SELF-ADAPTIVE ENSEMBLES FOR LEARNING UNDER CONCEPT DRIFT

JOBIN WILSON

DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY DELHI
HAUZ KHAS, NEW DELHI – 110016
JUNE 2024

© Indian Institute of Technology Delhi (IITD), New Delhi, 2024

Self-adaptive Ensembles for Learning Under Concept Drift

A THESIS

submitted by

Jobin Wilson
2015EEZ8419

under the guidance of

Prof. Santanu Chaudhury
Prof. Brejesh Lall

in partial fulfilment of the requirements for the award of the degree

of

Doctor of Philosophy

Department Of Electrical Engineering
INDIAN INSTITUTE OF TECHNOLOGY DELHI

June 2024

To my better half, for your selfless support.

THESIS CERTIFICATE

This is to certify that the thesis titled **Self-adaptive Ensembles for Learning Under Concept Drift**, submitted by **Jobin Wilson**, to the Indian Institute of Technology Delhi, for the award of the degree of **Doctor of Philosophy**, is a bona fide record of the research work done by him under our supervision. In our opinion, the thesis has reached the standards fulfilling the requirements of the regulations relating to the degree. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Prof. Santanu Chaudhury
Professor
Dept. of Electrical Engineering
IIT-Delhi, 110 016

Prof. Brejesh Lall
Professor
Dept. of Electrical Engineering
IIT-Delhi, 110 016

Place:

Date:

ACKNOWLEDGEMENTS

I express my heartfelt gratitude to my supervisor Prof. Santanu Chaudhury for allowing me to conduct this research under his guidance. I am indebted for the wisdom, knowledge, motivation and unwavering guidance I received from him, despite his demanding schedule. His passion for research, pursuit of perfection, and ability to instill persistence and patience sustained me in this academic pursuit and empowered me to challenge my limits.

I want to express my sincere and deepest gratitude to my co-supervisor Prof. Brejesh Lall for his invaluable suggestions and guidance throughout this research work. His consistent encouragement, positivity, motivation, and support, both personal and administrative, kept me focused even in the most challenging phases of this work.

I thank my SRC committee members, Prof. Sumantra Dutta Roy, Prof. Sumeet Agarwal and Prof. Parag Singla for their valuable suggestions and feedback that enabled me to enhance this work significantly.

I express my deepest gratitude to Dr. Prateek Kapadia, Dr. Vinod Vasudevan and Flytxt for giving me an opportunity to pursue my doctoral studies and for guiding and supporting me in every possible way to complete this work successfully.

I am grateful to staff members of Multimedia Lab and Samsung IoT Innovation Lab at IIT Delhi, particularly Amit Kumar, Usha Bhola, Sushila Devi (late), and Anita Bisht for their administrative and personal support. I thank all my friends from the institute and colleagues from Flytxt, who helped me during the course of this work. Special thanks to Dr. Ajith Tom James, Dr. Arun Unnikrishnan, Dr. Shruti Sharma, Shubham Choudhary, Bivin Vinod and Muhammad Arif for all the generous support and insightful discussions.

My heartfelt appreciation goes to my wife, Delma Momi. She acted as an invaluable sounding board and stood as a steadfast anchor to help navigate even the biggest obstacles. Without her, this thesis would not have been possible.

I take this opportunity to thank my dear parents and extended family for their love, care, and dedicatory prayers which provided me strength, motivation and perseverance.

I bow my head before the GOD Almighty, for blessing me with health, confidence, patience, and knowledge to carry out this work successfully.

Jobin Wilson

ABSTRACT

Recent advancements in the *Internet of Things* (IoT), computing infrastructure, and communication networks have given rise to a surge in the generation, organization, and consumption of streaming data. While these advances vastly increased the volume, velocity, and variety of data being produced, it has also accelerated the need to comprehend this data to enable timely decisions and actions. Though machine learning (ML) offers a promising solution by creating models that detect patterns and make predictions, specialized algorithms are required to handle large data volume, fast arrival rate, concept drift and computational resource constraints associated with data stream mining. Building real-world ML systems generally involve several laborious and error-prone manual tasks requiring ML experts with limited supply. While automated machine learning (AutoML) tools increasingly automate the creation of ML workflows, they typically operate under the assumption that the learning task is stationary and neglects the streaming and evolving nature of data. Therefore, a demand arises for self-maintaining learning systems which identifies the optimal model pipeline and adjusts to concept drift automatically while working within limited computational resources.

Given the above challenges, this thesis introduces self-adaptive ensembles to manage concept drift in data stream mining. These ensembles automate and integrate one or more steps among feature engineering, model selection, hyperparameter tuning, multi-model decision fusion, and drift detection to facilitate continuous learning from evolving data streams while adhering to computational resource limits. To this end, multiple approaches from stream mining, ensemble learning, multi-armed bandits (MABs), and AutoML research are systematically integrated to create four novel self-adaptive ensemble frameworks.

The first work introduces an adaptive self-optimized end-to-end ML pipeline that utilizes an ensemble of gradient-boosted decision trees incorporating Sequential Model-Based Optimization (SMBO) for automatic hyperparameter tuning. It classifies large volume high cardinality data streams experiencing concept drift while operating in a lifelong ML setting with strict computational resource limits. We then introduce two computationally efficient self-adaptive heterogeneous ensembles that exploit online model selection for concept drift adaptation by posing it as a non-stationary MAB problem. While one framework uses a formulation based on ϵ-greedy bandits to balance exploration and exploitation, the other uses a fixed-budget best-arm identification-based formulation that relies on Successive Halving Algorithm (SHA). Further, a deep reinforcement learning (RL) controller based on conservative Q-learning (CQL) automates the SHA budget selection. Lastly, we present a unified self-tuning ensemble framework that combines homogeneous and heterogeneous learning,

forming a single-pass hybrid classifier ensemble. Genetic algorithm is used to automate stream-specific base learner selection and hyperparameter optimization, whereas adaptive stacking is used for ensemble decision fusion, avoiding frequent manual reconfiguration of ensembles.

Our proposed self-adaptive ensembles are computationally efficient and capable of handling various drift types, while also providing competitive predictive performance compared to other state-of-the-art concept drift ensembles. Additionally, our approach eliminates the need for users to frequently re-configure and fine-tune their ML workflows for each data stream, making it an attractive solution for high-speed data stream mining problems in resource constrained settings.

© 2024, *Indian Institute of Technology Delhi*

सार

इंटरनेट ऑफ थिंग्स (आईओटी) कम्प्यूटिंग बुनियादी ढांचे और संचार नेटवर्क में हाल की प्रगति ने स्ट्रीमिंग डेटा के उत्पादन, संगठन और खपत में वृद्धि की है। जबकि इन प्रगति ने उत्पादित किए जा रहे डेटा की मात्रा, वेग और विविधता में काफी वृद्धि की है, इसने समय पर निर्णय और कार्यों को सक्षम करने के लिए इस डेटा को समझने की आवश्यकता को भी तेज कर दिया है। हालांकि मशीन लर्निंग (एमएल) पैटर्न का पता लगाने और भविष्यवाणियां करने वाले मॉडल बनाकर एक आशाजनक समाधान प्रदान करता है, डेटा स्ट्रीम माइनिंग से जुड़े बड़े डेटा वॉल्यूम, तेजी से आगमन दर, अवधारणा बहाव और कम्प्यूटेशनल संसाधन बाधाओं को संभालने के लिए विशेष एल्गोरिदम की आवश्यकता होती है। वास्तविक दुनिया के एमएल सिस्टम के निर्माण में आम तौर पर कई श्रमसाध्य और त्रुटि-प्रवण मैनुअल कार्य शामिल होते हैं जिनके लिए सीमित आपूर्ति वाले एमएल विशेषज्ञों की आवश्यकता होती है। जबकि स्वचालित मशीन लर्निंग (ऑटोएमएल) उपकरण तेजी से एमएल कार्यप्रवाह के निर्माण को स्वचालित करते हैं, वे आम तौर पर इस धारणा के तहत काम करते हैं कि सीखने का कार्य स्थिर है और डेटा की स्ट्रीमिंग और विकसित प्रकृति की उपेक्षा करता है। इसलिए, स्व-रखरखाव शिक्षण प्रणालियों की मांग उत्पन्न होती है जो इष्टतम मॉडल पाइपलाइन की पहचान करती है और सीमित कम्प्यूटेशनल संसाधनों के भीतर काम करते हुए स्वचालित रूप से अवधारणा ड्रिफ्ट को समायोजित करती है।

ऊपर दी गई चुनौतियों को ध्यान में रखते हुए, यह थीसिस डेटा स्ट्रीम माइनिंग में अवधारणा ड्रिफ्ट को प्रबंधित करने के लिए स्व-अनुकूल इनसेम्बल्स पेश करती है। ये इनसेम्बल्स फीचर इंजीनियरिंग, मॉडल चयन, हाइपरपैरामीटर ट्यूनिंग, मल्टी-मॉडल निर्णय फ्यूजन, और ड्रिफ्ट पहचान के बीच से एक या अधिक कदम को स्वचालित और एकीकृत करते हैं ताकि समायोजित संसाधन सीमाओं का पालन करते हुए बदलते डेटा स्ट्रीम्स से निरंतर सीखने को संभव बनाए। इस उद्देश्य के लिए, स्ट्रीम माइनिंग, इनसेम्बल लर्निंग, मल्टी-आर्म्ड बैंडिट्स (MABs), और ऑटोएमएल अनुसंधान से कई दृष्टियों को एकत्रित किया गया है ताकि चार नवीन स्व-अनुकूल इनसेम्बल फ्रेमवर्क बनाया जा सके।

पहला काम एक अनुकूली स्व-अनुकूलित एंड-टू-एंड एमएल पाइपलाइन का परिचय देता है जो स्वचालित हाइपरपैरामीटर ट्यूनिंग के लिए अनुक्रमिक मॉडल-आधारित अनुकूलन (एसएमबीओ) को शामिल करते हुए ढाल-बूस्टेड निर्णय पेड़ों के एक समूह का उपयोग करता है। यह सख्त कम्प्यूटेशनल संसाधन सीमाओं के साथ आजीवन एमएल सेटिंग में काम करते हुए अवधारणा ड्रिफ्ट का अनुभव करने वाली बड़ी मात्रा में उच्च कार्डिनलिटी डेटा धाराओं को वर्गीकृत करता है। इसके बाद हम दो कम्प्यूटेशनल रूप से कुशल आत्म-अनुकूली विषम समूह पेश करते हैं जो

एक गैर-स्थिर एमएबी समस्या के रूप में प्रस्तुत करके अवधारणा बहाव अनुकूलन के लिए ऑनलाइन मॉडल चयन का फायदा उठाते हैं। जबकि एक ढांचा अन्वेषण और शोषण को संतुलित करने के लिए ε-लालची डाकुओं पर आधारित सूत्रीकरण का उपयोग करता है, दूसरा एक निश्चित-बजट सर्वश्रेष्ठ-हाथ पहचान-आधारित सूत्रीकरण का उपयोग करता है जो क्रमिक हल्विंग एल्गोरिदम पर निर्भर करता है। (SHA). इसके अलावा, रूढ़िवादी क्यू-लर्निंग (सीक्यूएल) पर आधारित एक गहरा सुदृढीकरण शिक्षण (आरएल) नियंत्रक एसएचए बजट चयन को स्वचालित करता है। अंत में, हम एक एकीकृत सेल्फ-ट्यूनिंग एन्सेम्बल फ्रेमवर्क प्रस्तुत करते हैं जो सजातीय और विषम सीखने को जोड़ता है।जेनेटिक एल्गोरिदम का उपयोग स्ट्रीम-विशिष्ट बेस लर्नर का चयन और हाइपरपैरामीटर अनुकूलन को स्वचालित करने के लिए किया जाता है, जबकि एडेप्टिव स्टैकिंग का उपयोग एन्सेम्बल निर्णय मेलजोल के लिए किया जाता है, जो एन्सेम्बल की बार-बार मैनुअल पुनर्कॉन्फ़िगरेशन से बचता है।

हमारे प्रस्तावित स्व-अनुकूली एन्सेम्बल कम्प्यूटेशनल रूप से कुशल हैं और विभिन्न प्रकार के ड्रिफ्ट को संभालने में सक्षम हैं, जबकि अन्य अत्याधुनिक अवधारणा ड्रिफ्ट एन्सेम्बल की तुलना में प्रतिस्पर्धी भविष्यसूचक प्रदर्शन भी प्रदान करते हैं। इसके अतिरिक्त, हमारा दृष्टिकोण उपयोगकर्ताओं को प्रत्येक डेटा स्ट्रीम के लिए अपने एमएल वर्कफ़्लो को बार-बार फिर से कॉन्फ़िगर करने और ठीक करने की आवश्यकता को समाप्त करता है, जिससे यह संसाधन सीमित सेटिंग्स में उच्च गति डेटा स्ट्रीम खनन समस्याओं के लिए एक आकर्षक समाधान बन जाता है।

Contents

ACKNOWLEDGEMENTS **i**

ABSTRACT **ii**

LIST OF TABLES **ix**

LIST OF FIGURES **xii**

ABBREVIATIONS **xii**

1 Introduction **1**

1.1 Prologue 1

1.2 Motivation 3

1.3 Objectives 5

1.4 Scope 6

1.5 Major Contributions 7

1.6 Thesis Organization 8

2 Literature Review **11**

2.1 Learning Under Concept Drift: An Overview 11

2.1.1 Drift Adaptation Strategies 13

2.1.2 Drift Detection Techniques 14

2.1.3 Ensembles for Concept Drift 15

2.2 Self-adaptation and Continuous Learning 16

2.2.1 Automatic Hyperparameter Tuning and AutoML for Streams 16

2.2.2 Online Ensemble Selection for Drift Adaptation 17

2.2.3 Meta-learning for Classifier Selection 19

2.2.4 Multi-armed Bandits and Bandit Based Algorithm Selection 19

2.2.5 Online Bagging Ensembles 21

2.2.6 Continual Learning and Domain Adaptation 22
2.3 Summary 23

3 Automatically Optimized Gradient Boosting Trees 24
3.1 Introduction 24
3.2 NeurIPS 2018 AutoML Challenge 24
3.2.1 Prediction Problem 25
3.2.2 Competition Structure 25
3.3 Related Work 26
3.4 Methodology 28
3.4.1 Proposed Method 28
3.4.2 AutoGBT's Conceptual Architecture 30
3.5 Experiments and Results 32
3.5.1 NeurIPS 2018 AutoML Challenge: Final Results 32
3.5.2 Offline Experimentation 33
3.5.3 Results 34
3.6 Summary 38

4 Multi-armed Bandit Based Online Ensemble Selection 39
4.1 Introduction 39
4.2 Related Works 40
4.2.1 Motivations for the Proposal 41
4.3 Proposed Method 42
4.3.1 Space and Time Complexity 46
4.4 Experimental Methodology 47
4.4.1 Predictive Performance 48
4.4.2 Overall Execution Time 49
4.4.3 Memory Requirements 49
4.4.4 Hyperparameter Relevance 49
4.4.5 Datasets 50
4.5 Experimental Results 51
4.5.1 Predictive Performance 51

© 2024, *Indian Institute of Technology Delhi*

4.5.2 Overall Execution Time 54
4.5.3 Memory Requirements 54
4.5.4 Hyperparameter Relevance 56
4.6 Summary 58

5 Successive Halving Based Online Ensemble Selection 61
5.1 Introduction 61
5.2 Related Works 62
5.2.1 Motivations for the Proposal 63
5.3 Proposed Method 64
5.3.1 Successive Halving Ensemble 64
5.3.2 Meta-learning for Automatic Budget Selection 69
5.4 Experimental Methodology 72
5.4.1 Predictive Performance 74
5.4.2 Overall Execution Time 74
5.4.3 Memory Requirements 74
5.4.4 Hyperparameter Relevance 74
5.4.5 Effect of Automatic Budget Selection 75
5.5 Experimental Results 76
5.5.1 Predictive Performance 76
5.5.2 Overall Execution Time 79
5.5.3 Memory Requirements 81
5.5.4 Hyperparameter Relevance 82
5.5.5 Effect of Automatic Budget Selection 84
5.6 Summary 84

6 Homogeneous-Heterogeneous Hybrid Ensemble 86
6.1 Introduction 86
6.2 Related Work 87
6.3 Proposed Method 90
6.3.1 Computational Complexity Analysis 96
6.4 Experiments and Results 97

© 2024, *Indian Institute of Technology Delhi*

6.4.1 Methodology 97
6.4.2 Results 102
6.5 Summary 113

7 Conclusions and Future Work **114**
7.1 Conclusion 114
7.2 Potential Future Directions 115

List of Tables

3.1 Properties of feedback phase datasets 26

3.2 A comparison of AutoGBT with other relevant studies 27

3.3 AutoGBT hyperparameter configuration space for LightGBM 30

3.4 Leaderboard from the final phase 32

3.5 Benchmark dataset properties 34

3.6 Average normalized AUC scores on AutoML3 offline phase datasets 35

3.7 Model execution time (in seconds) on AutoML3 offline phase datasets . . . 35

3.8 Average normalized AUC on public datasets 36

3.9 Model execution time (in seconds) on public datasets 37

4.1 Datasets. [Type] R : real-world, S : synthetic; [Drift] G : gradual, A: abrupt, I : incremental, M : mixed, R: recurring, IN: intermediate, F: fast, S: slow, ? : unknown. 50

4.2 Classification accuracies of algorithms 52

4.3 Total execution time of algorithms in seconds (training and prediction) . . 55

4.4 Model size comparison (in MBs) of algorithms 57

4.5 OMS-MAB performance variation with ϵ (averaged from 10 runs) 59

5.1 Hyperparameter configuration for CQL and DDQN 75

5.2 Properties of *experience dataset* with meta-training examples gathered using random budget decisions from 5400 realizations of each stream. 76

5.3 Classification accuracies of different algorithms 78

5.4 Total execution time in seconds (training and prediction) for different algorithms 80

5.5 Model size comparison (in MBs) of different algorithms 81

5.6 Performance variation of SUHEN with budget B 83

5.7 Accuracy comparison among different budget selection strategies 85

6.1 A comparison of H3E with other relevant studies. 89

6.2 Search space for hyperparameter configurations associated with H3E's base learner algorithms 99

6.3 Datasets. [Type] R : real-world, S : synthetic; [Drift] G : gradual, A: abrupt, I : incremental, M : mixed, R: recurring, IN: intermediate, F: fast, S: slow, ? : unknown. 101

6.4 Classification accuracies corresponding to different ensemble algorithms . . 103

6.5 Accuracy differences among various ablation forms of H3E 107

6.6 Accuracy differences among self-evolving and fixed H3E variants 109

© 2024, *Indian Institute of Technology Delhi*

List of Figures

1.1 Continuous learning from infinite data streams 2

1.2 Motivation for computationally efficient self-adaptive concept drift ensembles 4

2.1 Illustration of different types of concept drifts 12

2.2 Visualization of types of concept drifts based on the speed of changes. Combination of a shape and color indicates a concept 12

2.3 A general architecture of concept drift ensembles 15

2.4 A general architecture of AutoML frameworks for batch ML problems . . . 16

3.1 Evaluation protocol for NeurIPS 2018 AutoML Challenge. Image source:[31] 25

3.2 AutoGBT overall ML pipeline structure 28

3.3 Feature Space Transformation 29

3.4 Conceptual Architecture of AutoGBT 31

3.5 Average AUC across batches on AutoML3 offline phase datasets 36

3.6 Average AUC across batches on public benchmark datasets 37

4.1 OMS-MAB ensemble architecture. A pool of heterogeneous ALs each with a builtin HDDM_A drift detector is maintained to solve the original prediction problem. A non-stationary MAB periodically selects the best AL for training and prediction based on past experience, while systematically exploring promising alternatives 42

4.2 Learning procedure for adaptive learners within OMS-MAB, based on [51] 45

4.3 Results of Nemenyi post-hoc tests (95% confidence level) 51

4.4 Performance estimation and active member selection of OMS-MAB 53

4.5 Execution time growth comparison (in seconds) across algorithms 54

4.6 Model size growth comparison (in MBs) across algorithms 56

4.7 Variation in average performance of OMS-MAB across datasets with ϵ . . . 58

5.1 SUHEN online ensemble selection procedure 64

5.2 Automatic budget selection procedure for SUHEN 70

5.3 Results of Nemenyi post-hoc tests (95% confidence level) 77

5.4 Performance estimation and active member selection of SUHEN on the datasets Adult, RBF_F and HYP_S 79

5.5 Execution time growth comparison (in seconds) across algorithms on the datasets Covtype, Rialto and KDD99 80

5.6 Model size growth comparison (in MBs) across algorithms on the datasets Covtype, Rialto and KDD99 82

5.7 Nemenyi post-hoc test (at 95% confidence level) comparing budget selection strategies 84

6.1 Architecture of H3E. Heterogeneous block is directly trained on input data, while the adaptive block undergoes training using online bagging with $\lambda = 6$. Adaptive block members evolve to learner types along with their appropriate hyperparameter configuration using GA. Drift-aware stacking combines short-term and long-term concept memories to generate the ensemble prediction 91

6.2 Procedure for training the stacking meta-classifier; based on [51] 93

6.3 Structure of a candidate solution vector representing a base learner and its hyperparameters 96

6.4 Stream-specific variation in performance for ensembles depending on the base learner choice. HT variants exhibit superior performance for *Elec* whereas NB variants perform well for *HYP_S* 102

6.5 Nemenyi post-hoc test results (at a 95% confidence level) corresponding to predictive performance comparison of ensembles. The x-axis denotes the mean rank of each algorithm across all datasets. 103

6.6 Comparing temporal variation of ensemble accuracy corresponding to various datasets affected by different drift types 104

6.7 Prequential accuracy variation of leading ensembles on additional datasets spanning different drift generators and combinations of drift speeds and types 106

6.8 Nemenyi post-hoc test results (at a 95% confidence level) corresponding to predictive performance comparison among ablation forms of H3E 108

6.9 Evolving pattern in AB learner type choices with time for multiple datasets: For *Adult* dataset, learner types from the tree family are increasingly taken up while learners based on NB are more frequently selected for *HYP_S* . . 110

6.10 Nemenyi post-hoc test result (at 95% confidence level) from the comparison among self-evolving and fixed H3E variants 110

6.11 Differences in the overall computational resource needs of leading algorithms across various datasets 111

6.12 Comparison of temporal growth in computational resource usage 111

6.13 Scalability comparison of ensembles 112

© 2024, *Indian Institute of Technology Delhi*

ABBREVIATIONS

AL	Adaptive Learner
ARF	Adaptive Random Forest
ASHT	ASHoeffdingTree
AutoML	Automated Machine Learning
BAI	Best Arm Identification
BO	Bayesian Optimization
CACM	Causally Adaptive Constraint Minimization
CASH	Combined Algorithm Selection and Hyperparameter Optimization
CD	Critical difference
CIL	Class Imbalanced Learning
CL	Continual Learning
CQL	Conservative Q-Learning
DA	Domain Adaptation
DDM	Drift Detection Method
DDQN	Double DQN
DESDD	Dynamic Ensemble Selection for Drift Detection
DG	Domain Generalization
DP	Diversity Pool
DQN	Deep Q-learning Network
DSM	Data Science Machine
DWM	Dynamic weighted majority
EACD	Evolutionary Adaptation to Concept Drifts
EDDM	Early Drift Detection Method
EFDT	Extremely Fast Decision Tree
EL	Evolutionary Learning
ELM	Extreme Learning Machines
EOCD	Ensemble Optimization for Concept Drift
EWMA	Exponentially Weighted Moving Average
FASE	Fast Adaptive Stacking of Ensembles
FHDDM	Fast Hoeffding Drift Detection Method
FSL	Few-shot Learning
GA	Genetic Algorithm
GBDT	Gradient Boosted Decision Trees
GBM	Gradient Boosting Machine

HAT	HoeffdingAdaptiveTree
HDDM	Hoeffding-based Drift Detection Methods
HDES-AD	Heterogeneous Dynamic Ensemble Selection based on Accuracy and Diversity
H-DRLN	Hierarchical Deep Reinforcement Learning Network
HDWM	Heterogeneous Dynamic Weighted Majority
HEES	HEterogeneous Ensemble Selection
HPO	Hyperparameter Optimization
HT	Hoeffding Trees
HTNB	Hoeffding Tree with Naive Bayes
IoT	Internet of Things
KUE	Kappa Updated Ensemble
LevBag	Leveraging Bagging
LML	Lifelong Machine Learning
MAB	Multi-armed bandit
MDP	Markov Decision Process
MLC	Multi-label classification
ML	Machine Learning
MOA	Massive Online Analysis
MVC	multi-valued categorical
NAS	Neural Architecture Search
NB	NaiveBayes
NELL	Never-ending Language Learner
OAML	Online Automated Machine Learning
OBag	OzaBag
OBoost	Online Boosting
PAC	Probably Approximately Correct
RL	Reinforcement Learning
ROSE	Robust Online Self-Adjusting Ensemble
SHA	Successive Halving Algorithm
SMAC	Sequential Model-based Algorithm Configuration
SMBO	Sequential Model-Based Optimization
SRP	Streaming Random Patches
SSPT	Single-pass Self Parameter Tuning
TPE	Tree-of-Parzen-Estimators
UCB	Upper Confidence Bound
VFDT	Very Fast Decision Tree
XAI	Explainable AI

© 2024, *Indian Institute of Technology Delhi*