G-IDS: Generative Adversarial Networks Assisted Intrusion Detection System

Md Hasan Shahriar*, Nur Imtiazul Haque*, Mohammad Ashiqur Rahman*, and Miguel Alonso Jr†

*Analytics for Cyber Defense (ACyD) Lab, Florida International University, USA
†School of Computing and Information Sciences, Florida International University, USA
{mshah068, nhaqu004, marahman, malonsoj}@fiu.edu

Abstract—The boundaries of cyber-physical systems (CPS) and the Internet of Things (IoT) are converging together day by day to introduce a common platform on hybrid systems. Moreover, the combination of artificial intelligence (AI) with CPS creates a new dimension of technological advancement. All these connectivity and dependability are creating massive space for the attackers to launch cyber attacks. To defend against these attacks, intrusion detection system (IDS) has been widely used. However, emerging CPS technologies suffer from imbalanced and missing sample data, which makes the training of IDS difficult. In this paper, we propose a generative adversarial network (GAN) based intrusion detection system (G-IDS), where GAN generates synthetic samples, and IDS gets trained on them along with the original ones. G-IDS also fixes the difficulties of imbalanced or missing data problems. We model a network security dataset for an emerging CPS technologies using NSL KDD-99 dataset and evaluate our proposed model's performance using different metrics. We find that our proposed G-IDS model performs much better in attack detection and model stabilization during the training process than a standalone IDS.

Index Terms—Generative Adversarial Networks, Cyber-Physical Systems Security, Intrusion Detection System

I. INTRODUCTION

Cyber-physical systems (CPS) can be referred to as contemporary systems with assimilated computational and physical capabilities that can communicate with humans in new modalities [1]. These systems have an enormous impact on various domains like environmental monitoring, intelligent transportation, production system, smart grid, smart home, smart city, and the smart healthcare system. All these domains are dependent on the network as they require remote data transfer for sending data from sensors to actuators via a control center. Communication in the wide-open network makes the system vulnerable and creates a humongous attack space for adversaries [2].

The Internet of things (IoT), one of the essential sub-domain of CPS, has taken a significant technological advancement to a whole new level where data is the main force. IoT has opened up a new dimension in conjunction with actuators, electronics, sensors, software, and connectivity to enhance connection, collection, and data exchange. In spite of wide acceptability, almost 80% IoT devices are vulnerable to a wide range of cyber attacks [3]. They are susceptible to various kinds of attacks like man-in-the-middle, data and identity theft, distributed denial of service (DDoS), device hijacking etc. To protect safety critical systems from the intruders, robust security measures must be taken into consideration to detect all kinds of known and unknown attacks [4]–[6]. Intrusion Detection Systems (IDSs), responsible for inspecting network traffic and

system data for malicious activities and issuing alerts, are the first and foremost part of the defense strategy in the CPS domain. Having the proper knowledge of exact place and time where specific abnormalities are creating hazards in the system helps to mitigate the impacts by taking appropriate actions, and thus intrusion prevention systems come into the picture. The intrusion prevention system works simultaneously with an intrusion detection system to prevent the attacker from doing any harm to the system.

Machine learning-based IDS can detect abnormalities in the system with substantial accuracy [7]. Though the emerging IT trends in CPS such as Industry 4.0, IoT, big data, and cloud computing are adding more momentum, they are introducing more vulnerabilities as well [8]. Besides, novel architectural compositions are adding complexity to the model due to unknown emergent behavior [9]. Individual IDS needs to be implemented to observe their interaction with this complex system, but insufficient data are limiting the model training. Moreover, most of these available datasets are imbalanced where different types of attack data are not available on a large scale compared to the normal data. Thus, to handle these limitations, we develop a generative adversarial network (GAN)-assisted IDS that can mitigate the dataset related limitations for all these emerging technologies.

Our contribution to the paper can be summarized in the following points:

- We model an IDS using artificial neural network (ANN) that can be trained on any database with high accuracy.
- We model an emerging CPS security dataset where standalone IDS is unable to predict with high accuracy.
- Most importantly, we propose a new security framework, G-IDS, where GAN generates more training data to solve the imbalanced and missing data problems. Evaluating the performance of standalone IDS (S-IDS) and G-IDS, we find that G-IDS outperforms S-IDS in different ways.
- We evaluate our proposed framework in a widespread network intrusion detection dataset KDD'99.

Organization: The rest of the paper is organized as follows: We add sufficient background information, and our motivation towards this research focus in Section II. The related works are discussed in Section III. We introduce our proposed G-IDS framework in Section IV. Section V discusses the technical details of the framework, along with the complete analysis of our algorithm. In Section VI, we explain the evaluation setup and dataset along with the data pre-processing and
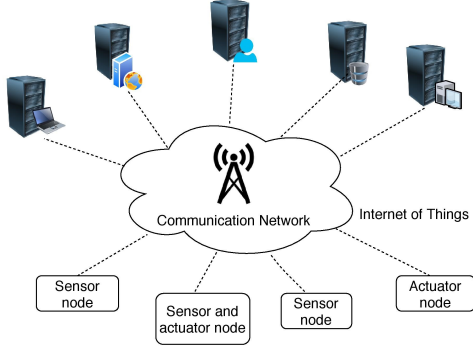
Fig. 1. Cyber-Physical System

data modeling steps. The empirical analysis and finding are formulated in Section VII. At last, we conclude the paper in Section VIII.

II. Background and Motivation

The term CPS refers to a new generation of systems with integrated computational and physical capabilities that can interact with human through state-of-the-art modalities. However, the seamless connections of IoT in CPS realm are opening the gates for the cyber attackers to launch different types of attack. To defend a CPS from being compromised, IDS is widely used. In this section we briefly discuss security aspects of CPS and the importance of IDS. We also give a short overview of GAN that we apply in our proposed solution. Finally, we discuss the motivation of this research.

A. Security Challenges in CPS

CPS is an amalgamation of computation, networking, and physical processes. IoT is another form of CPS which is a realm of connecting devices with sensing, computational, and actuating power. The emerging technologies, i.e., real-time data processing, machine learning, big data analytics, embedded systems, are shaping the IoT infrastructure nowadays.

Moreover, each day, lots of sophisticated and novel applications are being implemented. According to the statistics, in 2020, there will be 20.4 billion connected IoT devices, and by 2026 it will be around three trillion [10]. The recent invention of 5G technology can take the internet speed to a whole new level. Although the emerging CPS possess a humongous amount of data, the security and privacy of these complex systems face a significant amount of challenges due to non-homogeneous nature with other known systems and lack of related data and experimentation [11], [12].

CPS are adding new vulnerabilities due to: 1) replacement of traditional electro-mechanical relay-based controller by sophisticated software interaction-based microprocessor and operating systems, which enlarges the potential threat spaces, 2) growth of connected open-network consisting sensitive components, 3) public availability of internal system design and protocols, 4) radical increase in the number of functionalities, operations, and events, and 5) drastic surge of the technologically skilled cyber-criminals [13]–[15].

B. Intrusion Detection Systems in CPS

Along with the positive impacts of CPS in our lives, the risk of security breaches enhances parallelly. Moreover, IoT is being used in most of the critical infrastructures, which may cause a huge loss if an attacker gets control over them. An enormous amount of data is being processed and sent through the communication to keep the synchronization among all the equipment. A slight alteration in the data can cause severe destruction in the whole system. An IDS is a device or software application that monitors a network or system for harmful activities and rule violations. Thus, along with the device-level security, a dedicated IDS can ensure the high-level defense to the whole system.

Different types of IDS are widely implemented in the CPS domain. Data mining, machine learning, rule-based model, statistical modeling, protocol model, signal processing model, etc. are used in CPS [16]. Among them, anomaly-based IDS is mostly used. Anomaly-based IDS learns the distribution of data in the training process. If the new coming data is not from the same distribution, it is detected as an anomaly. Whenever the system has new training data, it trains the IDS model and keeps updated with the novel attacks. Thus, the defense mechanism of anomaly-based IDS is a dynamic process and needs more data for each label to train its model correctly [17].

Other than the anomaly-based model, various types of intrusion detection techniques are also used in the defense mechanism. Different machine learning and deep learning-based algorithms are showing high detection accuracy [18]–[21]. For classification type IDS, supervised learning is widely used, and unsupervised learning is usually used for clustering based IDS [22]. Besides, different data mining techniques are also used to extract important features from the bulk amount of data [23]. Statistical analysis also shows an efficient performance in the anomaly detector from the historic data [24]. Signal processing is also used to observe the pattern of the network and detect the abnormality [25]

C. Generative Adversarial Networks (GAN)

GAN was invented by Goodfellow et. al in 2014 [26]. It is one of the most powerful and promising tool in deep learning. GAN estimates a generative model though adversarial approach. It consists of two independent models: generator (G) and discriminator (D). The generative model G estimates the data distribution $p(g)$ over real data space x. Considering an input noise variable $p(z)$, the goal of G is to generate new adversarial sample $G(z)$ that comes from the same distribution of x. One the other hand, the discriminator model D returns the probability $D(x)$, that the given sample x is from real data set rather generated by G. The ultimate goal of G is to maximize the probability that D would mistakenly predict generated data as real one and for D the goal is to do the opposite. Thus, G and D play a two player minmax game and at the end they reach to an unique solution. The value function $V(G, D)$ is defined as follows:

$$\min_G \max_D V(D, G) =$$

$$\mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z} \sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

Thus, to achieve the optimum solution, these two game participants need to continually optimize themselves to improve their ability to reach the Nash equilibrium [27].

D. Research Motivation

The emerging CPS domains are going to reshape the world through technological advancement. Due to the uniqueness of each CPS domain, the attack patterns and threats also differ from each other. Most importantly, an emerging CPS always lacks such types of database to train the IDS to detect the probable attacks. Even if there is a database, the number of samples of each type of attack may not be enough for the IDS to get trained perfectly and detect with higher accuracy. Moreover, IDS is facing novel zero-day attacks more often as the cyber attackers are much more sophisticated nowadays and launch new types of cyberattacks regularly.

Thus, IDS can not be trained with high accuracy unless sufficient data is available. Besides, the size of normal data points is too large compared to the abnormal/attack data, which creates a data imbalance problem during the training. Moreover, even if there is a significant amount of data available, the distribution of the data space may not be apparent to IDS due to some missing data. On the other hand, GAN is a promising deep learning tool with the capability of learning the distribution of provided data and generates new similar samples.

Thus, to solve all the aforementioned problems, a GAN assisted IDS framework can be used to generate new data for all the labels that need improvements in prediction. Moreover, a comprehensive framework is also needed to deal with the imbalanced and missing data-related issues of the emerging CPS technologies.

III. RELATED WORK

There is a recent trend in the scientific community to study GAN's potential application in the CPS domains. GAN has promising applications in the security perspective of a CPS. Whereas some authors implemented GAN to attack, some utilized it to make the system robust.

Chhetri et al. proposed a conditional GAN based model to observe important security requirements by analyzing the relations between the cyber and physical domains in a CPS [28]. Yin et al. proposed a framework based on GAN along with the Botnet detection model, which improves the performance of the detection mechanism of a most formidable attack retaining the primary characteristics of the original detection model [29]. Seo et al. proposed a GAN-based model to reduce false positive rate in vehicle networks for enhancing driver's safety [30]. Some research works have attempted to find vulnerabilities of the systems using GAN. Li et al. used LSTM-RNN in GAN to capture the distribution of the multivariate time series of the sensors and actuators to detect abnormal working conditions for a sophisticated six-stage Secure Water Treatment (SWaT) system [31]. Usama et al. demonstrated that ML models are vulnerable to adversarial perturbation in the network traffic [32]. They mainly focused on deceiving machine learning-based IDS using the GAN-based adversarial model. Huang et al. proposed IDSGAN, which leverages a generator to transform original malicious traffic into adversarial malicious traffic examples [33].

On the other hand, GAN based anomaly detection model is gaining popularity. Ferdowsi et al. proposed a distributed adversarial network for providing a fully decentralized IDS for the IoT realm to detect anomalies, which is practical to conceal the user's sensitive data [34]. Belenko et al. explored the security breaches of a large scale CPS using generative adversarial ANNs [35]. GAN based models are also poplar to deceive detection models. Hu et al. proposed a GAN based framework named MalGAN to generate adversarial malware samples that can bypass any black-box machine learning-based detection models [36].

All of these works represents the application of GAN in CPS security. However, none of these works deals with imbalanced and limited data for emerging CPS technologies. Unlike the works mentioned above, we propose a comprehensive framework to synthesize training data that can improve the performance of IDS in cyber attack detection. Even though we evaluated our framework on the network intrusion detection dataset with ANN-based IDS, it is compatible with any IDS and different sensors and network datasets.

IV. PROPOSED G-IDS FRAMEWORK

We divide our proposed G-IDS framework into four different segments: 1. Database module 2. Intrusion detection system module 3. Controller module, and 4. Synthesizer module. Fig. 2 shows the block diagram of our proposed framework. At first, the database module collects real-world intrusion detection data from its data collector. Database module may also get synthesized data from the generator of the data synthesizer module. All of these data are accumulated parallelly and continuously and stored in the database with different flags to distinguish data sources. The synthesized data are sub-categorized into *pending* and *synthetic* classes. *Pending* data are uncommitted data that can be either accepted or deleted depending on the future decision of the controller module. On the other hand, data flagged as *synthetic* are already verified and stay permanently in the database. We consider a hybrid database that contains only *original* and *synthetic* data samples.

An ML-based model in the intrusion detection systems module gets trained twice on the data available in the database. The first one is trained on the hybrid dataset only, while the other model gets trained on both hybrid and pending datasets. The controller evaluates the performances of these two models of the IDS. Based on the evaluation, the controller decides to reject/accept the pending samples in the database. The pending samples are accepted and appended to the database as *synthetic* data if the model shows a better result after getting trained on them. Otherwise, the *pending* samples are rejected and removed from the database.

After that, controller module measures the detection rate for each class and makes a list of those for which the performance of the IDS falls below a certain threshold. Data samples for the weakly detected classes are sent to the data synthesizer module to generate more samples. Data Synthesizer module consists of a GAN, which generates new samples by learning the input data distribution. After generating potential samples, Data Synthesizer module feeds them in the database module with a *pending* flag. The overall process is repeated until the

Fig. 2. Proposed framework of G-IDS

label-wise performance of the IDS model becomes satisfactory to the controller. The Algorithm 1 provides a comprehensive summary of the complete framework.

V. G-IDS TECHNICAL DETAILS

This section provides the technical description of the four modules of the G-IDS framework.

A. Database Module (DM)

As shown in Fig. 2, DM consists of two important elements: data collector and database.

1) Data Collector (DC): In the proposed framework, we consider that the IDS is a network intrusion detector. Thus, DC collects real-world network data by capturing network packets. The packets are processed before feeding into the database. A packet capture application (i.e., Wireshark) runs within DC to collect packets and extract features from them. The responsibilities of DC is categorized into two types:

- Data Collection: Data collection is the initial stage of the network-based IDS, as shown in Fig. 2. The communication between the sensors and the actuators of a CPS generates a vast amount of network data, which mostly contains normal/good data. However, an adversary can try to compromise the sensors or the communication channels to launch a cyber attack. Thus, to make the system robust against any kind of cyber attack, DC works along with a firewall to label those packets. After collection, DC does some pre-determined prepossessing on them to make them feedable to ML models.

- Data Preprocessing: As the network packet analyzer collects a massive chunk of data from the network, some prepossessing is mandatory to gather information from those packets. In our proposed model we consider the following tasks:

  Encoding: Label encoding is the technique of encoding categorical values in the dataset. It transforms each category of a particular feature with a value between 0 to n-1, where n is the number of distinct categories of that feature.

  Feature Scaling: Feature scaling is a method used to normalize the range of features. A dataset containing features that are highly varying in magnitudes, units, and range can become a critical problem for various ML algorithms that use Euclidean distance between two data points in their computations. To overcome this problem, we apply standard scaling to normalize the features.

  Feature Extraction: Feature extraction (also known as feature reduction) remodels the high-dimensional space to fewer dimensions, where the transformation could be linear or non-linear. It helps to eliminate the redundant variables and makes the model a simpler one.

2) Database (DB): Database stores the collected data from DC and DSM, along with the labels and corresponding flags. Each label represents the type of data (i.e., normal/attack). In the case of attack data, the label shows the type of attack. The flag bit for each sample indicates the status for that particular data as explained below:

- ***Original:*** Data collected from DC are flagged as *original*.

Algorithm 1: Proposed Algorithm of G-IDS

---

initialization $Database$ = Data collected from DC;

for *true* **do**

    • Train IDS model on $HybridDatabase$

    • Update performance metrics, PM_H

    • Controller lists all labels with $PM_H < PM_{TH}$ and sends them to the DSM

    • Create $List\_of\_weak\_labels$ for $PM_H < PM_{TH}$

    for *each label l in $List\_of\_weak\_labels$* **do**

        for *number of epoch* **do**

            for *k steps* **do**

                • Create a temporary copy of $HybridDatabase$ replacing label l with 1 and the rest with 0.

                • Sample m noise $z^{(1)}, z^{(2)}, .., z^{(m)}$ from noise distribution $p_g(z)$

                • Sample batch of m examples $x^{(1)}, x^{(2)}, .., x^{(m)}$ from $HybridDatabase$

                • Update the discriminator model (D) by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{t=1}^{m} [logD(x^{(t)}) + log(1 - D(G(z^{(t)})))]$$

            end

            • Sample batch of m noise $z^{(1)}, z^{(2)}, ...z^{(m)}$ from noise distribution $p_g(z)$

            • Update the generator model by descending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{t=1}^{m} log(1 - D(G(z^{(t)})))]$$

        end

            • Generate p samples for label l and add them to $Database$ with flag p

    end

    • Train IDS on $Database$ with pending samples

    • Update performance metrics, PM_P

    if $PM_P > PM_H$ **then**

        • Accept pending (if any) samples for each label

        • Updates flags from *pending* to *synthetic*

        • add them to $HybridDatabase$

    else

        • Remove pending samples from $Database$

    end

end

• **Pending:** As mentioned in Section IV, DSM generates more data to improve the performance of IDS. However, due to the uncertainty of the GAN, every batch of the generated data does not guarantee the improvement of the performance of IDS. Hence, the generated data are flagged as *pending* in the first place, and the controller does the further inspection.

• **Synthetic:** If the *pending* data contribute to the improvement in the performance of IDS, the controller changes their flags from *pending* to *synthetic* and stores them permanently in the database. Whenever, the *pending* data convert to *synthetic*, they become the part of the hybrid dataset.

Table I demonstrates an example database for some samples along with the labels and flags before encoding.

TABLE I
AN EXAMPLE DATABASE

Duration	Protocol Type	Service	...	Label	Flag
0	tcp	http	...	normal	original
0	icmp	ecr_i	...	smurf	original
18848	tcp	telnet	...	normal	pending
199	tcp	telnet	...	spy	synthetic
⋮	⋮	⋮	⋮	⋮	⋮
18848	tcp	telnet	...	normal	synthetic

B. Intrusion Detection System Module (IDSM)

The core part of IDSM is an ML-based IDS which, in this case, is a multi-layer ANN-based model. It consists of 4 layers, where the input and output layers have 20 and 10 nodes, respectively, and both of the hidden layers consist of 50 nodes. IDS is trained twice to calculate the following two performance metrics which are useful to evaluate the synthesized data-

Performance metric without pending data (PM_H)**:** After collecting data from the data collection module, IDS trains the model on the hybrid database containing only original and synthetic data. Then, IDS evaluates the label-wise performance metrics and stores in PM_H, which is the current best possible performance of the module.

Performance metric with pending data (PM_P)**:** After obtaining synthesized data with the *pending* flag, IDS re-train the model on the hybrid database, including the pending data, to evaluate the additional contribution of the pending data points. The performance metric is stored in PM_P. Both of these evaluation metrics are available to the controller module based on which it takes a further decision on pending data.

C. Controller Module (CM)

CM performs two important tasks in parallel.

Sending data synthesize request to the DSM: Firstly, the controller analyzes PM_H and compare it with the PM_{TH}, which is the minimum threshold for the performance metric. Any class with an individual performance score below the PM_{TH} is considered as the weak class, which needs the aid of DSM in the generation of new samples to improve the detection rate. Each time the CM sends a request for one label with the hybrid database and continues for all the remaining weak ones.

Evaluating pending data requests from DSM: CM analyzes the evaluation metric PM_P and PM_H to update the database. Comparing these two metrics, CM suggests the database to remove the *pending* samples for a specific label if the performance degrades with them. On the other hand, if the performance improves, CM recommends accepting the pending data by updating the *pending* flags to *synthetic*.

D. Data Synthesizer Module (DSM)

The core part of the DSM is a GAN-based model. Fig. 2 shows the DSM, which consists of two ML models: 1. generator (G) and 2. discriminator (D). The controller selects one class and sends the whole database to the DSM, mentioning 1 as the label of that specific class and 0 for the rest of the classes. Thus, for DSM, it becomes a binary classification model and generates more samples after successful training. We model both the generator and discriminator using four-layer ANN. Each of the hidden layers has 50 nodes. The input and output layers of discriminator have 20 and 1 nodes, respectively. Besides, the generator takes a noise (*latent space*) as an input and produces artificial samples. Thus the input and output layers of the generator have 5 and 20 nodes, respectively. Once the GAN training is complete, it generates new samples with higher accuracy and asks the DB to add them with *pending* flags.

VI. EVALUATION METHODOLOGY

In this section, we introduce the experimental setup and necessary evaluation metrics to assess our proposed G-IDS framework's performance.

A. Dataset

The intrusion detection dataset of emerging CPS carries a lot of benign data comparing to attack data. Thus, we resemble the empirical data-limitation scenario by modifying KDD'99

TABLE II
CLASS LABELS AND THE NUMBER OF SAMPLES APPEARING IN THE DATASET

Label	Attack	Original Data	Class
1	Back	2003	DoS
2	Ipsweep probe	1047	PA
3	Perl u2r	1070	UTR
4	Normal	9707	Normal
5	Neptune dos	1070	DoS
6	Smurf dos	840	Do S
7	Spy r2l	1389	R2L
8	Warezclient r2l	779	R2L
9	Warezmaster r2l	820	R2L
10	Phf	550	R2L

dataset [37] and use it to evaluate our model by analyzing different attack detection.

The dataset contains ten different labels, which can be categorized into a. Denial of Service Attack (DoS), b. User to Root(UTR) Attack, c. Remote to Local Attack (R2L), and d. Probing Attack (PA), etc [38]. Table III demonstrates the number of supporting data for all attack labels. The characteristics of our dataset can be formalized as follows:

1) Multiclass: Our dataset in consideration is a multiclass dataset that contains nine different types of attacks from four different classes and one normal label, as shown in Table II.

2) Sparse: Sparsity is a process of determining the number of missing values present in the dataset. A variable with sparse data is one in which a relatively high percentage of the variable cells do not contain actual data. Data sparsity is widely considered as a key cause of unsatisfactory classification accuracy. Our dataset is purely cleaned one with no missing attributes and records. However, sparsity calculation using the following equation returns 70% sparsity, which is basically produced due to one-hot encoded features. Data sparsity can be calculated using:

$$Sparsity = 1 - \frac{non\_zero\_count(X)}{number\_of\_elements(X)}$$

where X denotes the dataset.

3) Imbalanced: Imbalanced data hinders the performance of a classification problem. The number of observations per class is not equally distributed, and there is an enormous amount of observations for a particular class (majority class) and significantly less number of samples for one or more other classes (minority classes). Our dataset is an example of an imbalanced dataset as the number of benign examples dominate the number of attack examples.

B. Data Preprocessing

Before training the model, some pre-processing is required to be performed. After a complete analysis, we perform the following techniques for data pre-processing.

1) Encoding: To deal with three categorical input features (namely, protocol_type, service, and flag) and one output feature, we encode the corresponding data by applying the label encoding technique.

2) Feature Scaling: Feature scaling is an essential step to deal with local optima, and skewness towards particular features. It also facilitates the ML-based IDS with faster training. We apply standard scaling, which replaces the values by their Z-scores. Z-score of a data point represents the measure of deviation from the mean value. z-score, z_i, is calculated using the following equation:

$$z_i = \frac{x_i - x_{mean}}{x_{std}}$$

where x_i represents the value of each feature, x_{mean} is the average value of the feature, and x_{std} denotes the standard deviation.

3) Feature Extraction: To reduce the number of features, we utilize principal component analysis (PCA)-based feature extraction method. PCA is an unsupervised, non-parametric statistical procedure that computes a new set of variables ("principal components") and expresses the data in terms of these new variables and uses an orthogonal transformation to transform a set of observations of possibly correlated variables [39]. Mathematically, PCA is calculated via linear algebra functions called eigendecomposition, a process of factorization of a matrix into a canonical form.

For selecting the number of principal components, we use the explained variance ratio, representing the amount of information retained after applying PCA. The total variance is the sum of variances of all individual principal components. We use PCA on all the original 42 features and shrink the dimension into 20 elements, keeping the total explained variance 91.14%.

C. Data Modeling

To mimic an emerging CPS, we have taken only 5000 samples for training, and the rest of the data are used for testing purposes. As the dataset in consideration is imbalanced, the training data set loses some labels if the dataset is randomly split. To overcome this problem, we randomly select data from all available classes.

D. Evaluation

We have considered the following metrics for evaluating the performance of our model.

1) Precision: Precision is the ratio of correctly predicted positive observations to the total positive observations.

2) Recall: Recall is the ratio of correctly predicted positive observations to the total actual positive observations.

3) F1 score: F1 score is a weighted average of precision and recall that takes both false positives and false negatives into account. In the case of unequal distribution of false Negative and false Positive values, F1 score can infer better comprehension of the model's performance.

4) Confusion Matrix: Confusion matrix is a table that intends to present the count of correct and incorrect predictions made by each class. It gives a clear insight into the prediction and shows the classification/misclassification type (e.g., True Positive, True Negative, False Positive, False Negative). Thus, the performance of an IDS can be analyzed using these four important key factors as shown in Table IV.

TABLE IV
EVALUATION FACTORS

Indicator	Actual Label	Prediction
True Positive, TP	Attack	Attack
False Positive , FP	Normal	Attack
True Negative, TN	Normal	Normal
False Negative, FN	Attack	Normal

Thus, from these four terms performance of an IDS can be easily calculated using the following equations [40]:

$$Precision, \quad P = \frac{TP}{(TP + FP)}$$

$$Recall, \quad R = \frac{TP}{(TP + FN)}$$

$$F1\ score = \frac{2 * P * R}{(P + R)}$$

As our dataset contains a lot of benign data, evaluating the system with a weighted average F1 score does not provide a clear picture. We consider the macro-average F1 score as it computes the metric independently for each label and then takes the average to treat all labels equally.

VII. EVALUATION RESULTS AND DISCUSSION

The experiments are conducted on Dell Precision 7920 Tower workstation with Intel Xeon Silver 4110 CPU @3.0GHz, 32 GB memory, 4 GB NVIDIA Quadro P1000 GPU. The performance of G-IDS is evaluated in two perspectives, contrasting with the performance of a standalone IDS (S-IDS).

A. Evaluation based on the Size of Training Dataset

To investigate the performance of S-IDS and G-IDS, we split the entire training dataset into five parts. We take 20%, 40%, 60%, 80%, and 100% of the training data to observe GAN's contribution with a growing dataset.

1) Label-wise Evaluation: Firstly, we train S-IDS model without the assistance of GAN, which gives good results in the prediction for a few of the specific labels. However, S-IDS struggles to predict a few of the classes accurately as the number of training samples for those particular classes is not sufficient. As shown in Fig. 3(a), the F1 score of S-IDS for label 5 is approximately 0.70 while considering 20% of data as the training data, and it increases gradually up-to 0.78 with the increasing training dataset. However, training G-IDS with the GAN generated data along with the original data boosts the performance of G-IDS drastically. As shown in the figure, the F1 score of G-IDS reaches up to 0.90 for 20% of data as training data and becomes 0.96 for 40%. It is clear from the figure that G-IDS performs much better than S-IDS in attack detection. However, the fall of the F1 score for G-IDS at 60% is not surprising as GAN's performance is highly random as it works by taking random noises as input. Similar improvements are also observed for label 6 and 9, shown in Fig. 3(b) and Fig. 3(c), respectively. Though there are improvements in the predictions for most of the labels, we only show these three labels as an example.
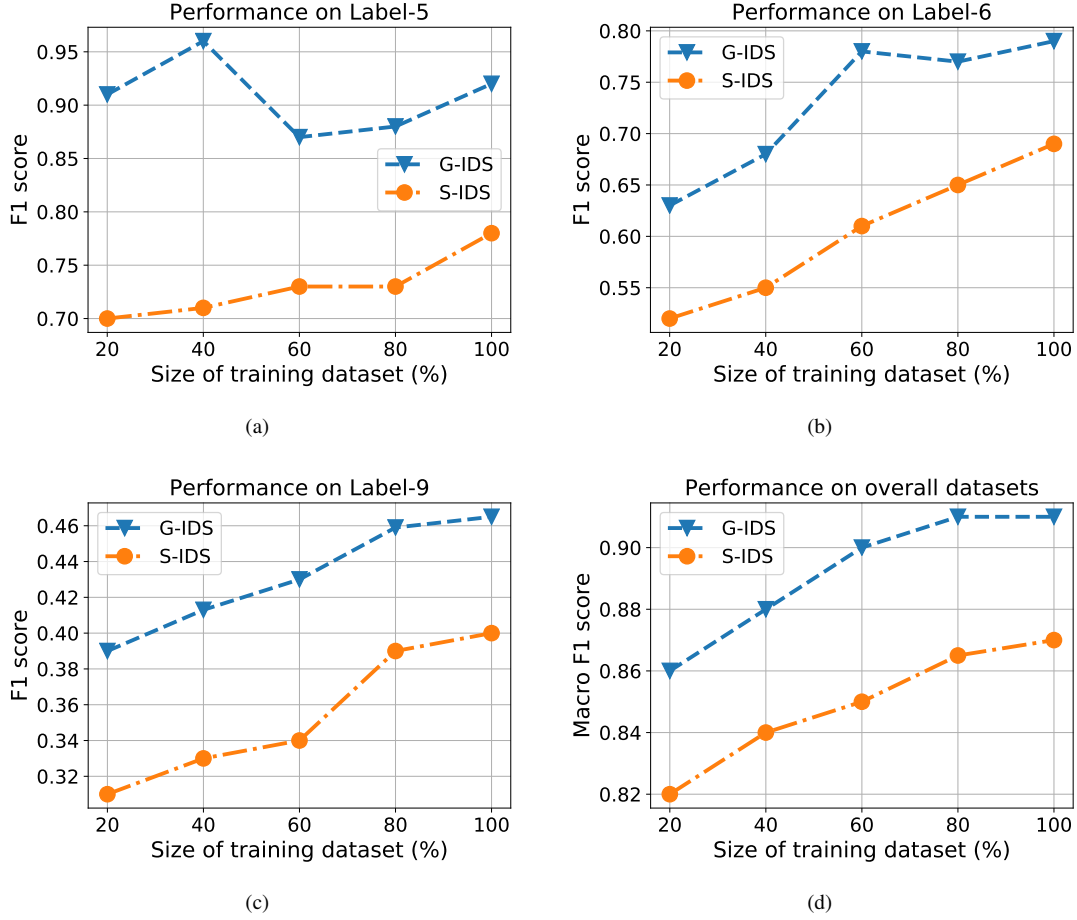
Fig. 3. Performance evaluation of G-IDS and S-IDS in the prediction (F1 score) with respect to the size of training dataset for (a) label 5, (b) label 6, (c) label 9, and (d) overall dataset.

2) Overall Evaluation: In this part, we contrast the performance S-IDS and G-IDS on overall prediction for all the classes. As mentioned in the previous section, we use the macro F1 score as the metric, as it gives better insight into the model's performance on distinct labels. Fig. 3(d) represents the macro F1 score of S-IDS and G-IDS on the same test data while trained on different sizes of training data. We see that the macro F1 score of S-IDS improves from 0.82 to 0.87, as training data increases from 20% to 100%. On the other hand, G-IDS's performance improves by approximately 4% as GAN-generated synthetic data are also considered in training. For the full training dataset, while S-IDS achieves a maximum macro F1 score of 0.87, G-IDS can take it at least up-to 0.91.

Table III summarizes the contrast of S-IDS and G-IDS performances when both are trained on 40% of the original data. The first three columns represent the performance metrics of S-IDS for each class. If the F1 score of any class is less than the threshold, which we consider as 0.98 in this case, the controller asks the data synthesizer module to generate more synthetic data for that specific class to improve the performance of IDS.

Initially, S-IDS has an F1 score of 0.85, 0.71, 0.55, and 0.33 for the labels of 2, 5, 6, and 9, respectively. These labels are

considered as *weak label* and are sent to the DSM to generate more training examples. GAN is trained on each of these labels and then generates 25% new samples for each of these labels. Once verified by the controller, the newly generated synthetic samples are added to the existing hybrid dataset. The updated hybrid dataset is used to train the IDS again, and the whole process is continued. The second part of Table III shows the performance of G-IDS, which is trained on the hybrid dataset. From the results, we see that the detection rates of most of the classes are improved. To be more specific, the F1 score of labels 2, 5, 6, and 9 increases up-to 0.93, 0.96, 0.68, and 0.41, respectively. Even though, by nature, GAN possesses the uncertainty of generating noisy unexpected data for one label that may overlap with the distribution of other labels, the controller module declines such kind of bad data and keeps the performance of G-IDS improving.

B. Evaluation of Improvement in the Training Stability

As well as improving the final detection rate for different attacks, GAN also enhances the stability of the IDS's training process. Fig. 4 shows the performance of the model during training for four random targets. To present the performance of our proposed framework, we plot the F1 score for the test

TABLE III
PERFORMANCE METRICS OF S-IDS AND G-IDS WHEN TRAINED WITH 40% OF ORIGINAL DATASET

Labels	S-IDS			G-IDS			Support
	Precision	Recall	F1	Precision	Recall	F1	
1	0.99	0.99	0.99	0.98	0.99	0.99	2003
2	*0.83*	*0.90*	*0.85*	*0.90*	*0.95*	*0.93*	*1047*
3	1.00	1.00	1.00	1.00	1.00	1.00	1070
4	1.00	0.98	0.99	1.00	0.99	0.99	9707
5	*0.55*	*0.98*	*0.71*	*0.93*	*0.99*	*0.96*	*840*
6	*0.38*	*0.98*	*0.55*	*0.52*	*0.97*	*0.68*	*1389*
7	1.00	1.00	1.00	1.00	1.00	1.00	2805
8	1.00	1.00	1.00	1.00	1.00	1.00	779
9	*0.20*	*0.98*	*0.33*	*0.27*	*0.90*	*0.41*	*820*
10	0.98	0.99	0.99	0.98	0.99	0.99	550
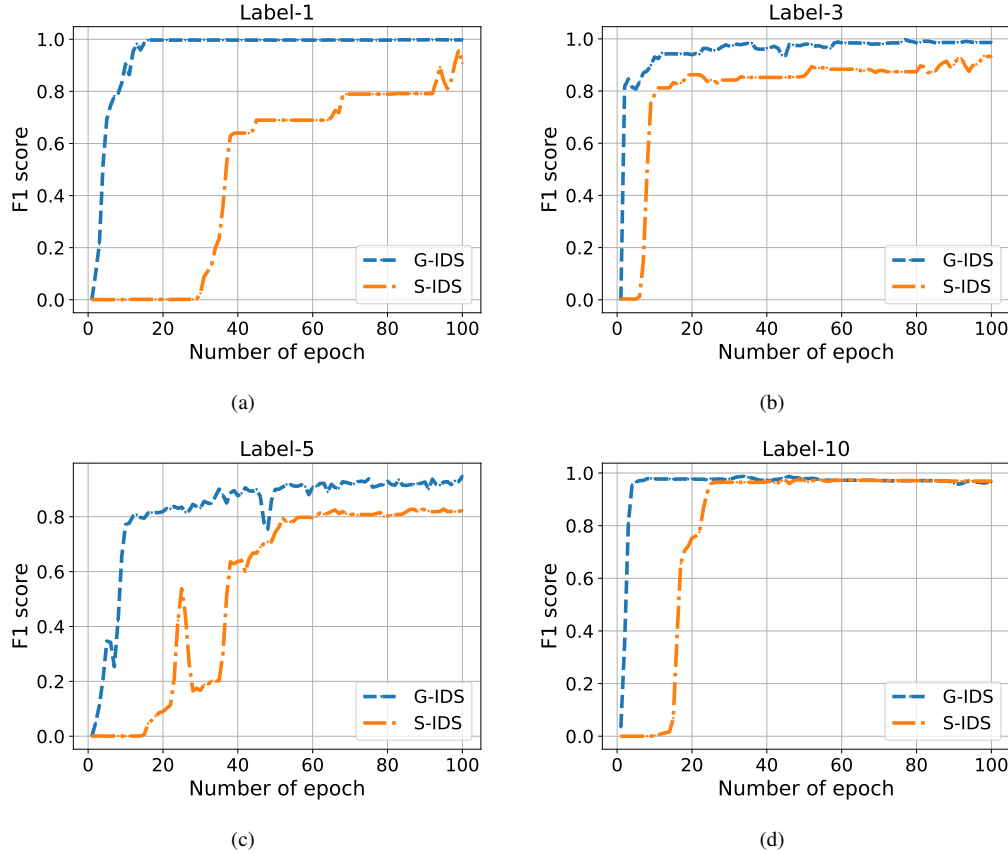
Fig. 4. Stability analysis of S-IDS and G-IDS during training process for (a) label 1, (b) label 3, (c) label 5 and, (d) label 10.

dataset for the increasing epoch. In the figure, we only show the first 100 epochs, which is sufficient to represent the contrast of the performance. From the figure, it is clear that S-IDS takes much more epochs to train the model and settle the F1 score to the final value. For S-IDS, most of the labels need approximately 50 to 100 epochs or even more to settle.

On the other hand, G-IDS does a pretty good job in the training stability. Almost all of the targets become stable within approximately 25 epochs. Thus the training process of G-IDS is much more stable than S-IDS. As the GAN-generated samples fill the gaps in the data distribution, it becomes easier for the

IDS to learn the distribution of the training data and get settled. Thus, in summary, G-IDS improves performance by balancing the imbalanced dataset as well as generating missing data.

VIII. CONCLUSION AND FUTURE WORK

GAN is an influential tool in the deep learning area. IDS is also another fundamental agent for the CPS domain. In this paper, we propose a GAN supported IDS to unite them, which performs better than standalone IDS for an imbalance dataset or in any emerging field of cyber-physical systems where very few amounts of data are available for model training. We implement our model in a benchmark dataset, NSL KDD'99. Experimental

analysis shows that our proposed G-IDS framework predicts with better accuracy than independent IDS, even after being trained with a small original dataset in the beginning. The centralized, computationally expensive and time consuming nature of G-IDS framework require further investigation. In future work, we aim to focus on creating a dynamic, efficient, and lightweight decentralized algorithm to implement it in the edge devices of the IoT realm.

REFERENCES

[1] Edward A Lee. Cyber physical systems: Design challenges. In *2008 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC)*, pages 363–369. IEEE, 2008.

[2] Hui Suo, Jiafu Wan, Caifeng Zou, and Jianqi Liu. Security in the internet of things: a review. In *2012 international conference on computer science and electronics engineering*, volume 3, pages 648–651. IEEE, 2012.

[3] Smart home: Threats and countermeasures. https://www.rambus.com/iot/smart-home/, 2019. Accessed: 2020-01-29.

[4] Constantinos Kolias, Georgios Kambourakis, Angelos Stavrou, and Jeffrey Voas. Ddos in the iot: Mirai and other botnets. *Computer*, 50(7):80–84, 2017.

[5] M Ashiqur Rahman, Md Hasan Shahriar, and Rahat Masum. False data injection attacks against contingency analysis in power grids: poster. In *Proceedings of the 12th Conference on Security and Privacy in Wireless and Mobile Networks*, pages 343–344, 2019.

[6] Amit Kumar Sikder, Leonardo Babun, Hidayet Aksu, and A Selcuk Uluagac. Aegis: a context-aware security framework for smart home systems. In *Proceedings of the 35th Annual Computer Security Applications Conference*, pages 28–41, 2019.

[7] Arwa Aldweesh, Abdelouahid Derhab, and Ahmed Z Emam. Deep learning approaches for anomaly-based intrusion detection systems: A survey, taxonomy, and open issues. *Knowledge-Based Systems*, 189:105124, 2020.

[8] Jin Ho Kim. A review of cyber-physical system research relevant to the emerging it trends: industry 4.0, iot, big data, and cloud computing. *Journal of industrial integration and management*, 2(03):1750011, 2017.

[9] Martin Törngren and Paul T Grogan. How to deal with the complexity of future cyber-physical systems? *Designs*, 2(4):40, 2018.

[10] https://ipropertymanagement.com/iot-statistics, 2019. Last accessed 08 September 2019.

[11] Kim-Kwang Raymond Choo, Mehran Mozaffari Kermani, Reza Azarderakhsh, and Manimaran Govindarasu. Emerging embedded and cyber physical system security challenges and innovations. *IEEE Transactions on Dependable and Secure Computing*, 14(3):235–236, 2017.

[12] AKM Newaz, Amit Kumar Sikder, Mohammad Ashiqur Rahman, and A Selcuk Uluagac. A survey on security and privacy issues in modern healthcare systems: Attacks and defenses. *arXiv preprint arXiv:2005.07359*, 2020.

[13] Abdulmalik Humayed, Jingqiang Lin, Fengjun Li, and Bo Luo. Cyberphysical systems security—a survey. *IEEE Internet of Things Journal*, 4(6):1802–1831, 2017.

[14] Mohammad Ashiqur Rahman, Md Hasan Shahriar, Mohamadsaleh Jafari, and Rahat Masum. Novel attacks against contingency analysis in power grids. *arXiv preprint arXiv:1911.00928*, 2019.

[15] Afsheen Ahmed, Rabia Latif, Seemab Latif, Haider Abbas, and Farrukh Aslam Khan. Malicious insiders attack in iot based multi-cloud e-healthcare environment: a systematic literature review. *Multimedia Tools and Applications*, 77(17):21947–21965, 2018.

[16] Shahid Anwar, Jasni Mohamad Zain, Mohamad Fadli Zolkipli, Zakira Inayat, Suleman Khan, Bokolo Anthony, and Victor Chang. From intrusion detection to an intrusion response system: fundamentals, requirements, and future directions. *Algorithms*, 10(2):39, 2017.

[17] Habtamu Abie. Cognitive cybersecurity for cps-iot enabled healthcare ecosystems. In *2019 13th International Symposium on Medical Information and Communication Technology (ISMICT)*, pages 1–6. IEEE, 2019.

[18] Yajie Song, Bing Bu, and Li Zhu. A novel intrusion detection model using a fusion of network and device states for communication-based train control systems. *Electronics*, 9(1):181, 2020.

[19] Amit Kumar Sikder, Hidayet Aksu, and A Selcuk Uluagac. 6thsense: A context-aware sensor-based attack detector for smart devices. In *26th {USENIX} Security Symposium ({USENIX} Security 17)*, pages 397–414, 2017.

[20] Ruchi Makani and BVR Reddy. Taxonomy of machine leaning based anomaly detection and its suitability. *Procedia computer science*, 132:1842–1849, 2018.

[21] N Neha, S Priyanga, Suresh Seshan, R Senthilnathan, and VS Shankar Sriram. Sco-rnn: A behavioral-based intrusion detection approach for cyber physical attacks in scada systems. In *Inventive Communication and Computational Technologies*, pages 911–919. Springer, 2020.

[22] AKM Iqtidar Newaz, Amit Kumar Sikder, Mohammad Ashiqur Rahman, and A Selcuk Uluagac. Healthguard: A machine learning-based security framework for smart healthcare systems. In *2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS)*, pages 389–396. IEEE, 2019.

[23] Solane Duque and Mohd Nizam bin Omar. Using data mining algorithms for developing a model for intrusion detection system (ids). *Procedia Computer Science*, 61:46–51, 2015.

[24] Syed Obaid Amin, Muhammad Shoaib Siddiqui, Choong Seon Hong, and Sungwon Lee. Rides: Robust intrusion detection system for ip-based ubiquitous sensor networks. *Sensors*, 9(5):3447–3468, 2009.

[25] Florin Vancea and Codruţa Vancea. Some results on intrusion and anomaly detection using signal processing and near system. In *2015 38th International Conference on Telecommunications and Signal Processing (TSP)*, pages 113–116. IEEE, 2015.

[26] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[27] John Nash. Non-cooperative games. *Annals of mathematics*, pages 286–295, 1951.

[28] S. R. Chhetri, A. B. Lopez, J. Wan, and M. A. Al Faruque. Gansec: Generative adversarial network modeling for the security analysis of cyber-physical production systems. In *2019 Design, Automation Test in Europe Conference Exhibition (DATE)*, pages 770–775, 2019.

[29] Chuanlong Yin, Yuefei Zhu, Shengli Liu, Jinlong Fei, and Hetong Zhang. An enhancing framework for botnet detection using generative adversarial networks. In *2018 International Conference on Artificial Intelligence and Big Data (ICAIBD)*, pages 228–234. IEEE, 2018.

[30] Eunbi Seo, Hyun Min Song, and Huy Kang Kim. Gids: Gan based intrusion detection system for in-vehicle network. In *2018 16th Annual Conference on Privacy, Security and Trust (PST)*, pages 1–6. IEEE, 2018.

[31] Dan Li, Dacheng Chen, Jonathan Goh, and See-kiong Ng. Anomaly detection with generative adversarial networks for multivariate time series. *arXiv preprint arXiv:1809.04758*, 2018.

[32] Muhammad Usama, Muhammad Asim, Siddique Latif, Junaid Qadir, et al. Generative adversarial networks for launching and thwarting adversarial attacks on network intrusion detection systems. In *2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC)*, pages 78–83. IEEE, 2019.

[33] Zilong Lin, Yong Shi, and Zhi Xue. Idsgan: Generative adversarial networks for attack generation against intrusion detection. *arXiv preprint arXiv:1809.02077*, 2018.

[34] Aidin Ferdowsi and Walid Saad. Generative adversarial networks for distributed intrusion detection in the internet of things. *arXiv preprint arXiv:1906.00567*, 2019.

[35] Viacheslav Belenko, Valery Chernenko, Maxim Kalinin, and Vasiliy Krundyshev. Evaluation of gan applicability for intrusion detection in self-organizing networks of cyber physical systems. In *2018 International Russian Automation Conference (RusAutoCon)*, pages 1–7. IEEE, 2018.

[36] Weiwei Hu and Ying Tan. Generating adversarial malware examples for black-box attacks based on gan. *arXiv preprint arXiv:1702.05983*, 2017.

[37] Mahbod Tavallaee, Ebrahim Bagheri, Wei Lu, and Ali A Ghorbani. A detailed analysis of the kdd cup 99 data set. In *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, pages 1–6. IEEE, 2009.

[38] Atilla Özgür and Hamit Erdem. A review of kdd99 dataset usage in intrusion detection and machine learning between 2010 and 2015. *PeerJ Preprints*, 4:e1954v1, 2016.

[39] Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987.

[40] Ilhame El Farissi, Mohammed Saber, Sara Chadli, Mohamed Emharraf, and Mohammed Ghaouth Belkasmi. The analysis performance of an intrusion detection systems based on neural network. In *2016 4th IEEE International Colloquium on Information Science and Technology (CiSt)*, pages 145–151. IEEE, 2016.