

TEST CASE GENERATION FOR SOFTWARE TESTING ON SMART TELEVISIONS WITH LSTM NETWORKS

A Thesis

by

Yusuf Çağlayan

Submitted to the
Graduate School of Sciences and Engineering
In Partial Fulfillment of the Requirements for
the Degree of

Master's Degree

in the
Department of Electrical and Electronics Engineering

Özyeğin University
December 2022

Copyright © 2022 by Yusuf Çağlayan

TEST CASE GENERATION FOR SOFTWARE TESTING ON SMART TELEVISIONS WITH LSTM NETWORKS


Approved by:

Assoc. Prof. Dr. Cenk Demirođlu, Advisor
Dept. of Electrical and Electronics Eng.
Özyeđin University

Assoc. Prof. Dr. Hasan sözer
Department of Computer Science
Özyeđin University

Prof. Dr. Ümit Güz
Dept. of Electrical and Electronics Eng.
Işık University

Date Approved: 1 December 2022



*To dedicate to Prof. Dr. Taner Abdullah Oğuzer, who is very precious
to me, a very knowledgeable teacher, precious brother, and father.*

ABSTRACT

Software testing for Smart TVs is an expensive and time-consuming process. Therefore, automatic and effective test case generation is important for reducing costs while improving software quality. In this study, we propose using the Long Short Term Memory (LSTM) networks for auto-generating new and effective test cases for smart TVs in a short amount of time. Training data for the network is prepared by an analysis of reset, crash, and hang errors that occurred in four different TV software during testing. The test cases for those errors were generated manually during the software approval processes at the Vestel Electronics TV factory. A total of 3000 errors were extracted from the test database, where each error corresponds to a sequence of actions, such as turning the TV on/off or opening the Netflix application, etc., that result in an error. A Long Short Term Memory (LSTM) network model is then trained to auto-generate test cases using those errors. The performance of the model is measured by how many errors could be generated given a fixed number of trial sequences. In experiments, auto-generated test cases were more successful in generating sequences that resulted in errors compared to the manually designed cases. Thus, significantly more errors could be generated in a given time frame using the proposed LSTM model, which resulted in more reliable and faster software.

ÖZETÇE

Akıllı TV'ler için yazılım testi, pahalı ve zaman alıcı bir süreçtir. Bu nedenle, yazılım kalitesini artırırken maliyetleri düşürmek için otomatik ve etkin test senaryosu oluşturma önemlidir. Bu çalışmada, akıllı TV'ler için kısa sürede otomatik olarak yeni ve etkili test senaryoları oluşturmak için Uzun Kısa Süreli Bellek (LSTM) ağlarının kullanılmasını öneriyoruz. Önerilen model için eğitim verileri, test sırasında dört farklı TV yazılımında meydana gelen sıfırlama, çökme ve kilitlenme hatalarının analizi ile hazırlanmıştır. Bu hatalara ilişkin test senaryoları, Vestel Elektronik TV fabrikasında yazılım onay süreçlerinde manuel olarak oluşturulmuştur. Test veritabanından, her bir hatanın, TV'yi açma ve kapatma veya Netflix uygulamasını açma gibi bir hatayla sonuçlanan bir dizi eyleme karşılık geldiği toplam 3000 hata ayıklandı. Bir Uzun Kısa Süreli Bellek (LSTM) ağ modeli daha sonra bu hataları kullanarak test senaryolarını otomatik olarak oluşturmak için eğitilir. Modelin performansı, sabit sayıda deneme dizisi verildiğinde ne kadar hata üretilebileceği ile ölçülür. Deneylerde, otomatik olarak oluşturulan test senaryoları, manuel olarak tasarlanmış vakalara kıyasla hatalara neden olan diziler oluşturma konusunda daha başarılıydı. Böylece, önerilen LSTM modeli kullanılarak belirli bir zaman diliminde önemli ölçüde daha fazla hata üretilebildi ve bu da daha güvenilir ve daha hızlı yazılımla sonuçlandı.

ACKNOWLEDGEMENTS

First, I would like to thank my esteemed thesis advisor Assoc. Prof. Dr. Cenk Demirođlu shared his knowledge, guided me with his valuable comments, and helped me with my missing points. Thanks to his valuable assistance, it was possible for me to finish this work.

I want to thank the department chief and employees of the Vestel Electronics Software Quality Assurance Department for allowing the necessary tests to be carried out and for their efforts to ensure the reliability and smooth functioning of the tests.

I would also like to thank Assoc. Prof. Dr. Hasan Sözer of the Department of Computer Science at Özyeđin University and as the second reader Prof. Dr. Ümit Güz of the Department of Electrical and Electronics Engineering at Işık university, and I am gratefully indebted to their lovely comments on this thesis.

Finally, thank my dear family for their unconditional help, always by my side and supporting me throughout my life.

TABLE OF CONTENTS

DEDICATION	iii
ABSTRACT	iv
ÖZETÇE	v
ACKNOWLEDGEMENTS	vi
LIST OF TABLES	viii
LIST OF FIGURES	x
I INTRODUCTION	1
II RELATED WORK	4
III METHODOLOGY	8
3.1 Recurrent Neural Networks(RNN) Model	8
3.2 Long Short Term Memory (LSTM) Model	10
3.3 Industrial Case Study	14
IV EXPERIMENTS	15
4.1 Experimental Setup	15
4.2 Application Management	16
4.3 Results and Discussion	20
V CONCLUSIONS	35
REFERENCES	37
VITA	40

LIST OF TABLES

1	Main Test Steps Revealed When 3000 Error Records Are Examined and Detailed Test Steps Included in These Main Test Steps.	25
2	Required Matrix Form to Input the Main Test Steps as Input to the LSTM Model.	26
3	Matrix Form of 2 Different Test Scenarios: The Matrix Form of 3000 Scenarios Seen in Manual and Automation Tests.	27
4	The summary of the first LSTM model used was designed using only one LSTM layer. This model repeats the input as many times as desired by using the RepeatVector layer. In addition, in the model, the final output of the LSTM layer, along with the step time, is obtained from the dense layer.	28
5	Table of Automation Test scenario created according to the first LSTM Model. The test scenario formed according to this model consists of 4 steps. The test steps are Netflix Application, Power Off\On, Standby Off\On, and the last step is Standby Off\On.	28
6	Summary of the second LSTM model used was designed using only 1 LSTM layer. In addition, using the repeat vector, the operations are repeated on the original LSTM layer, and the model is ready. The time-distributed option property, which is used in multi-layer situations, is used.	29
7	Table of Automation Test scenario created according to the Second LSTM Model. The test scenario formed according to this model consists of 3 steps. The test steps are Incremental Standby Off\On, Youtube Application, and Power Off\On.	29
8	Summary of The third LSTM model used was designed as a “Sequential Model,” and one more LSTM layer was designed by using a repeater on 1 LSTM layer.	30
9	Table of Automation Test scenario created according to the Third LSTM Model. The test scenario formed according to this model consists of 4 steps. The test steps are Web Browser Application, Incremental Power Off\On, Standby Off\On, and Youtube Application. . .	30
10	Summary of the fourth LSTM model used is designed as Multi-Layer LSTM Model, the output of the first LSTM layer is provided to the next, and at the same time, the final state of the second LSTM layer is provided as the initial state to the next LSTM layer.	31

11	Table of Test with Vestel TV Project 1(VTP1) with test scenarios with different LSTM Models. Test with Vestel TV Project 1(VTP1) with test scenarios with different LSTM Models. The LSTM Models used are the first, second, and third LSTM Models with the lowest accuracy. The automation test scenarios that emerged with these LSTM Models found errors parallel with their accuracy. The test scenario produced with the first LSTM model with the lowest accuracy found the least number of errors. The test scenario produced with the third LSTM Model, which had the highest accuracy among them, found the highest number of errors.	32
12	Automation Test Scenarios were obtained by Arranging the Last Layer of the LSTM Model as Probabilistic(TS2, TS3, TS4, TS5), and the Automation Test Scenario was given as the output of the program according to the LSTM Model(TS1).	33
13	Automation test scenarios have been tested in Vestel TV projects with different features (first TV project(VTP1), second TV project(VTP2), third TV project(VTP3), and fourth TV project (VTP4). Test scenarios obtained with the primary LSTM model (TS1), test scenarios obtained by the probabilistic randomization of the steps in the last layer of the primary LSTM model (TS2, TS3, TS4, TS5)), classical automation tests used in the Vestel SWQA department(VTS1, VTS2, VTS3, VTS4).	34

LIST OF FIGURES

1	<p>The figure of the RNN model. Boxes represent the neural network, and its called A in this figure; arrows represent memory, X represents input data, h represents output data, all inputs represent (X_t), and Outputs represent (h_t), previous inputs represent (X_{t-1}) and previous outputs represent (h_{t-1}). The expansion of the single figure on the left side of the equation is composed of all cells on the right side of the equation.</p>	9
2	<p>The figure of the Detailed RNN model. Boxes represent the neural network, and its called A in this figure; arrows represent memory, X represents input data, h represents output data, all inputs represent (X_t), and Outputs represent (h_t), previous inputs represent (X_{t-1}) and previous outputs represent (h_{t-1}). The activation function is the tanh activation function. This figure shows the activation function used in the Neural Network and its relationship with other cells.</p>	10
3	<p>The Figure of LSTM Model. In this Figure, the internal structure of the LSTM Model is shown and the functions used are shown in the Box. All Boxes here are Cells, and the working mechanism of these cells is visualized. The functions used and the signs in the Figure are shown. This Figure shows a simple structure of the LSTM model. . .</p>	11
4	<p>The figure of Detailed LSTM Model. In this figure, the structures in the LSTM Model are shown. The area in the green frame shows the Forget gate, the area in the red frame shows the Input Gate, the area in the blue frame shows the Output Gate and the area in the black frame shows the Cell State structures.</p>	12
5	<p>Test with Vestel TV Project 1(VTP1) with test scenarios with different LSTM Models. Test with Vestel TV Project 1(VTP1) with test scenarios with different LSTM Models. The LSTM Models used are the first, second, and third LSTM Models with the lowest accuracy. The automation test scenarios that emerged with these LSTM Models found errors parallel with their accuracy. The test scenario produced with the first LSTM model with the lowest accuracy found the least number of errors. The test scenario produced with the third LSTM Model, which had the highest accuracy among them, found the highest number of errors.</p>	22

6 The figure of the test result of TV Project 1 & TV Project 2 & TV Project 3 & TV Project 4. Automation test scenarios (TS) have been tested in Vestel TV projects with different features (VTP(first TV project(VTP1), second TV project(VTP2), third TV project(VTP3), fourth TV project (VTP4))). Test scenarios obtained with the primary LSTM model (TS1), test scenarios obtained by the probabilistic randomization of the steps in the last layer of the primary LSTM model (TS2, TS3, TS4, TS5), classical automation tests used in the Vestel SWQA department (VTS(VTS1, VTS2, VTS3, VTS4)). As expected, the test scenario that found the highest number of errors was the test scenario created with the LSTM model.



CHAPTER I

INTRODUCTION

In recent years, with the advancement of technological findings, consumer electronics has transformed from an electromechanical structure into complex and large software products. This transformation has two structures. First, all the features provided now pass from software to hardware. The second feature is the increasing number and variety of all consumer electronic products. Digital televisions (DTV), consumer electronic products, contain applications that require the internet, unlike ordinary and traditional televisions [1]. Some of these applications are Youtube application, Netflix application, Amazon Prime application, Web browser applications, etc. As a result of all these, the software's code is increasing. As a result of this increasing amount of code, software complexity is also increasing. Due to all these disadvantages, it becomes difficult to ensure and protect software security due to possible errors [2]. However, unlike critical security software, these errors do not cause irreparable results. Due to the competition in the consumer electronics market, it is influential in the formation of users' perception [3].

In addition to all these complexities, making radical decisions, especially in terms of cost, is necessary. Among these, labor and time play an essential role. All these restrictions prevent the software and software involved from being thoroughly tested. At some points, more is needed to remove the errors detected with the existing resources of the existing site. For this, it is necessary to develop an intelligent test program that includes the most optimal solution, taking into account the software and software versions. In light of all this, testing related products within the scope of consumer electronics should efficiently use the resources available.

In general, tests are done manually. However, this test method is insufficient due to time and labor limitations. Instead, it is crucial to plan and implement automation tests. Determining test scenarios is very important in planning automation tests. These scenarios should have a test mechanism that maximizes efficiency within a short test period.

In digital and smart television software issues, the most significant errors affecting the user are the crash\reset\hang errors in the software. These errors are a huge problem for users, and they can file potential customer complaints. These errors may not occur during software test checks because these errors occur in possible random scenarios and during continuous use. For all these reasons, producing comprehensive, practical, and fast scenarios for these errors is imperative.

This study aims to create a test scenario using the Long Short Term Memory (LSTM) algorithm to produce the most suitable test scenario for the mentioned crash\reset\hang errors. The data of the LSTM algorithm used for these scenarios are provided by VESTEL ELEKTRONIK SAN. ve TIC. A.Ş¹ test groups examined and processed 3000 crash\reset\hang errors in 4 different Vestel Smart TV projects. During this review, the scenarios of each error mentioned were separated one by one, and relevant data were created. The test scenario created with the LSTM algorithm is aimed to find the maximum number of crash\reset\hang errors in a minimum time. As a result, it is aimed to contribute to the stability of the Smart TV software. It has been observed that the test scenarios created using the LSTM algorithm according to the classical automation test scenarios found the maximum number of determining errors in a minimum time. It was started to be used in software tests by the Software Quality Assurance(SWQA) Department of Vestel Electronics.

¹<http://www.vestel.com.tr>

The organizational structure of the remainder of this thesis is as follows. In the next section, information about the summaries of the related studies is given. Chapter 3 includes our work, approach, and the reason why we work as an industrial case study. Chapter 4 describes our experimental setup and its comparison with the results. In the final chapter, Chapter 5, a summary of all results and future work will be described.



CHAPTER II

RELATED WORK

Long Short Term Memory Model is one of the Recurrent Neural Network mechanisms, one of the deep learning algorithms. In general, this algorithm examines data series as well as instant data. The LSTM algorithm has been used in multiple fields as an advantage of the memory mechanism and data analysis. The LSTM algorithm ensures that all possible bad scenarios can be predicted and precautions can be taken against them. For example, the LSTM algorithm is used to predict sewer overflows in troubled areas of municipalities [4]. In addition, the LSTM algorithm is used for defense against external interference by automatic control units, which are indispensable for electrical networks [5]. Together with all these results, the LSTM algorithm is used in wind energy production estimation [6], solar energy production estimation [7], traffic flow estimation, and health services trajectory estimation [8] [9], response rankings [10]. Because of the good results in other areas, we hope that using the LSTM algorithm for test case generation is an effective strategy.

Software validation and verification are significant in the software world. Various alternatives exist to test these two concepts [11]. Deep learning, one of these methods, is becoming widespread in software engineering (SE), and deep learning algorithms are used in software tests, which is one of the SE problems [12]. The selection and design of test cases for software testing are essential, and test case selection has been studied for over 30 years [13].

Different studies have been carried out and evaluated for this problem. Some of these studies are based on the similarity between test scenarios [14]. When the effects and results of the previous test scenarios made using the LSTM algorithm are made

as algorithm inputs, it has been observed that the efficiency increases in the results obtained [15]. Taking advantage of these results, automatic test model management was applied to be selected in factory acceptance tests for control engineering [16]. This approach in control engineering will have the effect of increasing productivity in production areas. As a result, it will be one step closer to error-free, perfect and fast production. By applying the deep learning method, vehicle driving scenarios can be created, which can be used in related tests [17]. This study, it is aimed that the scenarios that may affect people's lives are predicted and that the system will continue with minor damage in possible accident situations or that the users will get rid of the situation with minimum damage.

Machine learning can detect and prevent software vulnerabilities and even generate negative test cases. There are various studies conducted for this purpose. For example, deep learning algorithms are used to detect and prevent software security vulnerabilities [18]. Therefore, among the non-functional test types, security tests were made with artificial intelligence algorithms. Therefore, a safe and fast process is provided.

Nuclear reactors are quite severe structures. All kinds of security measures should be taken for these structures. Possible failure scenarios are very critical at this stage. Possible accident scenarios created using deep learning methods can be generated automatically [19]. These critical scenarios can prevent accidents that threaten people's lives, and new measures can be taken thanks to accident scenarios. This study, it was aimed to prevent the occurrence of bad situations.

Studies have focused on Web applications to create automated test scenarios [20]. These studies were categorized and generally divided into three different types [20]. These types are code-based white-box testing approaches, black-box testing approaches based only on end-user logic, and hybrid approaches [20]. Our study covers the end-user tests performed by taking the black-box test approach as a whole.

The black-box test approaches were obtained in two different ways. The first re-runs the recorded sessions over and over [21], or combines or merges and replaces all existing sessions and creates new test cases [20]. As a result, they do not learn a model like a user. However, our approach learns the scenarios that lead to user error and creates new scenarios.

Learning-based testing methods have been introduced in black box testing of reactive systems [22]. In this study, a model of the system in the form of a state machine is iteratively constructed using the tests carried out previously. In addition, this study used the model to explore related pathways. As a result, new test scenarios were created. Reactive systems are a set of architectural forms that cover applications with different tasks, consisting of more than a tiny service or a system with different task structures to provide better feedback and information to their environment and each other. Therefore, Smart Televisions are also a reactive system.

Based on the recorded activities of test engineers, studies supporting the test models have been carried out [23]. This way, a new test model has been developed as a state diagram. A new model has been developed based on this model, allowing it to be manually updated based on the situations and behaviors observed during free tests [24], [25], [26], [27]. Various studies have also been carried out to automatically update manual updates [28] and [29].

In addition, Vestel Electronics conducts an automation test scenario creation study using the LSTM model by learning the scenarios made by the testers[30]. In the related study, both test scenarios produced from a high-level model of the system for functional testing and registered robustness test inputs of experienced test technicians were used as training inputs to the LSTM network. However, since the results did not find enough crash errors, the software solution could not be taken, or the solution was delayed. Because of this situation, the necessity of doing a new study has arisen.

In this study, studies have been carried out to bring the previous studies to a better dimension and to use Artificial intelligence models most effectively and reliably. In the study, the error scenarios found by the Vestel electronic TV factory test groups were analyzed and taught to the model (LSTM Model). This training aims to capture the number and type of errors that will facilitate the resolution of stability errors. This way, relevant software groups can easily analyze errors and quickly integrate their solutions into the algorithm. For software tests of Vestel Smart TVs, an effort was made to produce test scenarios to be used in automation tests by using the LSTM algorithm to be used in black box tests, taking into account the results of previous studies.

CHAPTER III

METHODOLOGY

In this section, AI algorithm models will be discussed. These are Long Short Term Memory(LSTM) and Recurrent Neural Network(RNN). Artificial intelligence algorithms can solve many real-life problems. If there is a time-based or sequential relationship between the data, the problem can be solved by Recurrent Neural Networks (RNN) algorithms. The Long Short-Term Memory (LSTM) algorithm is also an RNN algorithm. The LSTM algorithm can process pictures, video, and audio and can be used to solve various problems.

The reason for using the LSTM Model in the study is to teach many data to the model in a time-series form and to enable it to predict a sequence from the model in the form of time series. At this stage, the LSTM Model was used. The general structure of the RNN Model is given in Section 3.1, and information about the structure of the LSTM Model is given in Section 3.2. In the last section, section 3.3, the solution approach of the LSTM Model for the problem in the industry is given. This solution creates test scenarios to find the maximum number of stability errors in the fastest time for the Black-Box tests of Smart TVs within Vestel Electronics.

3.1 Recurrent Neural Networks(RNN) Model

RNN is an artificial intelligence method. The main difference from other methods is that the connection between units creates a directed loop. Because of this loop, a time-dependent model can be turned into an ordered, dynamic structure. Recurrent neural networks are based on the work of David Rumelhart in 1986 [31]. A particular type of RNN called the Hopfield network was also developed by John Hopfield in 1982. In 1993, an RNN study accomplished a “profound learning” task that required more

than 1000 layers [32]. Long short-term memory (LSTM) networks were developed by Hochreiter and Schmidhuber in 1997 and recorded the best performances in various application areas [33].

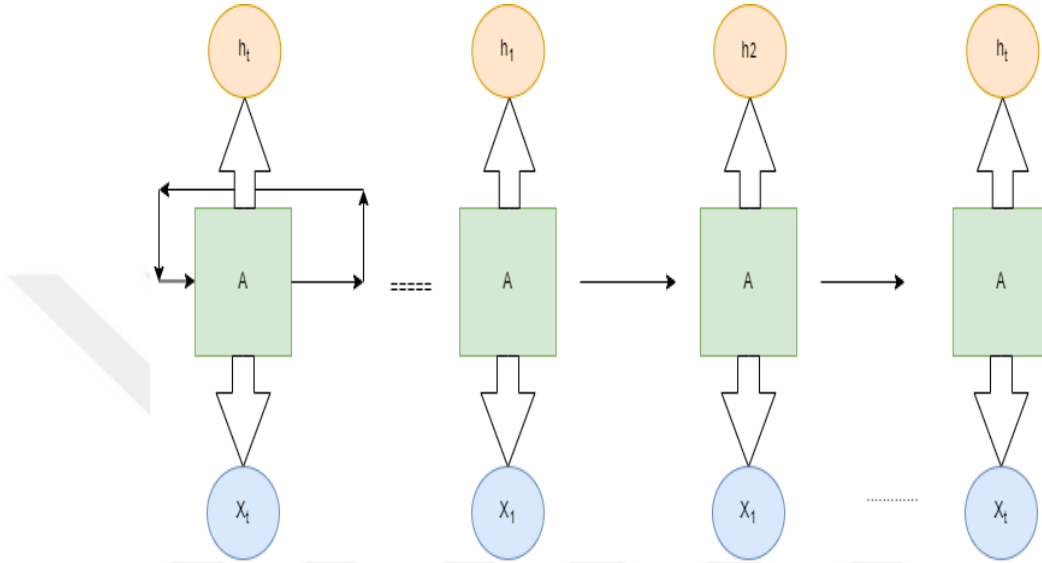


Figure 1: The figure of the RNN model. Boxes represent the neural network, and its called A in this figure; arrows represent memory, X represents input data, h represents output data, all inputs represent (X_t), and Outputs represent (h_t), previous inputs represent (X_{t-1}) and previous outputs represent (h_{t-1}). The expansion of the single figure on the left side of the equation is composed of all cells on the right side of the equation.

When Figures 1 and 2 are examined, green boxes represent the neural network; arrows represent memory; X represents input data, and h represents output data. As can be understood from this figure, the basic approach is based on using sequential information. That is, all inputs (X_t) and Outputs (h_t) are composed independently of the previous inputs (X_{t-1}) and previous outputs (h_{t-1}). The activation function is the tanh activation function.

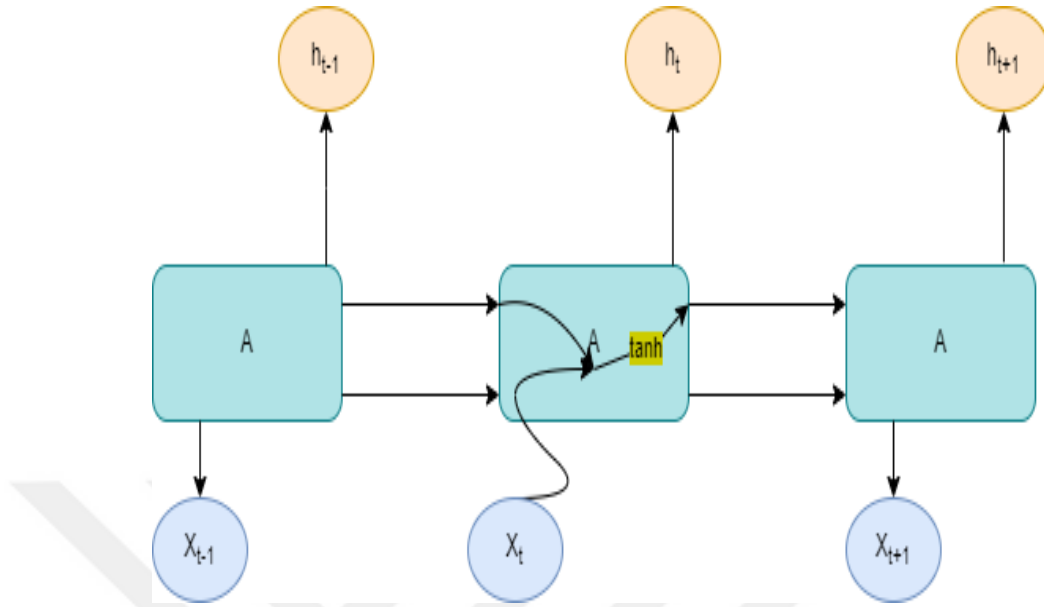


Figure 2: The figure of the Detailed RNN model. Boxes represent the neural network, and its called A in this figure; arrows represent memory, X represents input data, h represents output data, all inputs represent (X_t), and Outputs represent (h_t), previous inputs represent (X_{t-1}) and previous outputs represent (h_{t-1}). The activation function is the tanh activation function. This figure shows the activation function used in the Neural Network and its relationship with other cells.

However, this architecture needs to improve, especially in problems such as natural language processing problems, for which previous knowledge was essential. For this problem, it is necessary to use a structure with a memory layer, which creates a new output mechanism depending on the previous outputs.

3.2 Long Short Term Memory (LSTM) Model

The biggest problem encountered using RNN algorithms is the Vanishing Gradient Problem. The Vanishing Gradient Problem is that weights disappear with Backpropagation. In other words, the gradient values will decrease exponentially as the step progresses and eventually disappear. This will result in a significant memory problem. To solve this problem, it is necessary to use the LSTM model.

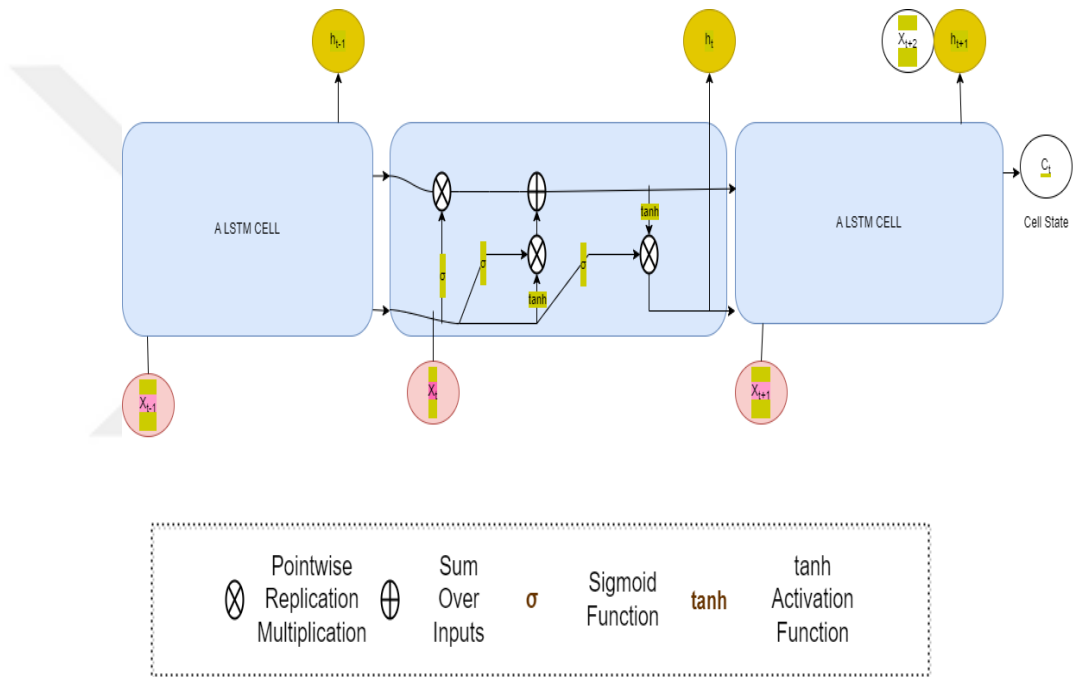


Figure 3: The Figure of LSTM Model. In this Figure, the internal structure of the LSTM Model is shown and the functions used are shown in the Box. All Boxes here are Cells, and the working mechanism of these cells is visualized. The functions used and the signs in the Figure are shown. This Figure shows a simple structure of the LSTM model.

The LSTM algorithm classifies, processes, and predicts time series data. There may be delays of unknown duration between significant events in the time series. LSTM, on the other hand, can easily overcome this problem with its existing memory. The structure of the LSTM Model shows in Figure 3.

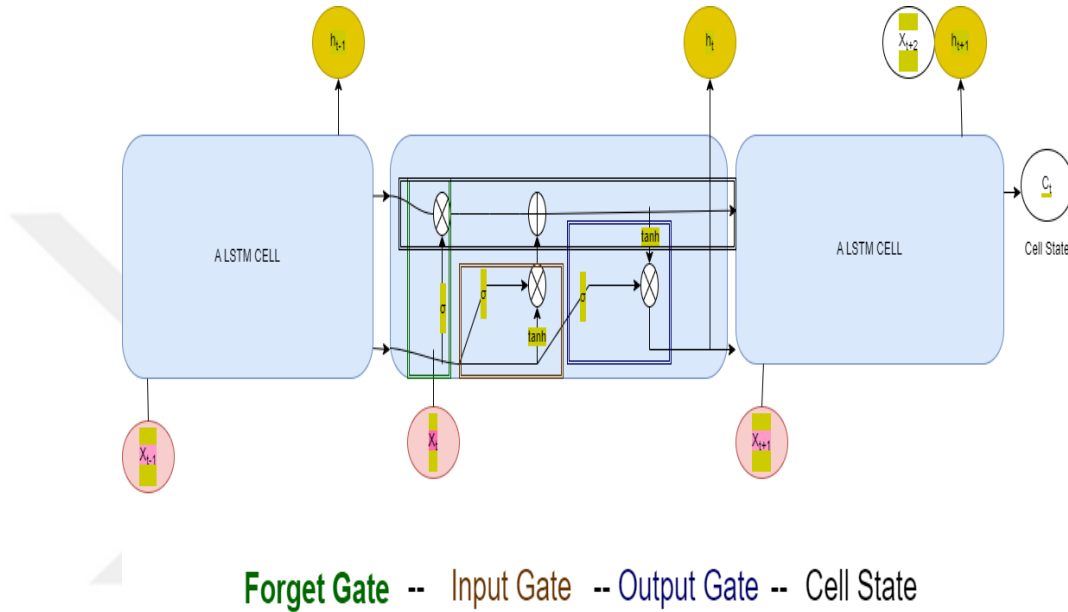


Figure 4: The figure of Detailed LSTM Model. In this figure, the structures in the LSTM Model are shown. The area in the green frame shows the Forget gate, the area in the red frame shows the Input Gate, the area in the blue frame shows the Output Gate and the area in the black frame shows the Cell State structures.

The LSTM model consists of 4 structures. These structures are shown in Figure 4 and are as follows:

- Forget Gate,
- Output Gate,
- Input Gate,
- Cell State.

The Cell State is a communication line that carries meaningful information across cells to make predictions. The cell state is simply the memory of the model. The short-term memory problem is solved thanks to the cell state, and old data is moved along the network chain. The information that needs to be carried out with the help of the Cell State is determined through the gates. Through these gates, it can determine what information is necessary or unnecessary. On the other hand, Gates uses the sigmoid function, which compresses the data to a range of 0 to 1. The sigmoid function is an activation function. As a result of the sigmoid function, the information that is 0 is forgotten, and the information that is 1 continues to progress with the Cell State.

Forget Gate is the gateway to decide which information to forget or keep. It processes the information from the previous cell (h_t) and the current information (x_t) with the sigmoid activation function. As a result of the operations, the information that is 0 is forgotten, and the information that is 1 continues to be transferred with the Cell State.

Input Gate assumes the function of updating the Cell State. It decides whether to update the previous and current information according to the result of the operation performed with the sigmoid function. Information with 0 is called unimportant, and information with 1 is essential. In addition, the tanh activation function, which compresses the data to the range of -1 and 1, is also used for the regulation process. In the gate region, as a final operation, the sigmoid and tanh function outputs are multiplied, and a decision is made on which information to update.

Finally, the Output Gate determines the next cell's input (h_{t+1}). Another task is to make predictions. First, the previous information and the current input information are passed through the sigmoid function. Then, the current information on the Cell State is passed through the tanh function. In the last stage, it is decided which information will be input (h_{t+1}) for the next cell by multiplying the two results.

When the gate operations for the current cell are completed, the Cell State that will go to the next cell and the Hidden State(h_t) information, defined as the cell's entrance information, are decided.

3.3 Industrial Case Study

This section describes the industrial case study conducted. Vestel Electronics, one of the largest TV manufacturers in Europe, produces TV and TV products for 157 brands from 145 different countries. In addition to producing approximately 5 million televisions annually, the software is tested by the relevant test groups for different innovative television projects. As the Software Quality Assurance Department is affiliated with Vestel Electronics, the main problem is creating test scenarios for software tests for automation tests. Because of the density of the software to be tested, the sufficient number of human resources, and the tight schedule of deadlines, software tests must be carried out in minimum time and with maximum effect. In line with this, our primary goal is to create automation test scenarios that find a maximum error in the minimum time.

Manual tests are more advantageous for finding faults, but the use of human resources and limited time are the most significant disadvantages of these tests. On the other hand, automation tests allow 24/7 testing without a workforce. Nevertheless, the probability of missing errors is high. In this direction, our research aims to create automation test scenarios that can be done in minimum time by increasing the defect detection rate for Black-box tests for Smart TVs.

Critical errors found in the minimum time are reported, allowing the stability of the TV software to be increased. By saving the time that needs to be spent, the time required for testing the software, which includes the solution of the related errors, is also gained. In the next section, we introduce our experimental setup.

CHAPTER IV

EXPERIMENTS

All test groups developed a new test scenario creation method at the Vestel Electronics TV factory using the Long Short Term Memory (LSTM) Model to eliminate errors in previous software tests. The LSTM model, one of the artificial intelligence models, was used in developing this management. It has been determined to use this model for a large amount of data and an ordinal output estimation.

In this section, first of all, the Experimental Setup(section 4.1) is introduced, and information is given about how the data is taken, how the model is trained, the technical characteristics of the data, and the integration of the data into the model. Then, the Application Management (section 4.2) is explained, the results and outputs of the model are emphasized, and the test scenarios are compared. Finally, the results are compared in the Result and Discussion section(section 4.3).

4.1 Experimental Setup

This study was conducted using an Asus ExpertBook computer with Intel(R) Core(TM) i5-10210U CPU @ 1.60GHz 2.11 GHz operating system. While using the programs, the Tensorflow library was used.

Our studies have been implemented in all innovative TV projects using the results obtained from the data of 4 different Smart TV projects made by Vestel Electronics. First, crash\reset\hang errors in previous manual and automation tests were examined. For this, 3000 errors were examined using the system where the error log was entered and the JIRA¹ system. As a result of this examination, it was found that the

¹<https://www.atlassian.com/software/jira>

scenarios of the related errors occurred in the following steps.

When Table 1 is examined, there are ten main test steps. However, these steps are also separated as test steps in themselves. Netflix application consists of 4 steps. Amazon Prime application is four steps. The Youtube application is four steps. The Web Browser application is three steps. Channel is two steps. Fast channel navigation is three steps, incremental power and standby steps consist of 2 steps, and standby and power steps consist of 2 steps. In the following sections, only the main test steps are written for convenience.

The main steps are converted to matrix form to give the time series as input to the LSTM algorithm. It is shown in Table 2 with its details. The encoding problem is solved for this data conversion. In addition, since the outputs from the Model are given as binary, the decoding problem has been solved to bring the binary outputs to a readable point. Briefly, encoding-decoding processes were carried out to integrate the data into the system and to re-analyze the data coming out of the system.

The analyzed 3000 data were taught to the LSTM algorithm in the form below. Crash\Reset\Hang problems are represented as 0000000000. In Table 3, it is shown in an example matrix form and the test steps it expresses.

Table 3 gives scenarios of crash\hang\reset problems arising in manual and automation tests. When Table 3 is examined, there are 6 test steps in total in the first scenario. One of the Crash\Reset\Hang problems occurred in Step 7. Therefore, its equivalent in matrix form is 0000000000. The second scenario has 4 test steps in total. One of the Crash\Reset\Hang problems occurred in Step 5, and its illustration is as in the Table.

4.2 Application Management

The Tensorflow library is used for the LSTM algorithm. The version of the TensorFlow library used is V2.1.0. Programmatically, the activation function was used as

the “softmax” function, the loss function as the “categorical cross-entropy” function, the “adam” function as the Optimizer, and the metric used as “accuracy.”

The softmax function transforms any number vector into a probability vector with appropriate scaling. The purpose of using this function is to use it to classify networks. Since linear or Sigmoid activation functions are unsuitable for multi-classification structures, the Softmax activation function is used.

Since our problem is a multi-class classification problem, it is necessary to select the loss function to be used according to this problem. Therefore, the categorical cross-entropy loss function is used because both multi-class outputs and their outputs have a one-hot-encoded system with 0s and 1s, as described in the previous section.

Finally, our problem involves too much data. The adam function is used as the optimizer function to run this data correctly and to make the algorithm work in a way that requires less memory.

Various trials have been made for the LSTM model used. Furthermore, these trials were determined as “accuracy” as a metric measurement. All tried and used LSTM models were tested and trained with 3000 data obtained in the past data. The training was done with 1000 datas out of 3000 data, and testing was done with the remaining 2000 data.

In light of this information, the LSTM models we use and compare with each other are as follows.

- The first LSTM model was designed using only one LSTM layer. This model repeats the input as many times as desired by using the RepeatVector layer. In addition, in the model, the final output of the LSTM layer, along with the step time, is obtained from the dense layer. A summary study of this LSTM design is shown in Table 4. As a result of the studies carried out with this LSTM model, the “accuracy” rate has increased to 37.19%. This result means that the relevant LSTM model could predict 37.197% of the 2000 data tested

correctly. This rate is meager, revealing the necessity of developing the model. The automation test scenario created with the relevant LSTM model is shown in Table 5. Accordingly, the first test step is Netflix Application, the second test step is Power off\on, the third test step is Standby off\on, and the fourth test step is again Standby Off\on.

- The second LSTM model used was designed using only 1 LSTM layer. In addition, using the repeat vector, the operations are repeated on the original LSTM layer, and the model is ready. The time-distributed property, which is used in multi-layer situations, is used. The summary work is shown in Table 6. As a result of the studies carried out with this LSTM model, the “accuracy” rate has increased to 46.73%. This result means that the relevant LSTM model could predict 46.73% of the 2000 data tested correctly. This rate is still shallow, revealing the necessity of developing the model. The automation test scenario created with the relevant LSTM model is shown in Table 7. Accordingly, the first test step is Incremental Standby Off\on, the second test step is Youtube Application, and the third test step is Power off\on.
- The third LSTM model used was designed as a “Sequential Model,” and one more LSTM layer was designed by using a repeater on 1 LSTM layer. A summary study of this LSTM design is shown in Table 8. As a result of the studies carried out with this LSTM model, the “accuracy” rate has increased to 61.34%. This result means that the relevant LSTM model could predict 61.34% of the 2000 data tested correctly. It is necessary to improve the Accuracy rate. The automation test scenario created with the relevant LSTM model is shown in Table 9. The first step is the Web Browser Application, the second step is Incremental Power Off\On, the third step is Standby Off\On, and the last test step is Youtube Application.

- The fourth LSTM model used is designed as a multi-Layer LSTM Model, the output of the first LSTM layer is provided to the next, and at the same time, the final state of the second LSTM layer is provided as the initial state to the next LSTM layer. A summary study of this LSTM design is shown in Table 10. As a result of the studies carried out with this LSTM model, the “accuracy” rate has increased to 69.77%. This result means that the relevant LSTM model could predict 69.77% of the 2000 data tested correctly. The scenario obtained with the LSTM model is named Test Scenario 1 (TS1).

With the automation test cases according to the first three LSTM Models, a trial was conducted in one Vestel TV project. The tested TV project is the same as the TV project used in the other tests, and its name is shown as Vestel TV Project 1 (VTP1). This experiment aims to see that although the accuracy of the first three LSTM models is low, the error rate found according to the accuracy rate is parallel when compared with each other. The results of the tests carried out regarding the situation are shown in Table 11 and Figure 5.

To diversify the test scenarios, the probabilities in the last layer of the LSTM model(The LSTM Model, which has the highest accuracy rate among the models described above, the Fourth LSTM Model) were compared, and four more different test scenarios were created. In this creation process, the test steps with the highest probability were removed from each test step, and the remaining test steps were randomized as probabilistic in the relevant test steps. The test scenarios that emerged as a result of these processes; It is named Test Scenario 2(TS2), Test Scenario 3(TS3), Test Scenario 4(TS4), and Test Scenario 5(TS5).

The automation test scenario created with the fourth LSTM model, which is the LSTM model with the highest accuracy rate and the test scenarios obtained by applying the probabilistic randomization process of the steps in the last layer using this model are shown in Table 12.

4.3 Results and Discussion

It has been seen that the type and structure of the LSTM Model to be used are critical in the efficient and effective operation of the system. It has been proven that the model's efficiency with multi-layer structures is noticeably high.

The tests were started with test scenarios created with the first 3 LSTM Models with the lowest accuracy rate. The results As expected, the error rates found by the systems according to the accuracy rate showed parallelism. In the tests conducted with Vestel TV project 1 (VTP1), produced at the Vestel TV Factory, the first LSTM Model with an accuracy rate of 37.197% found 31 errors as a result of the 3-day test. The second LSTM Model, with an accuracy rate of 46.73%, found 55 errors due to the 3-day test. The third LSTM Model 3, with an accuracy rate of 61.34%, found 125 errors due to the 3-day test. In short, the automation test scenario with the third LSTM model, which has the highest accuracy rate, found the highest number of errors. The results related to the situation are shown in Table 11 and Figure 5.

Tests continued with the fourth LSTM Model, the LSTM Model with the highest accuracy obtained, and comparisons were made with Vestel Tests, which are traditionally applied in the Vestel Electronics SWQA Department. The accuracy of the fourth LSTM model used is 69.77%. This is because the study aims to create a test case that finds the highest number of errors. Since the accuracy rate of the models and the number of errors found in the previous study were proven to be parallel, the second stage was started.

The tests were conducted specifically for four different Smart TV projects, which are Vestel products. These projects are named as TV project 1(TVP1), TV project 2(TVP2), TV project 3(TVP3), TV project 4(TVP4). The model and the test scenarios obtained by performing the probabilistic randomization process of the model

were compared with the fixed automation tests applied by the Software Quality Assurance (SWQA) Department of Vestel Electronics. The four automation test scenarios applied by the SWQA department are the torture test scenarios among the test scenarios used. The names of these scenarios are named as Vestel Test scenario 1 (VTS1), Vestel Test scenario 2 (VTS2), Vestel Test scenario 3 (VTS3), and Vestel Test scenario 4 (VTS4). Due to the confidentiality issue, the test steps of the torture test scenarios used are not given. A common test period of 3 days has been determined for the automation tests performed in each project. In total, nine automation tests were run for each project, and the duration of each test was determined as three days. All automation tests were completed in 45 days in total. The results of the test automation scenario tests obtained on the projects are given in Table 13. In addition, the graphical representation of this information is given in Figure 6.

When all the results are compared, the automation test scenario (TS1) created with the fourth LSTM model, which has the highest accuracy rate, found the highest number of errors in the tests of all projects. Immediately after, the results of the fourth LSTM model, which is relevant, were replaced by the test scenarios (TS2, TS3, TS4, TS5) obtained with the help of the probabilistic randomization process. Test cases used by the Software Quality Assurance Department (SWQA) were insufficient to find Crash\Reset\Hang errors when comparing automation test results. However, these errors are at a level that will affect the stability of the project. Because the relevant software teams resolve the errors found with manual and automation test scenarios after they are reported, these solutions sometimes cause side effects when transferred to the following software.

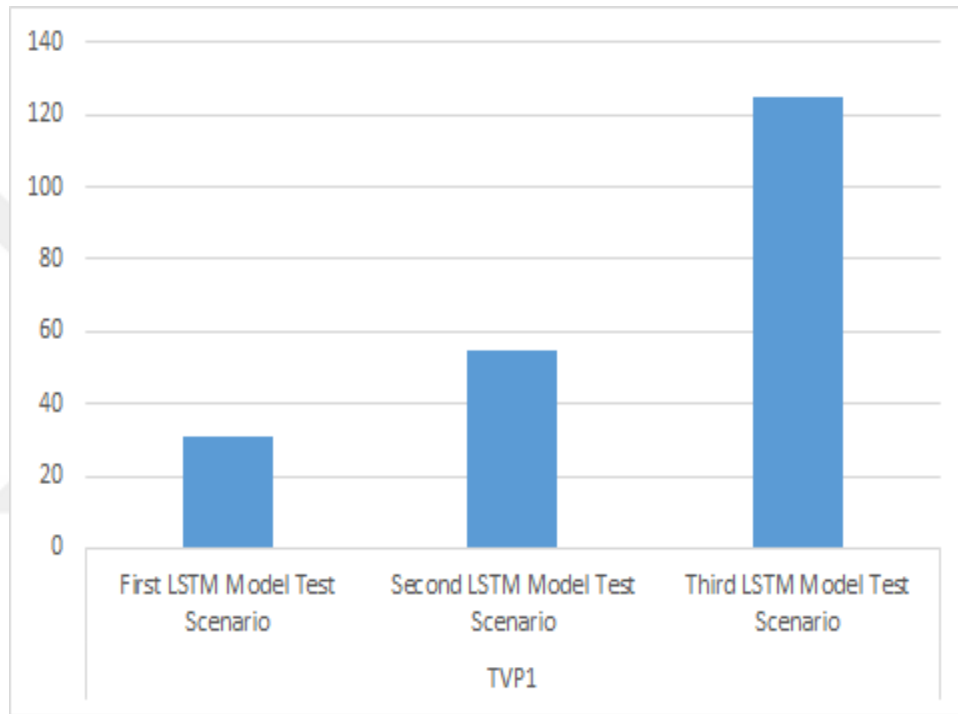


Figure 5: Test with Vestel TV Project 1(VTP1) with test scenarios with different LSTM Models. Test with Vestel TV Project 1(VTP1) with test scenarios with different LSTM Models. The LSTM Models used are the first, second, and third LSTM Models with the lowest accuracy. The automation test scenarios that emerged with these LSTM Models found errors parallel with their accuracy. The test scenario produced with the first LSTM model with the lowest accuracy found the least number of errors. The test scenario produced with the third LSTM Model, which had the highest accuracy among them, found the highest number of errors.

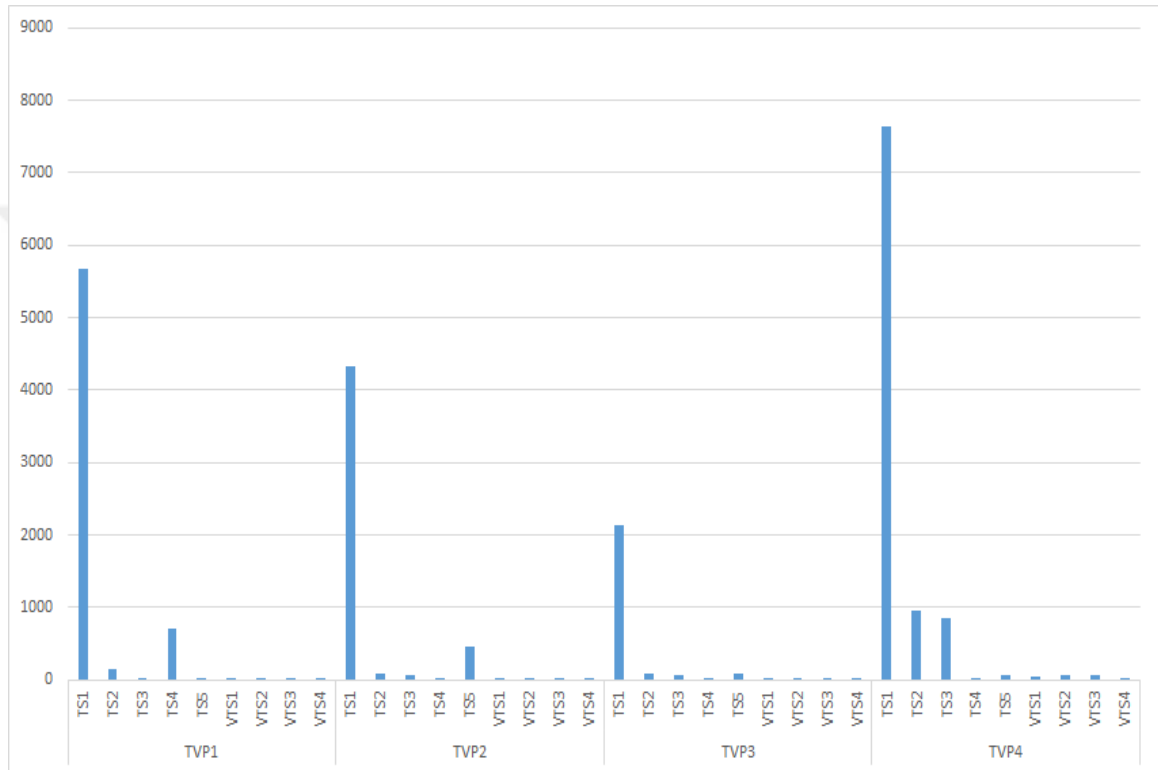


Figure 6: The figure of the test result of TV Project 1 & TV Project 2 & TV Project 3 & TV Project 4. Automation test scenarios (TS) have been tested in Vestel TV projects with different features (VTP(first TV project(VTP1), second TV project(VTP2), third TV project(VTP3), fourth TV project (VTP4))). Test scenarios obtained with the primary LSTM model (TS1), test scenarios obtained by the probabilistic randomization of the steps in the last layer of the primary LSTM model (TS2, TS3, TS4, TS5), classical automation tests used in the Vestel SWQA department (VTS(VTS1, VTS2, VTS3, VTS4)). As expected, the test scenario that found the highest number of errors was the test scenario created with the LSTM model.

When all the results were examined, two main effects emerged. First of all, in the tests made with test scenarios to be used in automation tests created with the LSTM model, errors belonging to different root causes were found. Secondly, these errors were multiplied, and detailed data were created to be given to the development group to solve the error. In other words, when the classically applied Vestel test scenarios were compared with the test scenarios created with the LSTM model, both different errors were found in the automation tests applied with the test scenarios created with the LSTM model, and a sufficient amount of data was generated for the solution by multiplying the found errors.

To summarize this section, the accuracy rate of different models made while establishing the LSTM model architecture and the error rate found with the automation test scenarios obtained from these models increase in parallel. In other words, the higher the accuracy of a model, the higher the test scenario and error. In addition, the increase in error rates and error types provides benefits to increase the stability of smart TVs, and the time for the related software groups to resolve errors is reduced. Therefore, the shorter the time required for testing and the time required to resolve bugs, the quicker new projects come into play and increase the system's stability. When all these results are compared, it is of great benefit to using test scenarios created with the LSTM algorithm instead of the classical tests of the SWQA department for stability tests. The following section is Conclusion, and the last section, a summary of the work done and information about future work, will be given.

Table 1: Main Test Steps Revealed When 3000 Error Records Are Examined and Detailed Test Steps Included in These Main Test Steps.

Main Test Steps	Detailed Test Steps
Netflix Application	Open Netflix Application
	Enter Launcher menu
	Select any content
	Play Video
Amazon Prime Application	Open Amazon Prime Application
	Enter Launcher menu
	Select any content
	Play Video
Youtube Application	Open Youtube Application
	Enter Launcher menu
	Select any content
	Play Video
Standby Off\On	Apply Standby off
	Apply Standby on
Power Off\On	Apply DC Power off
	Apply DC Power on
Incremental Standby Off\On	Tune any TV channel
	send incremental Standby command
Incremental Power Off\On	Tune any TV channel
	send incremental DC Power command
Channel Fast Navigation	Tune Any TV channel
	Program +(P+) is performed at short intervals for a certain period of time.
	Program -(P-) is performed at short intervals for a certain period of time.
Web browser Application	Open Web browser Application
	Enter any websites
	Select any content
Channel	Tune any TV channel
	Watch any content

Table 2: Required Matrix Form to Input the Main Test Steps as Input to the LSTM Model.

Main Test Steps	Matrix Representation
Youtube Application	1000000000
Netflix Application	0100000000
Web browser Application	0010000000
Channel fast navigation	0001000000
Standby off\on	0000100000
Incremental standby off\on	0000010000
Amazon Prime Application	0000001000
Inremental Power off\on	0000000100
Power off\on	0000000010
Channel	0000000001

Table 3: Matrix Form of 2 Different Test Scenarios: The Matrix Form of 3000 Scenarios Seen in Manual and Automation Tests.

Number of Test Scenario	Matrix form of Test Case of oldest Datas	Test Steps
1	0 0 0 0 0 0 0 0 0 1	Channel
	0 0 0 0 0 0 1 0 0 0	
	0 0 0 0 0 0 0 1 0 0	
	0 0 1 0 0 0 0 0 0 0	
	0 0 0 0 0 0 0 0 0 1	
	0 0 0 0 0 1 0 0 0 0	
0 0 0 0 0 0 0 0 0 0		
2	0 0 0 1 0 0 0 0 0 0	Channel Fast Navigation
	0 0 0 0 0 1 0 0 0 0	
	0 0 0 1 0 0 0 0 0 0	
	0 0 0 0 0 0 0 1 0 0	
	0 0 0 0 0 0 0 0 0 0	
	Amazon Prime Application	
	Incremental Power Off\On	
	Web Browser application	
	Channel	
	Incremental Standby Off\On	
	Incremental Standby Off\On	
	Channel Fast Navigation	
	Incremental Power Off\On	

Table 4: The summary of the first LSTM model used was designed using only one LSTM layer. This model repeats the input as many times as desired by using the RepeatVector layer. In addition, in the model, the final output of the LSTM layer, along with the step time, is obtained from the dense layer.

Layer (type)	Output Shape	Param #
lstm_1 (LSTM)	(None, 150)	96600
repeat_vector_1 (RepeatVector)	(None, 15, 150)	0
dense_1 (Dense)	(None, 15, 150)	1510
Total params: 98,100		
Trainable params: 98,100		
Non-trainable params: 0		

Table 5: Table of Automation Test scenario created according to the first LSTM Model. The test scenario formed according to this model consists of 4 steps. The test steps are Netflix Application, Power Off\On, Standby Off\On, and the last step is Standby Off\On.

Automation Test Scenario		
Test Scenario	Test Steps Number	Test Steps
First LSTM Model Test Scenario	1	Netflix Application
	2	Power Off\On
	3	Standby Off\On
	4	Standby Off\On

Table 6: Summary of the second LSTM model used was designed using only 1 LSTM layer. In addition, using the repeat vector, the operations are repeated on the original LSTM layer, and the model is ready. The time-distributed option property, which is used in multi-layer situations, is used.

Layer (type)	Output Shape	Param #
lstm_1 (LSTM)	(None, 150)	96600
repeat_vector_1 (RepeatVector)	(None, 15, 150)	0
time_distributed_1 (TimeDist)	(None, 15, 10)	1510
Total params: 98,110		
Trainable params: 98,110		
Non-trainable params: 0		

Table 7: Table of Automation Test scenario created according to the Second LSTM Model. The test scenario formed according to this model consists of 3 steps. The test steps are Incremental Standby Off\On, Youtube Application, and Power Off\On.

Automation Test Scenario		
Test Scenario	Test Steps Number	Test Steps
Second LSTM Model Test Scenario	1	Incremental Standby Off\On
	2	Youtube Application
	3	Power Off\On

Table 8: Summary of The third LSTM model used was designed as a “Sequential Model,” and one more LSTM layer was designed by using a repeater on 1 LSTM layer.

Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 150)	96600
repeat_vector (RepeatVector)	(None, 15, 150)	0
lstm_1 (LSTM)	(None, 15, 150)	180600
time_distributed (TimeDistri	(None, 15, 10)	1510
Total params: 278,710		
Trainable params: 278,710		
Non-trainable params: 0		

Table 9: Table of Automation Test scenario created according to the Third LSTM Model. The test scenario formed according to this model consists of 4 steps. The test steps are Web Browser Application, Incremental Power Off\On, Standby Off\On, and Youtube Application.

Automation Test Scenario		
Test Scenario	Test Steps Number	Test Steps
Third LSTM Model Test Scenario	1	Web Browser Application
	2	Incremental Power Off\On
	3	Standby Off\On
	4	Youtube Application

Table 10: Summary of the fourth LSTM model used is designed as Multi-Layer LSTM Model, the output of the first LSTM layer is provided to the next, and at the same time, the final state of the second LSTM layer is provided as the initial state to the next LSTM layer.

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	[(None,1,10)]	0	
lstm_2 (LSTM)	[(None,1, 150)]	(No 96600	input_1 [0][0]
lstm_3 (LSTM)	(None,1, 150)	180600	lstm_2 [0][0]
			lstm_2 [0][1]
			lstm_2 [0][2]
time_distributed_1 (TimeDistrib	(None,1, 10)	1510	lstm_3 [0][0]
Total params: 278,710			
Trainable params: 278,710			
Non-trainable params: 0			

Table 11: Table of Test with Vestel TV Project 1(VTP1) with test scenarios with different LSTM Models. Test with Vestel TV Project 1(VTP1) with test scenarios with different LSTM Models. The LSTM Models used are the first, second, and third LSTM Models with the lowest accuracy. The automation test scenarios that emerged with these LSTM Models found errors parallel with their accuracy. The test scenario produced with the first LSTM model with the lowest accuracy found the least number of errors. The test scenario produced with the third LSTM Model, which had the highest accuracy among them, found the highest number of errors.

TV Project	Test Scenario	Number of errors that occur
TVP1	First LSTM Model Test Scenario	31
	Second LSTM Model Test Scenario	55
	Third LSTM Model Test Scenario	125

Table 12: Automation Test Scenarios were obtained by Arranging the Last Layer of the LSTM Model as Probabilistic(TS2, TS3, TS4, TS5), and the Automation Test Scenario was given as the output of the program according to the LSTM Model(TS1).

Automation Test Scenario		
Test Scenario	Test Steps Number	Test Steps
TS1	1	Channel Fast Navigation
	2	Power Off\On
	3	Netflix Application
	4	Standby Off\On
	5	Channel Fast Navigation
	6	Incremental Power Off\On
TS2	1	Web browser Application
	2	Standby Off\On
	3	Incremental Power Off\On
	4	Amazon Prime Application
TS3	1	Youtube Application
	2	Power Off\On
	3	Incremental Power Off\On
	4	Channel Fast Navigation
	5	Netflix Application
TS4	1	Amazon Prime Application
	2	Youtube Application
	3	Incremental Standby Off\On
	4	Youtube Application
TS5	1	Web browser Application
	2	Channel Fast Navigation
	3	Standby Off\On
	4	Youtube Application
	5	Incremental Power Off\On
	6	Channel Fast Navigation

Table 13: Automation test scenarios have been tested in Vestel TV projects with different features (first TV project(VTP1), second TV project(VTP2), third TV project(VTP3), and fourth TV project (VTP4). Test scenarios obtained with the primary LSTM model (TS1), test scenarios obtained by the probabilistic randomization of the steps in the last layer of the primary LSTM model (TS2, TS3, TS4, TS5)), classical automation tests used in the Vestel SWQA department(VTS1, VTS2, VTS3, VTS4).

TV Project	Test Scenarios	Number of errors that occur
TVP1	TS1	5679
	TS2	144
	TS3	23
	TS4	701
	TS5	16
	VTS1	13
	VTS2	15
	VTS3	5
	VTS4	7
TVP2	TS1	4329
	TS2	97
	TS3	56
	TS4	28
	TS5	457
	VTS1	9
	VTS2	25
	VTS3	4
	VTS4	17
	TVP3	TS1
TS2		83
TS3		56
TS4		8
TS5		93
VTS1		27
VTS2		35
VTS3		3
VTS4		19
TVP4	TS1	7641
	TS2	959
	TS3	862
	TS4	13
	TS5	57
	VTS1	44
	VTS2	59
	VTS3	63
	VTS4	7

CHAPTER V

CONCLUSIONS

Two significant results emerged from all the studies conducted within the scope of this thesis. The first result is that according to the accuracy rate of the LSTM model used, the amount of error found by the automation test scenarios produced with the LSTM model increased in parallel. The second result is that the automation test scenarios created by applying the LSTM model with the highest accuracy rate and the probabilistic randomization of the appropriate test steps of this model found a higher rate and number of errors than the classically applied Torture and Stress test scenarios.

In light of all these results, when the previous error scenarios are taught to the LSTM model, a Deep Learning method, it has been proven that a very efficient test scenario is created to find new errors. The new test scenarios identified the system's shortcomings by finding the maximum number of errors in the relevant smart TV projects. In the solution of these deficiencies, the system will become more stable. Therefore, stability problems, one of the biggest problems of a software system, will be solved together with the automation tests included in the black-box test concept. It has been proven that the LSTM Model shape to be used is essential in the efficient operation of the system, and the effectiveness of the model with multi-layer structures is noticeably high.

The number and type of errors found with the LSTM algorithm are high. The abundance of the types of errors found is essential for the stability of the system. The high number of errors found will increase the amount of data to be given for the problem's solution, making it easier to solve the error. In addition, when new

software that contains the solution to these errors arrives, it will be able to find the risk of side effects arising from the resolution of the errors in the first test run.

Automation test scenarios created using the system suggested in this article, instead of classical automation test scenarios, are one step closer to completing software tests by finding the maximum number of errors in a minimum time. Therefore, the concept of the basic structure of stress and load tests will get closer together with the automation test scenario created with the LSTM algorithm. Due to the reduced time, the duration of software tests will be shortened, and additional time will be created to run other tests. These newly created test scenarios have been actively used in the SWQA department of Vestel Electronics.

In our future work, we want to work on test scenarios for manual and automation tests that more easily reveal all bugs that users can see, not just crash\reset\hang problems, but also video and audio errors. In addition, we aim to compare the results with different AI and LSTM models.

Bibliography

- [1] G. Sivaraman, P. Cesar, and P. Vuorimaa, “System software for digital television applications.,” in *ICME*, 2001.
- [2] C. Ş. Gebizli, A. Kırkıcı, and H. Sözer, “Increasing test efficiency by risk-driven model-based testing,” *Journal of Systems and Software*, vol. 144, pp. 356–365, 2018.
- [3] I. Luyk - de Visser, *Analyzing user perceived failure severity in consumer electronics products : incorporating the user perspective into the development process*. PhD thesis, Industrial Design, 2008.
- [4] H. Gudaparthi, R. Johnson, H. Challa, and N. Niu, “Deep learning for smart sewer systems: Assessing nonfunctional requirements,” in *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering: Software Engineering in Society*, pp. 35–38, 2020.
- [5] A. S. Musleh, G. Chen, Z. Y. Dong, C. Wang, and S. Chen, “Attack detection in automatic generation control systems using lstm-based stacked autoencoders,” *IEEE Transactions on Industrial Informatics*, 2022.
- [6] Y. Tao, H. Chen, and C. Qiu, “Wind power prediction and pattern feature based on deep learning method,” in *IEEE PES Asia-Pacific Power and Energy Engineering Conference (APPEEC)*, pp. 1–4, IEEE, 2014.
- [7] A. Gensler, J. Henze, B. Sick, and N. Raabe, “Deep learning for solar power forecasting—an approach using autoencoder and lstm neural networks,” in *IEEE international conference on systems, man, and cybernetics (SMC)*, pp. 002858–002865, IEEE, 2016.
- [8] W. Zhang, Y. Yu, Y. Qi, F. Shu, and Y. Wang, “Short-term traffic flow prediction based on spatio-temporal analysis and cnn deep learning,” *Transportmetrica A: Transport Science*, vol. 15, no. 2, pp. 1688–1711, 2019.
- [9] T. Pham, T. Tran, D. Phung, and S. Venkatesh, “Predicting healthcare trajectories from medical records: A deep learning approach,” *Journal of biomedical informatics*, vol. 69, pp. 218–229, 2017.
- [10] Z. Li, J. Huang, Z. Zhou, H. Zhang, S. Chang, and Z. Huang, “Lstm-based deep learning models for answer ranking,” in *IEEE First International Conference on Data Science in Cyberspace (DSC)*, pp. 90–97, IEEE, 2016.
- [11] O. Balci, “Verification validation and accreditation of simulation models,” in *Proceedings of the 29th conference on Winter simulation*, pp. 135–141, 1997.
- [12] X. Li, H. Jiang, Z. Ren, G. Li, and J. Zhang, “Deep learning in software engineering,” *arXiv preprint arXiv:1805.04825*, 2018.

- [13] E. Engström, P. Runeson, and M. Skoglund, “A systematic review on regression test selection techniques,” *Information and Software Technology*, vol. 52, no. 1, pp. 14–30, 2010.
- [14] E. Rogstad, L. Briand, and R. Torkar, “Test case selection for black-box regression testing of database applications,” *Information and Software Technology*, vol. 55, no. 10, pp. 1781–1795, 2013.
- [15] L. Xiao, H. Miao, T. Shi, and Y. Hong, “Lstm-based deep learning for spatial-temporal software testing,” *Distributed and Parallel Databases*, vol. 38, no. 3, pp. 687–712, 2020.
- [16] M. Barth and A. Fay, “Automated generation of simulation models for control code tests,” *Control Engineering Practice*, vol. 21, no. 2, pp. 218–230, 2013.
- [17] A. Demetriou, H. Alfsvåg, S. Rahrovani, and M. H. Chehreghani, “A deep learning framework for generation and analysis of driving scenario trajectories,” *arXiv preprint arXiv:2007.14524*, 2020.
- [18] S. K. Singh and A. Chaturvedi, “Applying deep learning for discovery and analysis of software vulnerabilities: A brief survey,” *Soft Computing: Theories and Applications*, pp. 649–658, 2020.
- [19] H. Kim, J. Cho, and J. Park, “Application of a deep learning technique to the development of a fast accident scenario identifier,” *IEEE Access*, vol. 8, pp. 177363–177373, 2020.
- [20] S. Elbaum, G. Rothmel, S. Karre, and M. Fisher II, “Leveraging user-session data to support web application testing,” *IEEE Transactions on Software Engineering*, vol. 31, no. 3, pp. 187–202, 2005.
- [21] S. Sprenkle, E. Gibson, S. Sampath, and L. Pollock, “Automated replay and failure detection for web applications,” in *Proceedings of the 20th IEEE/ACM International Conference On Automated Software Engineering*, pp. 253–262, 2005.
- [22] K. Meinke and M. A. Sindhu, “Lbtest: a learning-based testing tool for reactive systems,” in *IEEE Sixth International Conference on Software Testing, Verification and Validation*, pp. 447–454, IEEE, 2013.
- [23] H. Robinson, “Intelligent test automation,” *Software Testing & Quality Engineering Magazine*, pp. 24–32, 2000.
- [24] A. Tinkham and C. Kaner, “Learning styles and exploratory testing,” in *Proceedings Of The Pacific Northwest Software Quality Conference*, 2003.
- [25] C. Agruss and B. Johnson, “Ad hoc software testing: A perspective on exploration and improvisation,” in *Florida Institute Of Technology*, Citeseer, 2000.

- [26] J. Itkonen, M. V. Mantyla, and C. Lassenius, “Defect detection efficiency: Test case based vs. exploratory testing,” in *First International Symposium on Empirical Software Engineering and Measurement (ESEM)*, pp. 61–70, IEEE, 2007.
- [27] J. Itkonen *et al.*, “Empirical studies on exploratory software testing,” 2011.
- [28] C. S. Gebizli and H. Sözer, “Improving models for model-based testing based on exploratory testing,” in *IEEE 38th International Computer Software and Applications Conference Workshops*, pp. 656–661, IEEE, 2014.
- [29] C. Ş. Gebizli and H. Sözer, “Automated refinement of models for model-based testing using exploratory testing,” *Software Quality Journal*, vol. 25, no. 3, pp. 979–1005, 2017.
- [30] M. F. Kıraç, B. Aktemur, H. Sözer, and C. Ş. Gebizli, “Automatically learning usage behavior and generating event sequences for black-box testing of reactive systems,” *Software Quality Journal*, vol. 27, no. 2, pp. 861–883, 2019.
- [31] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.
- [32] J. Schmidhuber, “Habilitation thesis: System modeling and optimization,” *Page 150 ff demonstrates credit assignment across the equivalent of 1,200 layers in an unfolded RNN*, 1993.
- [33] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

VITA

Yusuf Çağlayan received a Bachelor of Science in Electrical and Electronics Engineering Department from Dokuz Eylul University in June 2019. He worked in a Software Quality Assurance area and continued his career in software testing. Currently, he is a Software Quality Assurance Specialist at the Software Quality Assurance Department of TV Factory at Vestel Electronics, one of Europe's largest TV manufacturers. His main area of research interest is Software Testing. In particular, he researches improving test case generation and prioritization to increase test efficiency.