancy to Turkey with departure from Vienna, find out how far is a given hotel from the beach, etc.).

To gather the ground truth labels of user confusion, we attached a software button with exclamation mark into the right top corner of the site. Figure 5 shows the button at the homepage. Participants were informed to use this button through whole session, anytime they need to. This approach was inspired by the work [10], which used the same technique to records moments of users' confusion. If the participants clicked the button, they were shown a questionnaire, in which they were asked to describe the reasons for pressing the button.

Throughout the study, the participants pressed the button 112 times in total. We used the study data to evaluate our method. The confusion prediction classifier was trained on a subset of all collected data. Before training, time windows were extracted and features were computed. We experimented with different size of time windows and decided to use the size of 10 seconds in the rest of our evaluation based on the achieved results. After this step, the data had to be balanced, because for the confused samples, we used only time windows directly preceding the button press. We ended up with 112 positives to more than 15,000 negative samples. As the solution we used *SMOTE* oversampling based on *k nearest neighbour* presented in work [2]. We choose *k* equal to 5. We experimented with the oversampling rate of minority class, namely oversampling by 200 and 500 percent. We also tried to oversample the minority class to majority class size. We used logistic regression

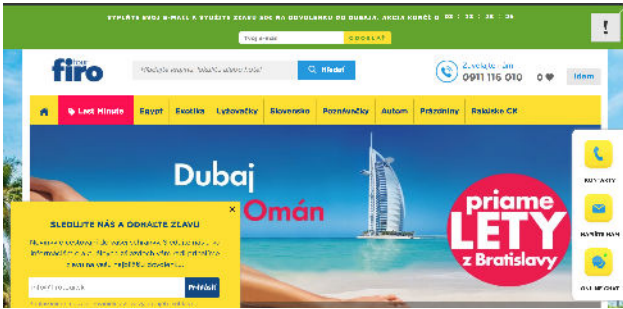


Figure 5: Confusion button (exclamation mark) in the right top corner of the screen for gathering ground truth labels. It was designed not to attract user’s attention, but to still be well visible.

for the prediction, because of its generally good results on binary classification problems.

The results of the proposed method for various oversampling setups are presented in Table 1. It is important to note that we used stratified 5-fold cross validation for evaluation of results and grid search was used for fine tuning the hyperparameters of the used logistic regression model (we used its implementation in Python’s scikit learn library [14]). Presented results are computed as average of each fold’s result.

By increasing the size of minority class we can improve the precision of confusion prediction. Our results show that we are able to predict the confusion with relatively high precision (which slightly outperforms the state-of-the-art methods) in real time. It could be even increased by gathering more ground truth labels of confusion. This can be a major advantage of YesElf tool, as it could lead to more personalized guides shown to users only when they really need them.

Table 1: Real time classification results using only mouse interaction data and logistic regression (1 stands for the positive class of the confusion instances) with time window of 10s. 200% oversampling (S200), 500% oversampling (S500) and oversampling to majority class (1to1) with SMOTE algorithm used. The best results of precision as well as recall on the minority (positive) class were achieved with 500% oversampling of the minority class without worsening the performance on the majority class.

	Prec(0)	Prec(1)	Rec(0)	Rec(1)
S200	0.99	0.40	0.97	0.38
S500	0.99	0.63	0.97	0.40
1to1	0.99	0.25	0.97	0.61

5 CONCLUSIONS

This work presents YesElf, an onboarding tool for web applications, enabling built-in pop-up hinting capability. YesElf can be easily integrated into any HTML-based interface. It is also easily configurable through visual editor. Lastly and most importantly, YesElf is able to detect confusion state based on user’s activity and thus display

the hints at the right moment. This makes the provided hinting capability (guides) intelligent and presents a good basis for their further personalization (e.g., in the form of user intent detection) in the future.

YesElf is especially suited for organizations, which rely on the use of complex software applications by their employees. It has been already successfully deployed to several companies where it catalyzed the onboarding of new employees and users alike (while saving the effort and cost on the organization side). It is also helpful during major application updates as it *adaptively* informs the employees about the changes.

To demonstrate the adaptive features of YesElf, we proposed and evaluated a method for user confusion prediction that we integrated into the YesElf and successfully deployed in the customer environment. The method estimates the exact moment of the user confusion. According to the results of our user study with 60 participants, by using logistic regression as a binary classifier we are able to predict the user confusion with 63% precision.

For the future work, we plan to recommend exact guides suitable for a given confused user based on his/her behavior. This could be very helpful because at the moment, the users have to manually choose from the guides provided to them at the moment of their confusion. We also plan to further analyze gathered data in order to improve precision of confusion prediction in real time.

ACKNOWLEDGMENTS

This work was partially supported by the projects APVV-17-0267, VG 1/0667/18, VG 1/0725/19, APVV-15-0508 and it is the partial result of education and research development project no. 002STU-2-1/2018.

REFERENCES

- [1] Roman Bednarik, Hana Vrzakova, and Michal Hradis. 2012. What Do You Want to Do Next: A Novel Approach for Intent Prediction in Gaze-based Interaction. In *Proceedings of the Symposium on Eye Tracking Research and Applications (ETRA '12)*. ACM, New York, NY, USA, 83–90. <https://doi.org/10.1145/2168556.2168569>
- [2] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. 2002. SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research* 16 (2002), 321–357.
- [3] Krishna Depura and Mohit Garg. 2012. Application of online gamification to new hire onboarding. In *3rd Int. Conf. on Services in Emerging Markets (ICSEM'12)*. IEEE, 153–156.
- [4] Markus Funk, Andreas Bächler, Liane Bächler, Thomas Kosch, Thomas Heidenreich, and Albrecht Schmidt. 2017. Working with Augmented Reality?: A Long-Term Analysis of In-Situ Instructions at the Assembly Workplace. In *Proceedings of the 10th International Conference on Pervasive Technologies Related to Assistive Environments (PETRA '17)*. ACM, New York, NY, USA, 222–229. <https://doi.org/10.1145/3056540.3056548>
- [5] Hugo Gamboa and Ana Fred. 2003. A User Authentication Technique Using a Web Interaction Monitoring System. In *Iberian Conference on Pattern Recognition and Image Analysis*. Springer, 246–254.
- [6] Hugo Gamboa and Ana Fred. 2004. A behavioral biometric system based on human-computer interaction. In *Biometric Technology for Human Identification*, Vol. 5404. International Society for Optics and Photonics, 381–393.
- [7] Bernhard Göschlberger and Peter A. Bruck. 2017. Gamification in Mobile and Workplace Integrated Microlearning. In *Proceedings of the 19th International Conference on Information Integration and Web-based Applications & Services (iiWAS '17)*. ACM, New York, NY, USA, 545–552. <https://doi.org/10.1145/3151759.3151795>
- [8] Suniti Hewett, Karen Becker, and Adelle Bish. 2019. Blended workplace learning: The value of human interaction. *Education+ Training* 61, 1 (2019), 2–16.
- [9] Peter Kratky and Daniela Chuda. 2018. Recognition of web users with the aid of biometric user model. *Journal of Intelligent Information Systems* 51, 3 (2018), 621–646.

- [10] Sébastien Lallé, Cristina Conati, and Giuseppe Carenini. 2016. Predicting Confusion in Information Visualization from Eye Tracking and Interaction Data. In *IJCAL*. 2529–2535.
- [11] Sébastien Lallé, Dereck Toker, Cristina Conati, and Giuseppe Carenini. 2015. Prediction of Users' Learning Curves for Adaptation while Using an Information Visualization. In *Proceedings of the 20th International Conference on Intelligent User Interfaces - IUI '15*. ACM Press, New York, New York, USA, 357–368. <https://doi.org/10.1145/2678025.2701376>
- [12] Allison Nederveld and Zane L Berge. 2015. Flipped learning in the workplace. *Journal of Workplace Learning* 27, 2 (2015), 162–172.
- [13] Jakob Nielsen. 2010. CHAPTER 1 - What Is Usability? In *User Experience Re-Mastered*, Chauncey Wilson (Ed.). Morgan Kaufmann, Boston, 3 – 22. <https://doi.org/10.1016/B978-0-12-375114-0.00004-9>
- [14] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in Python. *Journal of machine learning research* 12, Oct (2011), 2825–2830.
- [15] Jan Renz, Thomas Staubitz, Jaqueline Pollack, and Christoph Meinel. 2014. Improving the Onboarding User Experience in MOOCs. *Proceedings EduLearn* (2014).
- [16] Ngoc Thuy Thi Thai, Bram De Wever, and Martin Valcke. 2017. The impact of a flipped classroom design on learning performance in higher education: Looking for the best “blend” of lectures and guiding questions with feedback. *Computers & Education* 107 (2017), 113–126.
- [17] Paul Thomas. 2014. Using interaction data to explain difficulty navigating online. *ACM Trans. on the Web* 8, 4 (2014), 24.

Príloha G - Návrh článku na konferenciu

Identification of user confusion in a web application

Michal Hucko

Slovak University of Technology in Bratislava, UXI
Research Center, Slovakia
xhuckom@stuba.sk

Maria Bielikova

Slovak University of Technology in Bratislava, UXI
Research Center, Slovakia
maria.bielikova@stuba.sk

ABSTRACT

Web applications are daily visited by everyone of us. Confusion is a big issue hindering our user experience with the web application. By predicting the exact moment of the user confusion we would be able to mitigate its impacts, for example by providing helpful guides. In this paper we present a method for automated confusion detection in web application. Our method consists of two scenarios, in the first one we predict confusion from the whole session data. This approach can be used as retrospective evaluation of visitor sessions. In the second approach we predict confusion in real time trying to estimate the exact moment. We report promising results on data collected through a user study with a travel agency's website. For the classification of confusion we compare the logistic regression and the random forest classifier.

CCS CONCEPTS

• **Human-centered computing** → **Interaction paradigms**; *Web-based interaction*.

KEYWORDS

Onboarding; confusion detection; guides; personalization

ACM Reference Format:

Michal Hucko and Maria Bielikova. 2019. Identification of user confusion in a web application. In *Proceedings of ACM Conference (Conference'17)*. ACM, New York, NY, USA, 5 pages. <https://doi.org/10.1145/nnnnnnn.nnnnnnn>

1 INTRODUCTION

Web applications are closely coupled with our everyday life. We use them at work, for shopping or during our leisure time. One of the main goals of the web applications is to fulfil the needs of the users. To reach this goal a user friendly environment has to be established. In this environment every user should be able to find the information he or she needs.

Webster dictionary¹ describes confusion as "a situation in which people are uncertain about what to do or are unable to understand something clearly". When the user visiting the application reaches the tolerability limit of the time invested to find the desired information, he or she may leave the application and never come again.

¹www.merriam-webster.com

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

Conference'17, July 2017, Washington, DC, USA

© 2019 Association for Computing Machinery.

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM. . \$15.00

<https://doi.org/10.1145/nnnnnnn.nnnnnnn>

In [11], Thomas et al. describe this user state as a *disorientation* or that the user is *struggling*.

The occurrences of the user confusion hinder the business of organizations whose employees use a digital solution. Confusion slows down the work, frustrates the user and may lead to mistakes. It typically threatens the novice users (e.g., steep learning curves), but happens also to the experienced ones. Sometimes, the user needs to use some rarely-used feature or re-learn to use the application after a major update. In fact, the employees may be so entrenched in the ways they use a particular software, that its major update also means a major software adoption issue for the organization. This means that focusing solely on the novices onboarding is not enough; the applications have to provide the users with help and assistance when needed (but not bother them with any prompts in moments when these would be perceived as disturbing).

In this paper we leverage mouse interaction data for the prediction of user confusion in the web application. For the purpose of evaluation we conduct a user study with 60 participants on a travel agency's website, described next. We compare our results with predictors using eyetracking data. We discuss the possibility of real-time confusion prediction in production environment using machine learning models with just the mouse interaction data. We stand the hypothesis that from the mouse interaction data we are able to predict the confusion.

2 RELATED WORK

There have been just few works that deal with the automatic user confusion detection. In [8], Lalle et al. used eye tracker for the detection of user confusion. They conducted a user study with 136 participants, during which they were asked to notify about their confusion using a software button. Participants interacted with an interactive visualization to support multicriteria preferential choice. The resulting classification had a promising precision of 0.60; however, the limitation is the current low penetration of eye tracking technology. Authors proposed a method for prediction in real time by splitting the recorded data into time windows (setups with 5 and 10 seconds) where the results were similar. In [11], Thomas et al. used an approach where features derived from standard web server logs were used for the detection of struggling sessions (e.g., session duration, number of unique page visits, etc.). Although they reached the precision of 0.84, the selection of the input features makes the method difficult to use in real time, which is crucial for a timely intervention in case of detected user confusion.

In another interesting work [9, 10] Pentel and Avar focus on the confusion prediction using mouse interaction data in a computer game. The game was inspired with a advent calendar. Playground consisted of one screen filled with 24 numbered and randomly positioned rectangles from 1 to 24. The task was to clicked on each of this rectangle starting from 1 to 24 in the order. Through the game

players were asked to describe all the difficulties during finding the following rectangle. These descriptions were recorded and later manually evaluated. Based on them authors labeled confused moves during the game. They gathered 916 games together from 282 users. For the confusion prediction they used features based on number of micro direction changes during the target mouse move. They also used features based on mouse move duration. For the mouse move classification they used random forest classifier and they ended up with F1 score equals to 0,938. They claim that these method can be applied in web application for mouse movements between *html* elements.

2.1 Mouse implicit feedback

Working with the mouse interaction data we face the logs in the raw format. It is a sequence of mouse coordinates with the timestamps. To apply machine learning algorithms proper preprocessing has to be applied. One possible approach mentioned in the works [4, 5] is to split raw data into mouse moves. Gamboa et al. defined a move as a sequence of mouse coordinates where difference between each timestamp is less than 120ms. The move must consist of at least 4 records. After splitting the raw data into moves authors extract the mouse features. This approach was used in authentication of user, based on mouse interaction. This problem is very similar to ours, because in both cases we try to model user based on application independent data.

For the confusion prediction we use these mouse interaction features derived from raw data mentioned in works [4, 5].

- (1) *Horizontal velocity*. Value based on deviation of x coordinate through time $v_x = \frac{\delta x}{\delta t}$.
- (2) *Vertical velocity*. Value based on deviation of y coordinate through time $v_y = \frac{\delta y}{\delta t}$.
- (3) *Velocity*. Value based on deviation of both coordinates through time $v = \sqrt{v_x^2 + v_y^2}$.
- (4) *Acceleration*. Velocity derivation by time deviation $\dot{v} = \frac{\delta v}{\delta t}$.
- (5) *Jerk*. Acceleration deviation by time deviation $\ddot{v} = \frac{\delta \dot{v}}{\delta t}$.
- (6) *Distance*. Sum of distances through each coordinate $s = \sum_{i=1}^n \sqrt{x_i^2 + y_i^2}$.
- (7) *Duration*. Sum of moves duration $t = \sum_{i=1}^n \delta t_i$.

Features 1 to 5 were computed for each move within user data. To represent user behavior as whole means of given features were computed. Features 6 and 7 were computed as a sum to represent user behavior.

2.2 Eyetracker implicit feedback

Eyetracker interaction data are very similar to mouse interaction data. In both cases we are working with timeseries of coordinates. According to work [3], T Duchowski says that before actually computing the features we have to pre-process the raw coordinates into fixations and saccades. A saccade is reflexive move of the eye, where human is unable to read. Its duration is between 10 to 100ms. The fixation is a stable moment of eye. According to work of Duchowski, this moment is connect with some micro moves around the center of the fixation. Except these moves there are other characteristics connected with the eyetracker data such as head distance and pupil

dilatation. According to work [8], Lalle et al. showed that head distance features were the most significant features in confusion prediction from eyetracker data.

For the confusion prediction we use following features derived from computed fixations and saccades.

- (1) *Number of fixations*. Fixation count for given time interval.
- (2) *Number of saccades*. Saccade count for given time interval.
- (3) *Fixation ratio*. Fixation ratio to time interval.
- (4) *Saccade ratio*. Saccade ratio to time interval.
- (5) *Head distance*. Head distance from the eyetracker.
- (6) *Pupil deviation*. Pupil size deviation from average pupil size.

Each of the features has been mentioned in works [6, 7], where Kardan and Conati used it to model the user to determine his or her learning with interactive simulations.

2.3 Server logs

Http server is one of the standard solutions in modern web applications. Requests from client part of the applications are sent to it and processed afterwards. Most of the HTTPS servers store these information in logs. Mentioned in related work Thomas et al. used this data in work [11] to classify struggling sessions. We can use the data to classify confused sessions as well. Advantage of this approach is that collecting of this information is costless thanks to the server logs.

For the prediction we will use following features.

- (1) *Time spent*. Total time spent within prediction window.
- (2) *Total pages*. Number of unique pages visited for given window.
- (3) *Time per page*. Time spent divided by total pages.
- (4) *Cycles*. Number of cycles for given window. Cycle is an event when a user returns to url he or she already visited in the past.

It is important to note that during the experiment we did not have access to the http server logs at the web application. Mouse logs contains the *url* where the log happened. We used this *url* information to extract *HTTP* server metrics.

3 AUTOMATED CONFUSION DETECTION

To identify confused users in a web application we designed a method for automated confusion detection. We divided or problem into two scenarios. In the first one we predict the confusion based on the whole session data and in the second we try to predict confusion in real time. Considering labeled training data, the method consists of the following steps.

- (1) *Raw data preprocessing*. In this step we clean up the data. In case of the mouse interaction data we compute the mouse moves. In case of the eyetracker data we compute fixations and saccades, or we use proper filter. In case of real time prediction the data is split into the windows. Windows are collection of interaction data which are equally sized in terms of the time duration. This approach has been used in [8].
- (2) *Feature extraction*. In this step we compute features, summarised in section ?? from the preprocessed data. In case of highly imbalanced dataset, the oversampling of minority class should forego.

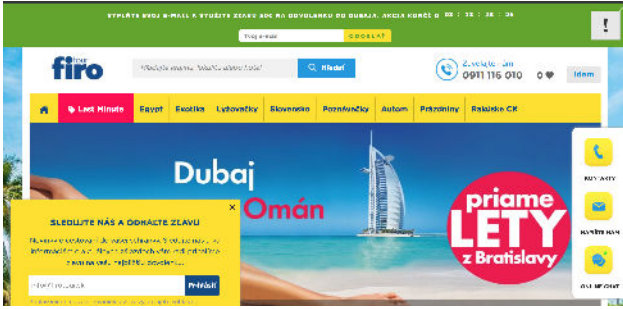


Figure 1: Confusion button (exclamation mark) in the right top corner of the screen for gathering ground truth labels. It was designed not to attract user’s attention, but to still be well visible.

- (3) *Model training.* Having all the features ready, machine learning model is being trained. For the classification we use logistic regression and random forest classifiers, because of their promising results on similar problems [8, 9, 11?].
- (4) *Prediction.* After the model being trained, the prediction is being applied on equally sized windows, in case of real time prediction, and on the whole session in the first scenario (while predicting on the session as whole). Again the same data pre-processing and feature extraction must be applied to the test data.

4 USER STUDY

To evaluate our method we conduct a study with 60 participants in the UXCenter at the Faculty of Informatics and Information Technologies at the Slovak University of Technology in Bratislava [1]. Participants solved 6 tasks at the travel agency portal Firotour². We used 20 computers equipped with the Tobii TX300 eyetracker to record eye moves, each one with frequency of 300Hz. For the recording of the mouse interaction data we use the mouse logger added directly to agency’s application with the frequency of 60Hz.

All the tasks of the study were designed with cooperation of the travel agency Firotour helpdesk. Each of the tasks represents common behavior of the application visitors. For example the participants should find the cheapest vacancy in Turkey with the departure from Bratislava, or find how far is given hotel from the beach etc.

The challenging part of the study was to record the exact moment of the participant’s confusion. We use this data as ground truth labels later in the results of this paper. In our case we modified approach used [8], where Lalle et al. used button with the text "I am confused" situated in the right top corner of the screen. Authors did the pilot studies with this button reported in work of Conati, mentioned earlier. To explain the usage of the button authors provide a descriptive text to the participants. In our case we use the same setup with the minor button text modification to just the exclamation mark. Sample screen in the figure 1 We did not want our participants to feel bad that they have some problems with the tasks. In our opinion the original text can evoke that feelings.

²www.firotour.sk

After each task participants had to answer three questions. The first was about the answer of the task they accomplished. Second was a boolean question whether they pressed the confusion button. The last question should be answered only if they pressed the confusion button. In this question we asked about the reasons of pressing. For the whole study setup we used iMotions³ software.

5 RESULTS

At the study we had 59 participants with 6 tasks each. After collecting the data minor technical issues were revealed, because of this only 336 valid records were used. The confusion button was pressed 112 times.

Before presenting results it is important to mention that we used python’s *scikitlearn* for all the computations. In each case of training of the model we used the grid-search (*GridSearchCV*) with stratified 3-fold cross validation for fine-tuning the hyper-parameters of the model. For the logistic regression we fine tuned parameters *C* (inverse of regularization strength) and *penalty* (norm in penalization). For the random forest classifier we tuned parameters: number of trees within the forest (*n_estimators*), the number of features to consider for the best split (*max_features*) and the maximum depth of the tree (*max_depth*). In the feature setup we used polynomial features where we fine-tuned the polynomial degree used. We also used *StandardScaler* for feature normalization and *SelectFromModel* with *LinearSVC* for feature selection. All presented result were computed on the test set with the stratified 3-fold cross validation. Predictions on each fold were collected and final metrics were computed upon all of them.

In the first step we tried to predict the confusion on the whole session. If the given session was confused, we used only the data before the button was pressed, otherwise the whole session data was used. In this case no oversampling had not required, because the training dataset was balanced (1:3). As the classifier we used logistic regression algorithm, because of its results on general binary classification problems. Table 1 shows the results. Here we can see, that the results using only the mouse data are comparable with methods using the eyetracker data and the http server data. For the purpose of evaluation we present a confusion matrix reached with the best (ALL) classification in the figure 2.

Table 1: Whole session classification results using eyetracker data, mouse data and http server data.

	Prec(0)	Prec(1)	Rec(0)	Rec(1)
Eyetracker	0.70	0.50	0.80	0.38
Mouse	0.80	0.60	0.86	0.40
HTTP	0.75	0.72	0.85	0.45
ALL	0.94	0.77	0.87	0.88

The second step was the real time prediction. In this case we split the data into equally sized windows. We tried various lengths, but the setup with size of 10 seconds seemed to reach the best results. Only seconds before pressing the confusion button were

³www.imotions.com

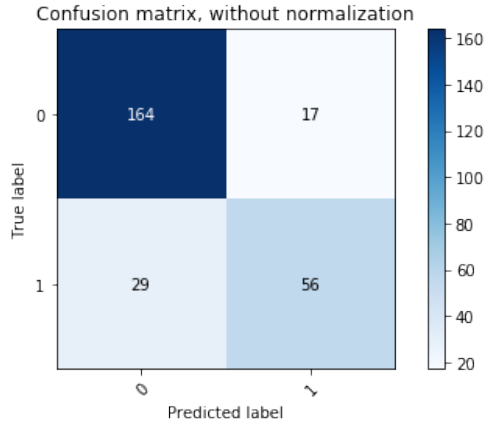


Figure 2: Confusion matrix using stratified 3-fold cross validation using logistic regression on whole session data using all features. Prediction of confused samples is marked with "1".

labeled as confused. Because of the sliding window usage on non-confused records, we ended up with highly imbalanced dataset. To overcome this problem we used *SMOTE* oversampling based on *k nearest neighbour* presented in work [2]. We used the parameter *k* of 5 for algorithm setup. We chose to over sample minority class by 200%, 500% and equal size to non-confused windows. Logistic regression and random forest have been used. Results are presented in the table 2. As we can see by increasing minority class size we are increasing recall on confusion prediction. Logistic regression reached the best results with 200% oversampling rate on minority class. For the comparison purpose we present also the confusion matrix for the best classification in the figure 3.

In the table 3 we can see results on real time prediction using 5 seconds windows and the same setup as before. As we can see by decreasing the amount of data used for feature extraction (from 10 seconds to 5 seconds) we are unable to predict confusion with decent precision. The best result we obtain were reached on the whole dataset.

Table 2: Real time classification results using only mouse interaction data and logistic regression (LR) and random forest (RF) with time window of 10 seconds and various oversampling setups using SMOTE algorithm

	Prec(0)	Prec(1)	Rec(0)	Rec(1)
LR	0.98	0.30	0.99	0.11
LR SMOTE 200%	0.98	0.36	0.09	0.09
LR SMOTE 500%	0.98	0.24	0.99	0.11
LR SMOTE 2000%	0.98	0.15	0.96	0.28
RF	0.98	0	1	0
RF SMOTE 200%	0.98	0.25	1.00	0.02
RF SMOTE 500%	0.98	0.27	1	0.07
RF SMOTE 2000%	0.98	0.08	0.95	0.25

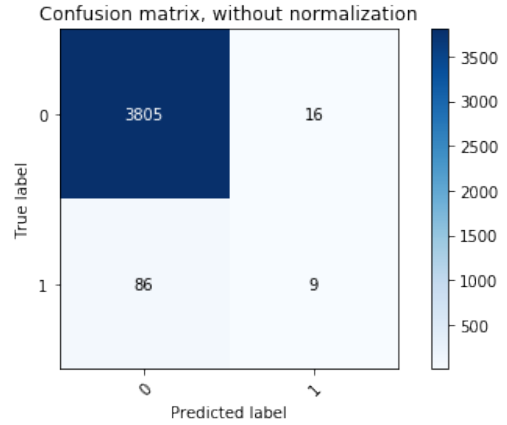


Figure 3: Confusion matrix using stratified 3-fold cross validation using logistic regression on windows of 10 second size using mouse features. Prediction of confused samples is marked with "1".

Table 3: Real time classification results using only mouse interaction data and logistic regression (LR) and random forest (RF) with time window of 5 seconds and various oversampling setups using SMOTE algorithm.

	Prec(0)	Prec(1)	Rec(0)	Rec(1)
LR	0.97	0	0.98	0
LR SMOTE 200%	0.97	0.03	0.97	0.03
LR SMOTE 500%	0.97	0.02	0.95	0.03
LR SMOTE 2000%	0.97	0.03	0.88	0.13
RF	0.97	0	1	0
RF SMOTE 200%	0.97	0	1	0
RF SMOTE 500%	0.97	0	1	0
RF SMOTE 2000%	0.97	0.02	0.97	0.02

6 CONCLUSION

In this paper we analysed confusion prediction using data gathered with a travel agency's web application *Firotour*. We compared various approaches in confusion prediction on whole session using mouse interaction, eyetracker and http server data. We also proposed a method for real time confusion prediction using only mouse interaction data. Our method uses logistic regression classifier and random forest classifier for the prediction.

Our results show that we are able to predict the confusion in the real time using only the mouse interaction data. 40% of samples predicted as confused actually were confused and 99% of not confused actually are not confused. Considering the fact that this model predicts on time windows this solution is easy to use in real production environment. We showed that by increasing the time window data used for the prediction of confusion we are able to even increase this precision.

Estimating the exact moment of user confusion is a very useful tool. We can use this information to help the user to overcome

troubles. Ideal tool in this issue are user guides with steps navigating the confused user through the web application. *Yeself*⁴ is a software tool providing these guides. Nowadays, we are working on the production module for *Yeself* solution for real time confusion prediction.

REFERENCES

- [1] Maria Bielikova, Martin Konopka, Jakub Simko, Robert Moro, Jozef Tvarozek, Patrik Hlavac, and Eduard Kuric. 2018. Eye-tracking en masse: Group user studies, lab infrastructure, and practices. *Journal of Eye Movement Research* 11, 3 (2018), 6.
- [2] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. 2002. SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research* 16 (2002), 321–357.
- [3] Andrew T Duchowski. 2007. Eye tracking methodology. *Theory and practice* 328 (2007).
- [4] Hugo Gamboa and Ana Fred. 2003. A User Authentication Technic Using a Web Interaction Monitoring System. In *Iberian Conference on Pattern Recognition and Image Analysis*. Springer, 246–254.
- [5] Hugo Gamboa and Ana Fred. 2004. A behavioral biometric system based on human-computer interaction. In *Biometric Technology for Human Identification*, Vol. 5404. International Society for Optics and Photonics, 381–393.
- [6] Samad Kardan and Cristina Conati. 2012. Exploring gaze data for determining user learning with an interactive simulation. In *International Conference on User Modeling, Adaptation, and Personalization*. Springer, 126–138.
- [7] Samad Kardan and Cristina Conati. 2013. Comparing and combining eye gaze and interface actions for determining user learning with an interactive simulation. In *International Conference on User Modeling, Adaptation, and Personalization*. Springer, 215–227.
- [8] Sébastien Lallé, Cristina Conati, and Giuseppe Carenini. 2016. Predicting Confusion in Information Visualization from Eye Tracking and Interaction Data.. In *IJCAI*. 2529–2535.
- [9] Avar Pentel. 2015. Employing think-aloud protocol to connect user emotions and mouse movements. In *2015 6th International Conference on Information, Intelligence, Systems and Applications (IISA)*. IEEE, 1–5.
- [10] Avar Pentel. 2015. Patterns of Confusion: Using Mouse Logs to Predict User's Emotional State.. In *UMAP Workshops*.
- [11] Paul Thomas. 2014. Using interaction data to explain difficulty navigating online. *ACM Transactions on the Web (TWEB)* 8, 4 (2014), 24.

⁴www.yeself.com

Príloha H - Zhodnotenie plánov práce

Náš plán práce na diplomový projekt 2 bol nasledovný:

1. *Kvantitatívna používateľská štúdia nad portálom alza*. Plánujeme zopakovať používateľskú štúdiu nad väčšou vzorkou účastníkov (do očakávaného dátumu 25.5.).
2. *Vylepšenie logera aktivity používateľa vo webovej aplikácii*. (do očakávaného dátumu 1.7.)
3. *Spolupráca s firmou yeself*. (do očakávaného dátumu 21.9.)
4. *Implementácia extrahovania dát z myši a klávesnice a algoritmu klasifikácie*. V tomto kroku sa zameriame na implementáciu automatizovanej extrakcie metrik zo surových dát o používaní myši a klávesnice. (do očakávaného dátumu 1.10.)
5. *Rozsiahlejšia používateľská štúdia*. (do očakávaného dátumu 1.11.)
6. *Analýza výsledkov používateľskej štúdie*. (do očakávaného dátumu 1.12.)
7. *Analýza dát z reálneho správania používateľov*. (do očakávaného dátumu 2.2.)

Každý zo spomenutých bodov sme stihli v predstihu. Dokonca sa nám podarilo vyhodnotiť získané dáta z pilotného experimentu. V pláne sme sa však odklonili v prípade domény používateľskej štúdie. Namiesto portálu *alza* sme zvolili dovolenkový portál *frotour*. Dôvod zmeny bol ten, že sme úzko spolupracovali s vlastníkmi portálu.

Náš plán práce na diplomový projekt 3 bol nasledovný:

1. *Vypracovanie dema na konferenciu CHI* ²⁴. Plánujeme napísať demo článok na konferenciu o aplikácii *yeself*. (do 7.1.)
2. *Vypracovanie článku na konferenciu IITSRC* ²⁵. (do 24. 2.)
3. *Hlbšie porovnanie dosiahnutých výsledkov s výsledkami článku [29]*. Plánujeme užšiu spoluprácu s autormi s cieľom porovnania dosiahnutých výsledkov. (do 28.2.)
4. *Nasadenie predikčného modlu do infraštruktúry yeself*. Integrácia natrénovaného modelu s ich softvérom. (do 30. 3.)
5. *Overenie na reálnych dátach z používania*. Po nasadení budeme monitorovať správanie zákazníkov. (do 30.4.)

V rámci tejto časti riešenia sme plánovali publikovať demo článok na konferenciu *CHI*. Bohužiaľ náš článok nebol prijatý. Na miesto toho sme podali článok na konferenciu *UMAP* ²⁶, ktorý bol prijatý a je súčasťou prílohy F. Taktiež sa nám podarilo publikovať článok na študentskej vedeckej konferencii na našej fakulte (príloha E). Okrem toho v prílohe G ponúkame návrh ďalšieho článku, ktorý plánujeme publikovať na vedeckej konferencii v blízkej budúcnosti. V rámci plánu

²⁴www.chi2019.acm.org/

²⁵www.fiit.stuba.sk/iit-src.html

²⁶www.cyprusconferences.org/umap2019/index.html

sme mali body, v ktorých sme chceli nasadiť naše riešenie do produkčnej aplikácie *YesElf*. Tento bod sa nám podarilo naplniť, náš klasifikátor je súčasťou aplikácie, avšak nepodarilo sa nám overiť jeho presnosť na produkčných zákazníkoch. Dôvodom bol zdĺhavý proces schvaľovania nasadzovania. Taktiež bolo treba celé riešenie poriadne otestovať.

Príloha I - Technická dokumentácia

V tejto prílohe uvádzame detaily k replikovaniu používateľskej štúdie a k vyhodnoteniu dát, ktoré sú súčasťou digitálneho obsahu našej práce. V prvom rade sumarizujeme zariadenia použité na získavanie dát o správaní používateľa počas používateľskej štúdie:

1. *Okulograf*. Zaznamenávanie dát o pohybe očí sme využili 20 okulografov *Tobii X2-60*, s frekvenciou 60Hz. Výstupom je *csv* súbor s pohybmi očí pre každého účastníka štúdie. Umiestnenie súborov dokumentujeme v prílohe D.
2. *Loger myši*. Na zaznamenanie pohybov myši sme použili vlastnú implementáciu logera myši, ktorá vznikla nad rámec diplomovej práce v rámci spolupráce našej fakulty a firmy *Brainware*. Loger je súčasťou aplikácie *YesElf*, ktorú sme opísali v prílohe C). Celá implementácia prebehla v spolupráci so študentom Matejom Váľkym. Výstupom boli záznamy o pohybe myši uložená v databáze *elasticsearch* umiestnené v rámci aplikácie *YesElf*.
3. *Dotazníky*. Po každej úlohe sme získavali odpovede účastníkov na otázky formou elektronických dotazníkov. Využili sme pri tom softvér *iMotions*²⁷. Výstupom je *txt* súbor s odpoveďami pre každého účastníka štúdie.

Pred prácou so skriptami na analýzu dát je potrebné nainštalovať knižnice jazyka *python*. Celý zoznam knižníc s verziami sa nachádza v koreňovom adresári v súbore *requirements.txt* priloženého digitálneho obsahu. Na inštaláciu je potrebné spustiť príkaz.

```
1 $ pip install -r requirements.txt
```

Následne je všetko pripravené na prácu so skriptami. Jednotlivé skripty sú organizované do logických celkov v každom z priečinkov a sú označené príponou *.ipynb*. Pre jednoduchšiu prácu odporúčame spúšťanie skriptov v prostredí *jupyter lab*²⁸, ktorého knižnica je súčasťou súboru *requirements.txt* a teda bola nainštalovaná. Pre spustenie prostredia vykonajte nasledujúci príkaz.

```
1 $ jupyter lab
```

Pred samotnou analýzou a trénovaním modelov odporúčame prejsť skripty použité na čistenie dát, ktoré sa nachádzajú v priečinkoch *MouseData*, *EyetrackerData* a *SurveyData*. Upozorňujeme, že vyhodnotenie dotazníkov nebolo automatizované, keďže bolo treba anotovať jednotlivé odpovede na otázky.

Ďalej pokračujeme extrakciou črt. V koreňovom adresári digitálneho obsahu sa nachádzajú 2 súbory *FeatureExtractionSession.ipynb* a *FeatureExtractionWindows.ipynb*. Obsahujú skripty použité na extrahovanie črt. Súbory sú doplnené o komentáre. Všetky skripty spojené s exploračnou analýzou extrahovaných črt, trénovaním modelov a ich vyhodnocovaním sa nachádzajú v priečinkoch:

²⁷www.imotions.com

²⁸www.jupyterlab.readthedocs.io/en/stable/getting_started/installation.html

1. *WholeSession*. Priechinok určený na skripty použité pri modeloch nad celou úlohou.
2. *Windows10secs*. Priechinok určený na skripty použité pri modeloch nad oknami veľkosti 10 sekúnd.
3. *Windows5secs*. Priechinok určený na skripty použité pri modeloch nad oknami veľkosti 5 sekúnd.

V každom z priechinok sa nachádzajú už extrahované črty (súbory s koncovkou *.csv*), skripty k exploratívnej analýze (*Analysis.ipynb*) a skripty k trénovaniu a vyhodnoteniu modelov (*Prediction.ipynb*).

Na záver chceme upozorniť na vysokú časovú a pamäťovú zložitosť v prípade priložených skriptov. Vykonávanie niektorých programov môže trvať aj niekoľko dní, keďže ide o spracovanie veľkých dát.