2024 IEEE 10th International Conference on Collaboration and Internet Computing (CIC) | 979-8-3503-8670-7/24/$31.00 ©2024 IEEE | DOI: 10.1109/CIC62241.2024.00021

Enhancing Explainability and Trustworthiness of Intrusion Detection Systems Using Competitive Learning

Jesse Ables[‡], Thomas Kirby[*], William Anderson[*],
Sudip Mittal[*], Shahram Rahimi[*], Ioana Banicescu[*], Maria Seale[†], Thomas Arnold[†], Joseph Jabour[†]
[*] Department of Computer Science & Engineering
Mississippi State University, Mississippi, USA,
(email: {jha92, tmk169, wha41}@msstate.edu, {mittal, rahimi, ioana}@cse.msstate.edu)
[†] U.S Army Engineer Research and Development Center
Vicksburg, Mississippi, USA, (email: {maria.a.seale, thomas.l.arnold, joseph.e.jabour}@erdc.dren.mil)
[‡] University of South Alabama
Mobile, Alabama, USA, (email: ables@southalabama.edu)

***Abstract*—Current AI-based Intrusion Detection Systems (IDS) primarily rely on untrustworthy black box methods. Traditionally, many of these black box IDS are built using Error Based Learning (EBL) algorithms such as neural networks. EBL algorithms can offer high accuracy but at the cost of explainability. White box algorithms, on the other hand, are far more explainable and trustworthy than black box EBL techniques. Our proposed solution is a white box Competitive Learning (CL) based eXplainable Intrusion Detection System (X-IDS), offering innate explainability. This architecture is built using DARPA's guidelines for explainable systems. We analyze the statistical and visual explanations generated by the CL models and demonstrate a method for understanding the explanations. Using these explanations, users could potentially make changes to the architecture to improve security. Lastly, a performance analysis using traditional accuracy metrics is performed using the NSL-KDD and CIC-IDS-2017 datasets. While achieving slightly lower accuracies (1%-3% less than EBL models) on NSL-KDD and CIC-IDS-2017 datasets, CL models provide enhanced explainability and trustworthiness.**

I. Introduction

Shifting away from the current trend of black box Intrusion Detection Systems (IDS) can lead to more trustworthy and transparent anomaly detection. Existing methods for AI enabled intrusion detection use Error Based Learning (EBL) algorithms to detect anomalies. EBL refers to models that train through minimizing a *loss* function, generally through the gradient descent algorithm. These models can achieve high detection rates, however they suffer from a few problems. First, these models are not easy to understand and are not innately explainable. Users who use these opaque models do not know how or why a prediction was computed. This can cause a lack of trust and prevent the adoption of AI IDS solutions [1], [2]. Second, many of these methods have high false positive rates which can harm the overall performance of a real-world IDS [3]. Without truly understanding the model, it is difficult to discover why the model is creating incorrect predictions.

eXplainable Intrusion Detection Systems (X-IDS) are a potential solution to the above mentioned problems [4]. The Defence Advanced Research Projects Agency (DARPA) defines an explainable system as an AI that can explain the *reasoning* for its decisions, characterize its *strengths and weaknesses*, and convey a sense of its *future behavior* [5]. Many methods can allow current EBL AI models to achieve these tenets. Solutions such as Local Interpretable Model-agnostic Explanations (LIME) [6], SHapely Additive exPlantions (SHAP) [7], and Layer-wise Relevance Propagation (LRP) [8] have the ability to convert black box models into semi-transparent, explainable models. However, the use of these types of solutions comes with downsides. One major downside to these techniques is their black box nature. Similar to the black box EBL models that they are used to explain, the user does not know how or why these explanation frameworks come to conclusions. If one of the goals of black box XAI is to generate trust in opaque models, how can we view explanations from opaque surrogate models as trustworthy? Black box surrogate explanation can be seen as less trustworthy than certain alternatives.

White box algorithms are a beneficial alternative to black box EBL models. One such set of white box algorithms is Competitive Learning (CL). CL algorithms, in contrast to EBL methods like deep neural networks and recurrent neural networks, employ a competitive learning process rather than weight adjustments to minimize loss. In CL-based techniques, nodes representing data samples compete against each other, with the winning node adjusting its weights to resemble the training sample, creating abstract representations of data. These nodes can be data-mined to create various visual and statistical explanations that users can use to understand the model's reasoning. Notably, the Self Organizing Map (SOM) and its variants, such as the Growing Self Organizing Map (GSOM) [9] and Growing Hierarchical Self Organizing Map (GHSOM) [10], constitute prominent CL algorithms, with the latter two enhancing the original SOM by dynamically

979-8-3503-8670-7/24/$31.00 ©2024 IEEE
DOI 10.1109/CIC62241.2024.00021

expanding the node map, enabling broader learning of abstract data patterns.

The use of CL algorithms in X-IDS offers several benefits, including enhanced transparency, explainability, and trustworthiness. This transparency enables users to formulate more confident responses to IDS-related tasks, leveraging the trust instilled by the CL algorithm's explanations. Security analysts can utilize model explanations to gain deeper insights into attacks, helping to strengthen the network's defenses. Moreover, machine learning engineers can identify and address model logic deficiencies, enhancing overall effectiveness by adjusting the architecture or introducing new training samples. Ultimately, these explanations contribute to bolstering the trust and credibility of the IDS, instilling users with greater confidence in their ability to fulfill their tasks effectively. Our previous work used the SOM algorithm to create an X-IDS. This work focuses on using the GSOM and GHSOM algorithms to create a more accurate, explainable intrusion detection system.

The major contributions presented in this paper are:

- An X-IDS architecture featuring three CL-based algorithms, built using DARPA's guidelines for an explainable system. We find that the innately explainable CL models have comparable accuracy to EBL models and that CL explanations can be more trustworthy than their black box counterparts.
- An analysis of statistical and visual explanations for an effective X-IDS. Our X-IDS architecture generates a collection of explainable visualizations ranging from global significance charts to fine-grained feature explanations. Users can use these explanations to understand how and why the model makes decisions.
- A performative analysis of our architecture using traditional accuracy metrics. We compare CL models to existing EBL models using the NSL-KDD and CIC-IDS-2017 datasets. CL models are 1% - 3% less accurate than EBL algorithms. Even though they are less accurate, their innate explainability and trustability make CL algorithms an important tool for X-IDSs.

The rest of the paper is outlined as follows - In Section II, we discuss background on IDS, XAI, and X-IDS. Section III describes the GSOM and GHSOM algorithms used in this paper. Section IV, outlines our CL based X-IDS with its architecture presented in Figure 1. Section V discusses our experimental results. Finally, the conclusion and future work are discussed in Section VI.

II. Related Work

In this section, we present some related work on Intrusion Detection Systems (IDS), Explainable Artificial Intelligence (XAI), and Explainable Intrusion Detection Systems (X-IDS).

A. Intrusion Detection Systems (IDS)

An *intrusion* refers to an action that obtains unauthorized access to a network or system [11]. An Intrusion Detection System (IDS) consists of tools, methods, and resources that help a Cyber Security Operation Center (CSoC) protect an organization by detecting an intrusion [12], [13]. IDS can be categorized into operation-based classes, such as signature, anomaly, and hybrid. Signature-based IDSs operate by preventing known attacks from accessing a network. The IDS compares incoming network traffic to a database of known attack signatures. Notably, this method has difficulty in preventing *zero-day* attacks [14]. Anomaly-based IDSs look for patterns in incoming traffic to recognize potential threats and leverage complex AI models [4], [15], [16]. A significant drawback of this approach is the tendency for such systems to categorize legitimate, unseen behavior as anomalous. Hybrid-based IDS incorporates the design philosophy of both signature-based and anomaly-based IDS to improve the detection rate while minimizing false positives [17], [18].

Current work on AI enabled anomaly-based IDSs can be further divided into black box and white box models [4]. White box models are considered *easy to understand* by an expert. This allows the expert to analyze the decision process and understand how the model renders its decision. This (semi-) transparent property allows white box models to be deployed in decision sensitive domains, where auditing the decision process is a requirement. White box models may use regression-based approaches [19], decision trees [20], and Self Organizing Maps (SOMs) [21]. Black box models, on the other hand, have an opaque decision process. This opaqueness property makes establishing the relationship between inputs and the decision difficult, if not outright impossible. Black box models comprise nearly all the AI enabled state-of-the-art approaches for IDS, as the focus is traditionally on model performance, not explainability. Examples of popular black box model techniques are Isolation Forest [22], One-Class SVM [23], and Neural Networks [24].

B. Explainable Artificial Intelligence Systems (XAI)

The notion of an Explainable Artificial Intelligence system (XAI) dates back to the 1970s. Moore et al. [25] surveyed works from the 1970s to the 1980s, detailing early methods of explanations. Some early explanations consisted of canned text and code translations, such as the 1974 explainer MYCIN [26]. We can find a more current definition of XAI by the Defense Advanced Research Projects Agency (DARPA) [5]. DARPA defines XAI as 'systems that are able to explain their reasoning to a human user, characterize their strengths and weaknesses, and convey a sense of their future behavior'. An XAI system that follows this definition offers some form of justification for its action, leading to more trust and understanding of the system. The explanations from an XAI system help the user not only in using and maintaining the AI model but also in helping users complete tasks in parallel with the AI system. Tasks can include doctors making medical decisions [26], [27], [28], credit score decisions [29], detecting counterfeit banknotes [30], advance maintenance [31], or CSoC operators defending a network [2], [5], [32].

The current literature consists of many different black box models being used alongside explanation techniques. Common

explainer modules for black box models are Local Interpretable Model-agnostic Explanations (LIME) [6], SHapely Additive exPlantions (SHAP) [7], and Layer-wise Relevance Propagation (LRP) [8]. Modern techniques for explaining black box models consist of creating surrogate models that generate explanations either locally or globally. Other methods involve propagating predictions backward in a neural network or decomposing a gradient. More novel approaches have also experimented with making datasets explainable [33] or making graphical user interfaces for explainable systems [34].

C. Explainable Intrusion Detection Systems (X-IDS)

Explainable Intrusion Detection Systems are still an emerging sub-genre. The need for explainability in IDS is becoming increasingly necessary. In decision sensitive domains, black boxes obfuscate the decision making process causing a lack of trust in predictions. The users need to be confident in the predictions or recommendations computed by an IDS. Understandable and trustworthy explanations allow users to perform their tasks correctly. The stakeholders of an IDS (e.g. CSoC operators, developers, and investors) are individuals who will be dependent on the performance of the system [4]. CSoC operators will be performing defense actions based on prediction and explanation results. Developers can use explanations to fortify the model in areas where it is weak. Investors may need explanations to help them make their company's budgeting decisions.

There are many examples of X-IDS being used in research today. A survey by Neupane et al. [4] describes in detail different X-IDS systems. Many black box implementations have been shown using libraries such as SHAP, LIME, or LRP [35], [36], [37]. There have also been more original explanation frameworks, such as one that involves using the CIA triad to generate explanations [33]. On the other hand, white box models have also been used to create strong X-IDS architectures. Notable entries have created explainable decision trees and linear regression models [19], [20]. In our previous work [2], we created a proof-of-concept X-IDS architecture that uses a Self Organizing Map (SOM). The architecture, based on DARPA's recommendation [5], is meant to be a good starting point for developing explainable IDS systems.

III. Competitive Learning Algorithms

The GSOM and GHSOM algorithms chosen for this paper are the Direct Batch Growing Self-Organizing Map (DBGSOM) and Directed Batch Growing Hierarchical Self-Organizing Map (DBGHSOM) [38]. Their pseudocode can be found in Alg. 1 and 2. Both take the same inputs: the dataset's dimensions (D), Spread Factor (SF), Learning Rate (LR), and Total Epochs (T). D is the number of features a dataset has. *SF* determines how quickly new nodes are generated. LR is the same as in the SOM. Another important variable that is not selected by the user is the Cumulative Error (CE). Each node in the GSOM has a CE value. CE is the sum of all the differences between a sample and its Best Matching Unit (BMU). This value slowly accumulates throughout training.

Algorithm 1 DBGSOM Algorithm

Input: Data Dimension (D), Spread Factor (SF), Learning Rate (LR), Total Epochs (T)
Output: Weights (W)
BEGIN
Initialization
Initialize 4 starter nodes with random Weights W [0,1]
Calculate Growth Threshold (GT): $GT = -D * ln(SF)$
Growing Phase
for Each Training Epoch in T **do**
  Reset Cumulative Error (CE) for all nodes to 0
  Present training samples
  Determine BMU using Euclidean Distance
  Update BMU and Neighboring weights
  Calculate CE for all BMUs
  for all non-boundary nodes **do**
    Distribute CE to neighbors
  end for
  for all boundary nodes CE $>$ GT **do**
    Grow depending on number of available neighbor positions
  end for
end for
return W
END

Algorithm 2 DBGHSOM Algorithm

Input: Data Dimension (D), Spread Factor (SF), Learning Rate (LR), Total Epochs (T)
Output: Weights (W)
BEGIN
Initialization
Same as Alg. 1
Horizontal Growing Phase
Same as Alg 1
Vertical Growing Phase
Calculate the Sum of all CE (SE)
Calculate the Vertical Threshold VT = LR $*$ SE
for All nodes with CE $>$ VT **do**
  Create new child DBGSOM
  Train new child DBGSOM using Alg. 1
end for
return W
END

DBGSOM and DBGHSOM follow similar tenets as the original algorithms. The main difference is that they generate new neighboring nodes in a batch process. Both are initialized with four starter nodes with randomized weights between 0 and 1. A growth threshold is calculated based on *SF* which is static throughout the training process. After the models are initialized, they enter the *Growing Phase* for the GSOM or

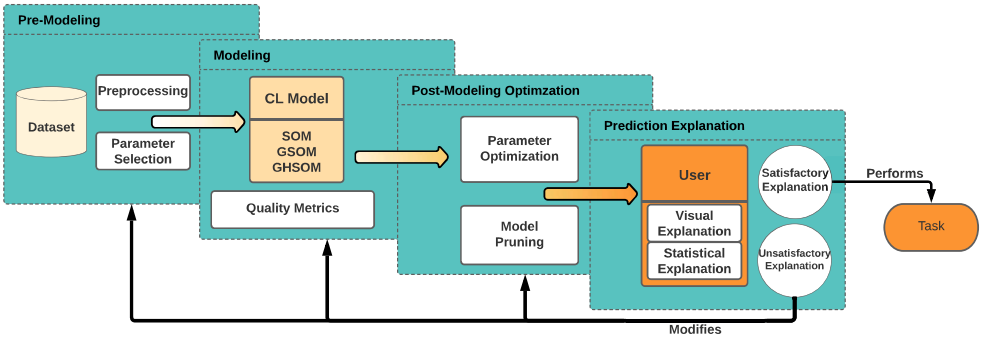


Fig. 1: A competitive learning based X-IDS architecture. The architecture is divided into four phases: Pre-Modeling, Modeling, Post-Modeling Optimization, and Prediction Explanation. Each phase contributes to translating raw input data into accurate predictions and useful explanations. Culminating in a user successfully completing an associated task or being required to make changes to previous steps in the architecture.

Horizontal Growth Phase for the GHSOM. All nodes have their *CE* reset to 0. Training the GSOM is now similar to training a SOM. Each training sample is presented to the map, and its respective BMU is found. The BMU has its weights and *CE* updated based on the training sample. Additionally, all neighbors of the BMU have their weights updated. After all of the training data has been used to update weights, we find all non-boundary nodes. For each of these nodes, we distribute their *CE* to their neighbors. Lastly, all boundary nodes for which $CE_i > GT$ have a new neighbor node generated next to it. An extra step is taken for the GHSOM where it checks if any node's CE is greater than the model's Vertical Threshold (VT). If any node's $CE_i > VT$, a new child GSOM is generated hierarchically "below" the parent GSOM with the offending node.

A. Competitive Learning and Intrusion Detection

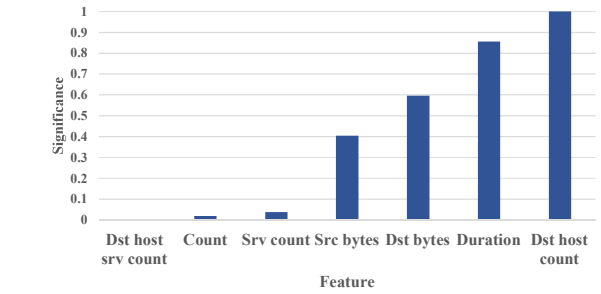
In the past, CL algorithms have been used to create many IDSs. These studies focused on building accurate IDSs and did not discuss explainability. Among these approaches, SOMs were used to create both host-based [39] and network-based [40], [41], [42] IDSs. The majority of these methods simply trained a SOM based IDS and illustrated mappings between data points and the associated BMU. The approaches described in [42], [43] use multiple SOMs in conjunction with one another to create a more effective IDS. Only one approach [40] discussed the false positive rate and accuracy of a SOM-based IDS. Their method for prediction involved assigning a label to BMUs based on the training dataset. In our previous work [2], we created an X-IDS architecture based on DARPA's recommended architecture. One of its main features is having user input for correcting or modifying the model or its explanations. Using this architecture, we were able to achieve an accuracy of 91% on NSL-KDD and 80% on CIC-IDS-2017.

In addition, we can look at instances of GSOM-based IDS. A multi-agent GSOM proposed by Palomo et al. [44] was created to be more accurate on datasets with many different attack types. The Growing SOM should be able to continuously grow as it discovers new attack types. Their IDS was able to achieve a 90% accuracy and a 1% false positive rate on the KDD CUP 1999 dataset using 38 different attacks. A novel GSOM algorithm was developed in [45] named Statistics-Enhanced Direct Batch Growth Self-Organizing Map (SE-DBGSOM). One of the goals of using this updated algorithm is to improve the efficiency of inserting new nodes. The authors note that their algorithm improves upon previous GSOMs by reducing the number of 'unnecessary' nodes. This improves both runtime and false positive rates. SE-DBGSOM was able to achieve a greater than 99% accuracy on KDD99 and CICIDS2017 datasets with false positive rates as low as .6%.

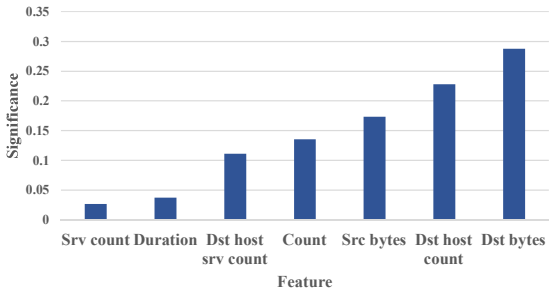
GHSOMs have also made an impact in the field of IDS. One inspiring work that created a GHSOM IDSs is from the authors Ippoliti et al. [46]. They create an Adaptive GH-SOM (A-GHSOM) that uses dynamic normalization scaling, adaptive growth thresholds, and confidence filtering to reduce inconsistent predictions. We can find other works that make other modifications like adding new metrics for numeric and symbolic data [47], enhancing map initialization and weight distribution [48], and changing growing conditions [49]. Many of these implementations were tested using KDD CUP 1999 or NSL-KDD to great effect.

IV. A Competitive Learning based Explainable Intrusion Detection Systems (X-IDS)

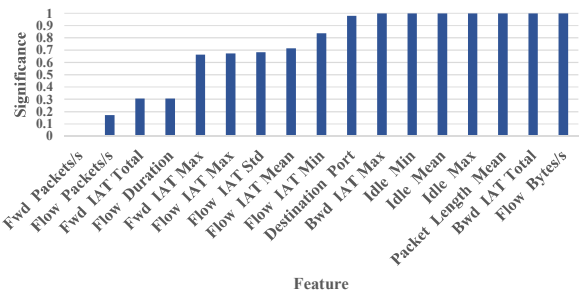
Explanations generated by the X-IDS should assist Cyber Security Operation Center (CSoC) operators in their mission to protect their organization. To help achieve this goal, we create the proof of concept Competitive Learning (CL) based X-IDS architecture in Figure 1. The proposed architecture is based on DARPA's recommended architecture for XAI systems [5]. The framework's components can be changed to suit the user's needs. The architecture is abstract enough, such that methods other than CL algorithms can be interchanged to create different X-IDSs. The architecture consists of four



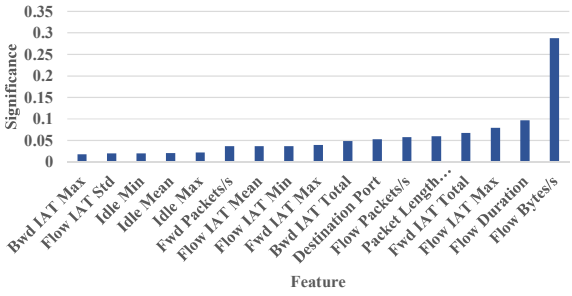
(a) NSL-KDD Local Anomaly Explanation



(b) NSL-KDD Global Feature Significance



(c) CIC-IDS-2017 Local Anomalous Explanation



(d) CIC-IDS-2017 Global Feature Significance

Fig. 2: Local and global feature explanations for NSL-KDD and CIC-IDS datasets. (a)(c) Demonstrate features chosen by the GSOM for malicious samples from NSL-KDD and CIC-IDS-2017 datasets. The closer a sample's feature value is to the BMU's feature value, the higher its significance. (b)(d) Global feature significance is calculated using Bayesian Probability of Significance [50]. Features with higher significance values are much more likely to cause predictions to be made for benign or anomalous. Global explanations apply to all tested models.

phases: pre-modeling, modeling, post-modeling optimization, and prediction explanation. In the pre-modeling phase, raw datasets are preprocessed and parameters are selected for the model. In the modeling phase, our CL algorithms are trained and quality metrics are recorded. In our proof of concept system, we are using the SOM family of CL algorithms. In the post-modeling optimization phase, models can then be optimized through various means described below. In the prediction explanation phase, data mining techniques are employed on the resulting models to generate explanatory visualizations that allow users to understand how predictions are generated.

A. *Pre-Modeling Phase*

The pre-modeling phase consists of preprocessing raw datasets and initial parameter selection. The parameters used to train the CL algorithms can be found in Table I. The preprocessing for our models includes feature selection and normalization. The feature selection algorithm that we have chosen to use is the 'Bayesian probability of significance' [50], which selects the most relevant features from each dataset. Feature selection is not used when training the GHSOM. The GHSOM is able to use all of the features in a dataset more effectively due to its hierarchical nature. Additionally, the datasets are preprocessed for binary classification. Lastly, the datasets are normalized to minimize feature bias and improve accuracy. After preprocessing is finished, the new, high-quality dataset can then be passed to the model. The next section details information about the selected datasets and their usefulness in testing IDSs.

In this work, NSL-KDD [51] and CIC-IDS-2017 [52] are used to test the explainability and effectiveness of our architecture. NSL-KDD is chosen because of its wide use in the literature. There are a few major benefits to using the NSL-KDD dataset. First, it allows our method to be compared to other existing methods for IDSs. Second, the dataset's relatively small size allows for quick testing and runtime comparisons against larger datasets. On the other hand, CIC-IDS-2017 includes more modern attacks and is useful for testing an unbalanced dataset. It was synthetically created over the course of five days to mimic the behavior of 25 users. The use of this dataset, allows us to show that our IDS is compatible with real-world data and to stress-test our systems for various performative metrics.

B. *Modeling Phase*

Using the high quality dataset and the parameters selected in the pre-modeling phase, we can train the set of CL models. We

	Parameter	NSL-KDD	CIC-IDS-2017
SOM	n	18	18
	m	18	18
	LR	.3	.3
	Epochs	1000	1000
GSOM	LR	.006	.006
	SF	.9	.9
	Epochs	100	40
GHSOM	LR	.006	.006
	SF	.3	.3
	Epochs	100	40

TABLE I: The selected parameters for each CL model.

utilize a subclass of CL models described in Section III. These models create clusters mimicking input data, and in doing so, they create a map that can be data-mined for explanatory purposes. The GSOM and GHSOM are more complex versions of the SOM algorithm that allow for dynamic growth, allowing them to have higher accuracies on more complex datasets.

There have been various metrics and measures proposed to evaluate the quality of a trained SOM. These include quantization error, topographic error, embedding accuracy, and convergence index. The quantization error was used by Kohenen [53], and measures the average distance between nodes and the data points. The topographic error measures how well preserved features are in the low dimensional output space. It is measured by evaluating how often the BMU and the second BMU are next to each other [54], [55]. The map embedding accuracy is similar to the quantization error. This metric measures how similar the distribution of the input data is with respect to that of the SOM units [56]. In order to measure both topographic preservation and distribution similarity between the input and SOM units, the convergence index was proposed to be a measure that linearly combines the map embedding accuracy and the topographic error [57]. Performative metrics are also important to include in an IDS architecture. These metrics include accuracy, F1-score, false positive rate, and false negative rate. We also opt to include training time and prediction speed since they can play an important role in intrusion detection. The experimental results using these performative metrics can be view in Section V. These measurements allow the architecture to be compared to the architecture of other existing IDS.

C. Post-Modeling Optimization Phase

In this phase, we optimize the model using Bayesian Search to get more accurate predictions and explanations. Bayesian search is a probabilistic hyper-parameter tuning method that limits its search space. It makes informed decisions about each set of parameters tested. On average, Bayesian search can provide a set of parameters faster than grid search and random search, two notable hyper-parameter search algorithms. The trade-off is that it may not find the best set of parameters as it doesn't search the entire parameter space.

D. Prediction Explanation Phase

Once the modeling and optimization phases have been completed, and the quality metrics have ensured that the model is a good representation of the data, the model can be used to perform a variety of explainability and visualizations. The models themselves are lists of nodes and the weights associated with the nodes. Visualizations include creating local and global explanations, U-Matrices, and feature heatmaps. Users can use explanations to perform tasks to better defend the network. When a user receives a subpar explanation, the user can modify the architecture where needed to help bolster the X-IDS. By using the explanations generated from the white box CL models, the user can build trust and confidence that the model is working as intended.

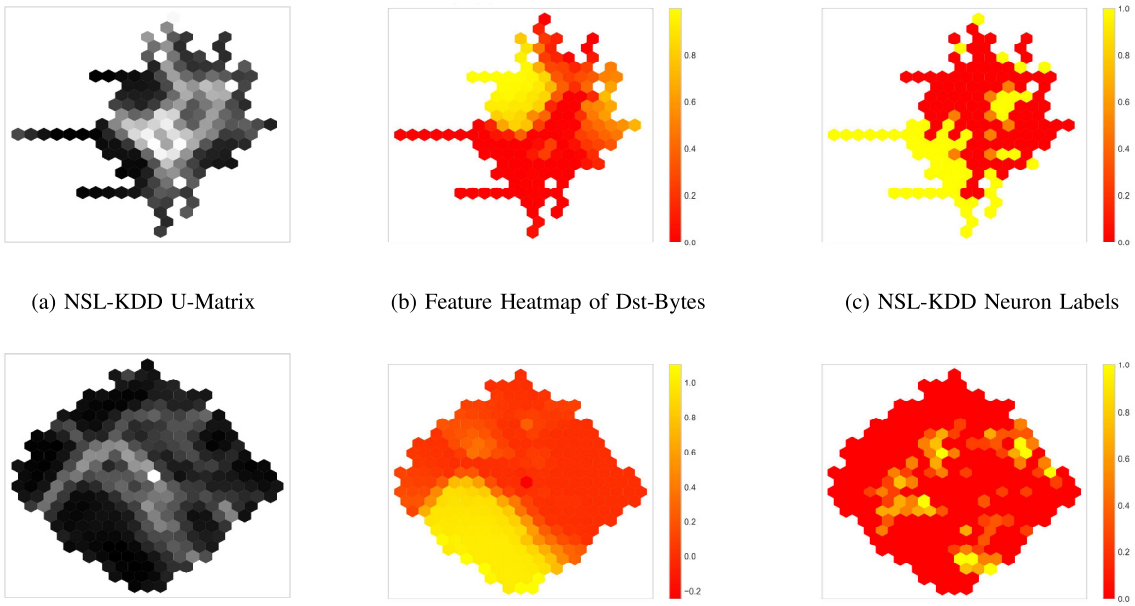
1) Local and Global Explanations: Global and local explainability can be achieved by examining important features of the trained CL algorithm, and then utilizing this information to generate an explanation for a specific data instance classification or cluster classification [58]. Global significance for NSL-KDD is shown in Figure 2b with higher values denoting that a feature has a higher probability of being important. The algorithm chosen to determine this variance was 'Bayesian probability of significance' [50]. Higher variance features increase the probability that a model will capture the dataset's structure. Through this graph, an analyst can understand which features are important to the overall SOM structure, allowing them to examine predictions at a local level based on globally important features.

Figure 2a shows the GSOM local explanations for a prediction on the NSL-KDD dataset. Each feature has a value representing its significance. Significance (S) is a calculation involving the *min-maxed* distances from a BMU inverted so that higher values are more important. The formula can be seen in Formula 1. Features with the highest significance are closer to the BMU, therefore, they played a large role in computing the predicted value. Seeing the specific features that influence predictions provides insight into samples labeled as malicious or benign and can further help users determine the reason for incorrect predictions. These features can also be further investigated with feature value heat maps.

$$S = 1 - (\frac{X - X_{min}}{X_{max} - X_{min}}) \tag{1}$$

2) Unified Distance Matrix (U-Matrix): The U-Matrix visualizes the distances between neighboring SOM nodes. With distances shown as a color gradient, nodes far apart will create light boundaries while areas with similar nodes will be darker. This can visually represent the natural clusters of input data. To enhance the standard U-Matrix, the starburst model uses connected component lines of nodes overlaid on the matrix to better represent clusters [59]. For a labeled data set, the user is able to visualize each BMU along with the associated label. Figure 3a shows clear clusters with boundaries separating malicious (1) and benign (0) behavior. Using this information users can investigate more visualizations and feature importance values to gain an understanding of why certain malicious network activities are being grouped together.

3) Feature Value Heat Map: A heat map applied to a feature shows general trends that a feature has on a model, in



(a) NSL-KDD U-Matrix (b) Feature Heatmap of Dst-Bytes (c) NSL-KDD Neuron Labels

(d) CIC-IDS-2017 U-Matrix (e) Feature Heatmap of Flow Bytes/s (f) CIC-IDS-2017 Neuron Labels

Fig. 3: Visualizations generated from a GSOM for models trained on NSL-KDD and CIC-IDS-2107. (a)(d) The U-matrix displays clusters in the map. Darker nodes denote nodes that are more similar to one another while lighter nodes denote separation. (b)(e) The feature value heatmaps display a specific feature's value on each GSOM node. Lighter values represent units with values closer to 1, while darker values show values closer to 0. (c)(f) The Neuron Label map shows the class label represented by a red or yellow color.

this case, the entire GSOM model. GSOM feature values are represented from 0 to 1, and the heat maps denote this with darker and lighter values, respectively. An example feature value heat map can be found in Figure 3b. In this example, the 'dst bytes' feature has a cluster of higher values in the top-right corner, while the rest of the GSOM consists of lower values. Users can use this information to form conclusions about the model. Feature value maps are more powerful when multiple are viewed at a time. The U-Matrix chart can then be referenced to make general decisions about the model. The heat maps work well as a fine-grained global explanation that helps users understand the overall model.

4) Users Performing Tasks: An important component of our architecture is its *user-in-the-loop* system. A 'user' is a network's stakeholder. There can be many kinds of stakeholders for an IDS. AI engineers who implement and maintain the X-IDS architecture, security analysts who protect the network, and investors who manage security expenses. Tasks are performed with the goal of protecting the network and are enhanced by the X-IDS's generated predictions.

When a satisfactory prediction has been created, a user can perform their task. Satisfactory explanations will cause the user to be able to perform their tasks more effectively. However, not all explanations will be useful. When an unsatisfactory explanation is created, a user can use that explanation to make changes to the parts of the architecture. This could be accomplished by changing how datasets are preprocessed, choosing a new ML model, modifying optimizations, or creating a new style of explanation.

V. Evaluation & Experimental Results

Our CL based architecture and its SOM variants were evaluated on both *traditional performative tests* and *explanation generation*. The datasets used to test our architecture were NSL-KDD and CIC-IDS-2017. In this section, we examine the performative results from the GSOM and GHSOM algorithms and compare them to the basic Self Organizing Map (SOM) algorithm and black box EBL algorithms.

A. Model Explainability

The explanations generated by the GSOM for the NSL-KDD and CIC-IDS-2017 datasets can be found in Figures 2 and 3. To understand the model, following a methodology can be beneficial. Users can begin by viewing the global feature significance charts in Figures 2b and 2d. Here we can see that the features with the most variability for the NSL-KDD dataset are 'Destination (Dst) bytes', 'Destination (Dst) host count', and 'Source (Src) bytes'. We can then look at many local significance explanations for a particular label. Figure 2a is an example of an anomalous explanation. Three of the four

NSL-KDD							
	SOM	**GSOM**	**GHSOM**	**NDNN Jia et al. [60]**	**CNN Mohammadpour et al. [61]**	**BGRU+MLP Xu et al. [62]**	**BAT-MC Su et al. [63]**
Accuracy	90.9%	96.7%	98.2%	95.0%	99.8%	99.3%	99.2%
Precision	97.2%	96.6%	98.0%	-	-	-	-
Recall	83.3%	96.5%	98.3%	97.4%	-	99.3%	-
F1	89.7%	96.6%	98.1%	91.4%	-	-	-
FPR	2.2%	3.1%	1.9%	-	-	0.8%	-
FNR	16.6%	3.5%	1.6%	-	-	-	-
Network Size	1	1	7288	-	-	-	-
Training Time (s)	8	60	692	-	-	-	-
Prediction Time (ms)	.03	.03	.06	-	-	-	-
CIC-IDS-2017							
	SOM	**GSOM**	**GHSOM**	**SDCNN Khan et al. [64]**	**DNN+RE Almutlaq et al. [65]**	**SS-Deep-ID Abdel-Basset et al. [66]**	**CNN-IDS* Halbouni et al. [67]**
Accuracy	79.4%	94.6%	96.7%	99.3%	97.4%	99.6%	99.6%
Precision	83.2%	83.7%	89.1%	99.1%	98.3%	99.5%	99.7%
Recall	42.0%	90.0%	94.5%	99.7%	99.2%	99.2%	99.4%
F1	55.8%	86.7%	91.7%	99.4%	98.3%	99.4%	99.7%
FPR	19.0%	4.3%	2.8%	1.0%	-	0.7%	0.5%
FNR	23.0%	10.0%	5.5%	1.0%	-	0.5%	-
Network Size	1	1	16894	-	-	-	-
Training Time (s)	260	1820	4299	-	-	-	-
Prediction Time (ms)	.03	.06	1.5	-	-	-	-

TABLE II: This table shows the results from testing our CL based X-IDS architecture. We compare our results with existing black box EBL models including deep neural networks, convoluted neural networks, and various ensemble methods. We do not provide the training and testing times for other models as they are not tested on the same machine.

most significant features (destination bytes, destination host count, and source bytes) coincide with the top features in the global feature significance explanation. A user, after having looked at many anomalous examples, would then be able to form some conclusions about how the model labels data.

Next, users can use the visual explanations in order to fine tune their understanding of the model. The U-matrix, feature component map, and label map can be found in Figure 3. Since features with higher variability are more likely to cause separation in clustering, users can use the global feature significance explanation as a starting point. The U-matrix in Figure 3a shows five to six separated clusters. Using the feature component map in Figure 3b, we can see that one of the clusters is associated with higher values of 'Destination (Dst) bytes'. Finally, the user can use the label map to formalize their conclusion. In this case, the user may conclude: "Higher values of Destination bytes are associated with benign traffic." Users would continue with this methodology with other features, helping them to create more complex conclusions. Using these explanations, it may be possible to determine why the model is losing accuracy and make adjustments to parts of the X-IDS architecture. This paper leaves the CIC-IDS-2017 explanations as an exercise for the reader.

Understanding the GSOM explanations using this methodology appears straightforward. However, the same cannot be said for the GHSOM. Table II shows that the GHSOM for NSL-KDD and CIC-IDS-2017 have a network size of 7,288 and 16,894 respectively. The above methodology not only requires the user to browse the single GSOM explanations but also many of the feature component explanations as well. The large size of the GHSOM, although considered explainable, is difficult for users to understand. Future works should look to making GHSOMs easier to grasp for humans.

B. Performative Tests

The model parameters can be found in Table I. The GSOM parameters were set to 100 and 40 training epochs for NSL-KDD and CIC-IDS-2017, respectively. We found that this, in addition to an aggressive Spread Factor (SF) of .9 created the best performative results. The GHSOM parameters were set to 100 epochs per GSOM created using an SF of .3 and a Learning Rate (LR) of .006. These settings were discovered using the parameter selection process outlined in Section IV-D. Using these parameters, we were able to create well trained, highly accurate models.

Table II shows the results from our CL-based X-IDS architecture. We compare the GSOM and the GHSOM to the basic SOM algorithm and black box EBL algorithms found in the literature. The GHSOM performs the best of the CL algorithms with an accuracy of 98.2% on the NSL-KDD dataset. The less complex GSOM algorithm loses some accuracy falling to 96.7%. Compared to the EBL models, these explainable, white box models lose about 1% accuracy on NSL-KDD. The results from CIC-IDS-2017 tell a different story. We see a similar trend where the more complex CL algorithms perform

better. However, we notice there is a much lower F1-score on this dataset. Using the explanations that we previously viewed, we can likely conclude that CIC-IDS-2017 model has overfit. The higher false negative rate is likely due to the bottom cluster that has a mix of benign and malicious nodes. This is likely corrected by stratifying the dataset in preprocessing. By making changes to parts of the X-IDS architecture, the CIC-IDS-2017 model may be able to overcome the 3% accuracy loss compared to the EBL algorithms.

VI. Conclusion

In this paper, we introduced an Explainable Intrusion Detection (X-IDS) architecture featuring Competitive Learning (CL) based algorithms. The architecture is built using DARPA's guidelines for explainable systems. Our architecture consists of four phases: Pre-Modeling, Modeling, Post-Modeling Optimization, and Prediction Explanation. By employing Growing Self Organizing Map (GSOM) and Growing Hierarchical Self Organizing Map (GHSOM), we demonstrated the efficacy of CL algorithms in achieving accuracies similar to Error Based Learning (EBL) algorithms. The architecture was able to achieve accuracies of 98.2% on NSL-KDD and 96.7% on CIC-IDS-2017. Despite being slightly less accurate than EBL algorithms, CL algorithms are far more trustworthy and explainable. We emphasized the importance of explainability, showcasing the CL algorithms' ability to provide interpretable explanations, furthering user trust and model improvement. Our findings show the potential of CL algorithms in advancing X-IDS by combining accuracy with interpretability, paving the way for more explainable and trustworthy security in the form of explainable intrusion detection systems.

VII. Acknowledgment

This work by Mississippi State University was financially supported by the U.S. Department of Defense (DoD) High Performance Computing Modernization Program, through the US Army Engineering Research and Develop Center (ERDC) (#W912HZ-21-C0058). The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the U.S. Army ERDC or the U.S. DoD.

References

[1] Alaa Marshan. Artificial intelligence: Explainability, ethical issues and bias. *Annals of Robotics and Automation*, pages 034–037, 08 2021.

[2] Jesse Ables, Thomas Kirby, William Anderson, Sudip Mittal, Shahram Rahimi, Ioana Banicescu, and Maria Seale. Creating an Explainable Intrusion Detection System Using Self Organizing Maps. In *IEEE Symposium on Computational Intelligence in Cyber Security*, 2022.

[3] Anna L Buczak and Erhan Guven. A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Communications surveys & tutorials*, 18(2):1153–1176, 2015.

[4] Subash Neupane, Jesse Ables, William Anderson, Sudip Mittal, Shahram Rahimi, Ioana Banicescu, and Maria Seale. Explainable intrusion detection systems (x-ids): A survey of current methods, challenges, and opportunities. *arXiv preprint arXiv:2207.06236*, 2022.

[5] David Gunning and David Aha. Darpa's explainable artificial intelligence (xai) program. *AI Magazine*, 40(2):44–58, 2019.

[6] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "Why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

[7] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.

[8] Alexander Binder, Grégoire Montavon, Sebastian Lapuschkin, Klaus-Robert Müller, and Wojciech Samek. Layer-wise relevance propagation for neural networks with local renormalization layers. In *International Conference on Artificial Neural Networks*, pages 63–71. Springer, 2016.

[9] Bernd Fritzke. Growing grid—a self-organizing network with constant neighborhood range and adaptation strength. *Neural processing letters*, 2(5):9–13, 1995.

[10] Michael Dittenbach, Dieter Merkl, and Andreas Rauber. The growing hierarchical self-organizing map. In *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium*, volume 6, pages 15–19. IEEE, 2000.

[11] Dorothy E Denning. An intrusion-detection model. *IEEE Transactions on software engineering*, (2):222–232, 1987.

[12] Rebecca Gurley Bace, Peter Mell, et al. Intrusion detection systems, 2001.

[13] Andrew McDole, Maanak Gupta, Mahmoud Abdelsalam, Sudip Mittal, and Mamoun Alazab. Deep learning techniques for behavioural malware analysis in cloud iaas. In *Malware Analysis using Artificial Intelligence and Deep Learning*. Springer, 2021.

[14] Ashu Sharma and Sanjay Kumar Sahay. Evolution and detection of polymorphic and metamorphic malwares: A survey. *arXiv preprint arXiv:1406.7061*, 2014.

[15] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Comput. Surv.*, 41:15:1–15:58, 2009.

[16] Andrew McDole, Mahmoud Abdelsalam, Maanak Gupta, and Sudip Mittal. Analyzing cnn based behavioural malware detection techniques on cloud iaas. In *International Conference on Cloud Computing*, pages 64–79. Springer, 2020.

[17] Mateusz Szczepański, Michał Choraś, Marek Pawlicki, and Rafał Kozik. Achieving explainability of intrusion detection system by hybrid oracle-explainer approach. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2020.

[18] Guansong Pang, Choubo Ding, Chunhua Shen, and Anton van den Hengel. Explainable deep few-shot anomaly detection with deviation networks. *arXiv preprint arXiv:2108.00462*, 2021.

[19] Basant Subba, Santosh Biswas, and Sushanta Karmakar. Intrusion detection systems using linear discriminant analysis and logistic regression. In *2015 Annual IEEE India Conference (INDICON)*, pages 1–6. IEEE, 2015.

[20] Basim Mahbooba, Mohan Timilsina, Radhya Sahal, and Martin Serrano. Explainable artificial intelligence (xai) to enhance trust management in intrusion detection systems using decision tree model. *Complexity*, 2021, 2021.

[21] Chet Langin, Michael Wainer, and Shahram Rahimi. Annabell island: a 3d color hexagonal som for visual intrusion detection. *Internation Journal of Computer Science and Information Security*, 9(1):1–7, 2011.

[22] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. *2008 Eighth IEEE International Conference on Data Mining*, pages 413–422, 2008.

[23] Bernhard Schölkopf, Robert C. Williamson, Alex Smola, John Shawe-Taylor, and John C. Platt. Support vector method for novelty detection. In *NIPS*, 1999.

[24] G.P. Zhang. Neural networks for classification: a survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 30(4):451–462, 2000.

[25] Johanna D Moore and William R Swartout. Explanation in expert systemss: A survey. Technical report, University of Southern California Marina Del Rey Information Sciences Inst, 1988.

[26] Edward Hance Shortliffe. Mycin: a rule-based computer program for advising physicians regarding antimicrobial therapy selection. Technical report, Stanford Univ Calif Dept of Computer Science, 1974.

[27] Andreas Holzinger, Chris Biemann, Constantinos S Pattichis, and Douglas B Kell. What do we need to build explainable ai systems for the medical domain? *arXiv preprint arXiv:1712.09923*, 2017.

[28] Leeanne Lindsay, Sonya Coleman, Dermot Kerr, Brian Taylor, and Anne Moorhead. Explainable artificial intelligence for falls prediction. In

International Conference on Advances in Computing and Data Sciences, pages 76–84. Springer, 2020.

[29] Ye Eun Chun, Se Bin Kim, Ja Yun Lee, and Ji Hwan Woo. Study on credit rating model using explainable ai. *The Korean Data & Information Science Society*, 32(2):283–295, 2021.

[30] Miseon Han and Jeongtae Kim. Joint banknote recognition and counterfeit detection using explainable artificial intelligence. *Sensors*, 19(16):3607, 2019.

[31] Subash Neupane, Ivan A Fernandez, Wilson Patterson, Sudip Mittal, and Shahram Rahimi. A temporal anomaly detection system for vehicles utilizing functional working groups and sensor channels. *IEEE International Conference on Collaboration and Internet Computing (IEEE CIC 2022)*, 2022.

[32] DARPA. Broad agency announcement explainable artificial intelligence (xai). *DARPA-BAA-16-53*, pages 7–8, 2016.

[33] Sheikh Rabiul Islam, William Eberle, Sheikh K Ghafoor, Ambareen Siraj, and Mike Rogers. Domain knowledge aided explainable artificial intelligence for intrusion detection and response. *arXiv preprint arXiv:1911.09853*, 2019.

[34] Chunyuan Wu, Aijuan Qian, Xiaoju Dong, and Yanling Zhang. Feature-oriented design of visual analytics system for interpretable deep learning based intrusion detection. In *2020 International Symposium on Theoretical Aspects of Software Engineering (TASE)*, pages 73–80. IEEE, 2020.

[35] Maonan Wang, Kangfeng Zheng, Yanqing Yang, and Xiujuan Wang. An explainable machine learning framework for intrusion detection systems. *IEEE Access*, 8:73127–73141, 2020.

[36] Izhar Ahmed Khan, Nour Moustafa, Dechang Pi, Karam M Sallam, Albert Y Zomaya, and Bentian Li. A new explainable deep learning framework for cyber threat discovery in industrial iot networks. *IEEE Internet of Things Journal*, 2021.

[37] Kasun Amarasinghe, Kevin Kenney, and Milos Manic. Toward explainable deep neural network based anomaly detection. In *2018 11th International Conference on Human System Interaction (HSI)*, pages 311–317. IEEE, 2018.

[38] Mahdi Vasighi and Homa Amini. A directed batch growing approach to enhance the topology preservation of self-organizing map. *Applied Soft Computing*, 55:424–435, 2017.

[39] Peter Lichodzijewski, A Nur Zincir-Heywood, and Malcolm I Heywood. Host-based intrusion detection using self-organizing maps. In *Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN'02 (Cat. No. 02CH37290)*, volume 2, pages 1714–1719. IEEE, 2002.

[40] Emiro De la Hoz, Andrés Ortiz García, Julio Ortega Lopera, Eduardo Miguel De La Hoz Correa, and Fabio Enrique Mendoza Palechor. Implementation of an intrusion detection system based on self organizing map. 2015.

[41] VK Pachghare, Parag Kulkarni, and Deven M Nikam. Intrusion detection system using self organizing maps. In *2009 International Conference on Intelligent Agent & Multi-Agent Systems*, pages 1–5. IEEE, 2009.

[42] Brandon Craig Rhodes, James A Mahaffey, and James D Cannady. Multiple self-organizing maps for intrusion detection. In *Proceedings of the 23rd national information systems security conference*, pages 16–19. MD Press Baltimore, 2000.

[43] Sahin Albayrak, Christian Scheel, Dragan Milosevic, and Achim Muller. Combining self-organizing map algorithms for robust and scalable intrusion detection. In *International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06)*, volume 2, pages 123–130. IEEE, 2005.

[44] Esteban J Palomo, Enrique Domínguez, Rafael M Luque, and Jose Munoz. A self-organized multiagent system for intrusion detection. In *International Workshop on Agents and Data Mining Interaction*, pages 84–94. Springer, 2009.

[45] Xiaofei Qu, Lin Yang, Kai Guo, Linru Ma, Tao Feng, Shuangyin Ren, and Meng Sun. Statistics-enhanced direct batch growth self-organizing mapping for efficient dos attack detection. *IEEE Access*, 7:78434–78441, 2019.

[46] Dennis Ippoliti and Xiaobo Zhou. A-ghsom: An adaptive growing hierarchical self organizing map for network anomaly detection. *Journal of Parallel and Distributed Computing*, 72(12):1576–1590, 2012.

[47] Esteban J Palomo, Enrique Domínguez, Rafael Marcos Luque, and José Muñoz. A new ghsom model applied to network security. In *International Conference on Artificial Neural Networks*, pages 680–689. Springer, 2008.

[48] Maher Salem and Ulrich Buehler. An enhanced ghsom for ids. In *2013 IEEE International Conference on Systems, Man, and Cybernetics*, pages 1138–1143. IEEE, 2013.

[49] Yahui Yang, Dianbo Jiang, and Min Xia. Using improved ghsom for intrusion detection. *Journal of Information Assurance and Security*, 5:232–239, 2010.

[50] Lutz Hamel and Chris Brown. Bayesian probability approach to feature significance for infrared spectra of bacteria. *Applied Spectroscopy*, 66:48–59, 1 2012.

[51] Mahbod Tavallaee, Ebrahim Bagheri, Wei Lu, and Ali A Ghorbani. A detailed analysis of the kdd cup 99 data set. pages 1–6, 2009.

[52] Iman Sharafaldin, Arash Habibi Lashkari, and Ali A Ghorbani. Toward generating a new intrusion detection dataset and intrusion traffic characterization. *ICISSp*, 1:108–116, 2018.

[53] Teuvo Kohonen. The self-organizing map. *Neurocomputing*, 21:1–6, 1998.

[54] Gregory Breard. Evaluating self-organizing map quality measures as convergence criteria. 2017.

[55] Jouko Lampinen and Erkki Oja. Clustering properties of hierarchical self-organizing maps. *Journal of Mathematical Imaging and Vision*, 2:261–272, 1992.

[56] Lutz Hamel. Som quality measures: An efficient statistical approach. volume 428, pages 49–59. Springer Verlag, 2016.

[57] Self-organizing map convergence. *Int. J. Serv. Sci. Manag. Eng. Technol.*, 9:61–84, 4 2018.

[58] Chathurika S Wickramasinghe, Kasun Amarasinghe, Daniel L Marino, Craig Rieger, and Milos Manic. Explainable unsupervised machine learning for cyber-physical systems. *IEEE Access*, 9:131824–131843, 2021.

[59] Lutz Hamel and Chris Brown. Improved interpretability of the unified distance matrix with connected components. *7th International Conference on Data Mining (DMIN'11)*, 4 2012.

[60] Yang Jia, Meng Wang, and Yagang Wang. Network intrusion detection algorithm based on deep neural network. *IET Information Security*, 13(1):48–53, 2019.

[61] Leila Mohammadpour, Teck Chaw Ling, Chee Sun Liew, and Chun Yong Chong. A convolutional neural network for network intrusion detection system. *Proceedings of the Asia-Pacific Advanced Network*, 46(0):50–55, 2018.

[62] Congyuan Xu, Jizhong Shen, Xin Du, and Fan Zhang. An intrusion detection system using a deep neural network with gated recurrent units. *IEEE Access*, 6:48697–48707, 2018.

[63] Tongtong Su, Huazhi Sun, Jinqi Zhu, Sheng Wang, and Yabo Li. Bat: Deep learning methods on network intrusion detection using nsl-kdd dataset. *IEEE Access*, 8:29575–29585, 2020.

[64] Adnan Shahid Khan, Zeeshan Ahmad, Johari Abdullah, and Farhan Ahmad. A spectrogram image-based network anomaly detection system using deep convolutional neural network. *IEEE Access*, 9:87079–87093, 2021.

[65] Samah Almutlaq, Abdelouahid Derhab, Mohammad Mehedi Hassan, and Kuljeet Kaur. Two-stage intrusion detection system in intelligent transportation systems using rule extraction methods from deep neural networks. *IEEE Transactions on Intelligent Transportation Systems*, 2022.

[66] Mohamed Abdel-Basset, Hossam Hawash, Ripon K Chakrabortty, and Michael J Ryan. Semi-supervised spatiotemporal deep learning for intrusions detection in iot networks. *IEEE Internet of Things Journal*, 8(15):12251–12265, 2021.

[67] Asmaa H Halbouni, Teddy Surya Gunawan, Murad Halbouni, Faisal Ahmed Abdullah Assaig, Mufid Ridlo Effendi, and Nanang Ismail. Cnn-ids: Convolutional neural network for network intrusion detection system. In *2022 8th International Conference on Wireless and Telematics (ICWT)*, pages 1–4. IEEE, 2022.