AutoConf: Automated Configuration of Unsupervised Learning Systems using Metamorphic Testing and Bayesian Optimization

Automated Software Engineering (ASE) Conference 2023

Lwin Khin Shar (Singapore Management University), Arda Goknil (SINTEF), Erik Johannes Husom (SINTEF), Sagar Sen (SINTEF), Yan Naing Tun (Singapore Management University), Kisub Kim (Singapore Management University)

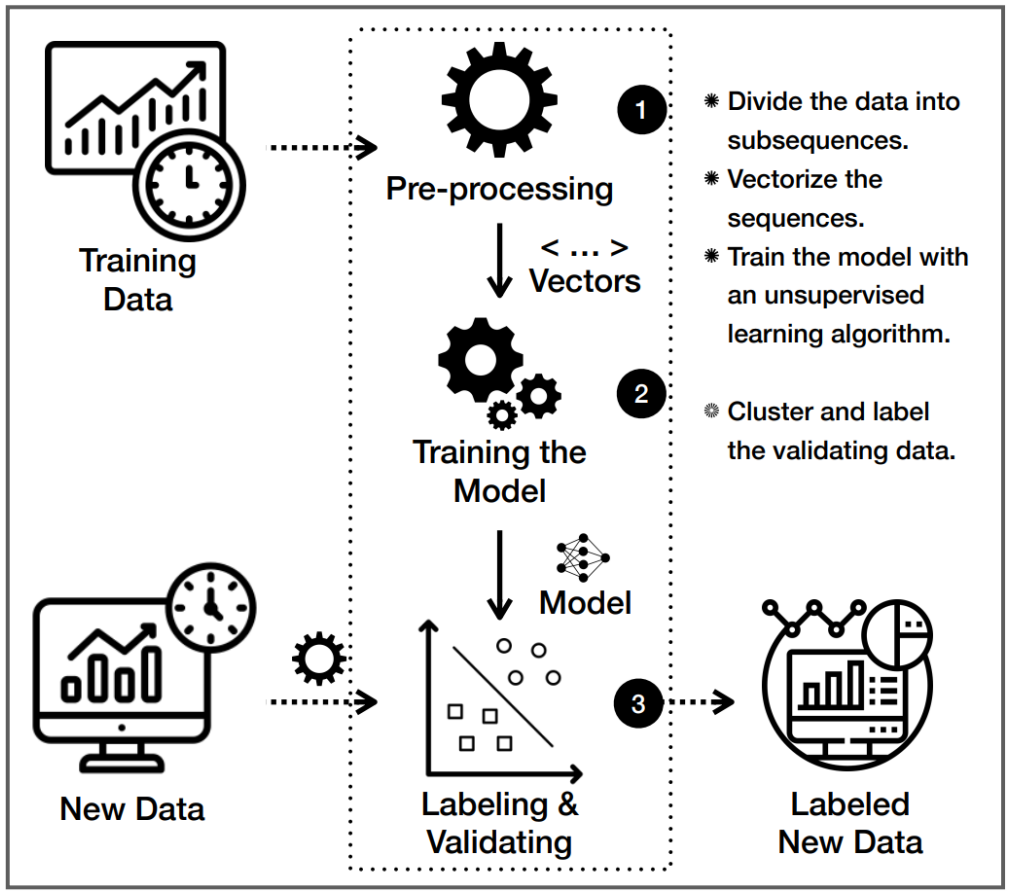
Luxembourg, 12th-14th of September 2023

Context: Configuration of Unsupervised Learning Systems

- In Step 1 (data preprocessing), the pipeline splits training timeseries data into subsequences and extracts statistical features from them.
- This step has three parameters:
  - optimal window size (the length of a sliding cutout of a time sequence of data)
  - overlap between the subsequences\windows
  - several features can be extracted from data subsequences to remove the temporal dimension

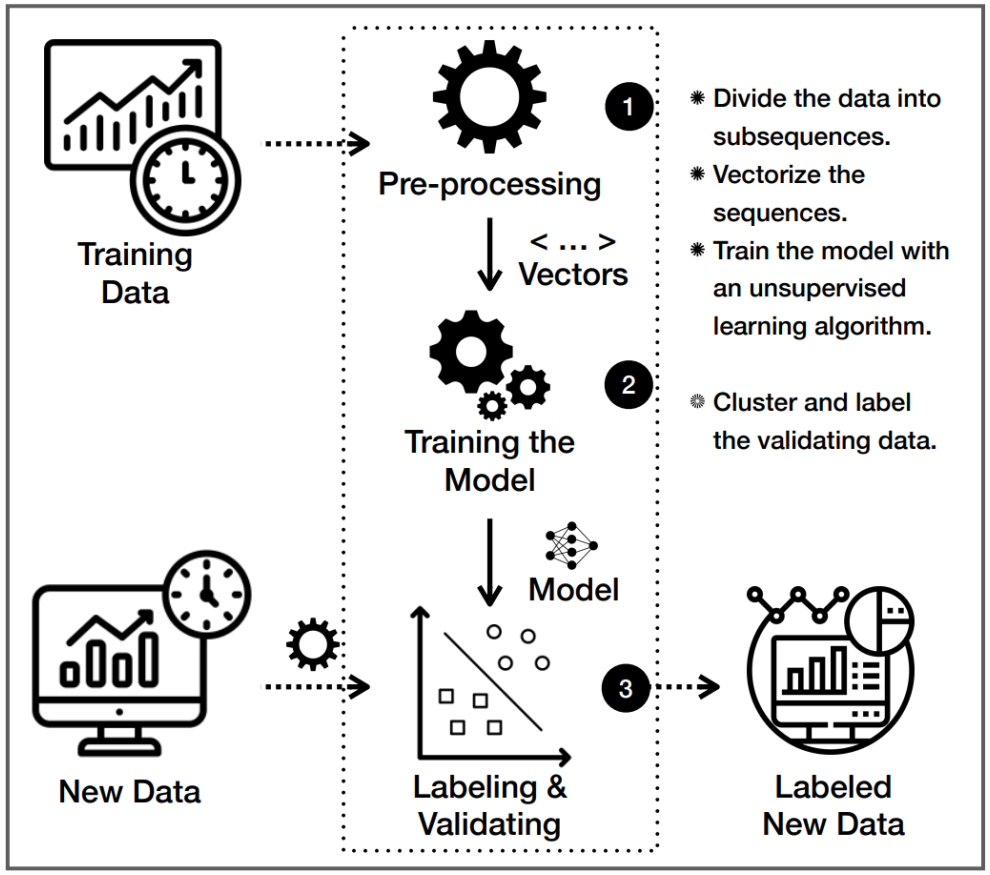
Feature name	Mathematical definition
Mean	$\mu = \frac{1}{w}\left(\sum_{i=1}^{w} x_i\right)$
Range	$\mathrm{r} = \max(\mathrm{x}) - \min(x)$
Gradient	$\nabla x = \frac{x_{i+1} - x_{i-1}}{2d}$
Variance	$v = \frac{1}{w}\sum_{i=1}^{w}(x_i - \mu)^2$
Frequency strength	$\nu = \lvert \mathrm{DFT}(w) \rvert_2$
Related quantities	**Symbol**
Sliding window size	w
Overlap of sliding windows	d

An example unsupervised learning system (pipeline) that automatically discovers anomalies in sensor data [1]



Context: Configuration of Unsupervised Learning Systems

- **In Step 2 (training the model)**, the pipeline performs cluster analysis on the feature vectors (assigns each feature vector to a cluster/category).
- This step has two parameters: **the clustering method (e.g., Kmeans, DBScan)** and **method's configuration parameters**.
  - For instance, Kmeans has the following parameters:
    (i) the number of clusters,
    (ii) predefined centroids (true or false),
    (iii) the initialization method (k-means++ or init),
    (iv) the number of runs with different centroid seeds,
    (v) the maximum number of iterations for a single run,
    (vi) the relative tolerance, and
    (vii) Kmeans algorithm (e.g., Lloyd or elkan)
- The feature vectors of the new data are being validated **in Step 3 (Labeling & Validation)**.

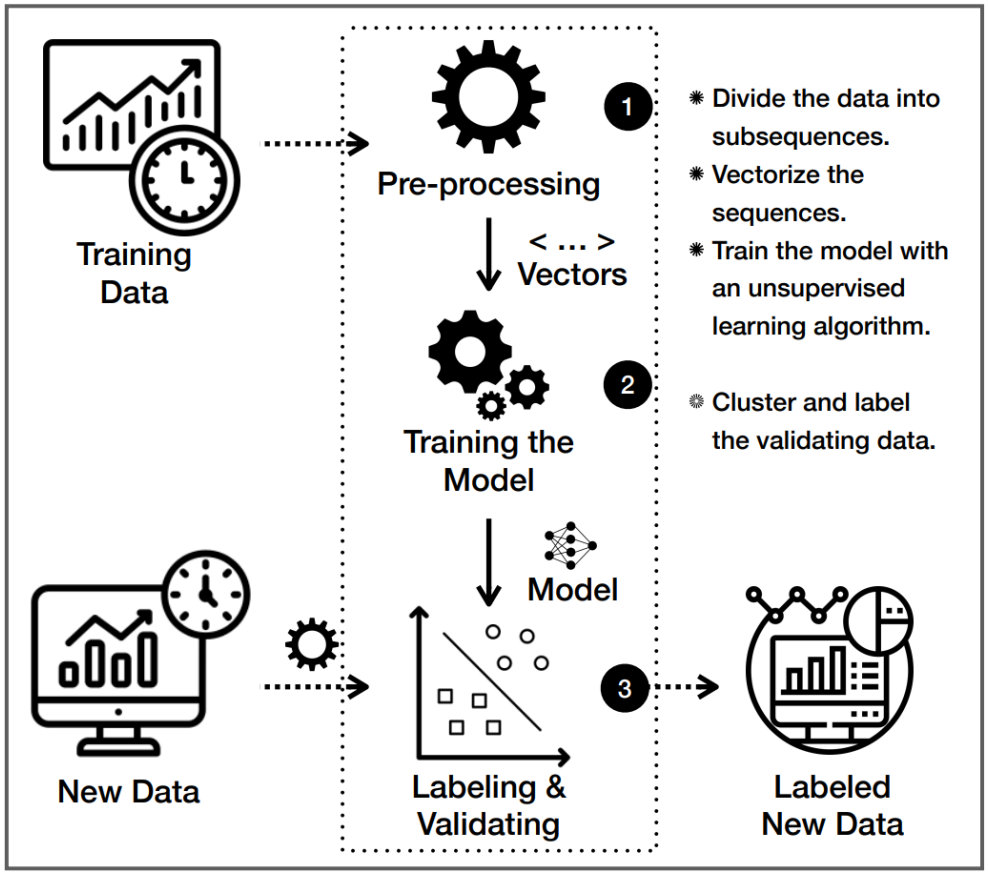


Context: Configuration of Unsupervised Learning Systems

- The effectiveness of such an unsupervised learning system highly depends on the **configuration of each step** in the system.
  - e.g., the selection of data preprocessing hyperparameters, features, and appropriate clustering algorithms and their hyperparameters

- Finding the right configuration is **challenging** due to
  - **large configuration space**
  - **numerous manual trials and errors**.

- **The absence of ground truth labels** due to the systems' unsupervised nature poses a challenge for automating the configuration process.



Related Work

- **Automated configuration support** for machine learning systems have been proposed in the context of AutoML [2]-[4]
  - Most AutoML approaches focus on **supervised learning systems**,

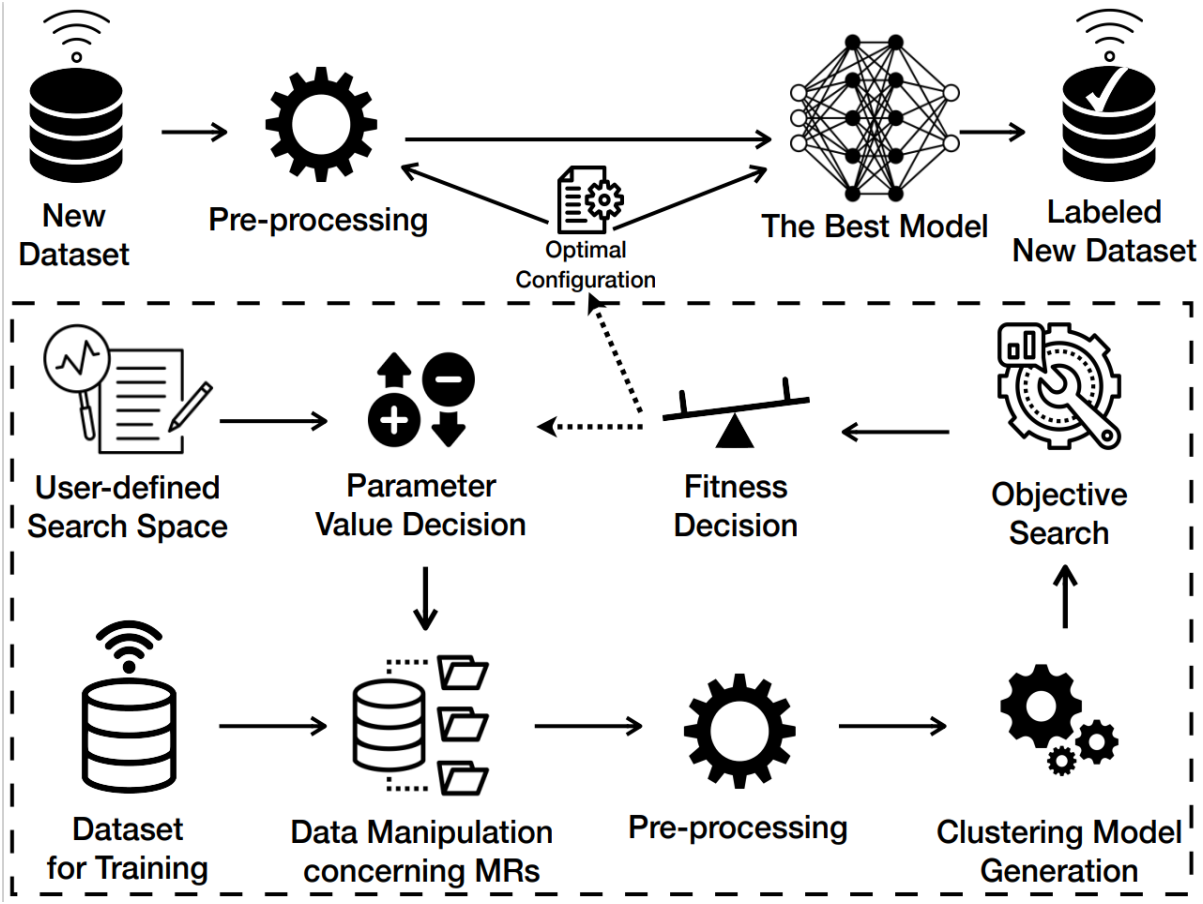
- Some others concentrate on either **cluster algorithm selection or hyperparameter tuning of the algorithms** [5]-[11]
  - Excluding hyperparameters for **other ML steps (e.g., data preprocessing)**

- To deal with **the lack of ground truth labels**, these approaches rely on **internal validity metrics (e.g., silhouette score [12])**
  - These metrics do not capture **the dynamic nature of the dataset**, and their performance is **sensitive to the data characteristics** (e.g., noise, density, and skewed distribution) [13] [14]

- **AutoConf employs metamorphic testing to address the lack of ground truth labels.**

Our Solution: AutoConf

- **Metamorphic testing** to address **the lack of ground truth labels** while evaluating the performance of configurations
- **Bayesian optimization** guided with metamorphic testing output as the objective function
  - To determine the optimal configuration
- **Tree Parzen Estimator (TPE)** approach for Bayesian Optimization
- AutoConf employs **five generic metamorphic relations (MRs)** proposed by Xie et al. [14] for testing clustering algorithms
- We present **six more custom MRs (for anomaly detection)** used by AutoConf



Our Solution: AutoConf

Algorithm 1 FindBestConfig

Require: $refData$
Require: $searchSpace : \{window, overlap, model, modelParams\}$
1: $losses, selectedValues \leftarrow \{\}$
2: $meanLoss \leftarrow 1$
3: **while** $\neg timeout \vee meanLoss \neq 0$ **do**
4: $selectedValues \leftarrow OptimizeSearch(searchSpace, meanLoss)$
5: $X \leftarrow ExtractFeatures(refData, selectedValues)$
6: $model \leftarrow BuildClusterModel(X, selectedValues)$
7: $silhouetteScore = ComputeSilhouetteScore(model)$
8: **if** $silhouetteScore < 1$ **then**
9: $losses \leftarrow |silhouetteScore|$
10: **else**
11: $losses \leftarrow 1 - silhouetteScore$
12: **end if**
13: $losses \leftarrow NumOfOutiers(model)/totalNumOfSamples$
14: **for all** $mr \in BenignMRs$ **do**
15: $X' \leftarrow GenerateFollowupDataset(X, mr)$
16: $model' \leftarrow BuildClusterModel(X', selectedValues)$
17: $losses \leftarrow EvaluateBenignMR(model, model')$
18: **end for**
19: **for all** $mr \in AnomalyMRs$ **do**
20: $X' \leftarrow GenerateFollowupDataset(X, mr)$
21: $model' \leftarrow BuildClusterModel(X', selectedValues)$
22: $losses \leftarrow EvaluateAnomalyMR(model, model')$
23: **end for**
24: $meanLoss \leftarrow mean(losses)$
25: **end while**

- *Optimize the search*
- *Extract the statistical features*
- *Build the clustering model*

- *Compute the silhouette score*
- *Calculate loss*

- *Evaluate benign and anomaly metamorphic relations*
- *Calculate loss*

Metamorphic Relations in AutoConf

- Overall, **eleven Metamorphic Relations (MRs)** in the search process
  - **Five generic MRs proposed by Xie et al.[14]** for testing clustering algorithms

- We propose **six new MRs (three benign and three anomaly MRs)** for anomaly detection in CPS
  - **Modifying attributes and clusters** (four MRs) **and adding new instances** (two MRs)
  - **Benign MRs** define the **same clustering model behavior** for source and follow-up models
  - **Anomaly MRs** define **different behaviors** for the two models

- **Example MR:** *Anomaly MR modifying Clusters*
  - We **modify the raw attribute of *n* consecutive instances** from few clusters to represent **anomalous behavior**
  - The new anomaly detection result should be **different** than the original result (**the modified instances should be flagged as anomalies**).

Evaluation – Research Questions

RQ1: *How does unsupervised learning systems perform with AutoConf?*

RQ2: *Is clustering-based anomaly detection configured by AutoConf more effective than baseline anomaly detection approaches?*

RQ3: *How does Bayesian optimization boost the efficiency of the search process in AutoConf?*

Configuration Domain and Datasets

Configuration Domain in our Experiments

Data preprocessing	Sliding window size w=[30,1500] Overlap d=[0,1) features={mean,range,gradient,variance}
Clustering method	{KMeans, Mini-batch KMeans, DBScan, Optics}
	Hyperparameters
KMeans	n_clusters=[1,15
Mini-batch KMeans	n_clusters=[1, max_iter={50, batch_size={2
DBScan & Optics	eps=[0.1,5] min_samples= metric=['euclid algo=['auto', '

Statistics of the Datasets used in our Experiments

Dataset	CPS	Anomaly	#Train	#Test
DJI-Windy	Drone	Extreme wind	10,016	20,000
DJI-VelFault	Drone	Faulty Sensor	5,000	2,000
PX4-Vibrate	Drone	Anomalous vibrations	43,547	10,887
Ardu-GyroFault	Drone	Faulty Sensor	144,176	2,000
Sleep-Apnea	ECG	Sleep Apnea	50,000	5,000
Bosch-CNC	CNC	Anomalous vibrations	59,393	99,400

RQ1 – Performance of AutoConf and Silhouette-only-approach

Best Configurations found by AutoConf

Dataset	w	d	Model	Model Params	Min. Loss
DJI-Windy	42	28	DBScan	eps=0.3	0.052
DJI-VelFau					
PX4-Vibrat					
Ardu-Gyrol					
Sleep-Apne					
Bosch-CNC	1100	330	DBScan	min_samples=1 metric=euclidea algo=aut eps=0.5 min_samples= metric=euclidea algo=aut	

Comparison of AutoConf with Silhoutte-Only-Approach

	AutoConf			*Silhouette-Only-Approach*		
Dataset	**Recall**	**Precision**	**F1-score**	**Recall**	**Precision**	**F1-score**
DJI-Windy	0.76	0.66	0.71	0.18	0.43	0.26
DJI-VelFault	1	1	1	1	1	1
PX4-Vibrate	1	0.92	0.96	1	0.87	0.93
Ardu-GyroFault	1	0.9	0.95	1	0.87	0.93
Bosch-CNC	0.76	0.72	0.74	0.5	0.66	0.57
Sleep-Apnea	0.84	0.85	0.85	1	0.65	0.78

Answer to RQ1: As each CPS may produce datasets with different characteristics, unsupervised learning systems should be configured differently for each CPS dataset. Thanks to metamorphic testing, *AutoConf* provides configurations yielding better results than the ones delivered by the baseline approach using an internal validity metric.

RQ2 – Performance of AutoConf and Baseline Anamoly Detection Approaches

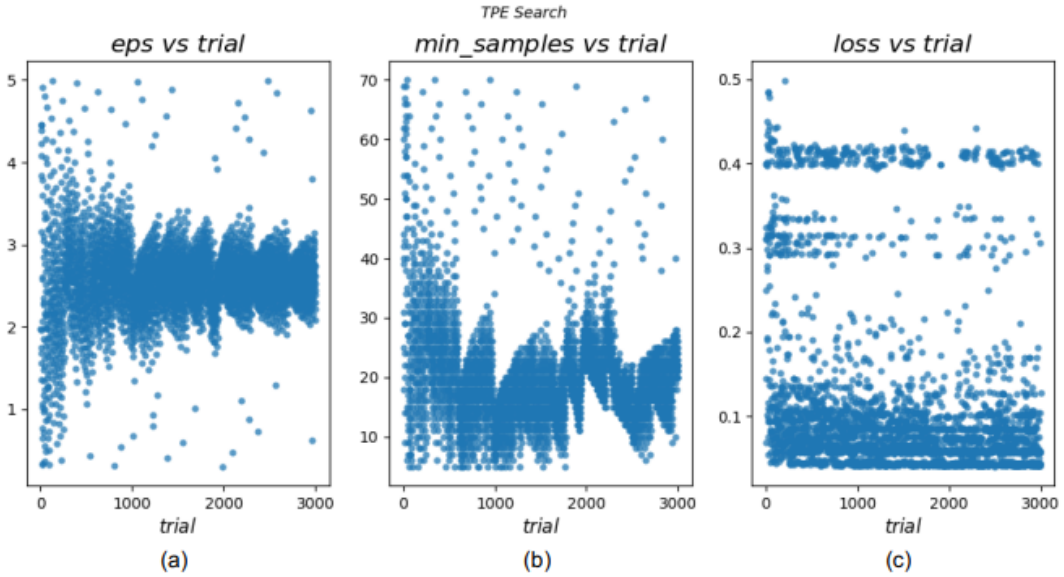
Comparison of AutoConf with Baseline Approaches

		AutoConf		*OneSVM*	*IF*	*LOF*
Dataset	**Recall**	**Precision**	**F1-score**	**F1-score**	**F1-score**	**F1-score**
DJI-Windy	0.76	0.66	0.71	0.51	0.65	0.49
DJI-VelFault	1	1	1	1	1	1
PX4-Vibrate	1	0.92	0.96	0.64	0.64	0.4
Ardu-GyroFault	1	0.9	0.95	0.93	0.9	0.9
Bosch-CNC	0.76	0.72	0.74	0.56	0.55	0.49
Sleep-Apnea	0.84	0.85	0.85	0.39	0.39	0.39

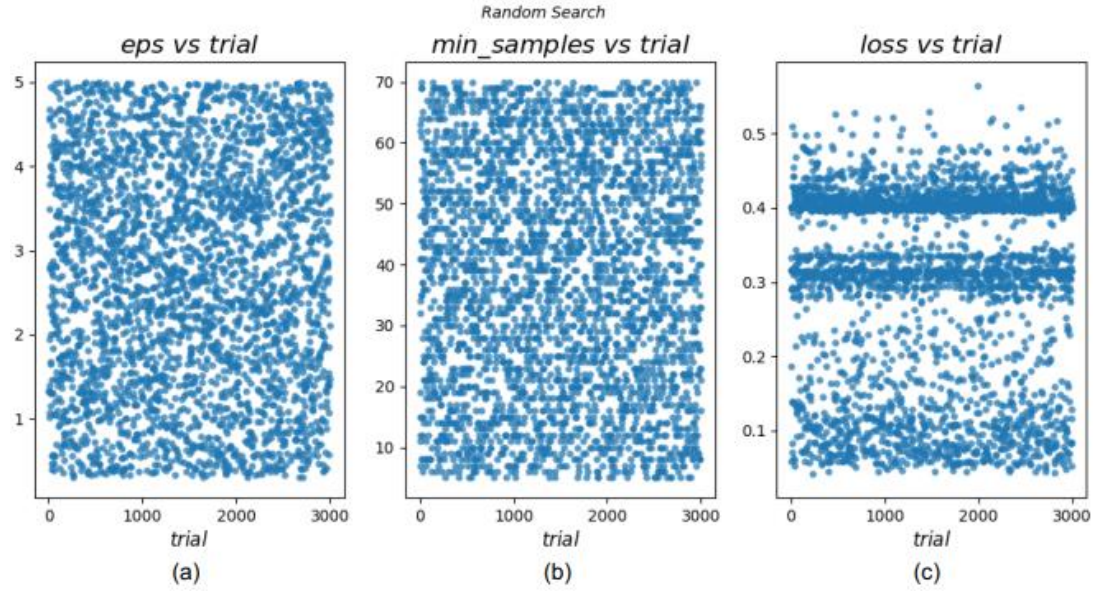
Answer to RQ2: *AutoConf* can identify configurations that yield similar or better anomaly detection results than the baseline anomaly detection approaches.

RQ3 – Efficiency of the Search in AutoConf

Bayesian Optimization Search



Random Search



Answer to RQ3: *AutoConf* leverages Bayesian optimization to achieve a superior guarantee of identifying the optimal configuration within a time budget. It efficiently guides the search toward the input space that minimizes the loss as determined by the prescribed loss functions.

Conclusion and Future Work

- Presented *AutoConf*, an automated approach to configure clustering-based unsupervised learning systems
  - Using Bayesian optimization and Metamorphic Testing
- Demonstrated the effectiveness of *AutoConf* in detecting anomalies through experiments conducted on six datasets
- *AutoConf* outperformed the baseline approaches
  - Achieving an average recall of 0.89 and a precision of 0.84

Future Work:

- Multi-objective Search
- Continuous Configuration
- Expanding AutoConf across Diverse Domains
- Assessing AutoConf on more Datasets

References (1)

[1] E. J. Husom, S. Tverdal, A. Goknil, and S. Sen, “Udava: An unsupervised learning pipeline for sensor data validation in manufacturing,” in Proceedings of the 1st International Conference on AI Engineering: Software Engineering for AI, 2022, pp. 159–169.

[2] X. He, K. Zhao, and X. Chu, “Automl: A survey of the state-of-the-art,” Knowledge-Based Systems, vol. 212, p. 106622, 2021.

[3] S. K. Karmaker, M. M. Hassan, M. J. Smith, L. Xu, C. Zhai, and K. Veeramachaneni, “Automl to date and beyond: Challenges and opportunities,” ACM Computing Surveys (CSUR), vol. 54, no. 8, pp. 1–36, 2021

[4] M. Bahri, F. Salutari, A. Putina, and M. Sozio, “Automl: state of the art with a focus on anomaly detection, challenges, and research directions,” International Journal of Data Science and Analytics, vol. 14, no. 2, pp. 113–126, 2022.

[5] E. Ditton, A. Swinbourne, T. Myers, and M. Scovell, “Applying semi-automated hyperparameter tuning for clustering algorithms,” arXiv preprint arXiv:2108.11053, 2021.

[6] R. ElShawi, H. Lekunze, and S. Sakr, “csmartml: A meta learningbased framework for automated selection and hyperparameter tuning for clustering,” in 2021 IEEE International Conference on Big Data (Big Data). IEEE, 2021, pp. 1119–1126.

[7] X. Fan, Y. Yue, P. Sarkar, and Y. R. Wang, “On hyperparameter tuning in general clustering problems,” in International Conference on Machine Learning. PMLR, 2020, pp. 2996–3007.

References (2)

[8] M. C. De Souto, R. B. Prudencio, R. G. Soares, D. S. De Araujo, I. G. Costa, T. B. Ludermir, and A. Schliep, "Ranking and selecting clustering algorithms using a meta-learning approach," in 2008 IEEE International Joint Conference on Neural Networks, 2008, pp. 3729–3735.

[9] D. G. Ferrari and L. N. De Castro, "Clustering algorithm selection by meta-learning systems: A new distance-based problem characterization and ranking combination methods," Information Sciences, vol. 301, pp. 181–194, 2015

[10] Y. Poulakis, C. Doulkeridis, and D. Kyriazis, "Autoclust: A framework for automated clustering based on cluster validity indices," in 2020 IEEE International Conference on Data Mining (ICDM). IEEE, 2020, pp. 1220–1225.

[11] M. Carnein, H. Trautmann, A. Bifet, and B. Pfahringer, "confstream: Automated algorithm selection and configuration of stream clustering algorithms," in Learning and Intelligent Optimization: 14th International Conference, LION 14, Athens, Greece, May 24–28, 2020, Revised Selected Papers 14. Springer, 2020, pp. 80–95.

[12] P. J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," Journal of computational and applied mathematics, vol. 20, pp. 53–65, 1987.

[13] Y. Liu, Z. Li, H. Xiong, X. Gao, and J. Wu, "Understanding of internal clustering validation measures," in 2010 IEEE international conference on data mining. IEEE, 2010, pp. 911–916.

[14] X. Xie, Z. Zhang, T. Y. Chen, Y. Liu, P.-L. Poon, and B. Xu, "Mettle: a metamorphic testing approach to assessing and validating unsupervised machine learning systems," IEEE Transactions on Reliability, vol. 69, no. 4, pp. 1293–1322, 2020

SINTEF

Technology for a better society