

LB/TH/01/2026
TH6097

**META-LEARNING-DRIVEN HYPERPARAMETER
TUNING FOR ROBUST AND SCALABLE
CONCEPT DRIFT TRACKING**

Dileesha Navindra Kannangara

248019U

Master of Science: Major Component of Research

Department of Computer Science and Engineering
Faculty of Engineering

University of Moratuwa
Sri Lanka

June 2025

META-LEARNING-DRIVEN HYPERPARAMETER TUNING FOR ROBUST AND SCALABLE CONCEPT DRIFT TRACKING

Dileesha Navindra Kannangara

248019U

Thesis submitted in partial fulfillment of the requirements for the degree
Master of Science: Major Component of Research

Department of Computer Science and Engineering
Faculty of Engineering

University of Moratuwa
Sri Lanka

June 2025

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).

06/09/2025

Signature:

Date:

The supervisors should certify the Thesis with the following declaration.

The above candidate has carried out research for the Master of Science: Major Component of Research Thesis under our supervision. We confirm that the declaration made above by the student is true and correct.

Name of Supervisor: Dr.Uthayasanker Thayasivam

Signature of the Supervisor:

Date: 08 Sep 2025

nanaruban Rajadurai

Name of Supervisor: Dr.Sumanaruban Rajadurai

I 10 Sep 2025

Signature of the Supervisor:

Date:

Name of Supervisor: Dr.Rajitha Navarathna

10 Sep 2025

Signature of the Supervisor:

Date:

ACKNOWLEDGEMENT

I would like to extend my heartfelt appreciation to my principle supervisor, Dr. T. Uthayasanker, for his invaluable guidance, steadfast mentorship, and unwavering support throughout the course of this research. His encouragement and insight were instrumental in navigating the challenges of this work, and his commitment to excellence continuously pushed me to elevate the quality of my efforts.

I would also like to express my deepest gratitude to my external supervisors, Dr. Sumanaruban Rajadurai and Dr. Rajitha Navarathna, whose invaluable guidance, insightful advice, and unwavering encouragement have been instrumental throughout the course of my research. Their mentorship not only enriched my academic journey but also inspired me to strive for greater heights. I am truly grateful for their patience, constructive feedback, and constant support, which have been pivotal in overcoming challenges and achieving this milestone.

I am also deeply grateful to the esteemed members of the progress review committee, Dr. Thanuja Ambegoda and Dr. Charith Chitraranjan, for their thoughtful feedback and constructive advice. Their expertise and perspectives played a pivotal role in shaping the direction and refinement of this study.

Finally, I wish to sincerely thank my family and friends for their constant encouragement, understanding, and belief in my abilities. Their unwavering presence has been a source of strength and inspiration, allowing me to persevere and remain committed to my academic goals.

ABSTRACT

Detecting concept drift is a critical task in machine learning, particularly when working with nonstationary real-world data streams that evolve. The ability to identify such drifts promptly and accurately is vital to ensuring that deployed machine learning models remain effective. However, the performance of drift detection algorithms is highly sensitive to the choice of hyperparameters, which must be appropriately configured for each application scenario. Despite this, existing approaches to hyperparameter tuning are often impractical, lack theoretical soundness, or fail to deliver optimal results. To tackle this problem, we introduce ProtoTune—a structured, zero-touch approach for selecting optimal deployment-time hyperparameters in error rate-based drift detection algorithms. Designed to function effectively across a wide range of data stream use cases, ProtoTune eliminates the need for both offline pre-tuning and online retrospective adjustments. This is particularly important because traditional tuning methods are frequently infeasible in real-world settings, where there is often limited prior knowledge, a lack of labeled drift points, and the constraints of streaming data that cannot be revisited once processed. At the core of our approach is an error behavior descriptor, derived from early profiling of a model’s error stream. This descriptor informs the identification of a suitable hyperparameter space. We then leverage a meta-learning framework using prototypical networks to learn the relationship between the error behavior descriptor and the ideal hyperparameter configurations. This enables us to provide a scalable and effective solution for deployment-time tuning. To our knowledge, this is the first method specifically developed to address this pressing challenge. Experimental results confirm that ProtoTune consistently surpasses current practical strategies, which often perform no better than random guessing. Our findings underscore the value of leveraging historical error information and offer a systematic solution to a growing operational challenge in adaptive machine learning.

Keywords: Concept Drift Detection, Hyperparameter Optimization, Online Learning, Prototypical Networks

TABLE OF CONTENTS

Declaration of the Candidate & Supervisor	i
Acknowledgement	ii
Abstract	iii
Table of Contents	iv
List of Figures	vi
List of Tables	vii
Abbreviations	ix
1 Introduction	1
1.1 Background	1
1.1.1 Types of Drift	1
1.2 Motivation	3
1.3 Objectives	5
1.4 Research Contributions	6
1.5 Thesis Structure	6
2 Literature Survey	8
2.1 Overview and Terminology in Concept Drift	8
2.2 Literature Review on Concept Drift Detection	9
2.3 Literature Review on Hyperparameter Tuning of Concept Drift Algorithms	11
2.4 Relationship between Error Stream Behaviour and Hyperparameters of Concept Drift Detection Algorithms	13
2.5 Learning Error-Hyperparameter Relationships with Prototypical Networks for Generalization to New Use Cases	14
3 Methodology	16
3.1 Problem Setting and Background	16
3.2 Overview of the Methodology	19
3.3 Error Behavior Descriptor Construction	20

3.4	Prototypical Networks for Learning Relationships	22
4	Experimental Settings	25
4.1	Data	25
4.2	Empirical Study Settings	27
4.2.1	SEA Synthetic Dataset	28
4.2.2	SINE Synthetic Dataset	28
4.2.3	Simulating Concept Drift via Boundary Shifts	29
4.3	ProtoTune Experimental Settings	29
4.3.1	Synthesizing Datasets to Test the Methodology	30
4.3.2	Synthesizing Different Use Cases	30
4.3.3	Evaluation Setup	31
5	Results and Discussion	33
5.1	Empirical Results	33
5.1.1	Performances against different decision boundary changes	34
5.1.2	General Discussion	38
5.2	ProtoTune Results	40
5.3	Comparison with practical baselines	42
5.4	Effect of Support Set Similarity	43
5.5	ProtoTune in Action	44
6	Conclusion	46
7	Future Work	47
	References	49

LIST OF FIGURES

Figure	Description	Page
Figure 1.1	General Machine Learning training pipeline	2
Figure 1.2	Why it is important to monitor model performance?	3
Figure 3.1	Error rate-based drift detection pipeline of a deployed machine learning model. The error stream of the model is used as input to the drift detectors to produce output.	16
Figure 3.2	Error rate-based drift detection pipeline of a deployed machine learning model. The error stream of the model is used as input to the drift detectors to produce output.	18
Figure 3.3	Error rate-based drift detection pipeline of a deployed machine learning model. The error stream of the model is used as input to the drift detectors to produce output.	20
Figure 3.4	Error rate-based drift detection pipeline of a deployed machine learning model. The error stream of the model is used as input to the drift detectors to produce output.	22
Figure 3.5	Error rate-based drift detection pipeline of a deployed machine learning model. The error stream of the model is used as input to the drift detectors to produce output.	23
Figure 4.1	Meta Task generation from each synthesized data point for one use case	31
Figure 4.2	Illustration of meta-task generation under varying experimental conditions such as dataset type, noise level, number of ways, number of shots, and more. This represents only a limited subset of the full configuration space.	32
Figure 5.1	Illustration of each algorithm's results across different decision boundary changes in the SEA data set. Performance is captured here using the detection delay metric.	38
Figure 5.2	Illustration of each algorithm's results across different decision boundary changes in the SEA data set. Performance is captured here using the false alarm rate metric.	38
Figure 5.3	Illustration of each algorithm's results across different decision boundary changes in the SINE data set. Performance is captured here using the detection delay metric.	39
Figure 5.4	Illustration of each algorithm's results across different decision boundary changes in the SINE data set. Performance is captured here using the false alarm rate metric.	39

LIST OF TABLES

Table	Description	Page
Table 4.1	Concept Drift Scenarios Used in Experiments	29
Table 5.1	Comparison of parameters for different drift detection algorithms. For each algorithm, the default parameter values, best chosen parameter values, and optimal parameter value ranges identified from our study are provided.	34
Table 5.2	Performance Comparison of Detection Delay (DD) and False Alarm Rate (FAR) for Concept Drift Detection Algorithms on the SEA Dataset Across Four Concept Changes (No Noise, Optimized Parameters). Results are presented as the mean and standard deviation ($\mu \pm \sigma$) from 10 conducted experiments.	35
Table 5.3	Performance Comparison of Detection Delay (DD) and False Alarm Rate (FAR) for Concept Drift Detection Algorithms on the SINE Dataset Across Four Concept Changes (No Noise, Optimized Parameters). Results are presented as the mean and standard deviation ($\mu \pm \sigma$) from 10 conducted experiments.	35
Table 5.4	Performance Comparison of Detection Delay (DD) and False Alarm Rate (FAR) for Concept Drift Detection Algorithms on the SEA Dataset Across Four Concept Changes (With Noise, Optimized Parameters). Results are presented as the mean and standard deviation ($\mu \pm \sigma$) from 10 conducted experiments.	36
Table 5.5	Performance Comparison of Detection Delay (DD) and False Alarm Rate (FAR) for Concept Drift Detection Algorithms on the SINE Dataset Across Four Concept Changes (With Noise, Optimized Parameters). Results are presented as the mean and standard deviation ($\mu \pm \sigma$) from 10 conducted experiments.	36
Table 5.6	Performance Comparison of Detection Delay (DD) and False Alarm Rate (FAR) for Concept Drift Detection Algorithms on the SEA Dataset Across Four Concept Changes (No Noise, Default Parameters). Results are presented as the mean and standard deviation ($\mu \pm \sigma$) from 10 conducted experiments.	37

Table 5.7	Performance Comparison of Detection Delay (DD) and False Alarm Rate (FAR) for Concept Drift Detection Algorithms on the SINE Dataset Across Four Concept Changes (No Noise, Default Parameters). Results are presented as the mean and standard deviation ($\mu \pm \sigma$) from 10 conducted experiments.	37
Table 5.8	Average Accuracy results obtained using our methodology for selecting hyperparameters for each algorithm, evaluated across different levels of meta tasks.	42
Table 5.9	Maximum Accuracy results obtained using our methodology for selecting hyperparameters for each algorithm, evaluated across different levels of meta tasks.	43
Table 5.10	Detection Delay (DD) and False Alarm Rate (FA) results for DDM, EDDM, and ECDDWT algorithms with default parameters (as commonly used in practice) and our proposed <i>protoTune</i> methodology with three different meta-learning configurations. Results are presented as mean and standard deviation ($\mu \pm \sigma$) across experiments on the SEA and SINE datasets with the 8 concept changes described earlier.	44

ABBREVIATIONS

Abbreviation	Description
DDM	Drift Detection Method
EDDM	Early Drift Detection Method
CNN	Convolutional Neural Network
AI	Artificial Intelligence
ADWIN	Adaptive Windowing
KS Test	Kolmogorov-Smirnov test
CUSUM	Cumulative Summation
CPU	Central Processing Unit
HDDM	Hoeffding Drift Detection Method
FHDDM	Fast Hoeffding Drift Detection Method
PSI	Population Stability Index
KL Divergence	Kullback-Leibler divergence
PH	Page-Hinkley
ECDD	Exponential Cumulative Drift Detection
RDDM	Reactive Drift Detection Method
ECCDWT	Exponential Cumulative Drift Detection with a Warning Threshold