Hybrid Evolutionary-ML Surrogate Models for Cyber-Attack Detection in Water Distribution Networks

Côme Frappé - - Vialatoux, Pierre Parrend

Hybrid Evolutionary-ML Surrogate Models for Cyber-Attack Detection in Water Distribution Networks

Côme Frappé - - Vialatoux[1,2] [0009−0000−9787−2638] and Pierre Parrend[2,1] [0000−0002−1680−1182]

[1] ICube, UMR 7357, Université de Strasbourg, Strasbourg, France
cfrappevialatoux@unistra.fr
[2] Laboratoire de Recherche de l'EPITA, 14-16 Rue Voltaire, Le Kremlin-Bicêtre, France
pierre.parrend@epita.fr

Abstract. Water distribution networks (WDNs), as critical infrastructure, face growing cyber-attack risks. While Industry 4.0 initiatives motivate the deployment of edge technologies for monitoring, efficient detection on distributed edge devices requires methods that reduce computational overhead without sacrificing classifier performance. We propose a hybrid evolutionary-machine learning (ML) approach that constructs probabilistic surrogate models of trained ML classifiers using an Estimation-of-Distribution Algorithm (EDA). Our *het-EDA* algorithm supports the analysis of heterogeneous network communication features, extending r-UMDA for categorical and PBIL-C for continuous data. It evolves class-wise surrogates which optimize the original classifier's output scores. Inference is performed via lightweight log-likelihood evaluation, making the method suitable for resource-constrained edge devices. Experiments on the WDN *a Hardware-in-the-Loop* dataset show that the surrogates preserve portions of the original decision boundaries, while highlighting limitations due to feature independence assumptions. These results demonstrate the potential of EDA-based probabilistic surrogates for efficient edge ML inference and motivate the development of more expressive EDAs suited for complex industrial network data.

Keywords: Estimation-of-Distribution Algorithms · Cyber-attack detection · Machine Learning · Water Distribution Networks · Edge computing

1 Introduction

With the democratization of Internet-of-Things (IoT) devices that accompany the modernization efforts of industries towards the 4.0 model [18], cyber-security stakes are at an all-time high. The increase in the number of interconnected devices, while offering better monitoring and control of industrial processes through edge devices [5], also represents a significant increase in the attack surface. This

greater exposure to cyber-threats is particularly evident in water distribution networks (WDNs), where the nature of security incidents has shifted [13] from on-site attacks by rogue employees, such as data theft or alteration of plant configuration, to remotely operated cyber-attacks such as ransomware carried out by external malicious actors. The high financial and societal impact of this type of attack makes detection an increasingly important research field [31].

With the recent framing of WDN security as part of cyber-physical systems (CPS) [12], considering both the information of the physical processes through sensors and Programmable Logic Controllers (PLCs), and the cyber layer composed of all connected devices that enable the supervision of the physical processes shows promising improvement for machine learning based detection [32]. This approach is especially relevant in the context of edge computing, where bringing the detection algorithms directly to the devices that generate physical data can reduce time-to-detection, the volume of data transferred, while providing an accurate location in case of attack. However, WDN often relies on aging infrastructures, which modernization progresses at a slow pace due to the high investment required [26], hindering the deployment of edge Machine Learning detection due to the low computing power of old PLCs.

To answer this problem of computational cost of Machine Learning (ML) detection for edge inference in WDN, we introduce a hybrid evolutionary-ML approach that reduces the inference cost of complex ML models by constructing a probabilistic surrogate model via Estimation-of-Distribution Algorithm (EDA). The property of EDAs to evolve explicit probabilistic models rather than single solutions allows to evolve the surrogate model by using the classifier's class-specific score as the fitness function to evolve one surrogate per class. We address the challenge of applying EDA to network communication data with the design of the Heterogeneous Estimation-of-Distribution Algorithm (*het-EDA*) that extends the multi-valued Univariate Marginal Distribution Algorithm (r-UMDA) [4] for categorical data, and Continuous Population Based Incremental Learning (PBIL-C)[29] for continuous data. Its design for GPU computation allows high population sizes for a wide exploration of the search space that suits the inherent diversity of network data. This process allows the inference to be reduced to selecting the class whose surrogate model assigns the highest log-likelihood to the input. Preliminary results show the feasibility of using *het-EDA* on a variety of ML model architectures by creating class-wise surrogates of XGBoost (XGB), Bagging and Multi-Layer-Perceptron (MLP) trained on the *a Hardware In The Loop* (HITL) dataset for cyber-attack detection on WDN [8].

The remainder of the paper is organized as follow: Section 2 describes the state of the art of ML-based attack detection with a focus on CPS and edge environment, Section 3 contains the formalization of *het-EDA* algorithm, followed by the description of the surrogate models creation process. Section 4 outlines the experiments led and analyses the obtained results, opening for the discussion in Section 5 before concluding in Section 6.

2 State of the art

This section first presents the state of the art of ML based cyber-attack detection, then describes the specificity of this detection for CPS environment followed by a focus on its application in edge computing.

2.1 ML detection for security

As the number of attacks against WDNs increases over the years, cyber-attacks such as ransomware and unauthorized remote access are becoming prevalent in the threat landscape, with impacts ranging from data loss to system paralysis [11] [13]. In this context of increased risk, different approaches to the detection of cyber-attacks have been developed by the scientific community. Signature based detection consists of the search for known patterns in the data, such as statistical properties [23] [20] or rule-based detection and indicators-of-compromise [2]. While providing a baseline for detection, their reliance on specific indicators extracted from past attacks renders them blind to new attacks [22].

Machine Learning (ML) based detection uses ML models trained to distinguish between normal and malicious data. The two main modalities for training ML models are supervised and unsupervised learning [10] [24]. With unsupervised learning, the ML model learns from data without attacks to recognize the normal state of the system. Examples algorithms are as 1-class Support Vector Machine (1-class SVM) [28] or Long-Short-Term-Memory (LSTM) [15]. With supervised learning, the model learns to classify data into classes according to labels provided for each data. Examples of algorithms are K-Nearest-Neighbors (KNN) [7] and MLP [27].

Unsupervised ML methods benefit from lighter data requirements for their deployments. These algorithms are built upon the idea of constructing a model capable of learning the representation of normal data through training, and then classify new data based on a proximity score to this learned representation. Multiple score metrics exist depending on the model used, such as reconstruction error on a predictive model in [21] with an Attention-based Spatio-Temporal Autoencoder, a distance to a learned geometric envelope of the normal data (1-class-SVM) in [17], or a threshold of unmet association rules (ML Invariants) in [34]. As the performances of this approach are tied to how well the normal state is learned, a common practice is to divide the data into smaller more homogeneous parts and fit a unique model on each of these subparts parts.

On the other hand, supervised ML models can be seen as an evolution over signature-based detection in their ability to detect known attacks but without the need for explicitly defined indicators. These ML models learn to associate each data instance with its correct class through the construction of complex representations derived from the labeled dataset used as training data. Recent work on supervised ML models for detection showed promising results across a wide range of algorithms such as Logistic Regressions, SVM or tree-based models (Decision Tree, Random Forest) [14], deep neural networks [25]. Approaches

combining multiple models in a single detection pipeline such as deep Convolutional Neural Networks with autoregressive moving average attributes [3].

2.2 Specificity of attack detection in CPS

The Cyber-Physical aspect of WDNs anchors attack detection in such systems at the boundary of Information Technology (IT) and operational Technology (OT). This duality led to the development of hybrid detection methods that encompass the entire attack perimeter of the WDN by combining the information present in OT and IT data. The different approaches to this information combination process shown in Figure 1 can be classified into two groups: model-wise and data-wise combinations.

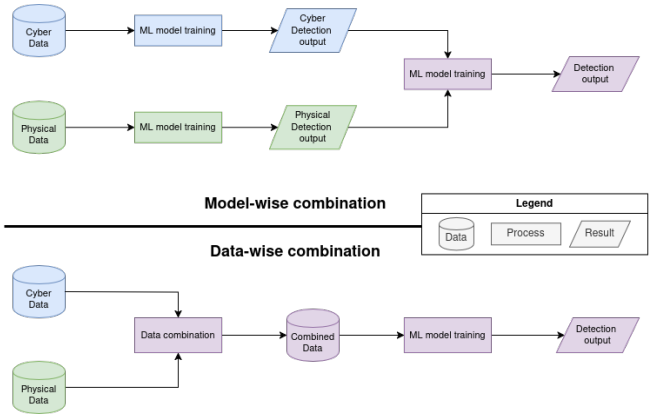

Fig. 1: Comparison of the Data-wise and Model-wise information combination methods for CPS

Model-wise combination means that the information from the physical and cyber layers of the CPS are fit into separate detection models, which outputs are then passed as input to a *meta-model* which detection will hence leverage information from the whole system. This approach has been used in [9] by feeding reconstruction error output from LSTM for the physical layer and autoencoders for cyber layer, to a Bayesian Network, allowing for an explainable detection on the whole system with an attack detection achieving 96% accuracy on a scenario involving attacks on the whole system. A drawback of this method is that by further adding models, it multiplies the costs associated with training and maintenance, with the added complexity that any update to one of the base models implies an update of the meta model.

Data-wise combination, on the other hand, proceeds to merge the data from both the physical and the cyber layer with a method ensuring consistency of the combined version. [32] used this method on 3 different datasets to compare

the detection performances of multiple supervised ML models trained on every data configuration, namely physical only, cyber only, and combined. The results showed that the use of combined data is associated with better performances for the best models on each dataset. However, the proposed combination method involves data duplication to ensure consistency, which comes with an increase in data size that can impact training and inference time of the ML models.

2.3 ML Detection in Edge computing

Edge computing is a computational paradigm where the processing is deported to the devices at the periphery of the network. It emerged as a response to the observation that the rate of adoption of IoT devices surpasses the increase of computational power of centralized architecture of cloud computing, thus leading to a bottleneck [30]. While providing a solution for reduced bandwith usage, lower latency and enabling more real-time application, edge computing main challenge lies in the limited resources at the disposal of edge devices [6].

As the attack perimeter of a network grows with the number of interconnected devices, adapting attack detection to the edge computing offers a scalable approach to the monitoring of very large interconnected networks, as each edge device is able to incorporate its own detection model. The approaches from the literature use signature based detection [33] unsupervised anomaly detection model such as LSTM [1] [35], or cloud-assisted supervised learning [19], but the use of supervised learning for direct inference on edge devices remains to be achieved.

3 Contribution

This Section presents the formalization of the proposed *het-EDA* algorithm, followed by the description of the surrogate models creation pipeline.

3.1 EDA algorithm for heterogeneous datatype

In order to use an EDA to model network communication data, the distribution models used must be adapted to the inherent heterogeneity of datatypes that encompass: categorical data such as IP and MAC addresses, continuous data such as packet sizes or counts. In addition, being compliant to GPU parallelization allows to generate sufficiently large population to reflect the diversity of these data during the optimization process and thus better explore the search space. Our proposed algorithm integrates the distribution models from different EDAs to represent each data type: r-UMDA [4] for categorical data, and PBIL-C[29] for continuous data. The distribution model used in r-UMDA is a frequency matrix that represents, for each categorical attribute taking r different possible values, the probability of each different value of r. In PBIL-C a Gaussian distribution describes each continuous attribute by its mean μ and its variance σ, which initial values are passed as vector arguments $\overrightarrow{\mu}$ $\overrightarrow{\sigma}$. Argument s corresponds to

the size of the selection of the best individuals at each generation, from which a new distribution will be extracted. Its value represents the trade-off between the exploration of the search space, with a high value of s, versus the exploitation of solutions, with a low value of s.

Algorithm 1 EDA for heterogeneous Data

Require: N : population size, n_{cat} : number of categorical attributes, r : number of different categories for categorical attributes, n_{cont} : number of continuous attributes, $\overrightarrow{\mu}$: PBIL-C initial means for continuous attributes, $\overrightarrow{\sigma}$: PBIL-C initial variances for continuous attributes, s : selection size,

Ensure: $P_{cat}^{(t_{max})}$: categorical distribution model after last iteration, $P_{cont}^{(t_{max})}$: continuous distribution model after last iteration

$P_{cat}^{(0)} \leftarrow \frac{1}{r} \times (i,j)_{\in [n_{cat}] \times [0..r-1]}$
$P_{cont}^{(0)} \leftarrow (\mu_i, \sigma_i)_{i \in [0..n_{cont}]}$
$t \leftarrow 0$
while *not* Stop criterion **do**
  $Pop_{cat}^{t} \leftarrow \text{Sample}(P_{cat}^{(t)}) \times N$
  $Pop_{cont}^{t} \leftarrow \text{Sample}(P_{cont}^{(t)}) \times N$
  $Pop^{t} \leftarrow \text{Concat}(Pop_{cat}^{t}[k],\ Pop_{cont}^{t}[k]), k \in [N]$ ▷ Merge each individual's *cat* and *cont* parts
  $S^{t} \leftarrow \text{Select}(Pop^{t})$ ▷ Select s individuals with best fitness values
  $S_{cat}^{t} \leftarrow$ Categorical elements of S^{t}
  $S_{cont}^{t} \leftarrow$ Continuous elements of S^{t}
  for $(i,j) \in [n_{cat}] \times [0..r-1]$ **do** ▷ Update categorical distribution
    $P_{cat}^{(t+1)}[i,j] \leftarrow \frac{1}{n_{cat}} \Sigma_{k \in [n_{cat}], i \in [s]} \mathbb{1}_{S_{cat}^{t}[i,k]=j}$
    $P_{cat}^{(t+1)} \leftarrow P_{cat}^{(t+1)}$ restricted to $[\frac{1}{(r-1)n_{cat}}, 1 - \frac{1}{n_{cat}}]$ as described in [4]
  end for
  for $i \in [1..s]$ **do** ▷ update continuous distribution
    $P_{cont}^{t+1}[i] \leftarrow (\frac{\sum_{j=1}^{n_{cont}} S_{cont}^{t}[i,j]}{n_{cont}}, \sqrt{\frac{\sum_{j=1}^{n_{cont}} (X_i^j - \bar{X}_i)_{X \in S_{cont}^{t}}}{n_{cont}}})$
  end for
  $t \leftarrow t + 1$
end while

The Algorithm starts by initializing the categorical distribution equiprobably for each value of r for each categorical attribute, as shown in Figure 2. The continuous Gaussian distribution of each continuous attribute is then initialized according to the value of arguments $\overrightarrow{\mu}$ and $\overrightarrow{\sigma}$ as shown in Figure 3.

The iteration number t starts at 0. The optimization loop then proceeds until the stopping criterion chosen by the user is met. The usual criteria are either a fitness value threshold or an iteration number. The first step of this optimization loop is to create a population of candidate solutions by independently sampling the categorical and continuous distribution until N individuals are generated. These individuals are then concatenated into an appropriate order for the evaluation function to run on them. The selection operator $Select(Pop^{t})$

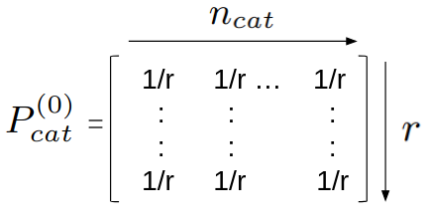

Fig. 2: The initialized matrix of categorical probabilities

$$P_{cont}^{(0)} = \begin{bmatrix} \mu_0 & \mu_1 & \cdots & \mu_{n_{cont}} \\ \sigma_0 & \sigma_1 & & \sigma_{n_{cont}} \end{bmatrix}$$

Fig. 3: The initialized matrix of means and variances of continuous attributes

consists of evaluating each individual and selecting the top s that will constitute the selected population S^t. The categorical and continuous components of this population are then separated into distinct variables S^t_{cat} and S^t_{cont} to be used separately to update their respective distributions. The categorical distribution is updated by counting the occurrences of each value of r of each attribute in the selected categorical population and dividing it by the number of categorical attributes n_{cat} to obtain a frequency vector that sums up to 1 on each categorical attribute. This distribution is then restricted to avoid the extremum values of 1 and 0 that would lock the attribute value for all subsequent iterations. The continuous distribution is updated by replacing the current value of μ and σ by the mean and variance of the attributes in the selected population S^t_{cont}. The iteration counter is updated, and the next iteration starts if a stopping criterion is not met.

Both these distribution models and optimization steps are well suited for GPU computation due to being easily adapted to "array-like" computations, however they assume independence between all variables as well as Gaussian distributions of continuous attributes, which are strong hypotheses that do not necessarily verify in network data.

3.2 Pipeline for Surrogate Model Creation

General overview The surrogate models creation pipeline is centered around the concept of using the classification score of a trained ML model as an optimization function. By trying to maximize the classification score for one of the output classes of a ML model, the optimization process will find solution inputs to that model that correspond most to what it learned as the target class.

To better represent the inherent diversity in the target class, we opted to use EDA as the optimization algorithm for their property of evolving probabilistic distribution models rather than single solutions. As a result, an EDA algorithm optimizing the classification score of a trained ML classification model will output a probabilistic model of the values that best characterize the target class learned by the ML model. Repeating this process for each class thus generates a catalogue of probabilistic models that can be extracted regardless of the ML

model architecture, as long as the model output score can be set as the optimization function. Finally, these profiles serve as surrogate probabilistic models of their ML model of origin by computing the log-likelihood of input to each surrogate model and attributing the class from which this input has highest probability of being draw.

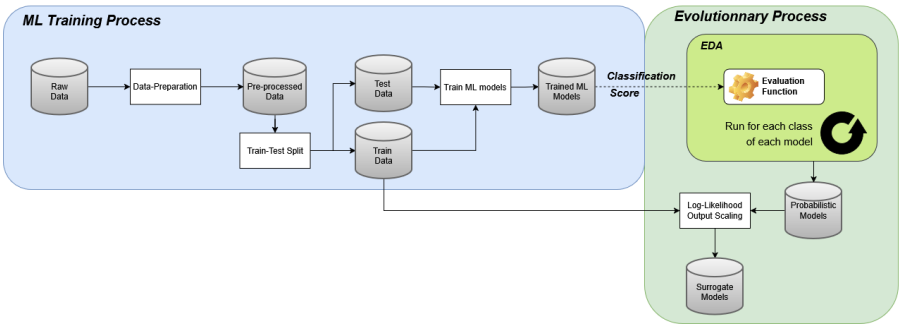

Fig. 4: Detailed surrogate model creation pipeline.

Detailed Description The distribution extraction pipeline represented in Figure 4 is composed of 2 main parts: the training of the ML classifiers and the extraction of the surrogate models through an evolutionary process.

First, the raw data undergo data preparation to be compatible with the constraints posed by the chosen machine learning models, as well as the EDA algorithm used. These data preparation steps can entail treatment of missing values, one-hot encoding of categorical columns, normalization and other appropriate operations. Once prepared, the data are split into training and testing sets for the training of ML classification models. It is important that the same sets are used for every model to ensure that the learned classes are based on the same base information, so the comparison of the extracted probabilistic distributions makes sense.

Next, the EDA optimization process is initialized. The probabilistic models' initial parameters are set to reflect no prior bias toward any particular outcome. In the context of our proposed *het-EDA*, this corresponds to assigning equal probability to each categorical value and initializing all continuous variables with a standard normal distribution ($\mu = 0, \sigma = 1$) to represent normalized data. In addition, a chosen class of the ML model is selected as the optimization function. This class stays fixed throughout the iterations. Then, the optimization process starts until a chosen stopping criterion is met: either after a set number of iterations or until convergence of the solutions. Once finished, the obtained probabilistic model for the class chosen as the optimization function is saved,

and the EDA optimization process is initialized and repeated for all remaining classes and then for all ML models.

The classification process computes the log-likelihood of an input x under each probabilistic model: for a given model C, this log-likelihood is the sum of the log-probabilities of the continuous features of x under the continuous distributions of C, and the log-probabilities of the categorical features of x under the categorical distributions of C (Eq. 1).

$$\log P(x \mid C) = \sum_i \log P(x_{\text{cont},i} \mid C) + \sum_j \log P(x_{\text{cat},j} \mid C) \tag{1}$$

The use of log-likelihoods instead of raw probability multiplications avoids numerical underflow caused by repeatedly multiplying probabilities less than one. Then, the class attribution can take two modalities: either by attributing the class with the absolute maximum log-likelihood, or, for each class, average the log-likelihoods of the surrogate models that compose it, then attribute the class with the highest average log-likelihood across its surrogates.

An important side-effect of using surrogates from different ML models is that the scale of the log-likelihood values can vary depending on the model architecture. For instance, decision tree models appeared to have more "extreme" probabilistic models, with categorical probabilities often reaching the maximum / minimum probabilistic thresholds (*i.e.* close to 1 and 0). A possible explanation for this phenomenon is that a decision tree architecture could have these variables appearing in only one branch, thus creating a binary condition on its values. The models with such probabilistic profiles have variations of log-likelihood that are orders of magnitude higher than the more "balanced" models. This scale difference biases the use of the maximum log-likelihood for the class attributions.

To solve this problem, each surrogate model is independently scaled by computing its log-likelihood values on every data of the training set, and the mean and standard deviation of these obtained values are stored and used as customized standardization values for their associated surrogate models' output. This process allows all surrogate models to have their log-likelihood values standardized, eliminating bias from their model of origin.

4 Experiments

The proposed distribution extraction pipeline was tested on the HITL cyber-physical dataset with the proposed *het-EDA* implemented with the EvoX library [16].

4.1 Experimental setup

For these experiments, the network communication data of Scenario 4 of the HITL dataset was used. The labels present are: Normal (51.51%), Denial-of-Service (DoS) (34.49%), Man-in-the-Middle (MITM) (8.53%), Physical Fault (5.46%) and Scan (<0.001%).

The data preparation steps included: conversion of port numbers to categories (well-known ports: $0 \rightarrow 1024$, registered ports: $1024 \rightarrow 49151$, dynamic ports: $49152 \rightarrow 65535$), one-hot encoding of categorical columns, standardization of continuous columns, aggregation of identical rows within each one-second interval and adding their occurrence count as a new column. The data were then split into train and test sets with an 80-20 ratio.

Table 1: ML model hyperparameters

Model	Library	Hyperparameters
XGB	`xgboost`	`eval_metric`: mlogloss, `learning_rate`: 0.01, `max_depth`: 10, `max_iter` = 3000, `early_stopping` = 10
MLP	`sklearn`	`hidden_layer_sizes`=(100, 50), `alpha`= 0.001, `max_iter`=100, `early_stopping`=10
Bagging	`sklearn`	`estimator`=DecisionTreeClassifier, `n_estimators`=10, `max_samples`=0.8, `max_features`=1.0

The ML models are trained in a multi-class setting, *i.e.*, each model is trained to classify among the five classes. The corresponding hyperparameters are listed in Table 1, and all random seeds are set to 42.

The EDA optimization process was run with an implementation of *het-EDA* with a population size of 200 000, a selection size of 10 000 and using five random seeds for each model-class combination.

4.2 Results

Table 2: Per class performance metrics for ML models

Class	Precision			Recall			F1-score			Support
	XGB	MLP	Bag.	XGB	MLP	Bag.	XGB	MLP	Bag.	
Normal	**0.78**	**0.78**	**0.78**	**1.00**	**1.00**	0.99	**0.87**	**0.87**	**0.87**	135,269
MITM	0.82	**1.00**	0.62	0.05	0.03	**0.06**	0.09	0.05	**0.11**	24,828
Physical Fault	0.00	0.00	**0.17**	0.00	0.00	**0.01**	0.00	0.00	**0.01**	13,197
DoS	0.97	**0.98**	0.96	**0.88**	0.85	**0.88**	**0.92**	0.91	**0.92**	11,768
Scan	**1.00**	**1.00**	**1.00**	**1.00**	0.43	**1.00**	**1.00**	0.60	**1.00**	7

The classification performances of the ML models summarized in Table 2 show overall similar performances between the different models, with a marginally better performance for Bagging, which has the best F1-scores across all classes. A notable point is the Physical Fault class being undetected by all models, as

shown by almost all-zero recall values, which is coherent with the models being trained only with network data, while this attack reflects on the physical data. The MITM attack also has very low detection rates that translate to very low recall values of 0.03 to 0.06; however, the MLP model's precision of 1 indicates a qualitatively learned representation of an extremely small portion of the samples.

The DoS detection metrics are almost identical for all models, with very high precision and relatively high recall, which points towards a portion of the class instances escaping all models' detection. Finally, the Scan class suffers from extremely low support, which makes the detection performance rapidly drop in case of error. This is apparent for the MLP model, whose precision and recall values of 1 and 0.43 can be directly converted to a correct classification of only 3 out of the 7 instances of the test set. The other models' perfect performances for the scan class indicate a good identification, but such a few instances do not allow to have an idea of the generalization capabilities of these models for this class. Overall, XGB and Bagging show slightly more stable generalization across classes, whereas MLP's lower recall on the low support scan class suggests higher sensitivity to data imbalance.

The classification performances of the surrogate models evolved with *het-EDA* from these ML models are summarized in Table 3. They have been evaluated on the test set with both the absolute maximum log-likelihood (AML) method and the average per class maximum likelihood (APCML) method described in Section 3.2.

Table 3: Per class performance metrics of surrogate models for classification

Class	Precision		Recall		F1-score		Support
	AML	APCML	AML	APCML	AML	APCML	
Normal	**0.75**	**0.75**	0.16	**0.26**	0.27	**0.38**	135,269
MITM	0.14	**0.15**	0.14	**0.17**	**0.15**	**0.15**	24,828
Physical Fault	**0.07**	**0.07**	**0.23**	0.19	**0.11**	**0.11**	13,197
DoS	**0.10**	0.09	0.31	**0.40**	0.12	**0.16**	11,768
Scan	**0.00**	**0.00**	**1.00**	**1.00**	**0.00**	**0.00**	7

The first observation to be made is that the overall detection performances are very inferior to those of the base ML models. By looking at the confusion matrix in Figure 5, we can more visually interpret the classification errors of the surrogate models: on one hand, the APCML method outperformed the AML method, with both a better recall for the DoS class as well as a higher precision for the Scan class, which do not appear in the metrics due to the very low support. As the ML models did all misclassify the MITM and Physical Fault as the Normal class, it is to be expected that the surrogate models are, in turn, showing confusion between these 3 classes. However, it is apparent from their

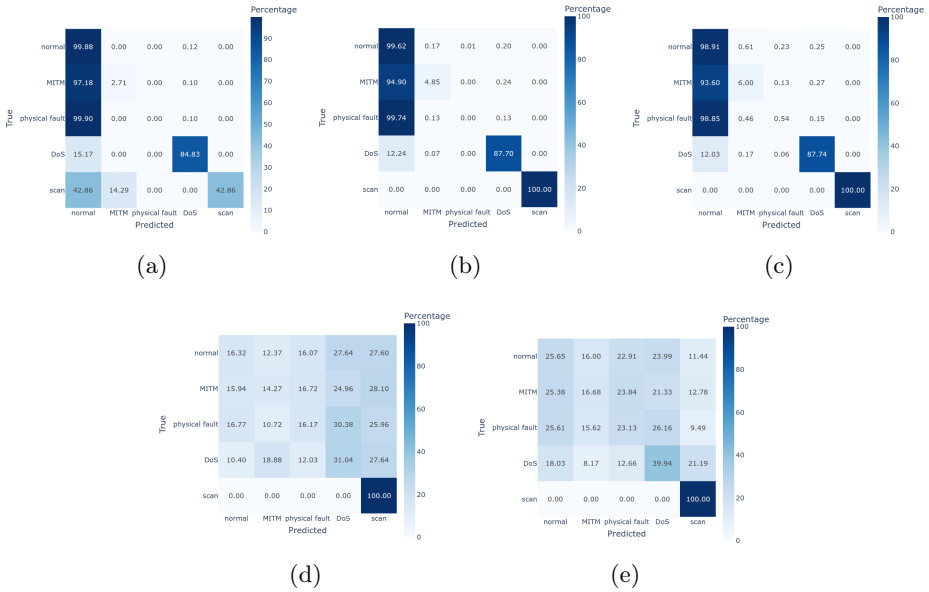

Fig. 5: Confusion matrix of classification performances on the test set of (a) MLP, (b) XGB, (c) Bagging, (d) surrogate AML, and (e) Surrogate APCML. The numbers in each cell represent the percentage of true classes predicted as the corresponding class on the x-axis.

error rates on the DoS and Scan classes that the fine class boundaries learned by the original models are only roughly approximated by the surrogates.

5 Discussion

The experimental results indicate that the proposed surrogate models can preserve a coarse class structure of the original ML classifiers, while remaining attractive for distributed edge deployment thanks to their lightweight log-likelihood-based inference. Nevertheless, the results also expose clear limitations of the current approach. The surrogates exhibit substantial overlap between class models, leading to a notable degradation in both precision and recall across all classes.

A major contributing factor to this drop in detection performance is the strong independence assumption enforced in the *het-EDA*, together with the use of a single normal distribution for continuous attributes. Although these choices enable heterogeneous modeling with minimal computational cost, they are highly restrictive for industrial network communication data, which often exhibits correlated features and heavy-tailed distributions. As a consequence, the surrogate models struggle to capture the complex dependencies that underlie real-world traffic in WDN cyber-physical systems.

These observations highlight the need for EDA capable of jointly modeling dependencies both within and across heterogeneous feature types, while remaining compatible with GPU-parallelization. Such advances would substantially improve the fidelity of EDA-based surrogates for ML decision regions.

Beyond classification, viewing ML decision boundaries through the lens of EDA opens promising research directions. First, sampling from class-wise probabilistic models could support targeted data augmentation. Second, the likelihood of an input under these models naturally lends itself to out-of-distribution detection for inputs that exhibit low log-likelihood for all modeled classes. Finally, replacing point-wise likelihoods with distances between probability distributions may support sequence-level classification, where the empirical distribution of observations in a time window is compared to learned class distributions.

6 Conclusions and Perspectives

We propose and evaluate a hybrid EDA-ML approach to construct probabilistic surrogate models of ML classifiers for cyber-attack detection. By replacing the original inference function with a lightweight log-likelihood-based surrogate, our method is better suited for deployment on edge devices in distributed WDN environments. To handle heterogeneous network communication data, we introduced the *het-EDA* algorithm, which supports both categorical and continuous features and uses trained ML classification scores as the optimization function to evolve class-wise surrogate models.

Experiments on the HITL cyber-physical dataset demonstrate that the surrogate models can preserve portions of the original decision boundaries, though they exhibit substantial overlap between classes. These results pave the way for broader applications of EDA-based probabilistic modeling for ML detection and highlight the need for EDAs capable of capturing the complex dependencies inherent in industrial network traffic, while providing lightweight inference suitable for edge deployment.

Acknowledgments. This study was funded by French Agence Nationale de la Recherche under grant ANR-22-CE39-0010 for CoRREau Project.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Alakkari, K., Subhi, A.A., Alkattan, H., Kadi, A., Malinin, A., Potoroko, I., Abotaleb, M., El-kenawy, E.S.M.: A Comprehensive Approach to Cyberattack Detection in Edge Computing Environments. Journal of Cybersecurity & Information Management **13**(1), 69 (Jan 2024). https://doi.org/10.54216/JCIM.130107, https://openurl.ebsco.com/contentitem/doi:10.54216%2FJCIM.130107?sid=ebsco:plink:crawler&id=ebsco:doi:10.54216%2FJCIM.130107

2. Asiri, M., Saxena, N., Gjomemo, R., Burnap, P.: Understanding Indicators of Compromise against Cyber-attacks in Industrial Control Systems: A Security Perspective. ACM Trans. Cyber-Phys. Syst. **7**(2), 15:1–15:33 (Apr 2023). https://doi.org/10.1145/3587255, https://doi.org/10.1145/3587255
3. Bakalos, N., Voulodimos, A., Doulamis, N., Doulamis, A., Ostfeld, A., Salomons, E., Caubet, J., Jimenez, V., Li, P.: Protecting Water Infrastructure From Cyber and Physical Threats: Using Multimodal Data Fusion and Adaptive Deep Learning to Monitor Critical Systems. IEEE Signal Processing Magazine **36**(2), 36–48 (Mar 2019). https://doi.org/10.1109/MSP.2018.2885359, https://ieeexplore.ieee.org/document/8653521
4. Ben Jedidia, F., Doerr, B., Krejca, M.S.: Estimation-of-Distribution Algorithms for Multi-Valued Decision Variables. In: Proceedings of the Genetic and Evolutionary Computation Conference. pp. 230–238. GECCO '23, Association for Computing Machinery, New York, NY, USA (Jul 2023). https://doi.org/10.1145/3583131.3590523, https://doi.org/10.1145/3583131.3590523
5. Bigliardi, B., Bottani, E., Casella, G.: Enabling technologies, application areas and impact of industry 4.0: a bibliographic analysis. Procedia Manufacturing **42**, 322–326 (Jan 2020). https://doi.org/10.1016/j.promfg.2020.02.086, https://www.sciencedirect.com/science/article/pii/S235197892030651X
6. Cao, K., Liu, Y., Meng, G., Sun, Q.: An Overview on Edge Computing Research. IEEE Access **8**, 85714–85728 (2020). https://doi.org/10.1109/ACCESS.2020.2991734, https://ieeexplore.ieee.org/abstract/document/9083958
7. Cover, T., Hart, P.: Nearest neighbor pattern classification. IEEE Transactions on Information Theory **13**(1), 21–27 (Jan 1967). https://doi.org/10.1109/TIT.1967.1053964, https://ieeexplore.ieee.org/document/1053964
8. Faramondi, L., Flammini, F., Guarino, S., Setola, R.: A Hardware-in-the-Loop Water Distribution Testbed Dataset for Cyber-Physical Security Testing. IEEE Access **9**, 122385–122396 (2021). https://doi.org/10.1109/ACCESS.2021.3109465, conference Name: IEEE Access
9. Faramondi, L., Flammini, F., Guarino, S., Setola, R.: A hybrid behavior- and Bayesian network-based framework for cyber–physical anomaly detection. Computers and Electrical Engineering **112**, 108988 (Dec 2023). https://doi.org/10.1016/j.compeleceng.2023.108988, https://www.sciencedirect.com/science/article/pii/S0045790623004123
10. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press (2016)
11. Gueye, T., Iqbal, A., Wang, Y., Mushtaq, R.T., Petra, M.I.: Bridging the Cybersecurity Gap: A Comprehensive Analysis of Threats to Power Systems, Water Storage, and Gas Network Industrial Control and Automation Systems. Electronics **13**(5), 837 (Jan 2024). https://doi.org/10.3390/electronics13050837, https://www.mdpi.com/2079-9292/13/5/837, number: 5 Publisher: Multidisciplinary Digital Publishing Institute
12. Gulzar, Q., Mustafa, K.: An analytical survey of cyber-physical systems in water treatment and distribution: Security challenges, intrusion detection, and future directions. SECURITY AND PRIVACY **7**(6), e440 (2024). https://doi.org/10.1002/spy2.440, https://onlinelibrary.wiley.com/doi/abs/10.1002/spy2.440, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/spy2.440
13. Hassanzadeh, A., Rasekh, A., Galelli, S., Aghashahi, M., Taormina, R., Ostfeld, A., Banks, M.K.: A Review of Cybersecurity Incidents in the Water Sector. Journal of Environmental Engineering **146**(5), 03120003 (May 2020). https://doi.org/10.1061/(ASCE)EE.1943-7870.0001686, https://ascelibrary.org/

doi/10.1061/%28ASCE%29EE.1943-7870.0001686, publisher: American Society of Civil Engineers

14. Hindy, H., Brosset, D., Bayne, E., Seeam, A., Bellekens, X.: Improving SIEM for critical SCADA water infrastructures using machine learning: International Workshop on the Security of Industrial Control Systems and Cyber-Physical Systems. Computer security pp. 3–19 (Mar 2019). https://doi.org/10.1007/978-3-030-12786-2_1, https://github.com/AbertayMachineLearningGroup/machine-learning-SIEM-water-infrastructure, publisher: Springer
15. Hochreiter, S., Schmidhuber, J.: Long Short-Term Memory. Neural Computation **9**(8), 1735–1780 (Nov 1997). https://doi.org/10.1162/neco.1997.9.8.1735, https://doi.org/10.1162/neco.1997.9.8.1735
16. Huang, B., Cheng, R., Li, Z., Jin, Y., Tan, K.C.: EvoX: A Distributed GPU-Accelerated Framework for Scalable Evolutionary Computation. IEEE Transactions on Evolutionary Computation **29**(5), 1649–1662 (Oct 2025). https://doi.org/10.1109/TEVC.2024.3388550, https://ieeexplore.ieee.org/document/10499977
17. Kadosh, N., Frid, A., Housh, M.: Detecting Cyber-Physical Attacks in Water Distribution Systems: One-Class Classifier Approach. Journal of Water Resources Planning and Management **146**(8), 04020060 (Aug 2020). https://doi.org/10.1061/(ASCE)WR.1943-5452.0001259, https://ascelibrary.org/doi/10.1061/%28ASCE%29WR.1943-5452.0001259, publisher: American Society of Civil Engineers
18. Klingenberg, C.O., Borges, M.A.V., Antunes, J.A.d.V.: Industry 4.0: What makes it a revolution? A historical framework to understand the phenomenon. Technology in Society **70**, 102009 (Aug 2022). https://doi.org/10.1016/j.techsoc.2022.102009, https://www.sciencedirect.com/science/article/pii/S0160791X22001506
19. Kozik, R., Choraś, M., Ficco, M., Palmieri, F.: A scalable distributed machine learning approach for attack detection in edge computing environments. Journal of Parallel and Distributed Computing **119**, 18–26 (Sep 2018). https://doi.org/10.1016/j.jpdc.2018.03.006, https://www.sciencedirect.com/science/article/pii/S0743731518302004
20. Kwon, H.Y., Kim, T., Lee, M.K.: Advanced Intrusion Detection Combining Signature-Based and Behavior-Based Detection Methods. Electronics **11**(6), 867 (Jan 2022). https://doi.org/10.3390/electronics11060867, https://www.mdpi.com/2079-9292/11/6/867, number: 6 Publisher: Multidisciplinary Digital Publishing Institute
21. Macas, M., Wu, C.: An Unsupervised Framework for Anomaly Detection in a Water Treatment System. In: 2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA). pp. 1298–1305 (Dec 2019). https://doi.org/10.1109/ICMLA.2019.00212, https://ieeexplore.ieee.org/document/8999210
22. Mboweni, I.V., Abu-Mahfouz, A.M., Ramotsoela, D.T.: A Machine Learning approach to Intrusion Detection in Water Distribution Systems – A Review. In: IECON 2021 – 47th Annual Conference of the IEEE Industrial Electronics Society. pp. 1–7 (Oct 2021). https://doi.org/10.1109/IECON48115.2021.9589237, https://ieeexplore.ieee.org/document/9589237, iSSN: 2577-1647
23. Miciolino, E., Bernieri, G., Panzieri, S., Pascucci, F., Polycarpou, M., Setola, R.: Fault Diagnosis and Network Anomaly Detection in Water Infrastructures.

IEEE Design & Test **PP**, 1–1 (Mar 2017). https://doi.org/10.1109/MDAT.2017.2682223
24. Murphy, K.P.: Machine Learning: A Probabilistic Perspective. MIT Press (Sep 2012), google-Books-ID: RC43AgAAQBAJ
25. Pandey, R.K., Das, T.K.: Anomaly detection in cyber-physical systems using actuator state transition model. International Journal of Information Technology **17**(3), 1509–1521 (Apr 2025). https://doi.org/10.1007/s41870-024-02128-x, https://doi.org/10.1007/s41870-024-02128-x
26. Rokstad, M.M., Ugarelli, R.M.: Minimising the total cost of renewal and risk of water infrastructure assets by grouping renewal interventions. Reliability Engineering & System Safety **142**, 148–160 (Oct 2015). https://doi.org/10.1016/j.ress.2015.05.014, https://www.sciencedirect.com/science/article/pii/S0951832015001635
27. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning representations by back-propagating errors. Nature **323**(6088), 533–536 (Oct 1986). https://doi.org/10.1038/323533a0, https://www.nature.com/articles/323533a0, publisher: Nature Publishing Group
28. Schölkopf, B., Platt, J.C., Shawe-Taylor, J., Smola, A.J., Williamson, R.C.: Estimating the Support of a High-Dimensional Distribution. Neural Computation **13**(7), 1443–1471 (Jul 2001). https://doi.org/10.1162/089976601750264965, https://doi.org/10.1162/089976601750264965
29. Sebag, M., Ducoulombier, A.: Extending population-based incremental learning to continuous search spaces. In: Eiben, A.E., Bäck, T., Schoenauer, M., Schwefel, H.P. (eds.) Parallel Problem Solving from Nature — PPSN V. pp. 418–427. Springer, Berlin, Heidelberg (1998). https://doi.org/10.1007/BFb0056884
30. Shi, W., Cao, J., Zhang, Q., Li, Y., Xu, L.: Edge Computing: Vision and Challenges. IEEE Internet of Things Journal **3**(5), 637–646 (Oct 2016). https://doi.org/10.1109/JIOT.2016.2579198, https://ieeexplore.ieee.org/document/7488250
31. Tuptuk, N., Hazell, P., Watson, J., Hailes, S.: A Systematic Review of the State of Cyber-Security in Water Systems. Water **13**(1), 81 (Jan 2021). https://doi.org/10.3390/w13010081, https://www.mdpi.com/2073-4441/13/1/81, number: 1 Publisher: Multidisciplinary Digital Publishing Institute
32. Vialatoux, C.F., Parrend, P.: Introducing Multi-Layer Concatenation as a Scheme to Combine Information in Water Distribution Cyber-Physical Systems. Procedia Computer Science **246**, 1840–1854 (Jan 2024). https://doi.org/10.1016/j.procs.2024.09.690, https://www.sciencedirect.com/science/article/pii/S1877050924027509
33. Xiao, Y., Jia, Y., Liu, C., Cheng, X., Yu, J., Lv, W.: Edge Computing Security: State of the Art and Challenges. Proceedings of the IEEE **107**(8), 1608–1631 (Aug 2019). https://doi.org/10.1109/JPROC.2019.2918437, https://ieeexplore.ieee.org/abstract/document/8741060
34. Yoong, C.H., Heng, J.: Framework for Continuous System Security Protection in SWaT. In: Proceedings of the 2019 3rd International Symposium on Computer Science and Intelligent Control. pp. 1–6. ISCSIC 2019, Association for Computing Machinery, New York, NY, USA (Jun 2020). https://doi.org/10.1145/3386164.3387297, https://doi.org/10.1145/3386164.3387297
35. Zhang, Y., Liu, Y., Guo, X., Liu, Z., Zhang, X., Liang, K., Zhang, Y., Liu, Y., Guo, X., Liu, Z., Zhang, X., Liang, K.: A BiLSTM-Based DDoS Attack Detection Method for Edge Computing. Energies **15**(21) (Oct 2022). https://doi.org/10.3390/en15217882, https://www.mdpi.com/1996-1073/15/21/7882