IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 8, 2021, accepted August 25, 2021, date of publication September 14, 2021, date of current version October 1, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3112397

Explainable Unsupervised Machine Learning for Cyber-Physical Systems

CHATHURIKA S. WICKRAMASINGHE[1], KASUN AMARASINGHE[2], DANIEL L. MARINO[1], CRAIG RIEGER[3], (Senior Member, IEEE), AND MILOS MANIC[1], (Fellow, IEEE)

[1]Department of Computer Science, Virginia Commonwealth University, Richmond, VA 23220, USA
[2]Carnegie Mellon University, Pittsburgh, PA 15213, USA
[3]Idaho National Laboratory (INL), Idaho Falls, ID 83415, USA

Corresponding author: Chathurika S. Wickramasinghe (brahmanacsw@vcu.edu)

This work was supported in part by the Commonwealth Cyber Initiative, an investment in the advancement of cyber research and development, innovation and workforce development (for more information about CCI, visit cyberinitiative.org).

ABSTRACT Cyber-Physical Systems (CPSs) play a critical role in our modern infrastructure due to their capability to connect computing resources with physical systems. As such, topics such as reliability, performance, and security of CPSs continue to receive increased attention from the research community. CPSs produce massive amounts of data, creating opportunities to use predictive Machine Learning (ML) models for performance monitoring and optimization, preventive maintenance, and threat detection. However, the ''black-box'' nature of complex ML models is a drawback when used in safety-critical systems such as CPSs. While explainable ML has been an active research area in recent years, much of the work has been focused on supervised learning. As CPSs rapidly produce massive amounts of unlabeled data, relying on supervised learning alone is not sufficient for data-driven decision making in CPSs. Therefore, if we are to maximize the use of ML in CPSs, it is necessary to have explainable unsupervised ML models. In this paper, we outline how unsupervised explainable ML could be used within CPSs. We review the existing work in unsupervised ML, present initial desiderata of explainable unsupervised ML for CPS, and present a Self-Organizing Maps based explainable clustering methodology which generates global and local explanations. We evaluate the fidelity of the generated explanations using feature perturbation techniques. The results show that the proposed method identifies the most important features responsible for the decision-making process of Self-organizing Maps. Further, we demonstrated that explainable Self-Organizing Maps are a strong candidate for explainable unsupervised machine learning by comparing its model capabilities and limitations with current explainable unsupervised methods.

INDEX TERMS Explainable artificial intelligence, self-organizing maps, interpretable machine learning, unsupervised machine learning.

I. INTRODUCTION

Cyber-Physical Systems (CPSs) are capable of seamlessly integrating computing and physical resources [1], [2]. Integration of physical components with cyber components allows resizing and reconfiguration of CPS, resulting in better scalability and flexibility than traditional standalone systems. Computing resources allows better information flow within CPSs, resulting in production efficiency (decreasing the production downtimes, increasing product quality, adjusting production planning) while reduced building and operations system costs [2]. Due to the various advantages of CPS, modern critical infrastructure has become heavily reliant on them. Therefore, it is important to research on building more efficient, reliable, and safe CPSs with innovative capabilities to address the needs of humans. Many independent agencies and national institutes such as the National Science Foundation (NSF), U.S. Department of Homeland Security (DHS), U.S. Department of Transportation (DOT), National Institute of Health (NIH), National Institute of Biomedical Imaging and Bio-engineering (NIBIB), National Cancer Institute (NCI), and European Commission (E.C.) have recently put their attention towards the advancements of CPS. Their interests include Internet of Things, Industrial Internet, Smart

The associate editor coordinating the review of this manuscript and approving it for publication was Fabrizio Marozzo.

Cities, Smart Grids, and ''smart'' anything (Manufacturing, Cars, Buildings) [2]–[6].

Due to the widespread usage and economic benefits of CPSs, ensuring the secure, reliable, resilient, and consistent performance of CPSs is crucial. One solution is to apply data-driven machine learning methods to the massive amount of data generated through these CPSs to improve their operation reliability, improve their performance (in terms of production capacity and cost), performance optimization, preventive maintenance, and threat detection [2], [3], [7]. Despite the tremendous benefits of machine learning (AI), many people hesitate to trust AI-based systems due to their black-box nature, which makes it difficult to get insight into the internal decision-making process of AI models [8]. Especially for human-in-the-loop systems, humans need to understand these algorithms such that they can trust these models. By addressing this question, the explainable machine learning (XAI) research area has been received a lot of attention. The goal of XAI is to provide reasoning for ML model outputs, allowing humans to understand and trust ML models' decision-making process. Currently, many entities have put their attention to XAI. DARPA is one of the first organizations that initiated XAI programs focusing on developing explainable models [9]. Their program is interested in developing a toolkit library consisting of machine learning and human-computer interface software modules that could be used to develop future explainable AI systems. Currently, many entities have put their attention to XAI such as European Commission, NSF, NIST, and IBM [10]–[12].

While XAI has become a trendy research topic, the majority of the work has been focused on supervised machine learning methods. However, real-world settings such as CPSs bring the challenge of dealing with high volumes of unlabeled data at a rapid pace. The manual labeling process is expensive, time-consuming, and requires the expertise of the data [13]. It has been found that the 25% of time allocated to machine learning projects is for data labeling. Further, supervised feature learning is not only unable to take advantage of the abundance of real-world unlabelled data, but it also can result in biases by relying on labeled data. These limitation has gained the focus towards unsupervised ML algorithms and is predicted to be far more important in the long term [14]. Given the abundance of real-world unlabelled data, it is important to focus on developing explainable unsupervised ML methods. However, in the current literature, very little work has been performed focusing on explainable unsupervised ML. Therefore, this paper focuses on unsupervised explainable ML.

In this paper, we explore Explainable Unsupervised Machine Learning on different aspects. Further, we propose a novel Explainable Unsupervised Machine Learning (XUnML) approach using the Self Organizing Map (SOM) algorithm, a widely used unsupervised algorithm with various Visual Data Mining (VDM) capabilities.

This paper has the following contributions:

- Brief overview of Supervised Machine Learning (SML), Unsupervised Machine Learning (UnML), and Explainable Machine Learning (XAI)
- Exploring initial desiderata towards Explainable UnML (XUnML), defining XUnML terminology based on the terminology used for XAI, and exploring the necessity of XUnML for CPSs
- Propose a novel methodology for explaining Self Organizing Map algorithm and exploring its advantages for specific requirements of CPSs

The rest of the paper is organized as follows. Section II provides the background and related work on Supervised Machine Learning (SML), Unsupervised Machine Learning (UnML), and Explainable Machine Learning (XAI); Section III presents the initial desiderata towards Explainable UnML (XUnML); Section IV presents the Explainable SOM approach; Section IV presents the experimental setup and results, and finally, Section V concludes the paper.

II. BACKGROUND

In this section, we discuss relevant literature briefly. We discuss SML, UnML, and different application areas of using them. Further, we discuss current literature on XAI and its terminologies.

A. SUPERVISED MACHINE LEARNING

Supervised Machine Learning (SML) algorithms require prior knowledge of data to train them and make predictions. SML is frequently used in data science due to its high predictive performance. However, the major drawback of these algorithms is that they cannot be trained with unlabelled data. SML algorithms can be categorized into main areas, namely classification algorithms and regression, which are briefly explained below [15].

- Classification: Classification algorithms require class labels as categorical variables. Therefore, it limits the number of possible prediction outcomes to a finite set of categorical variables. Widely used classification algorithms includes Support Vector Machines, Decision Trees, Random Forest, Neive-Bayes, K Nearest Neighbor, and Supervised Neural Networks [16], [17] These algorithms can be further categorized into binary classification and multi-class classification. Binary classification algorithms categorize data samples into two classes, whereas multi-class algorithms can categorize data samples into more than two classes.
- Regression: Regression algorithms can take data labels as real value and predict real value as an output. Hence, the outcomes of regression algorithms can have an infinite number of values. Widely used regression algorithms include linear regression, logistic regression, and polynomial regression [16], [17].

B. UNSUPERVISED MACHINE LEARNING

Unsupervised Machine Learning (UnML) has gained significant attention during the last decade. The main reason

for this is the large amount of unlabelled data generated to the public. To use these data effectively and efficiently, it is crucial to analyze these unlabeled data (exploratory data analysis) to identify hidden patterns within them and reduce the amount of data for high-level tasks such as labeling through dimensionality reduction [18]. In this way, UnML provides initial insight into data allowing domain experts to use them appropriately.

The traditional concept of UnML was mainly limited to the idea of exploratory data analysis and dimensionality reduction. The expansion of deep learning methods and data mining, combined with this era of big data, has given a much broader perspective to traditional unsupervised learning. Therefore, unsupervised learning is used not only for clustering and dimentionality reduction [18], but also for generative modelling [19], [20], auto-regressive modelling [21], [22] and represntation learning (unsupervised feature learning) [23]. Some of the widely used application areas of UnML techniques are discussed below.

- Clustering: Clustering is one of the most common uses of UnML, where it organizes data into sensible groups based on similarities and characteristics of data [24].
- Pre-trained models in transfer learning: This is the process of learning a machine learning model from a substantial amount of unlabeled data and using these pre-trained models for similar problem domains. These learned representations, have shown improved performance on downstream tasks for which the amount of data is limited, e.g., deep neural networks. [25].
- Unsupervised feature learning: This is the process of learning useful representations of data without manual annotations [26]. When the learned representation has a lower dimension than the input dimension, it is referred to as dimensionality reduction [27].
- Dimensionality reduction: This is the process of learning a low dimensional representation of the data set while preserving topological properties of data [28]. This low dimension can be either in the number of data points or the number of features in each data point.
- Association Rule Mining: This is the process of finding interesting associations (relationships, dependencies) in large sets of data items [29].
- Generative modeling: This is a typical use of unsupervised learning that models the probability distribution of data for generating new samples from the learned distribution [30]. These learned distributions are used to find good representations for large data sets and deal with missing data.
- Auto-regressive modeling: This is a process of time series modeling that uses previous observation from the previous timestamp as input to predict the value of the next timestamp [21].

C. EXPLAINABLE MACHINE LEARNING

As we discussed in Section I, the effectiveness of AI systems was limited by the inability to explain its decision-making process to human users (black-box behavior) [31]–[33]. This has triggered a new research area named Interpretable Machine Learning or Explainable Artificial Intelligence (XAI). XAI focuses on making machine learning models with the ability to explain their rationale, characterize their strengths and weaknesses, and convey an understanding of how they will behave in the future. It allows to produce AI models with high-performance levels while allowing users to understand, trust, and effectively manage machine learning algorithms [8], [31]. Explainable AI research can take two main approaches: 1) developing novel explainable machine learning algorithms, 2) modify the existing machine learning algorithms to make them understandable to humans.

Based on the literature, the need for XAI consists of four somewhat overlapping reasons [8]; Explain to Justify, Explain to Control, Explain to Improve, and Explain to Discover. Explain to justify refers to the need for reasons/justifications for a particular outcome, rather than providing a description of the inner workings or the logic of reasoning behind the decision-making process in general. It ensures that the AI-based decisions were not made erroneously. Explain to Control protect models from making wrongful outcomes by providing visibility of unknowns vulnerabilities, flows, and help to identify and correct errors through debugging. Explain to Improve refers to the fact that explainable and understandable models are easier to improved. Since the user knows why the model produces certain outcomes and flows, users can make models smarter through continuous improvements. Explain to Discover refers to explaining to learn new facts, gather information, and gain knowledge. The learned pattern from machine learning models can result in some new and hidden knowledge revealed through explanations. Explainable machine learning is a diverse research area that consists of many components. Figure 1 presents a taxonomy of XAI and a list of common terms used in XAI. They are briefly described below.

- Intrinsic or Extrinsic (post hoc): This distinguishes whether the model itself is interpretable or needs to apply methods that analyze models after training to achieve interpretability [34]. Intrinsic refers to simple, explainable models such as short decision trees. Extrinsic refers to the use of an interpretation method after training to achieve interpretability.
- Model Specific or Model Agnostic: This distinguishes whether the interpretation method is limited to a specific model or not [34]. Model-specific refers to methods and tools which are specific to a model (Ex: regression weights in a linear model, tools only work for neural networks). Model agnostic refers to methods that can use on any machine learning model to achieve interpretability. These models do not have access to internal model details such as weights or structural details.
- Local or Global: This distinguishes whether the interpretation method explains a specific data record or the entire behavior of a model [34]. Local refers to methods

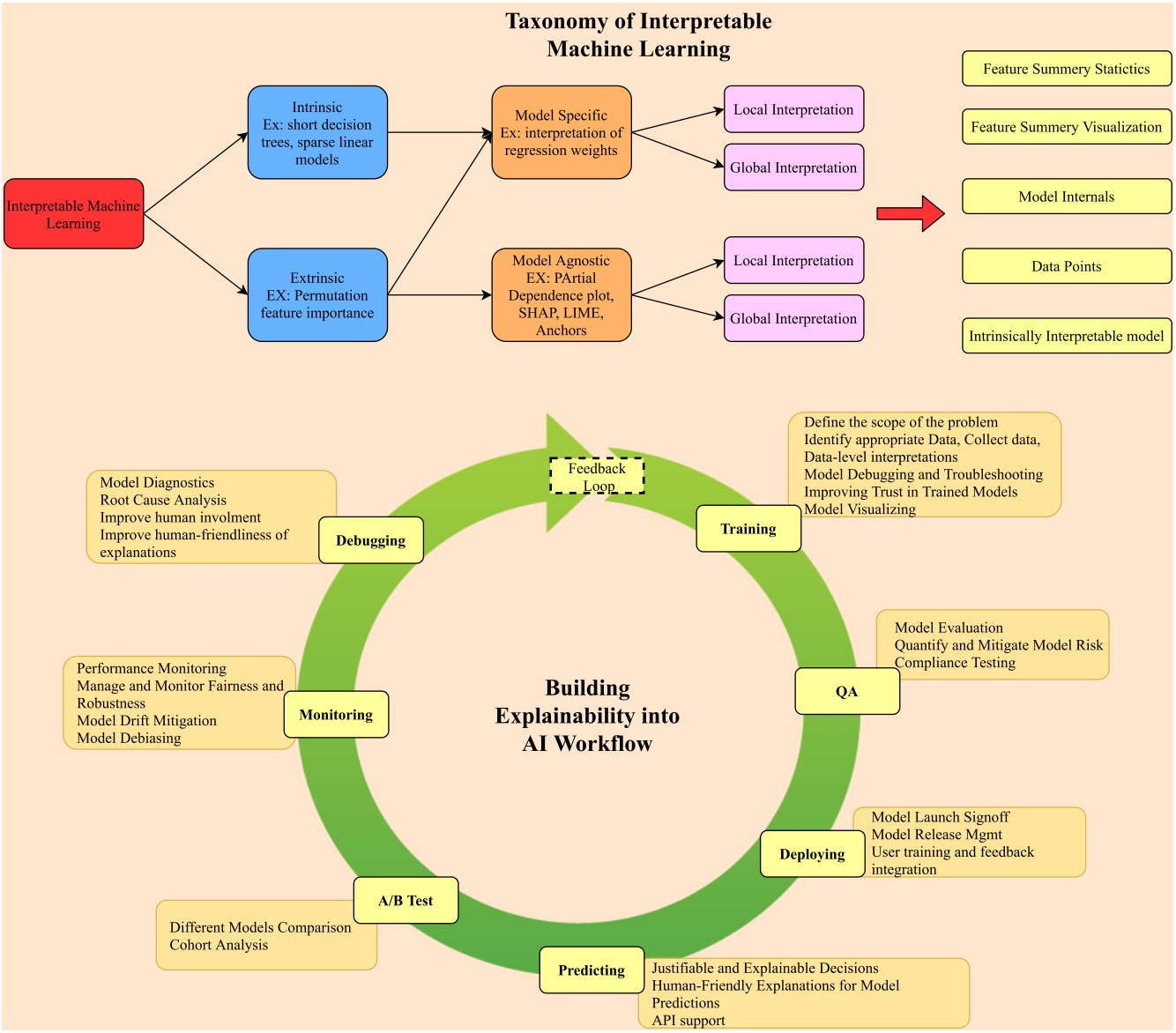

FIGURE 1. XAI concept and taxonomy.

that explain specific prediction, whereas global refers to methods and tools which provide interpretation for the entire model.

- Result of the interpretation method: The various interpretation methods result in various interpretation outcomes [34]. Some of them are listed below,
  - Feature summary statistic: Interpretation methods can result in feature summary statistics for a single feature or multiple features together. For example, it can be a feature importance score for each feature or pair-wise feature importance [34].
  - Feature summary visualization: Some feature statistics are meaningful only when presented visually. For example, partial dependence plots show the dependence between the output of the model and a set of input features. If this result is presented in tabular format, it is difficult to see the dependency between features and the model outcome [34].
  - Model internals (e.g., learned weights): Typically, intrinsic interpretable models result in model internals such as learned weights in linear models and tree structure of decision trees [34].
  - Data point: Some models' output already exists or newly created data points to make the model interpretable. For example, counterfactual explanation methods change the feature values of a data point to flip the class label of the data point [34].
  - Intrinsically interpretable model: Some black box models can be interpreted using interpretable

models. The result of this approach can be feature summary statistics or visualizations of the interpretable model [34].

Today, many companies such as Amazon, Google, NVIDIA Fiddler lab, IBM, and national institutes focus on adding explainability to the AI life cycle to ensure ethical and fair algorithms for their users. They point out that many people working within companies have no idea how to explain the inner workings of AI to customers. They are working towards bridging the gap between hardcore data scientists who are building the models and the business teams using these models to make decisions. Figure 1 present the idea of incorporating XAI into AI workflow. The idea here is to incorporate explainable techniques to different stages in the AI life cycle.

III. EXPLAINABLE UNSUPERVISED MACHINE LEARNING

As we discussed in the Introduction section, existing work on XAI is mainly concentrated on supervised learning algorithms. For domain areas such as CPSs, UnML is essential for assisting human decisions in building effective ML models. These systems generate a massive amount of unlabelled data at a rapid speed. Therefore, relying on SML alone is not sufficient for data-driven decision-making for CPSs. Further, unsupervised learning has a wide range of application areas, including model pre-training, auto-regressive modeling, and generative modeling. This section explores what XAI would look like in an unsupervised context, the need for unsupervised XAI methods, current literature on unsupervised XAI, and how unsupervised XAI can be used within the domain of CPSs.

A. DESIDERATA OF EXPLAINABLE UNSUPERVISED MACHINE LEARNING

As we discussed in the previous section, UnML offers a solution to analyze the large amount of real-world unlabelled data generated at a rapid speed. However, most of the existing UnML methods do not provide a way for people to understand their underlying decision-making process. Especially for non-domain experts, these models act as black-boxes. This black-box behavior leads to many drawbacks, including limiting the user's involvement with model improvement, limiting user input integration for model debugging, and harming the user trust in these models, making humans not deploy them in real-world environments [35]–[39]. Interpretable models are essential for high-risk environments where the model outcomes can result in severe consequences. For example, one use case is anomaly detection systems in critical infrastructures. On these systems, it is not enough to get the predictions (anomaly or not) of UnML models. It is crucial to produce an explanation of why it is an anomaly. This information is essential to identify where the anomaly occurred, possible catastrophic effects and make decisions to recover the system. Therefore, it is essential to analyze and explain the result obtained through these UnML models [35]–[39].

Analyzing and interpreting the results obtained through UnML is a very challenging process. This process often requires expert-based sophisticated manual inspection, which takes a significant amount of time [35], [37]. Further, complexities, high-dimensionality, and real-world data volume make it impossible to use manual expert-based data analysis. Existing unsupervised quality metrics such as Silhouette or Rank Index do not provide any explanations on why data record belongs to a specific cluster [35]. They only provide a structural insight that is not perceivable to non-domain experts. Further, many available methods are hard to explain, partially because they depend on all the data features in a complicated way, making it difficult to explain in a perceivable manner [38]. Other supervised quality metrics such as cluster purity requires labeled data, requiring expensive manual labeling. This approach is expensive and can result in partial, incorrect, or biased results [35]. Therefore, there is a crucial need to develop explainable UnML methods or develop methods to explain existing UnML methods.

B. MAPPING OF EXISTING EXPLAINABLE MACHINE LEARNING TERMS TO EXPLAINABLE UNSUPERVISED MACHINE LEARNING

As we described in the previous section, existing explainable AI mainly concentrates on supervised algorithms and is composed of many overlapping terms and concepts discussed. Therefore, it is essential to explore how these existing concept of XAI fits the unsupervised learning domain. Here we discuss our view on mapping from existing XAI concepts to the unsupervised domain.

The **Intrinsic or Extrinsic** model concepts can be used as it is in the domain of unsupervised learning. For example, unsupervised models like Principle Component Analysis visualized with two or three dimensions can be considered as an Intrinsic interpretable model. Association rule mining techniques can be considered as intrinsic models as they generate rules based on the conditions specified by the user. These conditions can utilize for generating interpretations. Unsupervised models like Mean Clustering go under Extrinsic interpretable models as they need external interpretation models after training to achieve interpretability.

The terms **Model Specific and Model Agnostic** can also be used as it is in the unsupervised domain. Small decision trees are one such example of Models Specific interpretable model as the splitting criteria used to explain decision trees are restricted to decision tree algorithms. Some existing agnostic models can be used to explain existing unsupervised clustering approaches. Typically, model agnostic models require labels on data records to achieve interpretability. We can use the cluster labels generated through unsupervised clustering algorithms as dummy labels to existing model agnostic methods. However, this area of research is still at a primitive stage.

In the unsupervised domain, **Local Interpretability** can be used to explain how a specific data point belongs to a given cluster or how to change the cluster label of a data point by changing its feature values. In auto-regressive modeling, we can present what features of the previous data records lead to predicting future data records. The **Global interpretation** can be defined as generating explanations on why a set of data points belongs to a specific cluster, the important features that decide the similarities between points within a cluster, and the feature value differences between different clusters.

Methods used in the supervised domain to present the result of interpretation models can also be mapped to the unsupervised domain. For example, important feature summary statistics can be presented using different visualization mediums such as bar charts, tabular format, and linguistic explanations for clustering tasks. Model internal values such as cluster centers of K-Means clustering can be used as a general representation for the data distribution.

It is important to notice that **qualitative and quantitative** analysis in unsupervised explainable models can be problematic. The main reason for this is, many existing model evaluation methods require some prior knowledge/labels of data. In the unsupervised domain, prior knowledge of data is not available. Further, described in previous section, available unsupervised quality metrics are not explainable. Therefore, the new evaluation mechanisms should be developed for explainable UnML methods.

One classic approach is to perform a human study, where machine learning experts apply the UnML method into an actual world application and provide global/local explanations to domain experts using appropriate visualization methods (Application-level evaluation). Domain experts can qualitatively evaluate explanations on whether the learned clusters represent some important similarities (human-level evaluation). Another approach is to use model fidelity which evaluates how truthfully the explanation represents the underlying model [40]. Model fidelity of UnML can evaluate by using the information on important subsets of features [41]s. These features can be perturbed, removed, or weighting can be used to get some notion of the truthfulness of features for the decision-making process on a model. For example, model faithfulness of clustering can be evaluated by checking how the cluster label changes when changing the feature values of data samples (quantitative).

C. CURRENT LITERATURE ON EXPLAINABLE UNSUPERVISED MACHINE LEARNING

Principle Component Analysis (PCA) has been used for interpreting the clusters by visualizing them across two or three dimensions [37]. However, it limits the number of dimensions that can be used for explaining the clusters, as visualizing is not possible when the number of dimensions increases. In [35], researchers have used existing supervised XAI methods for interpreting UnML approaches (EXPLAIN-IT). First, they cluster the input data using existing clustering methods such as K-Means or DBSCAN. A classifier is then trained on input data using the generated cluster labels as class labels for the classifier. Finally, the classifier is explained using existing model agnostic methods such as LIME. However, these can result in model biases, and the current research on this is at a primitive stage.

Interpretable tree-based clustering models have gained much attention recently as the decision tree model itself is an explainable model [42], [43]. In [42], [43], an explainable decision tree method was introduced by generating the smallest binary tree possible (threshold tree) with k leaves. Each node in the tree iteratively divides the input data into k clusters. By restricting to k leaves, they ensure that each such path accesses at most $k-1$ features. The explanations were generated using $k-1$ features. Also, in [36], researchers have proposed an explainable decision tree model (eUD3.5) where they have use compactness and separation of data clusters when evaluating feature splitting in the tree.

Deep Neural Networks (DNNs) have shown state-of-the-art performance in many areas such as computer vision and natural language processing. However, many DNNs are used as black-boxes. There are a couple of initial attempts toward explaining unsupervised DNNs such as Autoencoders. In [44]s, interpretable Variational AE has been presented. This is performed by analyzing the gradient contributed by each feature of a data record. Another interpretable VAE is presented in [45], and [46] by changing the decoder to embody explicit expert knowledge. Therefore, these architectures result in a latent space that has semantic meaning. Fuzzy logic combined with ML has also been used for achieving interpretability There are some initial attempts towards developing interpretable systems combining fuzzy logic systems with DNNs and clustering algorithms. However, majority of these system has some degree of the supervised learning process within their pipeline.

D. EXPLAINABLE UNSUPERVISED MACHINE LEARNING FOR CYBER-PHYSICAL SYSTEMS

As discussed in the Introduction, CPSs generate a high volume of unlabeled data at a rapid pace. Unsupervised machine learning is a viable solution to mine these data meaningfully, maintain and improve desired functionalities, and improve these systems' safety. In recent years, unsupervised learning has been used in CPSs mainly for three application areas of unsupervised leaning: clustering, unsupervised feature learning, and model pre-training. This section discusses these areas in brief and discusses the need for explainability. The main reason for this is, it is not possible to discuss explainability and the advantages of XAI in general without specifying a domain. Depending on the domain and application, the XAI models and mechanisms should be adapted to use them effectively. Therefore, we selected CPSs and above three UnML application areas for our discussion.

Clustering is the most commonly use of unsupervised learning method within CPSs. It groups samples based on some similarity criteria such that samples in the same group are similar to each other compared to samples in another

group. It has been applied to the abundance of unlabelled real-world CPS data for revealing hidden patterns and knowledge extraction. Clustering has been successfully used for many areas of CPSs including optimization the intelligent driver system performance [47], [48], prevention and detection of malicious activities [49], power consumption optimization [50], anomaly detection [51], [52], and performance degradation diagnostics and prognostics in CPSs [53]. Explainable clustering algorithms can have multiple advantages. Main advantage of explainable clustering is to summarize the input behavior patterns in the clusters allowing users to understand underlying commonalities of clusters. This is essential for domain experts to perform exploratory data analysis on unlabeled data, allowing them to understand the hidden patterns in data and outleirs within data. Further, clustering can be the first stage of identifying necessary data from large quantities of complex data in order to build ML systems for CPS. Meaningfully clusters data can be effectively use downstream ML tasks such as data labeling and applying supervised ML models. It also improve user trust in ML models allowing users to deploy them in real world applications.

Reliable and efficient modeling of high-dimensional data collected through CPSs require extracting only relevant and robust features through **unsupervised feature learning** techniques [27]. In the presence on unlabelled data, unsupervised feature learning techniques can be used. I.e., the process of converting a high-dimensional feature spaces into new embedded representation without using any prior knowledge or labels on data. Preferably, the learned new embedded representation will have lower dimension compared to the original dimension [13], [54]. Once relevant features are learned, these features are used for ML model training. Feature learning has been successfully used in CPSs for various tasks such as anomaly and threat detection [52], [55], mobile edge computing [56], heterogeneous data clustering [57], and intelligent manufacturing [58]. Widely used unsupervised feature learning techniques include variant of autoencoders (Stacked Autoencoder, Convolutional Autoencoders, ResNet Autoecoders), PCA, Locally Linear Embedding, and Singular Value Decomposition [27], [59], [60]. When looking at the need for explainability of unsupervised feature learning, the domain experts needs an explanations on why set of features are extracted, what is the linear or non-linear relationship between original feature space and the new embedded feature space, does learned features are actually important to some downstream task. This help domain experts and ML experts to perform meaningful feature learning and dimensionality reduction.

Pre-training is the process of learning a general representation using a substantial amount of labeled or unlabeled data and use the learned representation to improve performance on a downstream task where the data is limited [61], [62]. When the general representation is learned from unlabeled data, it is known as unsupervised pre-training [61]. Unsupervised pre-training has shown significant improvements across wide range of areas in CPSs including attack detection, security risk management, natural language processing, computer vision, and transfer learning [61]–[65]. It has been found that these pre-trained models perform better and can be efficiently re-trained to downstream tasks. Explainability is essential for these pre-trained models to understand what these models have actually learned so that domain experts can use improve these models by removing unwanted bias and to use these pre-trained models effectively in relevant applications.

IV. EXPLAINABLE SELF-ORGANIZING MAPS

This section discusses the Self Organizing Map (SOM) algorithm, advantages, and visual data mining capabilities of SOMs, and presents a novel explainable method for SOMs.

A. SELF-ORGANIZING MAPS

The Self-Organizing Map is a widely used unsupervised learning algorithm capable of mapping a high-dimensional data distribution onto a low-dimensional grid while preserving important topological, and metric relationships of the input data [66]–[68]. It consists of a topological neuron grid (typically 2D or 3D), with each neuron consisting of a weight vector. It adapts its neuron weight vectors to represent topological properties of input data using the unsupervised ''winner-take-all'' learning algorithm [69], [70]. Since SOMs can represent topological properties of input data, they have been widely used for visual data mining and dimensionality reduction [71], [72]. Other advantages of SOMs include ease of optimization [73], the better capability of revealing overlapping structures in clusters compared to other traditional clustering methods, and suitability for visualizing and mining high dimensional data [67]. SOMs have been successful in many areas, including speech recognition, robotics, telecommunication, and process optimization [69], [73]–[75].

The algorithm of SOM is presented in **Algorithm I** (Table 1). Each neuron in SOM maintains a weight vector $W = w_1, w_2, \ldots, w_m$ of m dimension, where m is the dimension of input feature vector. Input dataset can be represented as $X = x_1, x_2, \ldots, x_n$, where n is the number of records in the training set. For i^{th} input data record (x_i), the algorithm finds the closest neuron based some distance measurement calculation method (euclidean, Manhattan). This closest neuron is called as the Best Matching Unit (BMU). Then, the SOM neuron network updates the weights of the neurons in the neighborhood of the BMU so that the neighboring neurons move close to BMU (line 14-18 in **Algorithm I**). Figure 2 illustrates the structure of 2-D SOM architecture in the output space and the input space. The learning rate and the radius of the BMU neighborhood are used as the controlling hyperparameters. Typically, the neighborhood radius is halved and learning rate is decayed at each epoch.

$$\eta(t) = 0.49\left(1 - \frac{e}{epochs}\right) + 0.01 \tag{1}$$

where e is the current epoch and $epochs$ is the total number of epochs. The most crucial hyper-parameter of the SOM is

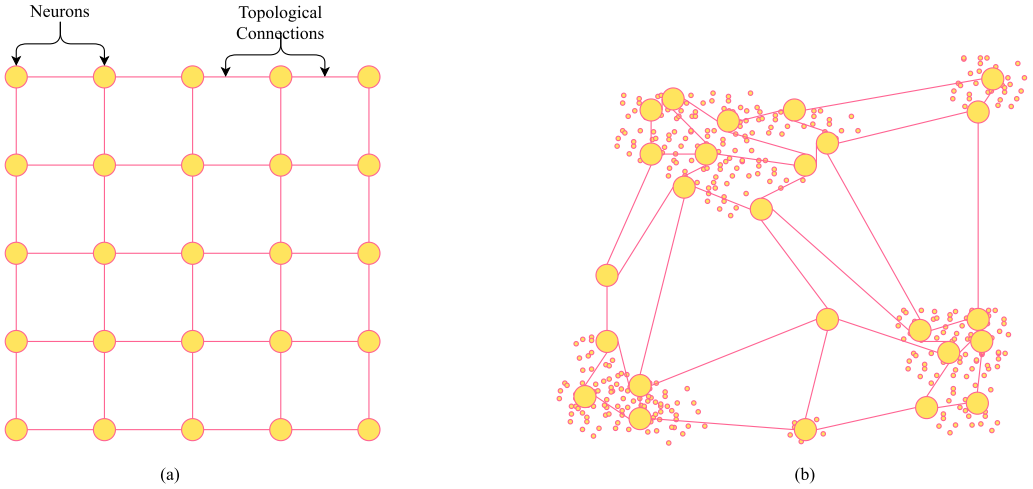

FIGURE 2. Self-Organizing Map displayed in the output space (a) and in the input space adapted to 2D distribution of input points (b).

TABLE 1. Algorithm for self-organizing map training.

Algorithm I: Training of SOM with Winner-Take-All algorithm	
Inputs: Training dataset (X)	
Outputs: Trained SOM	
1:	Initialization of SOM neuron weights randomly
2:	**FOR** each epoch e do
3:	**FOR** each data record in training data do
4:	$x \leftarrow$ pick random input data record from X
5:	$md \leftarrow$ initialize with very large float value
6:	**FOR** each neurons in SOM do
7:	$d_i \leftarrow \|X - W_i\|$ % find the BMU
8:	**if** $d_i < md$ do
9:	$BMU_x \leftarrow W_i$ % weight of BMU
10:	$BMUIndex_x \leftarrow i$ % index of BMU
11:	$md \leftarrow d_i$
12:	**END if**
13:	**END FOR**
	% update neuron weights
14:	**FOR** each neuron in SOM neighborhood do
15:	$n \longleftarrow e^{-\left(\frac{BMU_x - w}{2\delta t^2}\right)}$
16:	$\triangle W_i \longleftarrow W_i \times \alpha \times \eta \times (x - w)$
17:	$W_i \longleftarrow W_i + \triangle W_i$
18:	**END FOR**
	% Decay the neighborhood radius and learning rate
19:	**END FOR**
20:	**END FOR**

the size of the neuron map. The map size should be adequate enough not to over-fit or under fit to train data.

The following items outline exploratory data analysis capabilities of SOM.

- Histograms: These neuron histograms shows the data distribution of the 2D data topology of SOMs. It can be used as a visual indication for identifying whether the network can cluster the input data correctly. A properly trained network topically shows data grouped in some regions of the map (high data distribution density), making clear boundaries between clusters.
- T-distributed Stochastic Neighbor Embedding (t-SNE): This is a dimensionality reduction technique that is widely used for visualizing high-dimensional data. For SOMs, this can be used to represent the input data points and neuron weight together. It indicates to users that the network weights can represent the distribution of input data. Therefore, it is a clear indication to visually explore whether the trained network represents the trained data.
- Heat Maps: These are intensity representations of SOM network properties. Several types of heat-maps can be generated using cluster labeled or data labeled if available.
  - Class hits: This is a different visualization of Neuron Hit Histogram. This will represent the number of classes where each neuron fired for the whole SOM network topology. If majority of data fired a neuron belong to one class label, then it can be used as an indication for a well-trained SOM network.
  - Data hits: This represents the number of data points where each neuron fired for. If many neurons do not fire for any data point, then network size can be reduced. Therefore, this can be used to decide the SOM network size.
  - Class Percentages: This will represent the percentage purity of each neuron. Percentage purity can be used to ignore neurons with low purity allowing users to increase the quality of the network by retraining, redesigning, expanding the network. In case of tie, neighboring neurons are used to decide the class/cluster label.
- U-Matrix: Unified Distance Matrix (U-Matrix) is the standard visualization for SOMs representing the information regarding the distances between neighboring neurons. These maps are used to identify the naturally existing clusters and to identify well-separated clusters from overlapping clusters.

- Component Planes: This visualization shows the value of a single feature in each SOM neuron. A single component plane represents how a specific feature value changes across clusters. Further, by comparing multiple plans, it is possible to identify correlated features.
- U-Map: Similar to t-SNE, this also use to visualize high dimensional data. This builds a high dimensional graph representation of the input data then optimizes a low-dimensional graph to be as structurally similar as possible. For SOMs, this can be used to represent the input data points and neuron weight together. It indicates to users that the network weights can represent the distribution of input data.

As described above, SOM has many visual data mining capabilities which allows domain experts and non-domain experts to interact with SOM, making SOMs good candidates for exploring application in CPSs. Further, there have many improvements have been proposed that can be done on SOMs to improve it's capabilities. However, to the best of out knowledge, there is no efforts done towards making the model interpretable. Therefore, in the next section, we propose an approach toward developing an explainable SOM algorithms.

B. EXPLAINABLE SOMs

As we discussed above, SOM algorithm has many visual data mining capabilities. SOM is a unsupervised clustering method which is trained to produce a low dimensional representation of a large training dataset. U-Matrix of SOM neuron weights can represent any natural clusters available within training data. Component planes of SOM neuron weights can be used to visualize how the feature values change across clusters (feature summary visualization). In this paper, we used SOMs training approach (winner-take-all algorithm) together with the above discussed visual data mining capabilities of SOM to make the algorithm explainable. We propose a model-specific, post-hoc interpretable method for SOMs. The result of this method consists of feature summary statistics, model internals, and feature summary visualizations. The proposed approach is able to provide both global and local explanations. Further, we will discuss how each of these generated explanations can be used for CPS operations. Here we will discuss the steps for identifying most important feature list using SOM algorithm, model fidelity evaluation method, and generating interpretations.

1) Training of the SOM with dim *dXd*:
Trained the SOM with the winner-take-all algorithm presented in Algorithm 1.
2) Calculating the order of important features for each neuron (Algorithm II, line 1-13):
After training SOM with the training dataset, the trained SOM acts as a set of data points which represent the entire training dataset. Therefore, each neuron is a generalized representation of a set of training data records. We use the training data set and extract a set of data points (X_i') that selected i^th neuron as their BMU to calculate the ordered list of important features for i^th neuron. The importance of a feature is decided by calculating the standard deviation. We calculate the standard deviation for each feature j in x_i'.

$$\sigma f_j = \sqrt{\frac{\sum_{j=0}^{n}(X'_{i,j} - \bar{X'_{i,j}})}{n-1}} \tag{2}$$

3) Calculating the ordered list of important features for neuron i (Algorithm II, line 1-13):
Then the indices of features are ordered from lowest standard deviation to the highest standard deviation, representing the ordered list of feature from

TABLE 2. Proposed approach for explainable SOM.

Algorithm II: SOM Interpretation
Inputs: Trained SOM, Testing dataset (X), standard deviation threshold (th)
Outputs: Local interpretation, Global interpretation

```
% Calculating list of important features for each neuron in SOM
for each neuron i do
    X'_i ← initialize an empty array
    for each data record x in X do
        bmu ← calculate BMU using trained SOM
        if(bmu == i) : X'_i.append(x)
    end for
    for each feature j in n dimensional feature space
        % Calculating standard deviation for each feature
        σf_{i,j} = sqrt( Σ_{j=0}^{n} (X'_{i,j} − X̄'_{i,j}) / (n−1) )
        if(f_{i,j} > th) : f_{i,j} = INF
    end for
% Calculating the order of important features for each neuron in SOM
    f'_i ← sort indices j of σf_i in acceding order if f_{i,j}! = INF
end for
KMeans ← Apply K-Means clustering to SOM neurons and find optimal K
clus ← initialize an empty array for storing cluster labels
for each neuron i do
    clus_i ← apply KMeans to ith neuron and find its cluster label
end for
```

Algorithm III: Experiment I
Inputs: Trained SOM, X, f_i', $clus$
Outputs: Swap Percentages

```
X'_i ← initialize an empty array
count ← 0 %initialize a variable
for p number of features out of n where (p * 100/n)% <= 50%
    for each data record x in X do
        bmu ← calculate BMU for x using trained SOM
        x' ← change first p features of x from f'_b mu
        bmu' ← calculate BMU for x' using trained SOM
        if(clus_bmu ! = clus_bmu') : count + +
    end for
tot ← number of data records in X
Swap Percentage = count * 100/tot
end for
```

Algorithm IV: Experiment II
Inputs: Number of features (t), Trained SOM, X, f_i'
Outputs: Feature Percentages

```
list1 ← initialize an empty array
for each data record x in X do
    bmu ← calculate BMU for x using trained SOM
    l_d = |x − bmu| %Calculate L1 distance between x and bmu
    closets_features ← find closest t feature indices
    tot1 ← cardinality of closets_features
    tot2 ← cardinality of f'_bmu
    %percentage of t feature are in f'_bmu
    list1.append(tot1 * 100/tot2)
end for
tot ← number of data records in X
Percentage = sum(list1)/tot
end for
```

highest importance to lowest feature importance. Each neuron in SOM represents a set of data points, and low standard deviation of a feature represents low variation of a feature values, indicating that many data points have that feature value within a small range. The domain expert/user can decide on a threshold (*th*) standard deviation value so that any feature with equal or less standard deviation value is considered as the most important (active) feature. These ordered important features were used to achieve interpretability.

4) Cluster SOM neurons (Algorithm II, line 14):
To achieve interpretability, we clustered the trained SOM neurons. In this experiment, we used K-Means clustering. The number of clusters (K) was decided based on two cluster quality metrics: Silhouette Coefficient and Davies-Bouldin Index. Further, cluster quality metrics together with U-matrix and other visualization capabilities of SOM allows the domain expert to visually analyze the natural clusters of training data and evaluate the quality of K clusters to decide whether the results are reasonable.
5) Model fidelity evaluation:
We designed two experimentation to evaluate whether the identified features for each neuron are actually important for the decision-making process of SOM (fidelity test). This is performed by performing two experiments on the test data-set.
   - Changing the feature values of identified important features and checking whether the cluster labels (model outcomes) changes (Algorithm III, Experiment I). Through this experiment, we calculate the percentage of data points where the cluster label can be changes by changing their feature values of important features.
   - Calculating the percentage of important features which are included in identified important features (Algorithm IV, Experiment II) Through his experiment, we check whether the calculated order of importance feature lists are valid for the test data-set.

   These are discussed in detail in the next section.
6) Results interpretation: Once we identified the most important features and evaluated the features, we generated local and global explanations using SOM.
   - Local interpretability: Once an ordered set of important features are calculated for each neuron; it is used to generate local interpretation for a specific input record by providing the user a subset from important features and its value range (very low, low, medium, high, very high). It has to be noticed that the feature value granularity can be defined by users based on their preferences.
   - Global Interpretability: For each feature, we can visualize how the feature value is different across clusters and what features are active within clusters. Ordered important feature summary (features and values(range) of important features) for a set of neurons belonging to a particular cluster is used as a global interpretation.

V. EXPERIMENT SETUP AND RESULTS

In this section we discuss the five data sets we used for this experiment, the design of the evaluation methods, results, and discussion on results. First we will discuss the data sets used for this experiment.

KDD: This is a commonly used benchmark dataset for network intrusion detection and anomaly detection. It has around 2 million records divided into train and test sets. It consists of 41 features, and all the records are labeled into two classes, attack or normal. The attack data represent four categories, namely, Denial of Service (DOS), User to Root Attack (U2R), Remote to Local (RTL), and Probing Attack. For this experiment, we used normal records and DOS records (attack). This choice was made as other types of attacks have subcategories that do not include the test set. SOM algorithms, in general, does not handle huge variation in new data. This dataset has both categorical and numerical feature values.

German Credit: This dataset has 1000 data records with 20 features. It has both categorical and numerical feature values. Each record represents a person who takes a credit from a bank. Each person is labeled as good or bad credit risks.

Bank marketing: This dataset is derived from a marketing phone call campaign of a Portuguese banking institute. This data set has 45211 records, each with 12 features. Features are only numerical values. This has two classes, yes and no. The classification goal is to predict if the client will subscribe to a term deposit or not.

Adult Income: This dataset has 48842 records, each with 14 features. Features have both categorical and numerical features. The data set is labeled into two classes, representing whether the salary exceeds 50k or not based on the features. This dataset has missing values. In this experiment, we removed records with missing values.

DoHBrw-2020: Canadian Institute for Cybersecurity provides a set of datasets for building intrusion detection systems. For this experiment, we used the CIRA-CIC-DoHBrw-2020 dataset, which consists of benign and malicious records for DoH (DNS over HTTPS protocol) traffic along with non-DoH traffic. It consists of roughly around 250k records with 28 features. It has both categorical and numerical feature values. Benign and malicious records were considered as two classes.

Since these datasets have categorical variables, we used the frequency encoding method to convert categorical variables to nominal variables. Min-max scalar was used to scale the data into the 0-1 range. When there are separate train and test sets, they were used as it is for training and testing purposes. If the original data set is not divided into train and test,

70% of the data was randomly selected for training, and the rest was used for testing.

To decide the optimal number of clusters and dimension of the SOM, we used u-matrix together with two widely used clustering performance metrics, namely Silhouette Coefficient and Davies-Bouldin Index. They do not require labels to evaluate the clusters. They use different methods to calculate the compactness(density) of a cluster and separation (distance) between clusters.

- Silhouette Coefficient: This is calculated using the mean intra-cluster distance and the mean nearest-cluster distance for each sample. The score is higher when clusters are dense and well separated, which relates to a standard concept of a cluster. The score is bounded between -1 for incorrect clustering and $+1$ for highly dense clustering. Scores around zero indicate overlapping clusters.
- Davies-Bouldin Index: This index signifies the average 'similarity' between clusters, where the similarity is a measure that compares the distance between clusters with the size of the clusters themselves. A lower Davies-Bouldin index relates to a model with better separation between the clusters. Lower the better, Values closer to zero indicate a better partition.

Figure 3 shows the change in cluster quality matrices used for this experiment for the Bank Marketing data set. It shows how these matrices change with respect to SOM dimensions and the number of clusters. For a given SOM size, we calculate cluster quality metrics for SOM neuron weights (Blue) as well as for the training dataset (Orange). This analysis is used to identify the optimal SOM dimension and number of clusters. If the trained SOM neurons are a good representation of the whole data set, then cluster analysis of SOM weights and the whole data set should follow similar trends. Figure 3 shows that they follow the same trends when increasing the number of clusters. Based on the Silhouette Coefficient and Davies-Boulding index value, 3 to 5 clusters seem to be the best option for the tested SOM dimensions (8,16,2,40).

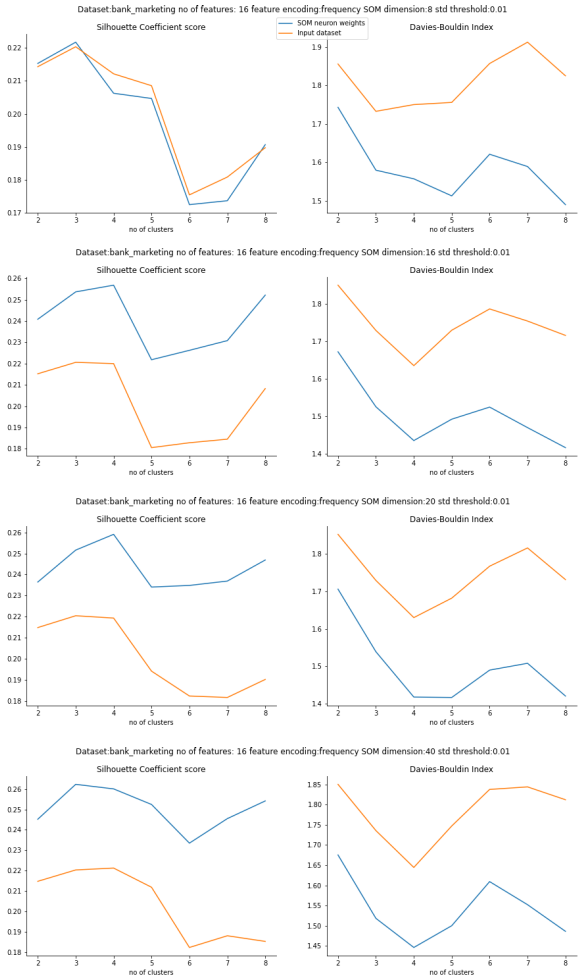

FIGURE 3. Cluster quality evaluation approach for K clusters using Silhouette Coefficient and Davies-Bouldin Index for different SOM map sizes.

A. MODEL FIDELITY

Once the ordered list of important features is calculated for each neuron in trained SOM, it is necessary to evaluate whether the identified ordered features are actually important using the a data perturbation experiments discussed in the previous section (Algorithm III). First, for each data point x in the test set, we check its BMU index i and the cluster label (m) of the BMU. Based on BMU index i, we have a list of most important features f_i'. It has to be noticed that different BMUs have a different number of important features based on a user-specified threshold (th) on standard deviation. To evaluate whether the identified ordered features are actually important, we change the feature values of important features of the test data record. First, we calculated the average feature values for each cluster using the neurons belong to that cluster. Then, the feature values of important features of x were replaced by the mean feature values of a cluster k where $k! = m$. Then we check whether the BMU of x is changed to another BMU, which does not belong to the original cluster label of that data point (m). Our hypothesize is that when we change the values of the most important features, the cluster label of the data point should change. We did this for the whole test data set and calculated the percentage of data records where we can change the original cluster label by changing the feature values of important features (Swap percentage). If it does, it confirms our hypothesis that the identified features decided the cluster label of that data point.

It is also essential to identify the minimum number of important features that define the cluster label of a data point. Explanations should be generated using a small number of features so that it is easy to perceive by the user rather than explaining with a higher number of features. Therefore, the cardinality of f_i' should be limited to a user-defined value. In this experiment, we tested with different cardinalities; 10%, 20%, 30%, 40%, and 50% of important features

out-of the total number of features of the dataset, which is also bounded by the threshold (*th*) of standard deviation (total number of identified important features for neuron *i*). To evaluate our hypothesis, we changed the feature values of randomly selected features and unimportant features (considering highest standard deviation to lowest). I.e., given the cardinality *p%*, we changed the values of p% number of most important features (features with lowest std values), p% number of randomly picked features, and p% number of most minor important features. For each data record in the test set, we checked whether it changes its cluster label when we change the feature value under the three scenarios described above. For each scenario, we checked the two cases; 1) What is the percentage of test data records where the cluster label can be swapped by at least one other cluster label, 2) What is the percentage of test data records where all other clusters can swap the cluster label. The reason for this is, for some data points, feature value perturbation using a close-by cluster features may be not strong enough to push it out of the original cluster. Therefore, we checked whether the cluster label of a given data point can be change by using the feature values of at-least one other cluster. For example, assume we have 4 clusters and a data point *j*, which belong to cluster 2. We replace its feature values with average feature values of clusters 1, 2, and 4 and check whether we can change its cluster label from 2 to some other cluster label. It has to be noted that lower cardinality *n%* and higher swapped percentages are expected. The result of swap percentage calculation for all the data sets is present in Figure 4. It can be seen that the best results for swapped percentages (tallest bar) were shown by important features (blue), and the second-best was shown by random features (brown bar) for all the data sets except for the second scenario (What is the percentage of test data records where all other clusters can swap the cluster label) of KDD dataset (second column, last raw). The reason for this can be the highly imbalanced classes of KDD data-set and higher differences between train data and test data, resulting poor performance. However, KDD also performce as expected for first scenario (What is the percentage of test data records where the cluster label can be swapped by at least one other cluster label). This empirical results confirms our hypothesis that the identified important features using the proposed approach for SOM decided the cluster labels of data records.

Another experiment was performed to check the percentage of selected K number of features included in the most important feature list of a BMU (Algorithm IV). For each data record in the test set, we calculated the feature-wise l1 distance between the data record and its BMU. Then the features were arranged based on the ascending order of l1 distances. Our hypothesis was that the closest features are the most important features of that data point, and they will be included in the identified important feature lists of its BMU. Once features distances are arranged in ascending order, K features are selected on three different strategies; 1) Closest, 2) Random, and 3) Furthest. We then calculated the percentage of

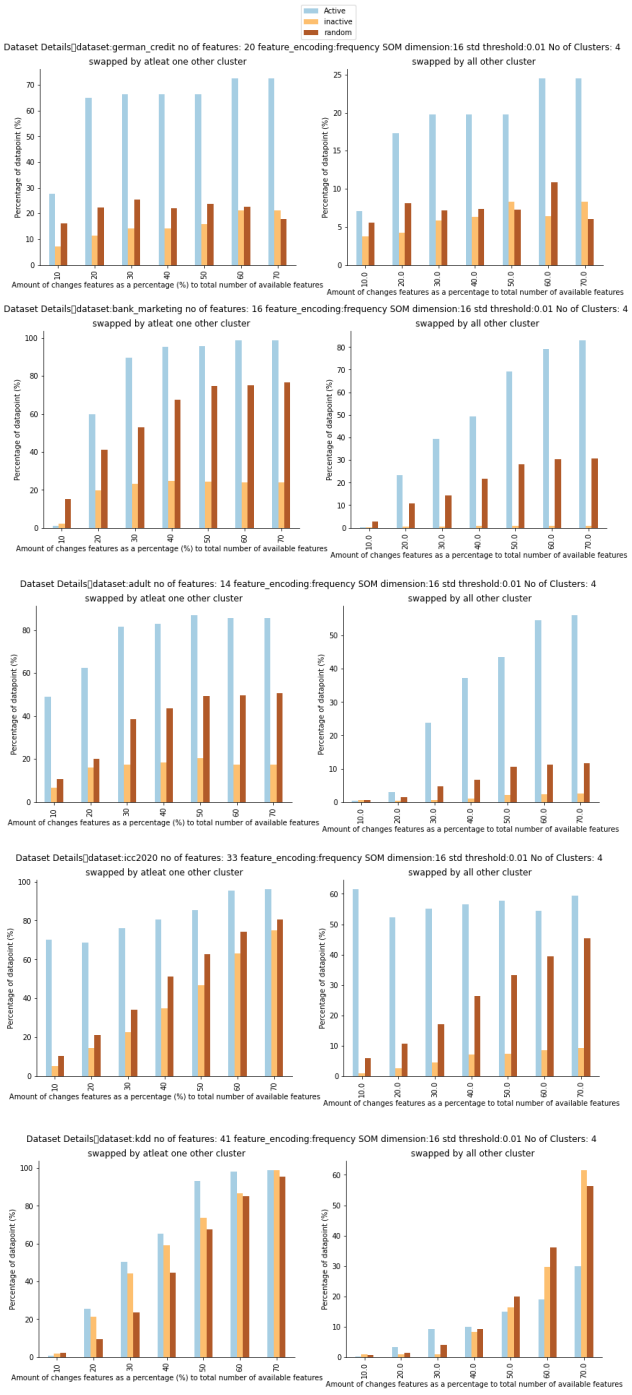

FIGURE 4. Fidelity test, Experiment I: Changed the values of p% number of most important (active) features, p% number of randomly picked features, and p% number of least important (inactive) features and calculated the percentage of data points where the cluster label changes after changing %p feature out of all the feature. we checked the two cases; 1) What is the percentage of test data records where the cluster label can be swapped by at least one other cluster label (left), 2) What is the percentage of test data records where all other clusters can swap the cluster label (right).

K features are included in the important feature list of the BMU. The results are presented in Figure 5 where the X-axis represents the K number of features, and Y-axis represents the percentage of K features that were included in the important

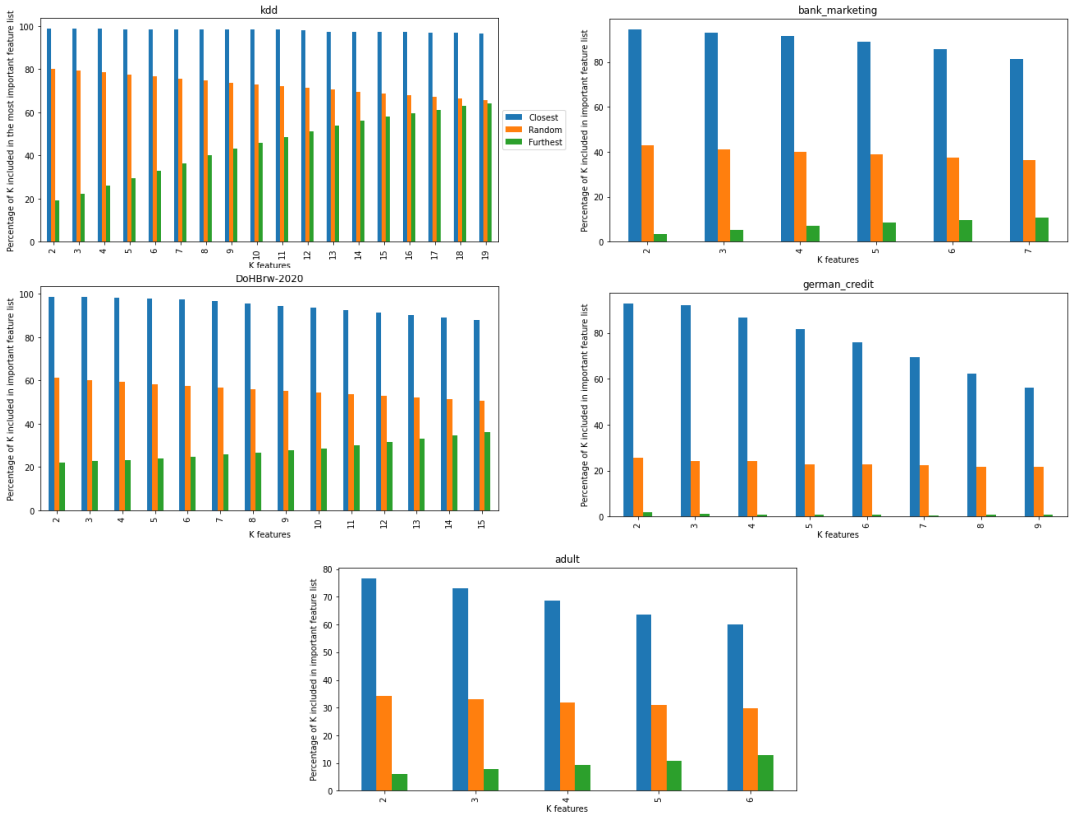

FIGURE 5. The percentage of closest K number of features included in the most important feature list of the BMU.

feature list. Blue color represents the closes feature, yellow represents the random features, and the green represents the furthest features. It can be seen that the blue bar shows the highest percentage for all K features, whereas yellow shows the second higher percentage. It infers that the closes features are included in the identified important feature lists of each BMU.

As described above, we identified the most important ordered set of features for each neuron in the SOM network and evaluated the model fidelity using two experimentations. Then we used the identified ordered list of important features to generate explanations.

B. LOCAL INTERPRETABILITY

For a given data point, a local explanation is generated based on the important features of its BMU, which were identified using Algorithm III. It has to be noticed that two different data points with the same BMU can have different orders of important features based on the l1 feature distance to the BMU. We provide a set of most important features and their feature values which are ordered based on the l1 feature distance calculated between the data point and its BMU. All the feature values are presented in several levels (very low, low, medium, high, very high). Low l1 distance indicates more important features specific to that data point. It has to be noticed that two data points can have the same set of important features, but the order of importance can be different.

Figure 6 shows the local explanation for a single data point of bank loan data set, generated using the proposed approach. The features are ordered based on l1 distance in ascending order (bottom to top) to its BMU. Thus, 'Education' is the furthest feature indicating the lowest importance, whereas the 'loan' is the closest feature indicating the highest feature importance. The most important features of the BMU are colored in green, whereas the rest is colored in red. It can be seen that for the given data point, the closes features are included in the set of the most important features of its BMU. In this manner, we can generate a local explanation of the important features that contributed to deciding the outcome of the SOM algorithm.

C. GLOBAL INTERPRETABILITY

Using the experiments above, we identified the model behavior of SOM, in terms or important features for each neuron in SOM map. Once we identify important features, then we can use them to explore and discuss how each feature behaves within a cluster. In this experiment, we used the neuron-wise important features and their value ranges for global interpretability to explain the clusters.

For each feature, we checked whether it is important for one cluster or multiple clusters. We observed that some

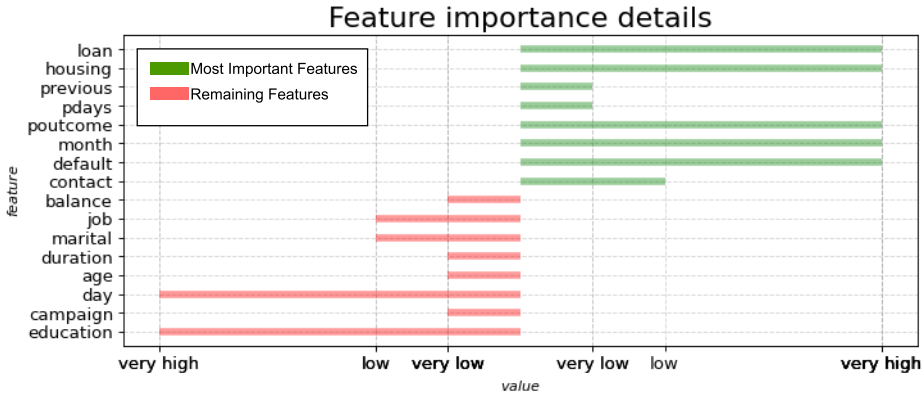

FIGURE 6. Local Interpretability; Explanation for a single data record, features are ordered from ascending order based on feature wise distance to BMU.

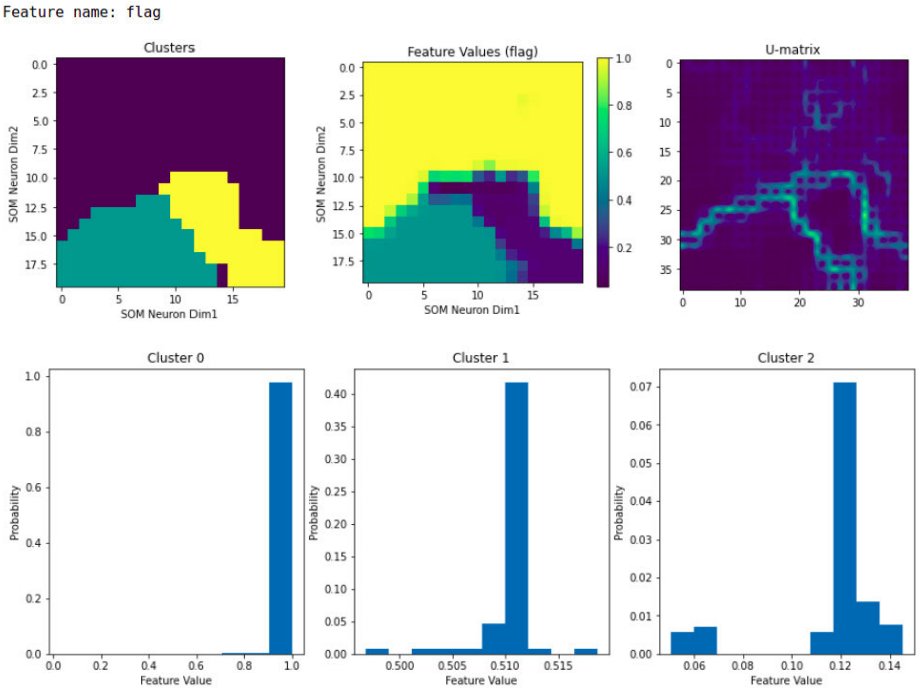

FIGURE 7. Global Interpretability; Feature behavior for 'flag' feature of KDD data set across clusters (SOM neurons were clustered into three categories, U-matrix visualize the distances between clusters and how well clusters are separated, the 'flag' feature value is different across clusters).

features are not important for any cluster, whereas some are important for one or more clusters. The feature value ranges of important features were visualized against the cluster assignments. Further, u-maps were used to check the separation between clusters. Figure 7 presents an example of the 'flag' feature of the KDD dataset. First raw, the first image shows the cluster separation of SOM neurons. The second image of the first row represents the feature value of the 'flag' feature across 3 clusters (component plans). It can be seen that the 'flag' feature value is different across 3 clusters. The third image of the first raw shows that the three clusters are well separated as there is a light color area that represents the distance between neurons. Lighter the area, better the separation between clusters. The second row of Figure 7 shows fine-grain visualization of feature value scale across clusters. It has to be noticed that a given cluster contains a set of neurons, and the feature value for a given feature can be different from one neuron to another, even within the same cluster. This information is essential for a domain expert to check whether how a given feature behave within a cluster. For the 'flag' feature, it shows a higher feature value (0.7-1.0) for cluster 0; for cluster 1, it shows an intermediate feature value range (0.45-0.52); for cluster 3, it shows a very low feature value range (0.15). Further, it shows the probability of having a specific feature value within a cluster as well. For example, for cluster 0, 90

D. USABILITY WITHIN CYBER PHYSICAL SYSTEMS

In previous sections, we described the need for XUnML in general terms for three areas of UnML for CPSs; clustering, feature learning, and model pre-training. In this section, we discuss how to use explainable SOMs for specific requirements of CPSs.

1) SAFETY AND SECURITY

One of the main challenges of CPSs is maintaining the safety and security of CPSs. Many modern critical infrastructures have CPSs at their core. Therefore, these systems are highly vulnerable to various attack vectors. Consequently, maintaining the safety and security of CPSs is a primary focus. One approach is to develop data-driven ML-based Anomaly Detection Systems (ADSs) to ensure the security of CPSs. For developing ADSs, data-driven ML algorithms require collecting data that represents the normal behavior of CPSs. SOMs can be trained with normal data records for this task and identify possible natural clusters (different normal behaviors) and interpretations for each cluster. The domain expert can analyze SOM based explanation to decide whether the collected data represent all status of the normal behavior, what are the dominant natural status in the system, what features are dominant in each cluster, and the amount of data record distribution among identified normal status are good enough to train ML algorithms. This initial information allows domain experts to take necessary actions such as collecting more data, avoiding possible data biases, initial ideas on important features, and reduce the data dimension. For example, if the train data set is too large for downstream tasks such as data labeling, then neuron weights can be used as a data set representing the input data distributions.

2) PROCESS OPTIMIZATION

CPSs such as Intelligent Transportation Systems focus on improving the fuel efficiency of vehicles. It has been found that driver behavior is one of the highly influential factors on fuel efficiency. One approach to achieve fuel efficiency is developing data-driven Intelligent Driver Systems, which develops for changing driver behaviors to follow fuel-efficient velocity profiles. However, different drivers have different driving behaviors. Therefore, it is necessary to identify different driver patterns and develop optimal velocities for different driver categories rather than develop systems assuming that all drivers have similar capabilities. SOM-based clustering can identify different driver clusters, thus helping ML experts develop different velocity profiles representing different clusters. Further, explanations on individual clusters can be used to generate recommendations for drivers, which improved user trust in ML systems. Domain experts can use global feature behaviors to evaluate different driver categories and their unique behaviors. Further, the local explanations can be used to provide personalized recommendations to a particular driver.

3) SALES STRATEGIES

Another main use of clustering is discovering customer groups in companies. Many large companies today need sales strategies targeting different customer groups. SOM can be used to identify cluster groups; then SOM based explanations can be used to identify why a set of customers belongs to a specific cluster. Domain experts can use SOM-based global explanations and evaluate whether the cluster explanations are meaningful. These identified meaningful explanations can be used towards building marketing strategies targeting meaningful customer clusters. Further, SOM local explanation allows to analyze individual customers and provide customer-specific customization.

4) GENERALIZABILITY

Lack of generalizability is one main problem in CPS as data-driven ML models mode for one system may not be useful to other CPSs even when both have many similarities. One approach is to retrain and re-purpose models used within one CPSs to another by using pre-trained ML models. SOMs can be used as pre-trained models as SOM can arrange their neuron weights to represent the input data distribution. Therefore, a trained SOM for one task can be used, retrained efficiently for another similar task. However, to use pre-trained models effectively, it is essential to evaluate whether the trained SOMs represent meaningful clusters through explanations. Therefore, domain experts can evaluate these clusters using global explanations and decide whether the clusters are meaningful and where to use these trained SOMs efficiently. It allows domain experts to make meaningful reusing of trained models.

5) REAL-TIME OPERATIONS

In CPSs, a large amount of high-dimensional data is generated at a rapid speed. For example, in power grids, large volumes of readings come from physical components of the system (voltages, currents) and cyber components (network flow features such as packet rate, payload size, flag). When it comes to high-dimensional data, training ML algorithms can be very expensive, generating outcomes from real-time high-dimensional data can be computationally expensive, and storing data can be difficult due to large volumes. Further, it can be impossible to perform real-time processing of these vast volumes of data generated at a rapid speed. In such situations, feature learning is beneficial as it reduces the dimension of input feature space, reducing the number of computations in downstream ML tasks. Further, it reduces the storage requirements for storing data. SOM-based global explanations can be used to identify feature correlations in these situations as it shows how different features values change across clusters. This allows domain experts to identify and remove highly correlated features, resulting in low dimensional feature spaces. Consequently, reducing the store requirements and computational cost of downstream ML tasks. Especially when it comes to real-time operations, faster interpretation (simple and easy to perceive) of data

TABLE 3. Comparison between XUnML methodologies.

Explainable SOM	**EXPLAIN-IT**	**Interpretable Trees**	**PCA**	**Variational AE**
Interpretation Approach:				
Model-specific	Model agnostic	Model-specific	Model-specific	Model-specific
Used for:				
Clustering	Clustering	Clustering	Dimension Reduction	Dimension Reduction
Data Distribution Visualization Capability:				
Inbuild capability to visualize input data distribution: High dimensional data space to a low dimensional grid (2 dimensions)	NA	NA	Inbuild capability to visualize input data distribution: High dimensional data to low dimensional data (2 dimensions)	NA
Visual data mining capabilities:				
Many visual data mining capabilities: histograms,component plane	Limited	Limited	Limited	Limited
Model quality evaluation:				
Can apply unsupervised quality matrix such as adjusted mutual information, adjusted random score, completeness, Fowlkes-Mallows, homogeneity, silhouette, and V-measure	Can apply unsupervised quality a matrix such as adjusted mutual information, adjusted random score, completeness, Fowlkes-Mallows, homogeneity, silhouette, and V-measure	Splitting criteria such as information gain and entropy	Reconstruction error measurements (Variability)	Reconstruction error measurements (MSE)
Local vs Global Explanation:				
Both local and global explanations	Both local and global explanations	Both local and global explanations	NA	NA
Integrating clustering capability:				
Can work with any clustering algorithm	Can work with any clustering algorithm	NA	NA	NA
Time Complexity with respect to the number of training samples (n):				
O(n)	Depend on the model used for clustering	O($nlog_2n$)	O(n^3)	O(n)
Limitations:				
Depending on which type of distance metric used, the result may very	Model biases can occur, Explainability is dependant on other models makes this approach complicated	Split evaluation measure are required	Principle components of the model are not interpretable, Data should be standardize	High model complexity, Need expert knowledge base for training

samples is essential to avoid catastrophic failures in CPSs. The SOM-based global explanation can reduce the feature space, and local explanations can be used to produce faster interpretations for real-time data samples.

E. DISCUSSION

It has to be noticed that the desired outcomes and evaluation methods for explainable machine learning methods are different based on many factors, including domain areas, applications, user groups, expected performance criterion, and medium of explanation. Therefore, it is not easy to establish a set of generalized requirements or outcomes of explainable machine learning systems. Further, evaluating explainable algorithms and their effectiveness is complicated as there is no clear way of measuring it [34]. Especially in the unsupervised domain, there was no clear way of measuring and comparing the quality of the explanation methods. One classic approach for that is doing a human study with existing unsupervised explainable ML approaches for a specific problem domain, which is out of this paper's scope. However, we explore the model-specific features, limitations, and usability of the proposed approach with other existing explainable unsupervised ML approaches, presented in Table 3.

When looking at Table 3, it can be noted that different unsupervised XAI methods have different usability, features, and

limitations. The current literature of XUnML is mainly concentrated on clustering and dimensionality reduction. When looking at Explainable SOM, the main advantage of it comes from its many Visual Data Mining capabilities, which are described in Section III. All the other methods discussed above have very limited visual data mining capabilities, limiting their usage in tasks that require VDM capabilities. A human study will be performed in future work to analyze the above methods to explore the advantages and limitations of the above methods.

Bias is a frequently addressed topic in the machine learning community. Bias in machine learning can exist in many shapes and forms, such as data biases (ex: measurement biases, representation biases, data processing biases), algorithmic biases (ex: algorithmic design choices related biases), and user biases (ex: user interaction biases and evaluation biases). Interpretable unsupervised machine learning can be used to address some forms of bias in ML models. Unsupervised models that perform clustering and dimensionality reduction can be used to eliminate data biases, revealing what such data actually represents (data clusters), how clusters are different, and how clusters are correlated/overlapped. Thus, using the proposed SOM-based global and local explanations, users can understand which features the model depends on, feature behaviors on different clusters, and because of which features the model decides the data point belongs to a specific cluster. This information allows machine learning experts and domain experts to understand what the training data represents, helping them preprocess data appropriately to improve the data quality, hence reducing data biases.

It is also necessary to understand the difference between SOM neural networks and typical Feed-Forward Neural Networks (FFNNs) in terms of the learning approach, visualization capabilities, and global/local interpretability. SOMs use the winner-take-all algorithm for training while preserving the input space's topological properties. Thus the trained set of neurons in SOM represents the topological properties of input data distribution. Whereas FFNNs use error-correction learning (such as backpropagation with gradient descent) for training which does not have the capability of representing the topological properties of input data using trained neurons. FFNNs are trained to perform classification and regression, whereas SOMs are trained to perform clustering tasks. As discussed in the previous section, SOM has many in-build visual data exploration approaches for visualizing feature behaviors, whereas FFNNs have very limited inbuilt VDM capabilities.

The difference between FFNNs and SOMs in terms of global and local interpretability are: 1) The presented interpretation technique for SOMs generates local/global interpretations for clustering tasks, whereas the most popular interpretation techniques for FFNNs generate local/global interpretations for classification and regression tasks; 2) Local interpretability: Most popular local interpretation techniques used for FFNNs produce relative feature importance scores, whereas the presented technique for SOMs does not generate relative feature importance scores. It only generates a sorted features list indicating the most important features to the least important feature; 3) Global interpretability: Most popular global interpretation techniques used for FFNNs produce a set of IF-THEN rules for explaining the model behavior for different classes, whereas the presented technique for SOMs generates a set of component planes and feature value distributions for explaining how different features behave across different data clusters; 4) SOMs carry an inherent topological understanding of data and clusters. This inherent topology directly reflects notions of local and global belonging of data to clusters and addresses local vs global interpretability, unlike FFNNs that do not have the topological understanding of the data.

VI. CONCLUSION

The motivation for this paper is two-pronged: 1) Unsupervised Machine Learning (UnML) has gained significant attention due to large amounts of unlabelled data generation at rapid speed, and 2) majority of the work on explainable/interpretable AI is focused on supervised machine learning, but real-world settings brings the challenge of dealing with unlabelled data, making supervised machine learning alone is not sufficient for data-driven decision making. Therefore, in this paper, we investigated the need for Explainable Unsupervised Machine Learning (XUnML). We explored and revealed that the current literature has limited work on XUnML. We refined the terminologies in explainable machine learning in the unsupervised domain, exploring current terms in XAI towards achieving XUnML. We specifically focused on the Cyber-Physical Domain (CPS) domain as these systems generate a large amount of unlabelled data at rapid speeds. Therefore, unsupervised ML is a viable option to extract knowledge from the data coming from CPSs. We observed from the recent literature that three unsupervised approaches are being widely used within CPSs: Clustering, Unsupervised Feature Learning, and Model pre-training. Under each approach, we discussed the need for XunML and explored the advantages.

We proposed a novel model-specific explainable method for the Self-Organizing Map (SOM) algorithm, generating local and global explanations. Through feature value perturbation, we evaluated the model fidelity and showed that the proposed approach identifies the most important feature used by the decision-making process of SOMs. We showed that the changing of features values of important features affects the model outcomes of SOMs. We presented the proposed approach as a strong candidate as a XUnML method by comparing it with current XUnML methods in terms of model-specific features, limitations, and usability. Further, we explored how to apply the proposed method for specific requirements of CPSs such as safety and security, process optimization, sales strategies, the generalizability of models, and real-time operations. We discussed the usability and advantages of generated local and global

explanations for each identified requirement, showing that explainable SOMs are highly beneficial for distinct needs in CPSs. In future work, the proposed approach will be further evaluated through a human study.

ACKNOWLEDGMENT

This work was supported in part by the Commonwealth Cyber Initiative, an investment in the advancement of cyber R&D, innovation and workforce development. For more info about CCI, visit cyberinitiative.org.

REFERENCES

[1] J. Shi, J. Wan, H. Yan, and H. Suo, "A survey of cyber-physical systems," in *Proc. Int. Conf. Wireless Commun. Signal Process. (WCSP)*, Nov. 2011, pp. 1–6.

[2] *Cyber-Physical Systems (CPS)*. Accessed: Mar. 27, 2021. [Online]. Available: https://www.nsf.gov/pubs/2021/nsf21551/nsf21551.htm

[3] Anonymous. (Oct. 2020). *Cyber-Physical Systems*. [Online]. Available: https://ec.europa.eu/digital-single-market/en/cyber-physical-systems

[4] Y. Zhang, I.-L. Yen, F. B. Bastani, A. T. Tai, and S. Chau, "Optimal adaptive system health monitoring and diagnosis for resource constrained cyber-physical systems," in *Proc. 20th Int. Symp. Softw. Rel. Eng.*, Nov. 2009, pp. 51–60.

[5] W. U. Guangyu, J. Sun, and J. Chen, "A survey on the security of cyber-physical systems," *Control Theory Technol.*, vol. 14, no. 1, pp. 2–10, 2016.

[6] R. Rajkumar, I. Lee, L. Sha, and J. Stankovic, "Cyber-physical systems: The next computing revolution," in *Proc. Design Automat. Conf.*, Jun. 2010, pp. 731–736.

[7] Kristy.thompson@nist.gov. (Nov. 2019). *Cyber-Physical Systems*. [Online]. Available: https://www.nist.gov/el/cyber-physical-systems

[8] A. Adadi and M. Berrada, "Peeking inside the black-box: A survey on explainable artificial intelligence (XAI)," *IEEE Access*, vol. 6, pp. 52138–52160, 2018.

[9] *Explainable Artificial Intelligence (XAI)*. Accessed: Mar. 29, 2021. [Online]. Available: https://www.darpa.mil/program/explainable-artificial-intelligence

[10] (Feb. 2020). *White Paper on Artificial Intelligence: A European Approach to Excellence and Trust*. [Online]. Available: https://ec.europa.eu/info/publications/white-paper-artificial-intelligence-european-approach-excellence-and-trust_en

[11] Robin.materese@nist.gov. (Mar. 2021). *Artificial Intelligence*. [Online]. Available: https://www.nist.gov/artificial-intelligence

[12] Thelma.allen@nist.gov. (Jan. 2021). *AI Foundational Research–Explainability*. [Online]. Available: https://www.nist.gov/artificial-intelligence/ai-foundational-research-explainability

[13] M. Längkvist, L. Karlsson, and A. Loutfi, "A review of unsupervised feature learning and deep learning for time-series modeling," *Pattern Recognit. Lett.*, vol. 42, pp. 11–24, Jun. 2014.

[14] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.

[15] A. Mujumdar and V. Vaidehi, "Diabetes prediction using machine learning algorithms," *Proc. Comput. Sci.*, vol. 165, pp. 292–299, Jan. 2019.

[16] A. Singh, N. Thakur, and A. Sharma, "A review of supervised machine learning algorithms," in *Proc. 3rd Int. Conf. Comput. Sustain. Global Develop. (INDIACom)*, Mar. 2016, pp. 1310–1315.

[17] T. Jiang, J. L. Gradus, and A. J. Rosellini, "Supervised machine learning: A brief primer," *Behav. Therapy*, vol. 51, no. 5, pp. 675–687, Sep. 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0005789420300678

[18] A. A. Mohamed, "An effective dimension reduction algorithm for clustering Arabic text," *Egyptian Informat. J.*, vol. 21, no. 1, pp. 1–5, Mar. 2020.

[19] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Red Hook, NY, USA: Curran Associates, Inc., 2014, pp. 2672–2680.

[20] I. J. Goodfellow, "NIPS 2016 tutorial: Generative adversarial networks," 2017, *arXiv:1701.00160*. [Online]. Available: https://arxiv.org/abs/1701.00160

[21] P. Luc, N. Neverova, C. Couprie, J. Verbeek, and Y. Lecun, "Predicting deeper into the future of semantic segmentation," *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 648–657.

[22] G.-Y. Chen, M. Gan, and G.-L. Chen, "Generalized exponential autoregressive models for nonlinear time series: Stationarity, estimation and applications," *Inf. Sci.*, vol. 438, pp. 46–57, Apr. 2018.

[23] E. Aljalbout, V. Golkov, Y. Siddiqui, M. Strobel, and D. Cremers, "Clustering with deep learning: Taxonomy and new methods," 2018, *arXiv:1801.07648*. [Online]. Available: http://arxiv.org/abs/1801.07648

[24] M. Khanam, T. Mahboob, W. Imtiaz, H. A. Ghafoor, and R. Sehar, "A survey on unsupervised machine learning algorithms for automation, classification and maintenance," *Int. J. Comput. Appl.*, vol. 119, pp. 34–39, Jan. 2015.

[25] S. Schneider, A. Baevski, R. Collobert, and M. Auli, "wav2vec: Unsupervised pre-training for speech recognition," 2019, *arXiv:1904.05862*. [Online]. Available: https://arxiv.org/abs/1904.05862

[26] F. Wang, H. Liu, D. Guo, and F. Sun, "Unsupervised representation learning by invariancepropagation," 2020, *arXiv:2010.11694*. [Online]. Available: https://arxiv.org/abs/2010.11694

[27] C. S. Wickramasinghe, D. L. Marino, and M. Manic, "ResNet autoencoders for unsupervised feature learning from high-dimensional data: Deep models resistant to performance degradation," *IEEE Access*, vol. 9, pp. 40511–40520, 2021.

[28] P. Bojanowski and A. Joulin, "Unsupervised learning by predicting noise," in *Proc. 34th Int. Conf. Mach. Learn.* (Proceedings of Machine Learning Research), vol. 70. D. Precup and Y. W. Teh, Eds. Sydney, NSW, Australia: JMLR.org, Aug. 2017, pp. 517–526. [Online]. Available: http://proceedings.mlr.press/v70/bojanowski17a.html

[29] K. J. Cios, R. W. Swiniarski, W. Pedrycz, and L. A. Kurgan, "Unsupervised learning: Association rules," in *Data Mining: A Knowledge Discovery Approach*. Boston, MA, USA: Springer, 2007, pp. 289–306, doi: 10.1007/978-0-387-36795-8_10.

[30] Z.-Y. Han, J. Wang, H. Fan, L. Wang, and P. Zhang, "Unsupervised generative modeling using matrix product states," *Phys. Rev. X*, vol. 8, no. 3, Jul. 2018, Art. no. 031012. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevX.8.031012

[31] D. Gunning and D. Aha, "DARPA's explainable artificial intelligence (XAI) program," *AI Mag.*, vol. 40, no. 2, pp. 44–58, Jun. 2019. [Online]. Available: https://ojs.aaai.org/index.php/aimagazine/article/view/2850

[32] R. Roscher, B. Bohn, M. F. Duarte, and J. Garcke, "Explainable machine learning for scientific insights and discoveries," *IEEE Access*, vol. 8, pp. 42200–42216, 2020.

[33] A. Barredo Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. Garcia, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, and F. Herrera, "Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI," *Inf. Fusion*, vol. 58, pp. 82–115, Jun. 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1566253519308103

[34] C. Molnar, *Interpretable Machine Learning, A Guide for Making Black Box Models Explainable*. Lulu.com, 2020. [Online]. Available: https://christophm.github.io/interpretable-ml-book/

[35] A. Morichetta, P. Casas, and M. Mellia, "Explain-it," in *Proc. 3rd ACM CoNEXT Workshop Big DAta, Mach. Learn. Artif. Intell. Data Commun. Netw. (Big-DAMA)*, 2019. [Online]. Available: http://dx.doi.org/10.1145/3359992.3366639

[36] O. Loyola-Gonzalez, A. E. Gutierrez-Rodriguez, M. A. Medina-Perez, R. Monroy, J. F. Martinez-Trinidad, J. A. Carrasco-Ochoa, and M. Garcia-Borroto, "An explainable artificial intelligence model for clustering numerical databases," *IEEE Access*, vol. 8, pp. 52370–52384, 2020.

[37] E. Horel, K. Giesecke, V. Storchan, and N. Chittar, "Explainable clustering and application to wealth management compliance," 2020, *arXiv:1909.13381*. [Online]. Available: https://arxiv.org/abs/1909.13381

[38] M. Moshkovitz, S. Dasgupta, C. Rashtchian, and N. Frost, "Explainable K-means and K-medians clustering," in *Proc. 37th Int. Conf. Mach. Learn.* (Proceedings of Machine Learning Research), vol. 119, H. D. III and A. Singh, Eds. PMLR, Jul. 2020, pp. 7055–7065. [Online]. Available: http://proceedings.mlr.press/v119/moshkovitz20a.html

[39] K. Sokol and P. Flach, "Explainability fact sheets: A framework for systematic assessment of explainable approaches," in *Proc. Conf. Fairness, Accountability, Transparency*, Jan. 2020, pp. 56–67, doi: 10.1145/3351095.3372870.

[40] A. Papenmeier, G. Englebienne, and C. Seifert, "How model accuracy and explanation fidelity influence user trust," 2019, *arXiv:1907.12652*. [Online]. Available: http://arxiv.org/abs/1907.12652

[41] C.-K. Yeh, C.-Y. Hsieh, A. Suggala, D. I. Inouye, and P. K. Ravikumar, "On the (in) fidelity and sensitivity of explanations," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, H. Wallach, H. Larochelle, A. Beygelzimer, F. d' Alché-Buc, E. Fox, and R. Garnett, Eds. Red Hook, NY, USA: Curran Associates, 2019, pp. 1–12.

[42] S. Dasgupta, N. Frost, M. Moshkovitz, and C. Rashtchian, "Explainable k-means and k-medians clustering," in *Proc. ICML*, 2020.

[43] N. Frost, M. Moshkovitz, and C. Rashtchian, "ExKMC: Expanding explainable K-means clustering," 2020, *arXiv:2006.02399*. [Online]. Available: https://arxiv.org/abs/2006.02399

[44] Q. P. Nguyen, K. W. Lim, D. M. Divakaran, K. H. Low, and M. C. Chan, "GEE: A gradient-based explainable variational autoencoder for network anomaly detection," in *Proc. IEEE Conf. Commun. Netw. Secur. (CNS)*, Jun. 2019, pp. 91–99.

[45] M. Neumeier, A. Tollkühn, T. Berberich, and M. Botsch, "Variational autoencoder-based vehicle trajectory prediction with an interpretable latent space," 2021, *arXiv:2103.13726*. [Online]. Available: https://arxiv.org/abs/2103.13726

[46] M. Curi, G. A. Converse, J. Hajewski, and S. Oliveira, "Interpretable variational autoencoders for cognitive models," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2019, pp. 1–8.

[47] C. Lv, X. Hu, A. Sangiovanni-Vincentelli, Y. Li, C. M. Martinez, and D. Cao, "Driving-style-based codesign optimization of an automated electric vehicle: A cyber-physical system approach," *IEEE Trans. Ind. Electron.*, vol. 66, no. 4, pp. 2965–2975, Apr. 2019.

[48] C. S. Wickramasinghe, K. Amarasinghe, D. Marino, Z. A. Spielman, I. E. Pray, D. Gertman, and M. Manic, "Intelligent driver system for improving fuel efficiency in vehicle fleets," in *Proc. 12th Int. Conf. Human Syst. Interact. (HSI)*, Jun. 2019, pp. 34–40.

[49] Y. Zhao, S. K. Tarus, L. T. Yang, J. Sun, Y. Ge, and J. Wang, "Privacy-preserving clustering for big data in cyber-physical-social systems: Survey and perspectives," *Inf. Sci.*, vol. 515, pp. 132–155, Apr. 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0020025519309764

[50] W. Wei, X. Xu, W. Marcin, F. Xunli, D. Robertas, and L. Ye, "Multi-sink distributed power control algorithm for cyber-physical-systems in coal mine tunnels," *Comput. Netw.*, vol. 161, pp. 210–219, Oct. 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128618310673

[51] S. Yin, S. X. Ding, X. Xie, and H. Luo, "A review on basic data-driven approaches for industrial process monitoring," *IEEE Trans. Ind. Electron.*, vol. 61, no. 11, pp. 6418–6428, Nov. 2014.

[52] C. S. Wickramasinghe, D. L. Marino, K. Amarasinghe, and M. Manic, "Generalization of deep learning for cyber-physical system security: A survey," in *Proc. 44th Annu. Conf. IEEE Ind. Electron. Soc. (IECON)*, Oct. 2018, pp. 745–751.

[53] Z. Wu, H. Luo, Y. Yang, X. Zhu, and X. Qiu, "An unsupervised degradation estimation framework for diagnostics and prognostics in cyber-physical system," in *Proc. IEEE 4th World Forum Internet Things (WF-IoT)*, Feb. 2018, pp. 784–789.

[54] Y. Ren, K. Hu, X. Dai, L. Pan, S. C. H. Hoi, and Z. Xu, "Semi-supervised deep embedded clustering," *Neurocomputing*, vol. 325, pp. 121–130, Jan. 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231218312049

[55] P. Schneider and K. Böttinger, "High-performance unsupervised anomaly detection for cyber-physical system networks," in *Proc. Workshop Cyber-Phys. Syst. Secur. PrivaCy* New York, NY, USA: ACM, 2018, pp. 1–12, doi: 10.1145/3264888.3264890.

[56] Y. Chen, Y. Zhang, S. Maharjan, M. Alam, and T. Wu, "Deep learning for secure mobile edge computing in cyber-physical transportation systems," *IEEE Netw.*, vol. 33, no. 4, pp. 36–41, Jul. 2019.

[57] F. Bu, "A high-order clustering algorithm based on dropout deep learning for heterogeneous data in Cyber-Physical-Social systems," *IEEE Access*, vol. 6, pp. 11687–11693, 2018.

[58] C. Liu and P. Jiang, "A cyber-physical system architecture in shop floor for intelligent manufacturing," *Proc. CIRP*, vol. 56, pp. 372–377, Jan. 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2212827116310514

[59] W. Sun, S. Shaoa, R. Zhaob, R. Yana, X. Zhangc, and X. Chen, "A sparse auto-encoder-based deep neural network approach for induction motor faults classification," *Measurement*, vol. 89, pp. 171–178, Jul. 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0263224116300641

[60] D. Zhang, Y. Sun, B. Eriksson, and L. Balzano, "Deep unsupervised clustering using mixture of autoencoders," Dec. 2017, *arXiv:1712.07788*. [Online]. Available: https://arxiv.org/abs/1712.07788

[61] S. Schneider, A. Baevski, R. Collobert, and M. Auli, "wav2vec: Unsupervised pre-training for speech recognition," in *Proc. Interspeech*, Sep. 2019, pp. 3465–3469.

[62] D. Erhan, A. Courville, Y. Bengio, and P. Vincent, "Why does unsupervised pre-training help deep learning?" in *Proc. 13th Int. Conf. Artif. Intell. Statist.* (Proceedings of Machine Learning Research), vol. 9. Y. W. Teh and M. Titterington, Eds. Sardinia, Italy: JMLR, May 2010, pp. 201–208. [Online]. Available: http://proceedings.mlr.press/v9/erhan10a.html

[63] S. Li, F. Bi, W. Chen, X. Miao, J. Liu, and C. Tang, "An improved information security risk assessments method for cyber-physical-social computing and networking," *IEEE Access*, vol. 6, pp. 10311–10319, 2018.

[64] Y. Chen, F. Chen, T. Wu, W. Hu, and X. Xu, "A deep learning model for secure cyber-physical transportation systems," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Apr. 2018, pp. 1–2.

[65] D. Hendrycks, K. Lee, and M. Mazeika, "Using pre-training can improve model robustness and uncertainty," in *Proc. 36th Int. Conf. Mach. Learn.* (Proceedings of Machine Learning Research), vol. 97, K. Chaudhuri and R. Salakhutdinov, Eds. Curran Associates, Jun. 2019, pp. 2712–2721. [Online]. Available: http://proceedings.mlr.press/v97/hendrycks19a.html

[66] T. Kohonen, E. Oja, O. Simula, A. Visa, and J. Kangas, "Engineering applications of the self-organizing map," *Proc. IEEE*, vol. 84, no. 10, pp. 1358–1384, Oct. 1996.

[67] C. Budayan, I. Dikmen, and M. T. Birgonul, "Comparing the performance of traditional cluster analysis, self-organizing maps and fuzzy C-means method for strategic grouping," *Expert Syst. Appl.*, vol. 36, no. 9, pp. 11772–11781, Nov. 2009.

[68] A. Rauber, D. Merkl, and M. Dittenbach, "The growing hierarchical self-organizing map: Exploratory analysis of high-dimensional data," *IEEE Trans. Neural Netw.*, vol. 13, no. 6, pp. 1331–1341, Nov. 2002.

[69] T. Kohonen, "The self-organizing map," *Proc. IEEE*, vol. 78, no. 9, pp. 1464–1480, Sep. 1990.

[70] C. S. Wickramasinghe, K. Amarasinghe, and M. Manic, "Deep self-organizing maps for unsupervised image classification," *IEEE Trans. Ind. Informat.*, vol. 15, no. 11, pp. 5837–5845, Nov. 2019.

[71] J. Vesanto and E. Alhoniemi, "Clustering of the self-organizing map," *IEEE Trans. Neural Netw.*, vol. 11, no. 3, pp. 586–600, May 2000.

[72] C. Ferles, Y. Papanikolaou, and K. J. Naidoo, "Denoising autoencoder self-organizing map (DASOM)," *Neural Netw.*, vol. 105, pp. 112–131, Sep. 2018.

[73] T. Graepel, M. Burger, and K. Obermayer, "Self-organizing maps: Generalizations and new optimization techniques," *Neurocomputing*, vol. 21, nos. 1–3, pp. 173–190, Nov. 1998.

[74] M. Cococcioni, B. Lazzerini, and S. L. Volpi, "Robust diagnosis of rolling element bearings based on classification techniques," *IEEE Trans. Ind. Informat.*, vol. 9, no. 4, pp. 2256–2263, Nov. 2013.

[75] T. Kohonen, "Self-organization of very large document collections: State of the art," in *Proc. ICANN*, L. Niklasson, M. Bodén, and T. Ziemke, Eds. London, U.K.: Springer, 1998, pp. 65–74.

CHATHURIKA S. WICKRAMASINGHE received the B.Sc. degree in computer science from the University of Peradeniya, Sri Lanka, in 2016. She is currently pursuing the Ph.D. degree in computer science with Virginia Commonwealth University, Richmond. She is working as a Research Assistant at Virginia Commonwealth University. Her research interests include machine learning, unsupervised learning, explainable AI, generalization, and visual data mining.

KASUN AMARASINGHE received the B.Sc. degree in computer science from the University of Peradeniya, Sri Lanka, in 2011, and the Ph.D. degree in computer science from Virginia Commonwealth University, Richmond, USA, in 2019. He is currently a Postdoctoral Research Associate at Carnegie Mellon University (CMU). His research interests include explainable machine learning, algorithmic fairness, and applications of machine learning in the domain of public policy.

DANIEL L. MARINO received the B.Eng. degree in automation engineering from La Salle University, Colombia, in 2015. He is currently pursuing the Ph.D. degree with Virginia Commonwealth University. He is a Research Assistant at Virginia Commonwealth University. His research interests include stochastic modeling, deep learning, and optimal control.

CRAIG RIEGER (Senior Member, IEEE) received the B.S. and M.S. degrees in chemical engineering from Montana State University, Bozeman, MT, USA, in 1983 and 1985, respectively, and the Ph.D. degree in engineering and applied science from Idaho State University, Pocatello, ID, USA, in 2008. He has 20 years of software and hardware design experience for process control system upgrades and new installations. He also has been a Supervisor and a Technical Lead for control systems engineering groups having design, configuration management, and security responsibilities for several INL nuclear facilities and various control system architectures. He is currently the Chief Control Systems Research Engineer and a Directorate Fellow with Idaho National Laboratory (INL), Idaho Falls, ID, USA, pioneering interdisciplinary research in the area of next-generation resilient control systems. In addition, he has organized and chaired 11 Institute of Electrical and Electronics Engineers technically cosponsored symposia and one National Science Foundation Workshop in this new research area. He has authored more than 50 peer-reviewed publications.

MILOS MANIC (Fellow, IEEE) is currently a Professor with the Computer Science Department and the Director of VCU Cybersecurity Center, Virginia Commonwealth University. He completed over 40 research grants in data mining and machine learning applied to cyber security, critical infrastructure protection, energy security, and resilient intelligent control. He has given over 40 invited talks around the world. He has authored over 200 refereed articles in international journals, books, and conferences, and holds several U.S. patents. He is the IES Officer and a Senior AdCom Member. He is an Inductee of U.S. National Academy of Inventors (class of 2019) and a fellow of the Commonwealth Cyber Initiative (specialty in AI and Cybersecurity). He has won 2018 Research and Development 100 Award for Autonomic Intelligent Cyber Sensor (AICS), one of top 100 science and technology worldwide innovations, in 2018. He was a recipient of the IEEE IES 2019 Anthony J. Hornfeck Service Award, the 2012 J. David Irwin Early Career Award, and the 2017 IEM Best Paper Award. He was the Founding Chair of the IEEE IES Technical Committee on Resilience and Security in Industry and the General Chair of the IEEE IECON 2018 and the IEEE HSI 2019. He served as an Associate Editor for IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS. He serves as an Associate Editor for IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS and IEEE OPEN JOURNAL OF THE INDUSTRIAL ELECTRONICS SOCIETY.