

**LOGLEARN: PREDICTING COMPUTER NODE  
FAILURES USING CONTINUOUS MACHINE  
LEARNING**

Kumararatnam Kabilesh

219347D

Master of Science in Computer Science

Department of Computer Science  
Faculty of Engineering

University of Moratuwa  
Sri Lanka

July 2023



**LOGLEARN: PREDICTING COMPUTER NODE  
FAILURES USING CONTINUOUS MACHINE  
LEARNING**

Kumararatnam Kabilesh

219347D

Thesis submitted in partial fulfillment of the requirements for the degree  
Master of Science in Computer Science

Department of Computer Science  
Faculty of Engineering

University of Moratuwa  
Sri Lanka

July 2023

## **DECLARATION**

I declare that this is my own work and this Thesis does not incorporate without acknowledgement any material previously submitted for a Degree or Diploma in any other University or Institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text. I retain the right to use this content in whole or part in future works (such as articles or books).

Signature:

Date: 29-07-2023

The supervisor should certify the Thesis with the following declaration.

The above candidate has carried out research for the Master of Science in Computer Science Thesis under my supervision. I confirm that the declaration made above by the student is true and correct.

Name of Supervisor: Prof. Indika Perera

Signature of the Supervisor:

Date:



## **ACKNOWLEDGEMENT**

I would like to express my sincere gratitude to everyone who contributed to the completion of this thesis.

First and foremost, I would like to thank my supervisors, Dr. Gayashan Amarasinghe and Prof. Indika Perera, for their guidance, patience, and support throughout my research. Their invaluable advice and feedback were instrumental in shaping my research and improving my writing skills.

I would thank all the members of the faculty and staff of the Department of Computer Science, Faculty of Engineering, University of Moratuwa for providing a supportive and stimulating academic environment that allowed me to pursue my research interests.

Finally, I would like to express my heartfelt gratitude to my friends and family for their unwavering encouragement and support throughout my academic journey. Their words of encouragement and motivation kept me going during the challenging times, and I am forever indebted to them.



## ABSTRACT

Ensuring reliability, availability, and fault-tolerance is crucial in modern computer systems. Despite the substantial efforts put into the development, testing, and operation, failures still occur during runtime, leading to significant consequences. To address this issue, a proactive approach is necessary to predict and prevent failures before they happen. System and software logs provide essential data for monitoring systems and their performance during runtime. However, processing this information in real-time poses a unique challenge for machine learning because of the properties of streaming big data such as logs.

Therefore, this study utilizes the continuous machine learning paradigm to develop a failure prediction model called LogLearn, which uses system log data. The design and development of LogLearn consider the drawbacks and limitations of current continuous machine learning models to provide a more efficient and accurate approach to predicting computer node failures and their potential root cause with a high lead time.

The LogLearn model is implemented with an online failure prediction method, which is evaluated using multiple algorithms. Logistic regression showed the best performance in prediction. The LogLearn model outperformed previous studies' models in terms of accuracy, precision, recall, and f1-score. Additionally, an online time-series prediction model using the SNARIMAX algorithm was implemented to forecast the potential time of failure. Although previous studies have shown promising results, their lead times were insufficient to fix the underlying cause of failure in advance. Thus, LogLearn provides a viable alternative approach for failure prediction in computer systems.

**Keywords:** System logs, data streams, failure prediction, anomaly detection, continuous machine learning





## TABLE OF CONTENTS

Declaration of the Candidate & Supervisor	i
Acknowledgement	iii
Acknowledgements	iii
Abstract	v
Abstract	v
Table of Contents	vii
1 Introduction	1
1.1 Background and context	1
1.2 Research problems and objectives	1
1.2.1 Challenges in processing big data streams for machine learning	1
1.2.2 Problems in current approaches for predicting failures of computer systems	2
1.2.3 Research objectives	3
1.3 Thesis structure	4
2 Related Work	5
2.1 Understanding servers, nodes, and virtual machines in computing environments	5
2.2 System logs for node failure prediction	6
2.2.1 Linux system logs	7
2.3 Continuous machine learning for system failure prediction	8
2.3.1 What is continuous machine learning?	8
2.3.2 Advantages of using continuous machine learning for system failure prediction	9
2.4 Concept drift in continuous machine learning	10
2.5 Catastrophic forgetting in continuous machine learning	14
2.6 Characteristics of continuous machine learning models	16
2.7 Categorization of continuous machine learning paradigms	17

2.7.1	Rehearsal methods	17
2.7.2	Regularization methods	17
2.7.3	Architectural methods	18
2.7.4	Memory methods	21
2.7.5	Knowledge distillation methods	21
2.7.6	Hybrid methods	22
2.8	Anomaly detection using continuous machine learning	23
2.9	Failure prediction using continuous machine learning	24
3	Methodology	31
3.1	LogLearn: A model for predicting computer node failures using continuous machine learning	31
3.2	Streaming data to the model	31
3.2.1	Streaming system logs	31
3.2.2	Streaming system performance metrics	31
3.3	Parsing and labeling logs	32
3.3.1	Log parsing	32
3.3.2	Labeling failure logs	34
3.4	Training and prediction	38
3.4.1	Failure classification	38
3.4.2	Lead time prediction	38
3.4.3	Root cause prediction	43
4	Experiments and evaluation	45
4.1	Experiments on model performance	45
4.1.1	Evaluation of failure classification	45
4.1.2	Evaluation of lead time prediction	46
4.2	Results and discussion	58
5	Conclusion	61
5.1	Conclusion	61
5.2	Future work	62
	References	63

## LIST OF FIGURES

Figure 2.1	System log directory in Ubuntu	7
Figure 2.2	Illustrative example of concept drift [1]	11
Figure 2.3	Real and virtual concept drifts [2]	11
Figure 2.4	Different types of drifts depending on sharpness and frequency of change [2]	12
Figure 2.5	Prequential scheme of drifts [1]	13
Figure 2.6	Framework for concept drift detection [3]	14
Figure 2.7	Preventing catastrophic forgetting in continual learning of natural language tasks [3]	16
Figure 2.8	Preventing catastrophic forgetting in federated continuous-class learning [3]	16
Figure 2.9	Training and inference using episodic memory [4]	18
Figure 2.10	SOINN - 01 [5]	20
Figure 2.11	SOINN - 02 [5]	20
Figure 2.12	SOINN - 03 [5]	20
Figure 2.13	Adaptive Deep Forest [6]	22
Figure 2.14	DeepLog Architecture [7]	23
Figure 2.15	LogLens Architecture [8]	24
Figure 2.16	Aarohi - Offline and Online [9]	25
Figure 2.17	Bayesian network for determining the root cause [10]	27
Figure 2.18	Time series of the signals in the streaming data [10]	27
Figure 2.19	Desh with LSTM [11]	28
Figure 2.20	Hardware failure prediction [12]	29
Figure 2.21	Offline model for training [13]	30
Figure 2.22	Online model for prediction [13]	30
Figure 3.1	Streaming Data to Prediction Model	32
Figure 3.2	Failure cluster 1	37
Figure 3.3	Failure cluster 2	37
Figure 3.4	Warning cluster	37
Figure 3.5	Information log cluster	37
Figure 3.6	Failure Prediction Model	39
Figure 3.7	Actual CPU usage vs Time	40

Figure 3.8	Actual average CPU usage between 13.00-16.00 vs Time	41
Figure 3.9	Actual memory usage vs Time	41
Figure 3.10	Actual average memory usage between 13.00-16.00 vs Time	42
Figure 3.11	Markov model for root cause prediction [14]	43
Figure 4.1	CPU Usage vs Time	47
Figure 4.2	Memory Usage vs Time	47
Figure 4.3	CPU usage (Dickey-Fuller test)	49
Figure 4.4	Memory usage (Dickey-Fuller test)	49
Figure 4.5	Rolling mean and standard deviation for CPU usage (KPSS test)	51
Figure 4.6	Rolling mean and standard deviation for memory usage (KPSS test)	52
Figure 4.7	Forecast after 100 messages	53
Figure 4.8	Forecast after 500 messages	54
Figure 4.9	Forecast after 1000 messages	54
Figure 4.10	Forecast after 5000 messages	55
Figure 4.11	Forecast after 10000 messages	55
Figure 4.12	Forecast after 20000 messages	56
Figure 4.13	Forecast after 30000 messages	56
Figure 4.14	Forecast after 40000 messages	57
Figure 4.15	Forecast after 50000 messages	57

## LIST OF TABLES

Table 2.1	Properties of continuous machine learning	17
Table 3.1	System logs	33
Table 3.2	Parsed system logs	33
Table 4.1	Model evaluation for failure prediction	46
Table 4.2	Results of Dickey-Fuller Test	48
Table 4.3	Results of KPSS Test	51
Table 4.4	Lead time to failure in previous studies	58
Table 4.5	Performance of prediction models by previous studies	59

## LIST OF ALGORITHMS

1	SOINN+ Algorithm [15]	19
---	-----------------------	----