REPUBLIC OF TURKEY
YILDIZ TECHNICAL UNIVERSITY
GRADUATE SCHOOL OF SCIENCE AND ENGINEERING

AUTOMATED TEST SCRIPT GENERATION BASED ON HIDDEN MARKOV MODELS TRAINED ON USER BROWSING BEHAVIORS

İzzettin ERDEM

MASTER OF SCIENCE THESIS
Department of Computer Engineering
Program of Computer Engineering

Supervisor
Prof. Dr. Mehmet Sıddık AKTAS

April, 2023

REPUBLIC OF TURKEY
YILDIZ TECHNICAL UNIVERSITY
GRADUATE SCHOOL OF SCIENCE AND ENGINEERING

AUTOMATED TEST SCRIPT GENERATION BASED ON HIDDEN MARKOV MODELS TRAINED ON USER BROWSING BEHAVIORS

A thesis submitted by İzzettin ERDEM in partial fulfillment of the requirements for the degree of **MASTER OF SCIENCE** is approved by the committee on 04.04.2023 in Department of Computer Engineering, Program of Computer Engineering.

Prof. Dr. Mehmet Sıddık AKTAS
Yildiz Technical University
Supervisor

Approved By the Examining Committee

Prof. Dr. Mehmet Sıddık AKTAS, Supervisor
Yildiz Technical University _____

Prof. Dr. Oya KALIPSIZ, Member
Yildiz Technical University _____

Prof. Dr. Hasan SOZER, Member
Ozyegin University _____

I hereby declare that I have obtained the required legal permissions during data collection and exploitation procedures, that I have made the in-text citations and cited the references properly, that I haven't falsified and/or fabricated research data and results of the study and that I have abided by the principles of the scientific research and ethics during my Thesis Study under the title of Automated Test Script Generation Based on Hidden Markov Models Trained on User Browsing Behaviors supervised by my supervisor, Prof. Dr. Mehmet Sıddık AKTAS. In the case of a discovery of false statement, I am to acknowledge any legal consequence.

İzzettin ERDEM

Signature

This study was supported by the Scientific and Technological Research Council of Turkey (TUBITAK) under project id: 3191534.

Dedicated to my family

ACKNOWLEDGEMENTS

First, I would like to express my gratitude to my advisor, Assoc. Prof. Dr. Mehmet Sıddık Aktaş, for his patience, humility, support, and interest throughout my entire education process and for his significant contributions and support in the publications we produce. The academic methodology and scientific perspective methods I learned from him helped me overcome the issues I struggled with during my thesis process.

In addition, I would like to thank my colleagues with whom we worked together during the whole studying period, and I would like to thank my family and friends who have supported me throughout my life.

I am grateful to TUBITAK for supporting this research (under project id: 3191534) and Saha Bilgi Teknolojileri Egitim Danismanlık San. Tic. A.S. R&D Center for providing the computational materials that made this study achievable.

İzzettin ERDEM

TABLE OF CONTENTS

LIST OF SYMBOLS viii

LIST OF ABBREVIATIONS ix

LIST OF FIGURES x

LIST OF TABLES xi

ABSTRACT xii

ÖZET xiii

1 INTRODUCTION 1

1.1 Abstract 1

1.2 Literature Review 2

1.3 Objective of the Thesis 4

1.4 Purpose 4

1.5 Hypothesis & Contributions 5

1.6 Organization 5

2 RESEARCH PROBLEMS 6

3 PROPOSED METHODOLOGY 7

3.1 Crawling Module 7

3.2 Encoding-Rules Generation Module 7

3.3 Sessionization Module 8

3.4 Hidden Markov Model Module 9

3.5 Unseen Data Sequence Generator Module 10

3.6 Decoding Module 10

3.7 Test Script Generator Module 11

3.8 Selenium Plugin Module 11

4 PROTOTYPE IMPLEMENTATION 12

4.1 Datasets 12

4.1.1 Test Crawl Datasets: 12

4.1.2 Session-based User-activity Datasets: 13
4.2 Test Design 14
4.2.1 Test 1 15
4.2.2 Test 2 16

5 EVALUATION **17**
5.1 Dataset 17
5.2 Experimental Study Design 18
5.3 Evaluation Results 19
5.3.1 Results of Test 1 19
5.3.2 Results of Test 2 20
5.4 Threats to Validity of Results 21

6 CONCLUSION **22**

REFERENCES **24**

PUBLICATIONS FROM THE THESIS **28**

LIST OF SYMBOLS

N	number of words
w_i	i^{th} word in the current sentence

LIST OF ABBREVIATIONS

CPU	Central Process Unit
GPU	Graphics Processing Unit
HMM	Hidden Markov Models
UI	User Interface
URL	Uniform Resource Locator

LIST OF FIGURES

Figure 3.1 Encoding-Rules Generation Process 8
Figure 3.2 Sessionization Process 9
Figure 3.3 Overview of the Hidden Markov Model 10
Figure 3.4 Decoding URL Sequences by Using Viterbi Algorithm 11
Figure 4.1 Evaluation of Test-1 15
Figure 4.2 Evaluation of Test-2 16
Figure 5.1 Levenshtein Distance Formula 19
Figure 5.2 Edit Distances. In each chart, y-axis means shows the percentage in the interested dataset, x-axis means specifies the edit distance value. 21
Figure 5.3 Bleu scores. In each chart, y-axis means shows the percentage in the interested dataset, x-axis means specifies the bleu scores. . . . 21

LIST OF TABLES

Table 4.1 Properties of Session-based User Activity Dataset 14
Table 5.1 Differences between actual and generated URL Sequences 19

ABSTRACT

Automated Test Script Generation Based on Hidden Markov Models Trained on User Browsing Behaviors

İzzettin ERDEM

Department of Computer Engineering
Master of Science Thesis

Supervisor: Prof. Dr. Mehmet Sıddık AKTAS

User interface (UI) testing is a necessary process to evaluate whether the developed web applications meet the software requirements defined by the end users. This thesis proposes an e-business workflow that automates tests performed on user interfaces using Hidden Markov Models. The recommended workflow is designed to generate test scripts based on user navigation behaviors.

In this regard, a prototype application was developed to demonstrate the usability of the proposed workflow, and experimental studies were conducted to measure the quality of newly created test scripts to facilitate testing of this prototype application. In these studies, the similarity between the navigation behaviors performed by the users and the newly created navigation behaviors was analyzed.

In the results we obtained throughout the study, we discovered that the workflow we proposed can generate new and unprecedented user navigation behaviors with high similarities to real user navigation behaviors. This shows that the proposed system can generate new test scripts that can provide comprehensive testing of web applications.

Keywords: Automated UI testing, comprehensive testing, Hidden Markov Models, Viterbi algorithm

ÖZET

Kullanıcı Gezinti Davranışları ile Eğitilmiş Gizli Markov Modellerine Dayalı Otomatik Test Betiği Üretimi

İzzettin ERDEM

Bilgisayar Mühendisliği Anabilim Dalı
Yüksek Lisans Tezi

Danışman: Prof. Dr. Mehmet Sıddık AKTAŞ

Kullanıcı arayüzü (UI) testi, geliştirilen web uygulamalarının son kullanıcılar tarafından tanımlanan yazılım gereksinimlerini karşılayıp karşılamadığını değerlendirmek için gerekli bir süreçtir. Bu tez, Gizli Markov Modellerini kullanarak kullanıcı arayüzlerinde gerçekleştirilen testleri otomatikleştiren bir e-iş iş akışı önermektedir. Önerilen iş akışı, kullanıcı gezinti davranışlarına dayalı olarak test betik dosyaları oluşturmak için tasarlanmıştır.

Bu minvalde önerilen iş akışının kullanılabilirliğini göstermek için bir prototip uygulama geliştirilmiştir ve bu prototip uygulamanın test edilmesini kolaylaştırmak için yeni oluşturulan test betiklerinin kalitesini ölçen deneysel çalışmalar yürütülmüştür. Bu çalışmalarda kullanıcıların gerçekleştirdiği gezinti davranışları ile yeni oluşturulan gezinti davranışları arasındaki benzerlik analiz edilmiştir.

Çalışma boyunca elde ettiğimiz sonuçlarda önerdiğimiz iş akışının, gerçek kullanıcı gezinti davranışlarıyla yüksek benzerliklere sahip yeni ve daha önce görülmemiş kullanıcı gezinti davranışları üretebildiğini keşfettik. Bu da önerilen sistemin web uygulamalarının kapsamlı testini sağlayabilecek yeni test betiklerini oluşturabildiğini göstermektedir.

Anahtar Kelimeler: Otomatik kullanıcı arayüzü testi, kapsamlı test, Gizli Markov Modelleri, Viterbi algoritması

YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

1
INTRODUCTION

Today, people's use of technology is increasing in direct proportion to the development of technology. This increase has also increased the size of the user's behavioral data stored on the web. This web usage data is now generally referred to as "big data" since it has the following three properties: volume, velocity, and variety. Here, variety refers to the different structures of the web usage data that can be found in the log files, volume refers to the size of the web usage data, and velocity refers to how fast such user-web interaction data grows. In this study, we investigate the ways of extracting knowledge from web usage data to understand better how users interact with web applications.

Launching a real-time application on the web and storing and controlling data on this application is a difficult task. E-commerce websites are the systems where people interact with the system with high frequency. Having an automated test framework that can be applied to such systems can reduce maintenance costs and difficulties. Naturally, unexpected results may occur if the systems are taken to the production environment without being tested. Therefore, testing every system following the software development life cycle has been adopted as a suitable method in the software industry.

1.1 Abstract

As the users interact with the front-end of a web application, click-stream data is generated and recorded in the log files to trace the user activities. Web-based applications are tested for User Interface functionalities to identify and detect errors and bugs that users may encounter during their interactions with the application. Previously introduced User Interface testing methods focused on writing unit tests and service tests that were utilized to carry out the testing. However, automated test execution tools for web applications such as Selenium and Cucumber emerged in recent years. In this study, we propose an e-business workflow that complements

these tools by generating the testing scripts in an automated way.

By utilizing the tools like Selenium or Cucumber, the UI testers usually test the browsing patterns that frequently occur on the websites. Here, users' visit sequences on the websites are used to create specific test patterns. There exist studies that identify the frequent patterns in the browsing data in the form of data sequences [1]. These studies utilize the data sequences, i.e., URL sequences, to generate test scripts. However, they do not focus on generating unseen (new) data sequences that need to be tested. In this study, we focused on designing and implementing automated testing workflows that can generate test scripts to test unseen URL sequences.

1.2 Literature Review

Hidden Markov Model is a statistical model consisting of observation and latent state variables, initial state distributions, transition, and observation distributions[2]. HMM is generally used in speech recognition and bioinformatics[3]. The Viterbi algorithm is a computationally efficient technique used to determine the most probable path in a Markov chain. According to Andrew J. Viterbi, the oldest application for which the algorithm was first proposed is the maximum likelihood decoding of convolutionally encoded digital sequences transmitted over a noisy channel [4] [5]. This study uses the Viterbi algorithm to find the most likely URL sequence path, namely Viterbi Path, available on the HMM for a given topic sequence.

In this study, it has been seen that different methods have been used to generate UI test scripts. One of these is reinforcement learning. Juha Eskonen et al. argue that the system they propose in their study can learn from scratch with less data and therefore is more effective than supervised learning algorithms[6].

There are studies in the literature to learn about user behaviors using click-stream navigation data [7] [8]. These studies aim to represent user behaviors with specific patterns and show users with similar patterns in common clusters with supervised learning techniques. In addition, it is seen that there are studies to represent user click-stream data and to find hidden sequences by using unsupervised learning techniques [1]. However, these studies are not in a format that can be used in UI testing. In our study, unlike these studies, we obtain unprecedented URL sequences from click-stream sequences made by users and recommend an end-to-end automated UI testing framework.

In our previous study, we presented a way to group the URLs obtained from the log files of a website [9]. The previous study used unsupervised learning algorithms to group

URLs. We also worked on a study in which we clustered URL sequences extracted from the log files [10].

Today, many applications are created based on the web. There are many studies in the literature for web-based tests on these applications and tools [11]. Selenium is one of the most common tools used in these studies. Wang et al. have developed a framework where testers can create and run these tests on different web browsers and test applications [12]. In their study, Arya et al. prepared a framework for testers to prepare test scripts with keywords. They stated that converting the keyword list prepared using Microsoft Excel into test scripts offers an environment where testers can test their applications even though they do not know any programming language [13]. Although these studies provide convenience to the testers, the testers still decide which test scripts to run. Unlike these studies, we offer a framework that automatically generates test scripts and eliminates the need to write test scripts specified in previous studies by testers.

There are studies in different domains using user behavior data. Yang et al. proposed an equivalent algorithm to the Page Rank algorithm and analyzed web links through user behavior using a semi-Markov model [14]. Some studies produce recommendation systems based on user browsing data. Peska et al. have proposed an e-shop recommendation system called UPComp using user browsing data [15]. Xu et al. have proposed a collaborative Web recommendation framework that extracts connections between user sessions and topics using Latent Dirichlet Allocation [16]. Unlike these studies, we use user data to create test scripts.

User behavior can be analyzed using a HMM. Kawazu et al. used clickstream user data in their study. They created the transition and emission matrices and clustered the user data in the matrix with this data [17]. Chuang et al. have proposed designing and implementing a GUI test case generation mechanism powered by user behavior. They proposed a User Computer Interaction Hidden Markov Model (UCIHMM) to model user behaviors. They designed the HMM as a model that can distinguish user behaviors and calculated the most likely user behavior for a given observation using the Viterbi algorithm [18]. Unlike these studies, we used HMM to generate URL sequences from unseen topic sequences.

There exists studies focusing on software quality and testing [19–23]. Different from these studies, in the thesis we mainly focus on designing and creating automated test script generator frameworks.

There exist studies that focus on designing distributed software infrastructures for varying purposes in different domains such as metadata management [24],

information services [25, 26], provenance data management [27–29], SOA based computational science environment [30–32], wheather forecasting modeling [33], Internet of Things [34].

Unlike these studies, we only focus on designing distributed workflow software designed for UI testing. Also, some studies focus on test case prioritization [35, 36]. However, we leave out the prioritization of test cases for future studies. We focus on a framework designed for testing UIs.

1.3 Objective of the Thesis

Our aim in this study is to create a framework that can perform end-to-end UI tests on websites. It is one of our primary goals to perform these tests using navigation URLs that have not been encountered before but are likely to be encountered. To achieve this goal, a prototype application testing the proposed e-workflow has been developed, and the results have been analyzed. Looking at these results, we can say that the proposed methodology works and produces successful results.

Motivation: The test structure and processes are critical in the field of software, and we find the studies carried out to increase the quality of the tests very important. There are two basic structures in a web application; backend and frontend. Just as it is vital to test the developments made on the backend, it is also essential to test the developments made on the frontend, because the primary place where users interact is the UI. Although UI tests are partially automated through prepared scripts, an end-to-end automated UI testing process is required. But more than the automated testing process is needed; proper test scripts must be created.

Our motivation is to create a flow that can automatically generate test scripts and automate testing processes to ensure UI testing happens quickly and accurately. In this flow, we aim to automatically produce test scripts and create a plugin for frameworks such as Selenium.

1.4 Purpose

Creating UI tests brings additional effort and cost to software teams. For the UI to be properly tested, test scripts must be created and adequately run using URLs within the website. At the same time, these tests should have a high inclusiveness, and the tested navigation sequences should have a high probability of occurrence. We have prepared a workflow to automate this testing process from start to finish to close the gap we see in the literature and find solutions. We generate unprecedented URL sequences based

on users' navigations and propose a process flow that tests these sequences.

1.5 Hypothesis & Contributions

Our work proposes a UI testing framework that automates testing processes and generates new and unprecedented test sequences for any website using users' navigations. With this proposed framework, any web UI can be tested thoroughly, and even possible sequences that users in the future can perform can be determined and included in the testing processes.

1.6 Organization

Chapter 2 describes the research problems we focused on, while Chapter 3 presents the proposed methodology. Chapter 4 details the prototype of the proposed methodology. Chapter 5 represents the evaluation metrics and the results of the proposed framework. Finally, chapter 6 concludes the thesis.

2
RESEARCH PROBLEMS

In this study, we address the research challenges related to the methodologies that can understand the user behavior from browsing data and then use this information to generate test scripts for unseen user behavior automatically. To this end, we outline the research questions for this study as follows:

1. What is a design of an e-business workflow that can generate unseen URL sequences and UI test scripts based on the knowledge extracted from users' browsing data?
2. How can we generate URL sequences from topic sequences utilizing Hidden Markov Models?
3. What is a good methodology to evaluate the success of URL sequence generation?

To address the research questions above, we propose an e-business workflow that can be used to generate IU test scripts in an automated way. The contributions of this study are as follows:

We propose an e-business workflow as an end-to-end automated UI test script generation tool. We discuss how to represent users' URL sequences as topic-based data sequences and generate unseen data sequences based on the knowledge extracted from users' browsing behaviors. We demonstrate using Hidden Markov Models (HMM) to map topic-based unseen data sequences to URL sequences.

We implemented a prototype application to demonstrate the proposed workflow's usability. We applied the prototype to browsing data collected from the log files of an e-commerce website. We introduce an evaluation methodology that can be used to understand the quality of newly generated unseen URL sequences. The results show that the workflow can generate new and unseen user-browsing behaviors that have high similarities with the actual user-browsing behaviors.

3
PROPOSED METHODOLOGY

This study introduces an e-business workflow that can generate automated UI test scripts from browsing graph data. The proposed workflow is an end-to-end UI test script generation tool. The proposed methodology generates unseen URL sequences, which are, in turn, used in generating test scripts. The proposed e-business workflow consists of 9 modules. This workflow receives browsing graph data (i.e., session-based, click-stream-based URL sequences) as an input and produces the test scripts as the output. The proposed workflow can produce test scripts that can be used as an input for automated test execution tools such as Selenium and Cucumber. The following subsequent sections give the details of each module.

3.1 Crawling Module

In order to understand the users' behavior on a website, there is a need to identify the categories of the web pages within that site. To do this, we implemented a crawler, which crawled the web pages from an e-commerce website. This module takes the URL of the e-commerce website. By crawling the web pages within the website, the proposed system obtains the content of all the web pages and the structure of the website. The crawled data is used as an input to the next module in the proposed workflow.

3.2 Encoding-Rules Generation Module

By utilizing this module, we create the mapping rules that can map each URL obtained within the crawl data to its corresponding topic. This module takes the crawl data as the input and produces regex rules as its output. Here, we identify regular expression (regex) based rules that can be used to map each URL to a topic. Figure 3.1 illustrates some example URLs and their corresponding topics.

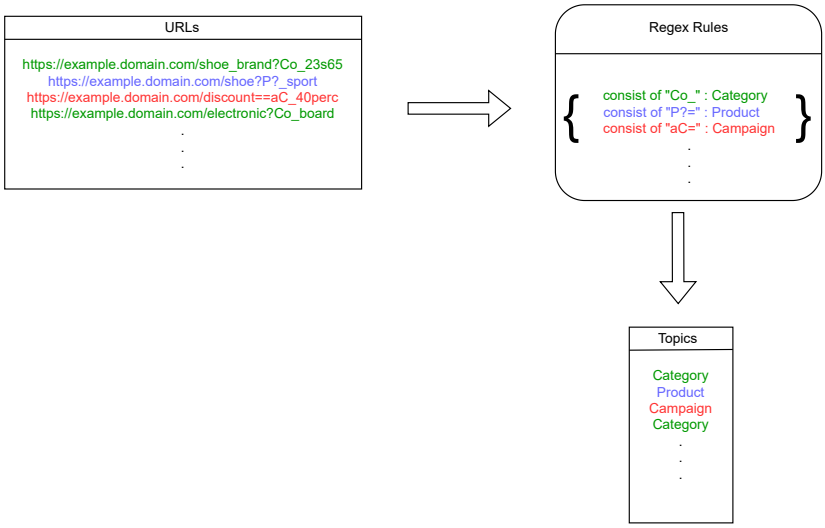


Figure 3.1 Encoding-Rules Generation Process

3.3 Sessionization Module

Sessionization is the process of segmenting the user activity record of each user into sessions, each representing a single visit to the site. The goal of a sessionization is to reconstruct, from the clickstream data, the actual sequence of actions performed by one user during one visit to the site. This module is responsible for creating topic sequences from initial URL sequences. The URLs sequences are mapped to more understandable topic sequences using the regex rules obtained in the previous step in the workflow. Fig 3.2 illustrates an example session data and URL sequences. This module takes the user-browsing data as its input and produces URL sequences, each corresponding to a session.

The sessionization steps we have done on the URL sequences dataset are as follows:

1 Preprocess data by removing unnecessary columns (i.e xpath, OS type, cursor location)
2 Filter only the URLs that the user clicked in the resulting sequences and then remove the type field.
3 Sort the data obtained as a result of filtering by timestamp, group by user and remove the timestamp field.
4 Finally, map each URL to topics using regex rules and create user topic activity sequences.



Figure 3.2 Sessionization Process

3.4 Hidden Markov Model Module

The transition and emission matrices required for the HMM are calculated in this module. First, the user URL navigation sequences obtained in the data preprocessing module and the user topic sequences obtained in the user action sequences module are used. Second, a dictionary is created from the unique data in the URL sequences, a number is assigned to each URL, and the same process is applied to the topic sequences. The transition and emission matrices are initialized using these dictionaries. If N is the total count of unique URLs and M is the total count of unique topics, the size of the transition matrix to be created will be N*N, and the size of the emission matrix will be M*N.

Here, URLs represent hidden states, and topics represent observation states. URL transitions for each user's navigation are examined, the number equivalents of these URLs in the dictionary are determined, and the value of the cell corresponding to these numbers in the transition matrix is increased by 1. Afterward, the transition matrix is created by calculating the probabilities of the values in the matrix.

Similarly, the URL navigations of the users are examined, and the related topics are found. The number values corresponding to these URL-topic binaries are found in the dictionaries, and the cell corresponding to these numbers is increased by 1 in the emission matrix. Then the emission matrix is created by calculating the probabilities of the values in the matrix. Finally, the stationary distribution is calculated using the transition matrix. This module takes URL and topic navigation sequences made by real users as input and produces an HMM module as output. The HMM creation steps

can be followed in Figure 3.3.

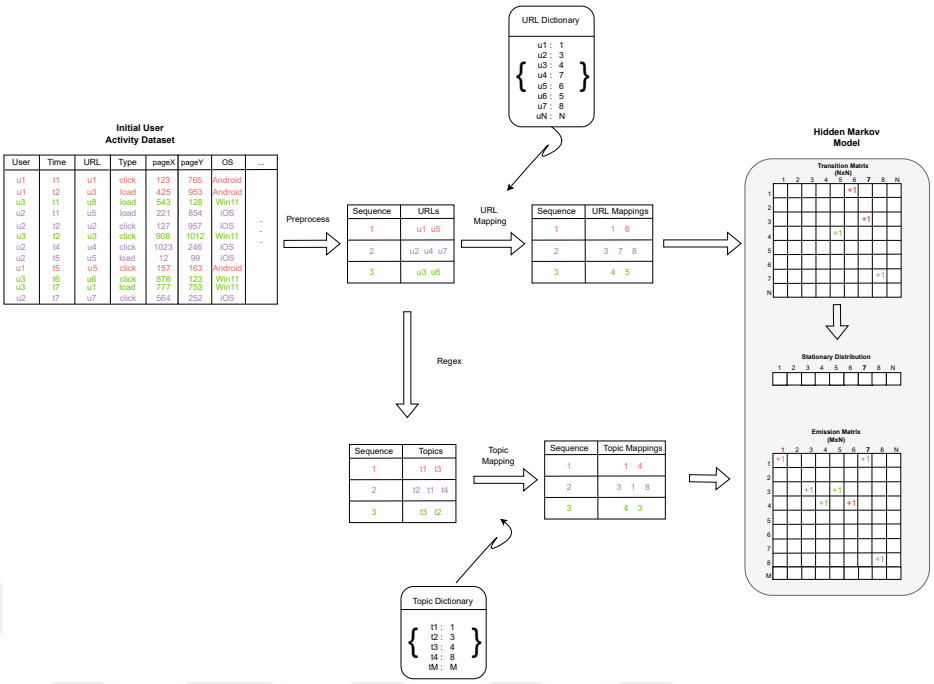


Figure 3.3 Overview of the Hidden Markov Model

3.5 Unseen Data Sequence Generator Module

In our earlier study [37], we implemented a module that can generate unseen data sequences by utilizing generative deep learning algorithms. This study uses our previous implementation as the Unseen Data Sequence Generator Module. The input to this module is the data sequences that actually happened in the browsing graph data. Moreover, the output is the unseen data sequences that are generated by the generative deep learning algorithms.

3.6 Decoding Module

This module is responsible for creating URL sequences from the given topic sequences by using Viterbi algorithm. The Viterbi algorithm finds the most probable hidden sequence for the given observation sequence by maximizing the finite probability, and it was used to calculate the path that can be reached from these inputs with the highest probability. This module generates URL sequences as output for the given topic sequences by using the probabilities in the HMM model and Viterbi algorithm.

HMM and Viterbi's primary purpose is not to create new artificial URL sequences but to find the most suitable URL sequences for newly released sequences.

In our study, the observation sequences were the topic sequences we produced with GAN; artificial sequences were URL sequences that are decoded with Viterbi and produced as output. Figure 3.4 shows a brief explanation of generated topic sequences' decoding process by using the Viterbi algorithm. Details of the Viterbi algorithm can be viewed at [2].

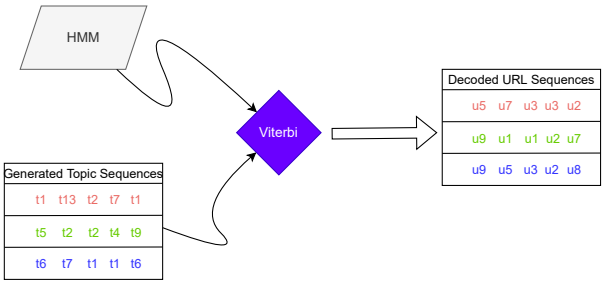


Figure 3.4 Decoding URL Sequences by Using Viterbi Algorithm

3.7 Test Script Generator Module

This module is responsible for creating test scripts from the URL sequences. The input to this module is the URL sequences that are unseen in the browsing data; the output of this module is the test scripts that can be used as an input to the test execution tools. In this study, we utilize the Selenium tool as the test execution tool. The URL sequences obtained from the previous step in the workflow were converted into a specific format and made to be input to tools such as Selenium.

3.8 Selenium Plugin Module

This module is responsible for executing the test scripts generated in the previous step of the workflow. The test scripts, which are the output obtained from the previous module, are executed using the Selenium tool in an automated way. This enables the automated test execution of the test scripts generated by the proposed workflow.

4
PROTOTYPE IMPLEMENTATION

In this study, we developed a prototype implementation to show the usability of the proposed e-business workflow in Python programming language [38]. In the implementation, we used several python based open source libraries. We used scrapy [39] (version: 2.5.1) library in the crawling module; regex [40] (version: 2021.4.4) library for regular expression based rules in encoding rules generation module and sessionizatin module; numpy [41] (version: 1.21.2) and pandas [42] (version: 1.2.4) libraries for matrix calculations and data frames in HMM module, Decoding Module, Test Script Generator Module; and pickle library for data serialization. We used a virtual machine on the cloud with a powerful GPU to generate the topics and a workstation with a 12-core CPU to calculate probabilities for the HMM.

4.1 Datasets

We worked on three different e-commerce websites' datasets to implement our proposed workflow. The first of these is an e-commerce site selling mother and baby products. The other two sites are Loadium and Testinium datasets, which also belong to Testinium company. We also used two different datasets for each web site, the dataset that obtained by crawling website and the dataset that contains users navigation sequences.

4.1.1 Test Crawl Datasets:

In order to implement the proposed e-business workflow, we first crawled the e-commerce websites that mentioned above. In turn, this enables us to extract the site's tree structure and URLs that are available within the website domain. We tagged each URL with its related topic. Here, we tagged the URLs which we could not put in any category, with the 'other' and 'unknown' tags. The categories/topics are shown below.

On Mother&Baby products web site, we obtained a total of 14 different topics.

- /product/, /category/, /search/, /category_brand/, /corporate/, /check-out/orderConfirmation/id/, /my-account/order/id/, /my-account/edit-address/id/, /discount/, /campaign/, /province/shopping_center/, /province/, /home/ and other

On Loadium web site, we obtained a total of 19 different topics.

- /test_update/, /home/, /test_jmeter/, /account_user/, /test/, /test_run/, /test_new/, /test_report/, /running-tests/, /test_detail/, /projects/, /account_payment/, /test_public-report/, /compare-test/, /test_project/, /code/, /payment-status/, /unknown/

On Testinium web site, we obtained a total of 12 different topics.

- /scenario/, /plan/, /report/, /home/, /project/, /automated/, /create/, /search/, /account/, /user/, /public_report/, /unknown/

4.1.2 Session-based User-activity Datasets:

In this study, we also obtained web usage log files from the e-commerce websites from which we obtained the crawl datasets. We extracted the session-based user-activity dataset using the web usage log files. This dataset contains information such as the URL that each user navigates on the website, type of access to this URL, timestamp of the navigation, session ID of the user, and structure of other components on the page. Here, we created the final version of the user-activity dataset by utilizing four different fields (i.e. 'session-id', 'timestamp', 'URL', and 'type' fields) from the initial user-activity datasets used in the prototype implementation. Within these datasets, there exist user sessions with different sequence lengths.

Table 4.1 shows the information about user-activity datasets after the preprocessing steps. For the first dataset, we used 25,000 of 165,000 user sequences because of two reasons, first is to solve resource problems, and the second is to check results to get if our framework can simulate future users or not.

We created two different dictionaries from this datasets, as shown in Figure 3.3. In the increasing order, we assigned integer values to each unique URL and unique topic. We used these dictionaries to calculate the HMM probabilities.

Table 4.1 Properties of Session-based User Activity Dataset

Dataset	# Sessions	# Unique URLs	# Unique Topics	# Total URLs
Mother & Baby Products	25,000	44,952	14	875,360
Loadium	16,762	6,803	19	134,030
Testinium	32,957	23,212	12	536,456

4.2 Test Design

To implement the workflow we propose, we need the structures of the websites whose data we will use. By using a crawler, we can navigate websites and reveal the structure of the website. Also, crawlers can be used to sample some of the trips a user might take on that website. In our study, scrapy was used to crawl related websites and URLs were found along with the structure of the websites.

In addition, the navigations made by real users were conveyed to us by websites. Both the crawled data and the data given to us by the websites have been made ready for use by subjecting some pre-processes. Also, we conducted a test using two methods consisting of different steps to better analyze our proposed workflow's performance. All of these steps are summarized below:

- Determine the parts of the data that are advanced by click,
- Determine regexes,
- Apply the regex rules for the user action sequence,
- Calculate the transition, emission matrices, and stationary distribution required for the HMM using user action sequences,
- Test-1) After finding the most common 500 URL sequences in initial user action sequences;
  - Find their topic counterparts by using regex module
  - Produce URL sequences with these topic sequences by using HMM+Viterbi
  - Compare URL sequences which generated by Viterbi with the original ones
- Test-2) Produce topic sequences by using GAN method which is explained in our previous study [37];
  - Produce URL sequences with these topic sequences by using HMM+Viterbi

  - Calculate Edit Distances between URL sequences which generated by Viterbi with the URLs in initial user action dataset in topic basis

- Prepare test scripts that can be used in tools such as Selenium and Cucumber as it progresses.

After these steps, URL sequences produced were converted into test scripts and used on test tools such as Selenium or Cucumber.

4.2.1 Test 1

For each dataset, transition and emission matrices were created using user-action sequences to calculate the probability matrices and distributions required for the HMM. The stationary distribution was calculated over the created transition matrix. Before proceeding to the URL sequence generation step, the most frequently topic sequences among the user-action sequences were found, and new URL sequences were obtained from these topic sequences by using the HMM and Viterbi algorithm. A one-to-one comparison was made between these obtained URL sequences and the URL sequences whose belongs to most frequent topic sequences , and the different ones were marked.

Since the Viterbi algorithm guarantees us to find the most probable path, the differences between URL sequences mean that there are more possible URL sequences that represent the topic sequences of the URL navigations users have taken so far. Therefore, future users are more likely to proceed through these URL sequences. The importance of this inference for us is that the test scripts can be produced using the Viterbi algorithm and HMM to represent future user behaviors so that the generated test scripts can enable us to anticipate the problems we may encounter in the future. Our results show us the success of HMM and Viterbi algorithm in a more concrete way. Figure 4.1 briefly shows the testing operation.

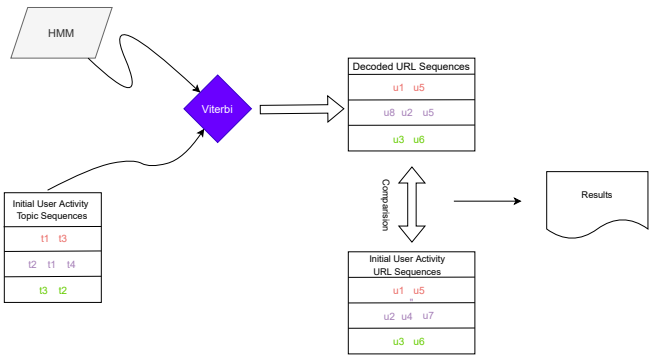


Figure 4.1 Evaluation of Test-1

4.2.2 Test 2

As in Test 1, after calculating the necessary probabilities for HMM, for each dataset, we produced unseen topic sequences by using GAN. We gave these topic sequences to HMM and Viterbi as input and produced URL sequences. After generating these URL sequences, we obtained their corresponding topic sequences using the regex module and compared them with the sequences performed by real users. However, the comparison here was made using topic sequences, unlike Test 1. Although switching from URL sequences back to topic sequences at first sight may sound complicated, we argue that the URLs and the topic equivalents of these URLs represent the same sequences for us. Therefore, as a different approach, we aim to compare and analyze topic sequences.

First, we compared these topics by calculating Edit Distance between produced topics and the initial topics. The purpose of using Edit Distance is to calculate how many different changes we need to make from the topic sequences we produce to the sequences made by real users. Second, we compared these topics by calculating Bleu Score between produced topics and initial ones.

The result of these calculations between topics give information about how much the URL sequences we produce are related/similar to the real sequences. Figure 4.2 briefly shows the calculation process as pseude code.

```
evaluation_test_2 (generated_topic_sequences, initial_topic_sequences)
    bleu_scores = []
    edit_dists = []

    for seq in generated_topic_sequences:
        bleu_scores.append(most_similar(calc_bleu_score(seq, initial_topic_sequences)))
        edit_dists.append(most_similar(calc_edit_dist(seq, initial_topic_sequences)))

return bleu_scores, edit_dists
```

Figure 4.2 Evaluation of Test-2

5
EVALUATION

We conducted an experimental study to test the prototype implementation and demonstrate our proposed workflow's usability. In the results we obtained within the scope of this study, Bleu Score and Edit Distance have been used as metrics to evaluate the similarity and convergence between the input user activities and output new/unseen user activities. Detailed results of the experimental study have described below.

5.1 Dataset

We used three different e-commerce websites to implement our proposed workflow. For each website, we have obtained the data we need in 2 different ways in order to implement the prototype. First, we gathered the URL tree structure of the websites using a crawler. During this process, we only crawled up to 7 nodes to avoid being blocked by websites for security reasons and not to increase the complexity of the tests we will perform. By examining this data we obtained, we listed all possible topics corresponding to each URL and created the regex rules that enabled this conversion. In addition, using this data, we produced new topic sequences with the method we suggested in our previous study[37].

Since we aim to generate new test scripts based on user behavior, we need more than the crawl dataset. That is why we used their data by collaborating with a website that sells motherbaby products. In this data, the navigations made by a certain number of users are available in detail, and this is the primary dataset we use. Meaningful sequences were extracted from the dataset and used by making the necessary preprocessing.

5.2 Experimental Study Design

To illustrate the testing of the system, we designed an experimental study. We compared the input URL sequences (obtained from the initial user-activity dataset) and the output URL sequences (generated by the prototype implementation to be used in test script generation). We analyzed the similarity between the actual URL sequences and generated URL sequences on topic basis by using the two similarity metrics: Bleu Score and Levenshtein metrics.

A good test script generation tool should create test scripts that can be used to test the unseen user activities, while these unseen user activities should reflect the similar user behavior that can be observed in the actual user-activity dataset. To this end, we examine whether generated URL sequences represent the same behavior as the input URL sequences.

Evaluation Metrics There are many computational metrics by which we can calculate similarity or distance. We chose two of these metrics to calculate the similarity between the actual user navigation sequences and the sequences we generated; Bleu Score and Levenshtein Distance.

We use the Bleu Score metric [43] to evaluate the similarity between the generated and actual URL sequences. The Bleu Score metric values can range from 0 to 1. Results close to 0 mean that the conversion between sequences is not very good, while results close to 1 mean that it is good. We calculated the Bleu Score metric value by comparing/contrasting the real URL sequences and newly generated URL sequences. The Bleu Score metric is be calculated with the following equation:

$$BLEU_c = \prod_{i=1}^{N} P(i)^{w_i} \tag{5.1}$$

where $\sum_{i=1}^{N} w_i = 1$ and c is the n-gram order

We also use the Levenshtein distance metric to explore the similarity between the generated URL sequences and actual URL sequences [44]. Figure 5.1 shows the calculation formula of Levenshtein distance.

$$\mathrm{lev}_{a,b}(i,j) = \begin{cases} \max(i,j) & \text{if } \min(i,j) = 0, \\ \min \begin{cases} \mathrm{lev}_{a,b}(i-1,j) + 1 \\ \mathrm{lev}_{a,b}(i,j-1) + 1 \\ \mathrm{lev}_{a,b}(i-1,j-1) + 1_{(a_i \neq b_j)} \end{cases} & \text{otherwise.} \end{cases}$$

Figure 5.1 Levenshtein Distance Formula

5.3 Evaluation Results

As stated in the previous sections, we ran two different tests to measure the success of our proposed workflow. In our first test, the 500 most repetitive sessions were selected, and their topic sequence equivalents were found. URL sequences were obtained by decoding these topic sequences with Viterbi and HMM, and these URL sequences were compared with 500 URL sequences selected at the beginning. In our second test, to generate new topic sequences, the method we suggested in our previous study has been used, and to the similarity of the URL sequences obtained by decoding these topic sequences by using Viterbi with the initial user action URL sequences has been calculated. Test results were calculated using two different metrics. The first is Bleu Score, and the second is Levenshtein (Edit) Distance.

Our results show that our proposed workflow can successfully generate new URL sequences. Since these sequences can be converted into test scripts, test processes can be automated using a test automation tool such as Selenium or Cucumber.

5.3.1 Results of Test 1

It has been mentioned in the previous sections how the HMM was created from the dataset consisting of sessions, including URL sequences and navigations made by real users. In our first test, the 500 most repetitive topic sequences made by users were found, URL sequences were obtained by decoding these topic sequences with Viterbi and HMM, and 500 decoded URL sequences compared with the URL sequences which are navigated by users. Since the Viterbi algorithm guarantees us the most probable path, 500 decoded URLs can be considered the most probable sequences corresponding to the given topic sequences. The result obtained as a result of the test is as follows.

Table 5.1 Differences between actual and generated URL Sequences

	Mother & Baby Products	Loadium	Testinium
Same	95	275	213
Different	405	225	287
Differences (%)	81	45	57

Using Viterbi with HMM, it has been seen that many of navigations can be performed

on different URLs with a higher probability. There may be many reasons for this; such as users progressing according to a specific pattern or using sub-patterns more in other sequences. But besides all this, this result shows us that; if these topic sequences continue to be used in the future, this is how we can get the most probable URL navigations for future users. In instance, for the first dataset, we have approximately 140,000 sequences which we did not use. When we inspect the 405 different URL sequences of the first dataset, we saw that 17 of them navigated by users which registered after 25,000th user. This shows us the ability to calculate and test the navigations of future users.

5.3.2 Results of Test 2

Different from the above, we made an edit distance calculation for each dataset between the topic sequences we produced, and the topic sequences of the navigations made by real users. According to the results we obtained, we can obtain topic sequences made by real users with an average of three changes on most of the topic sequences we produced for the first dataset, an average of three changes for the Loadium dataset, and an average of three changes for the Testinium dataset. This means that although the topic sequences we produce are different from the actual topic sequences, they are similar enough.

In 5.2, we have three charts. In the first chart, it is seen that 18% of the produced topic sequences can be converted with a single change, on 55% with two changes, on 25% with three changes, on 2% with the four changes, and so on to their original form which are topic sequences navigated by real users.

In the second chart, it is seen that approximately 4% of the produced topic sequences can be converted with a single change, on 25% with two changes, on 48% with three changes, on 18% with the four changes and so on to their original form which are topic sequences navigated by real users.

In the third chart, it is seen that approximately 16% of the produced topic sequences can be converted with a single change, on 41% with two changes, on 34% with three changes, on 8% with the four changes and so on to their original form which are topic sequences navigated by real users.

These results show that the newly generated URL sequences are in fact similar (distribution probability-wise) to URL sequences that are used as input by the proposed workflow. The results indicate that by using the proposed workflow, we can create new URL sequences can be used in UI test scripts. Detailed information is

shown in Figure 5.2. These charts show the edit distance between the topic sequences we produced, and the topic sequences navigated by real users. The first chart shows the Mother & Baby products dataset, the second chart shows the Loadium dataset, and the third chart shows the Testinium dataset.

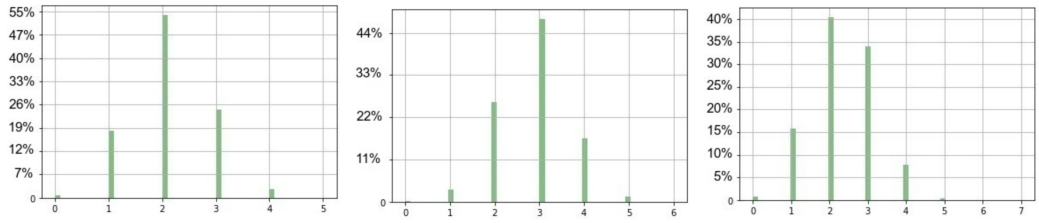


Figure 5.2 Edit Distances. In each chart, y-axis means shows the percentage in the interested dataset, x-axis means specifies the edit distance value.

Also, we made a Bleu score calculation for each dataset between the topic sequences we produced, and the topic sequences of the tours made by real users. Our results show that more than 70% of generated topic sequences have 0,7 or higher Bleu scores. This means that our topic sequences are highly similar to actual topic sequences.

Detailed information is shown in Figure 5.3. These charts show the bleu scores between the topic sequences we produced and the topic sequences navigated by real users. The first chart shows the Mother & Baby products dataset, the second chart shows the Loadium dataset, and the third chart shows the Testinium dataset.

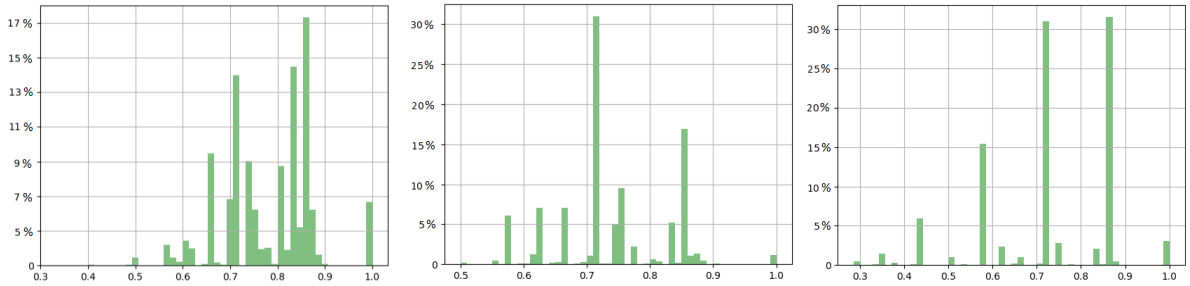


Figure 5.3 Bleu scores. In each chart, y-axis means shows the percentage in the interested dataset, x-axis means specifies the bleu scores.

5.4 Threats to Validity of Results

The results we obtained in this study depend on the website URL structure of the websites we have carried out. However, the proposed end-to-end automated ui testing framework's workflow that we have used constitutes a step-by-step mechanism independent of data. Moreover, we have conducted experiments on three websites to ensure our introduced workflow is accurate. The framework flow is also detailed and described above. The proposed workflow can also be applied to different datasets.

6
CONCLUSION

This study proposes an e-business workflow that can be used as a test-script generation tool. The proposed framework learns from the users' behaviors (by analyzing the users' s browsing graph data). Then it generates test scripts, which can be used to test new/unseen users' activities.

We provide a prototype implementation to show the proposed workflow's usability. To facilitate testing the prototype, we use a real-life user-activity (i.e., URL sequences) datasets obtained from an e-commerce site, one of the leading stores selling mother and baby products, and two of the leading testing firms. By utilizing our prototype implementation, we generated new/unseen user activities. We used the new/unseen user-activities output and converted them to test scripts. Then, the test scripts were integrated into the Selenium UI test execution tool.

We also conduct an experimental study to demonstrate our methodology's effectiveness. Here, we measured the success of the proposed workflow by evaluating the similarity and convergence between the input user activities and output new/unseen user activities.

The results indicate that while the URL sequences generated by the proposed workflow are different from the URL sequences in the actual data, there is a high similarity between the actual URL sequences and unseen URL sequences in topic basis. Thus, based on the promising results, the proposed framework can be used to successfully generate new test patterns based on the knowledge obtained from the user browsing graph data. Also, the results show that our proposed framework can generate new test scripts that will be navigated by future users.

In our previous work, we stated that we wanted to use different datasets in a future study [45]. In this study, we achieved this goal by testing HMM and Viterbi on three different datasets in test script generation. In our future works, we want to investigate the alternative methodologies that can be used to generate new/unseen

user activities.

REFERENCES

[1] E. Olmezogullari, M. S. Aktas, "Pattern2vec: Representation of clickstream data sequences for learning user navigational behavior," *Concurrency and Computation: Practice and Experience*, e6546, doi:10.1002/cpe.6546, 2021.

[2] M. I. M. Yusoff, I. Mohamed, M. R. A. Bakar, "Hidden markov models: An insight," in *Proceedings of the 6th International Conference on Information Technology and Multimedia*, 2014.

[3] X. Deng, H. Ali, "A hidden markov model for gene function prediction from sequential expression data," in *Proceedings. 2004 IEEE Computational Systems Bioinformatics Conference*, 2004.

[4] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IIEEE Transactions on Information Theory*, vol. 13, pp. 260–269, 1967.

[5] A. J. Viterbi, "A personal history of the viterbi algorithm," *IEEE Signal Processing Magazin*, vol. 23, pp. 120–142, 2006.

[6] J. Eskonen, J. Kahles, J. Reijonen, "Automating gui testing with image-based deep reinforcement learning," 2020.

[7] Q. Su, L. Chen, "A method for discovering clusters of e-commerce interest patterns using click-stream data," *Electronic Commerce Research and Applications*, vol. 14, pp. 1–13, 2014.

[8] I.-H. Ting, C. Kimble, D. Kudenko, "Ubb mining: Finding unexpected browsing behaviour in clickstream data to improve a web site's design," in *The 2005 IEEE/WIC/ACM International Conference on Web Intelligence (WI'05)*, 2005.

[9] Y. Uygun, R. F. Oguz, E. Olmezogullari, M. S. Aktas, "On the large-scale graph data processing for user interface testing in big data science projects," in *2020 IEEE International Conference on Big Data (IEEE Big Data)*, 2020, pp. 2049–2056.

[10] E. Olmezogullari, M. Aktas, "Representation of click-stream datasequences for learning user navigational behavior by using embeddings," in *2020 IEEE International Conference on Big Data (IEEE Big Data 2020)*, 2020, pp. 3173–3179.

[11] L. Prasad, R. Yadav, N. Vore, "A systematic literature review of automated software testing tool," in *Abraham A., Castillo O., Virmani D. (eds) Proceedings of 3rd International Conference on Computing Informatics and Networks. Lecture Notes in Networks and Systems*, vol. 167, Singapore: Springer, 2021, pp. 101–123. DOI: https://doi.org/10.1007/978-981-15-9712-1_10.

[12] F. Wang, W. Du, "A test automation framework based on web," in *2012 IEEE/ACIS 11th International Conference on Computer and Information Science*, IEEE, 2012.

[13] K. V. Arya, H. Verma, "Keyword driven automated testing framework for web application," in *2014 9th International Conference on Industrial and Information Systems (ICIIS)*, IEEE, 2015.

[14] S. Yang, J. Tao, J. Meng, "Ranking algorithm based on user behavior model," in *2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, IEEE, 2020.

[15] L. Peska, A. Eckhardt, P. Vojtas, "Upcomp - a php component for recommendation based on user behaviour," in *2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*, IEEE, 2011.

[16] G. Xu, Y. Zhang, X. Yi, "Modelling user behaviour for web recommendation using lda model," in *2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, Australia: IEEE, 2008.

[17] H. Kawazu, F. Toriumi, M. Takano, K. Wada, I. Fukuda, "Analytical method of web user behavior using hidden markov model," in *2016 IEEE International Conference on Big Data (Big Data)*, IEEE, 2016.

[18] K.-C. Chuang, C.-S. Shih, S.-H. Hung, "User behavior augmented software testing for user-centered gui," in *RACS '11: Proceedings of the 2011 ACM Symposium on Research in Applied Computation*, ACM, 2011, pp. 200–208.

[19] E. Guveyi, M. Aktas, O. Kalipsiz, "Human factor on software quality: A systematic literature review," in *ICCSA 2020*, 2020.

[20] M. Kapdan, M. Aktas, M. Yigit, "On the structural code clone detection problem: A survey and software metric based approach," in *ICCSA 2014*, 2014.

[21] M. Uzun-Per, A. B. Can, A. V. Gurel, M. S. Aktas, "Big data testing framework for recommendation systems in e-science and e-commerce domains," in *IEEE Big Data 2021*, 2021.

[22] M. S. Aktas, M. Kapdan, "Structural code clone detection methodology using software metrics," in *IJSEKE vol 26 issue 2*, 2016.

[23] M. Sahinoglu, K. Incki, M. Aktas, "Mobile application verification: A systematic mapping study," in *ICCSA 2015*, 2015.

[24] M. Aktaş, G. Fox, M. Pierce, "Managing dynamic metadata as context," in *The 2005 Istanbul International Computational Science and Engineering Conference (ICCSE2005), Istanbul, Turkey*, 2005.

[25] M. Aktas, G. Fox, M. Pierce, "Information services for dynamically assembled semantic grids," in *The First International Conference on Semantics, Knowledge and Grid (SKG 2005), Beijing, China*, 2005.

[26] M. Aktas, G. Aydin, G. Fox, H. Gadgil, M. Pierce, A. Sayar, "Information services for grid/web service oriented architecture (soa) based geospatial applications," in *The First International Conference on Semantics, Knowledge and Grid (SKG 2005), Beijing, China*, 2005.

[27] Y. Tas, M. Baeth, M. Aktas, "An approach to standalone provenance systems for big social provenance data," in *12th International Conference on Semantics, Knowledge and Grids (SKG)*, 2016.

[28] J. Baeth, M. Aktas, "Detecting misinformation in social networks using provenance data," in *13th International Conference on Semantics, Knowledge and Grids*, 2017.

[29] I. Yazici, E. Karabulut, M. Aktas, "A data provenance visualization approach," in *14th International Conference on Semantics, Knowledge and Grids*, 2018.

[30] G. Aydin, M. Aktas, G. Fox, H. Gadgil, M. Pierce, A. Sayar, "Servogrid complexity computational environments cce integrated performance analysis," in *2005 6th International Workshop on Grid Computing (GRID)*, 2005.

[31] M. Aktas, et al., "Implementing geographical information system grid services to support computational geophysics in a service-oriented environment," in *NASA Earth-Sun System Technology Conference 2005*, 2005.

[32] M. Pierce *et al.*, "The quakesim project: Web services for managing geophysical data and applications," *Journal of Pure and Applied Geophysics*, vol. 165, pp. 635–651, 2008. DOI: 10.1007/s00024-008-0319-7.

[33] A. Tufek, A. Gurbuz, O. Ekuklu, M. Aktas, "Provenance collection platform for the weather research and forecasting model," in *14th International Conference on Semantics, Knowledge and Grids*, 2018.

[34] B. Dundar, M. Astekin, M. Aktas, "A big data processing framework for self-healing internet of things applications," in *12th International Conference on Semantics, Knowledge and Grids*, 2016.

[35] T. Boyar, M. Oz, M. Aktas, "A novel approach to test case prioritization for software regression tests," in *International Conference on Computational Science and Its Applications (ICCSA)*, 2021.

[36] S. Ozturk, N. Yagci, M. Aktas, M. Ozbek, F. Paligu, "Test case prioritization related to code quality," in *Proceedings of the International Conference on Software Engineering Research and Practice (SERP)*, 2013.

[37] R. F. Oguz, M. Oz, E. Olmezogullari, M. S. Aktas, "Extracting information from large scale graph data: Case study on automated ui testing," in *Euro-Par 2021*, 2021.

[38] "Python software foundation. python programming language." (Oct. 2021), [Online]. Available: `https://www.python.org/` (visited on 10/22/2021).

[39] "Scrapy, fast high-level web crawling and web scraping framework." (Oct. 2021), [Online]. Available: `https://scrapy.org/` (visited on 10/22/2021).

[40] "Regular expression operations." (Oct. 2021), [Online]. Available: `https://docs.python.org/3/library/re.html` (visited on 10/22/2021).

[41] C. R. Harris, K. J. Millman, et al., "Array programming with NumPy," *Nature*, vol. 585, pp. 357–362, 2020. DOI: 10.1038/s41586-020-2649-2.

[42] "Pandas; fast, powerful, flexible and easy to use open source data analysis and manipulation tool." (Oct. 2021), [Online]. Available: `https://pandas.pydata.org/` (visited on 10/22/2021).

[43] K. Papineni, S. Roukos, T. Ward, W.-J. Zhu, "Bleu: A method for automatic evaluation of machine translation," in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, Philadelphia, 2002, pp. 311–318.

[44] P. E. Black. "Levenshtein distance." (May 2021), [Online]. Available: https://www.nist.gov/dads/HTML/Levenshtein.html (visited on 05/15/2019).

[45] I. Erdem, R. F. Oguz, E. Olmezogullari, M. S. Aktas, "Test script generation based on hidden markov models learning from user browsing behaviors," in *2021 IEEE International Conference on Big Data (Big Data)*, 2021, pp. 2998–3005. DOI: 10.1109/BigData52589.2021.9671312.

PUBLICATIONS FROM THE THESIS

Papers

1. Oguz, Ramazan Faruk, Izzettin Erdem, Olmezogullari, E. and Aktas, M.S., 2022. End-to-End Automated UI Testing Framework for Web Sites with Intensive User-System Interactions. International Journal of Software Engineering and Knowledge Engineering (Vol. 32, No. 10, pp. 1477-1497)

Conference Papers

1. Erdem, I., Oguz, R.F., Olmezogullari, E. and Aktas, M.S., 2021, December. Test Script Generation Based on Hidden Markov Models Learning From User Browsing Behaviors. In 2021 IEEE International Conference on Big Data (Big Data) (pp. 2998-3005). IEEE.