# A Comparative Assessment of Unsupervised Deep Learning Models for Detecting GPS Spoofing Attacks on Unmanned Aerial Systems

Tala Talaei Khoei[1], Khair Al Shamaileh[2], Vijaya Kumar Devabhaktuni[3], and Naima Kaabouch[4]
[1]Khoury College of Computer Sciences, Roux Institute at Northeastern University, Portland, ME 04106, USA
[2]Electrical and Computer Engineering Department, Purdue University Northwest, Hammond, IN 46323, USA
[3]Electrical and Computer Engineering Department, Illinois State University, Normal, IL 61761 USA
[4]School of Electrical Engineering and Computer Science, University of North Dakota, Grand Forks 58201, ND, USA

2024 Integrated Communications, Navigation and Surveillance Conference (ICNS) | 979-8-3503-9309-5/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICNS60906.2024.10550633

***Abstract*— Unmanned Aerial Vehicles (UAV) are prone to cyber threats, including Global Positioning System (GPS) spoofing attacks. Several studies have been performed to detect and classify these attacks using machine learning and deep learning techniques. Although these studies provide satisfactory results, they deal with several limitations, including limited data samples, high costs of data annotations, and investigation of data patterns. Unsupervised learning models can address these limitations. Therefore, this paper compares the performance of four unsupervised deep learning models, namely Convolutional Auto-Encoder, Convolutional Restricted Boltzmann Machine, Deep Belief Neural Network, and Adversarial Neural Network in detecting GPS spoofing attacks on UAVs. The performance evaluation of these models was done in terms of Gap static, Calinski harabasz score, Silhouette Score, homogeneity, completeness, and V-measure. The results show that the Convolutional Auto-Encoder has the best performance results among the other unsupervised deep learning models.**

***Keywords*— Artificial neural network, deep learning, Global positioning system, machine learning, unsupervised learning, unmanned aerial systems.**

## I. INTRODUCTION

The Global Navigation Satellite System (GNSS) plays a crucial role in the positioning and navigation of Unmanned Aerial Vehicles (UAVs). Despite significant advancements in automation and control of UAVs, their security has been overlooked. UAVs are vulnerable to various cyber threats, including GPS Spoofing attacks, which can have a significant impact on the safety of people and infrastructure q [1, 2]. These attacks have been observed during the last decade, especially during conflicts in Ukraine, Russia, China, and Iraq, where malicious actors transmitted false Global Positioning Signals (GPS) signals to falsify the position, time, and velocity information. These signals are designed to mimic genuine satellite signals and can be difficult to detect depending on the attack sophistication [3].

In the realm of securing GPS signals, certain studies have centered around cryptography techniques that aim to encrypt GPS signals; while this approach may present a good security solution, it is not practical for civilian applications that require unencrypted GPS signals. Other studies were based on the assumption that signals arriving at a different angle from the GPS constellation are due to spoofing attempts; thus, several angle-of-arrival determination-based techniques have been proposed. These techniques require additional hardware at the antenna architecture level (antenna array, circular antenna design. etc.) [5]. As a result of these constraints, Artificial Intelligence-based approaches have been proposed to detect and classify these attacks on UAVs. Traditional and ensemble machine learning (ML) [reference] and Deep Learning (DL) [] models, such as support vector machine, decision tree, gradient boosting, random forest, bagging, and Naive Bayes, were used for the detection. Despite the fact that these models provide high performance, several issues need to be addressed. To begin, the field of study is still in its early stages, and studies in the literature indicate that detection and misdetection rates need to be improved. Moreover, various research in the literature address overfitting/underfitting difficulties, which result in erroneous predictions. Furthermore, current studies in the literature have largely concentrated on supervised models; nevertheless, these models need large, labeled datasets which is tedious and time consuming [6, 7].

Annotating datasets and performing data pre-processing steps are costly, resulting in increasing the computational complexity of the training process. In addition, supervised models do not discover hidden patterns of the given data. Furthermore, there are very limited datasets related to GPS spoofing. Therefore, to address these challenges, unsupervised models, are proposed in this paper. These models can detect patterns and relationships in data without the need for labels or classifications. This allows algorithms to learn from the data and make predictions based on the patterns they detect, resulting in low complexity and faster processes.

Thus, in this paper, we provide a comprehensive assessment of unsupervised deep learning models to detect GPS spoofing attacks. The advantage of using DL models over ML models is their ability to do feature engineering on their own, which leads to higher scalability, self-learning capability, and cost-effectiveness. Four unsupervised models, Convolutional Auto-Encoder (CAE), Convolutional Restricted Boltzmann Machine (CRBM), Deep Belief Neural Network (DBNN), and Generative adversarial neural network (GAN) are chosen for this study. These models are evaluated in terms of Gap static, Calinski harabasz score, Silhouette Score, homogeneity, completeness, and V-measure.

The remainder of this paper is organized as follows: Section II discusses the methodology used in this study. Section III indicates the results of this study. The conclusion and future work are outlined in Section IV.

## II. Related Works

Several studies have been performed on the detection, classification, and mitigation of GPS spoofing attacks on UAVs, as illustrated on Table I. In general, the proposed methods can be classified into three categories, namely UAV- characteristics, signal processing, and AI-based techniques. For example, the authors of [8] proposed IMU measurements, such as angle, velocity, and acceleration along with GPS data, longitude and latitude, to detect attacks on UAVs. In [9], the authors developed another UAV-characteristic based approach to detect GPS spoofing attacks on UAVs. In this approach, the authors mainly used Gyroscope Measurement and GPS data to detect attacks and avoid hijacking scenarios. In [10], the authors proposed another UAV-characteristic based approach that is highly dependent on the error calculated from IMU and GPS receiver.

Several other studies have focused on vision-based approaches to detect, classify, and mitigate GPS spoofing attacks on UAVs. In [11], the authors proposed a vision-based approach based on the vision sensor, UAV's sensor, IMU, and monocular camera. In another study [12], the authors used Visual Odometry by employing UAV's camera to obtain fake GPS signals in image format. The UAV trajectory can be obtained from these images using Visual Odometry. Then, the extracted trajectory can be compressively compared with the existing flight trajectory data from GPS positions to detect GPS spoofing attacks. Other studies focused on using AI-based approaches to detect, classify, and mitigate these attacks on UAVs. In these studies, malicious attackers can still coordinate complex spoofing assaults using some cutting-edge technologies. These technologies can be used by ML and DL models to bypass standard detection techniques, as summarized in table I.

It is now widely recognized that machine learning is a powerful tool to detect anomalies and attacks, particularly in heterogeneous and uncertain environments. For instance, the authors of [13] used a supervised ML model, support vector machine (SVM), to detect GPS spoofing attacks. In this study, the authors used temporal drift of the receiver clock and the time derivative of the clock offset. In [14], the authors used several supervised ML models, namely Naïve Bayes, linear regression, decision tree, random forest, and SVM, and proposed a learning approach. They used Jitter and shimmer and their subcategories as input features. The extracted features consist of the GPS signal fundamental frequency and amplitude variations. K-learning was applied as a voting technique for the developed ML models. In [15, 16], the authors compared the performance of four tree-based supervised ML models, namely Random Forest, Gradient Boost, Extreme Gradient Boosting, and Light Gradient Boosting, along with several instance-based supervised ML models, namely Support Vector Machine (SVM), Linear-SVM, Nu-SVM, K-nearest neighbor (KNN), and Radius Neighbors in detecting and classifying GPS spoofing attacks on UAVs.

In [17], the authors compared the performance of ensemble supervised ML models, namely bagging, boosting, and stacking for detecting these attacks on UAVs. In [18], the authors proposed two dynamic selection algorithms, namely Metric-Optimized Dynamic and Weighted Metric-Optimized Dynamic to select the best performance model among a group of supervised models. All these approaches only perform well on structured data; however, real-world data are mostly not pre-processed and structured; therefore, pre-processing the data and transferring it to the proper format is time-consuming and costly. To solve this problem, DL models have been proposed to address the limitations of traditional and convolutional ML models.

One of the critical key factors of using DL models is to automatically learn complex and abstract representations from large amounts of data, which can be useful for identifying patterns or anomalies in network traffic or other types of data associated with cyber-attacks. Additionally, other used ML approaches in literature may deal with some shortcomings, such as a high rate of error and bias, low detection, and high misdetection rate. For this purpose, several studies have focused on DL models to detect and classify these attacks on UAVs. For instance, the authors of [19] proposed a GPS replay attack detection method based on a supervised DL model, namely ANN. In this study, the authors showed the effect of several extracted features from the received signal on detection performance. The best results were obtained by combining five parameters, namely satellite vehicle number, pseudo-range, carrier phase, Doppler shift, and signal-to-noise ratio.

In [20], the authors used three signal properties as input features of a supervised DL model, a multi-layer neural network. These three input features are early-late phase, delta, and signal level. The proposed method has been evaluated using software-based GPS simulators. In [21], the authors proposed Long Short-Term Memory which monitored the derived PVT information from the GPS signal using this DL model. In [22], the authors used a supervised DL model, namely a Convolutional Neural Network-based model, Residual Neural Network. As discussed previously, the current studies in literature are widely focused on supervised ML and DL models. Despite these proposed studies indicating satisfactory results, there are still several limitations that need to be addressed.

## III. Materials and Methods

The proposed GPS spoofing detection framework is depicted in Figure 1. The framework contains three main steps, data acquisition and preprocessing, implementation DL models, and GPS spoofing classification. In the data acquisition and pre-processing step, real-time experiments were conducted to gather GPS signals, and several simulations were performed to develop GPS spoofing attacks, namely simplistic, intermediate, and sophisticated attacks. The features were extracted, and the class labels of the given data were discarded. To perform the data preprocessing, the non-stationary data was transformed to stationary, and several pre-processing techniques, such as data correlation, data imputation, and transformation, were used to improve the quality of the data. The second step, implementation of DL models, mainly focuses on using selected unsupervised models and optimizing the results.

TABLE I. CURRENT STUDIES ON DETECTING AND CLASSIFYING GPS SPOOFING ATTACKS ON UAVS USING MACHINE LEARNING AND DEEP LEARNING.

| Category | Method(s) | Study highlights | Limitations |
|---|---|---|---|
| UAV Characteristic-based | IMU [8] | Providing high detection rate, using GPS data features and IMU characteristics. | Detected attacks with the same behavior during the training process. |
| | Gyroscope Measurement [9] | Using Gyroscope Measurement along GPS data to detect spoofed signals. | Requiring motion sensors, which are power hungry. |
| | Acceleration error [10] | Providing better performance using acceleration magnitude. | Providing constant false alarm rate. |
| Signal Processing-based | Vision [11] | Detecting attacks using vision sensors with the IMU data. | Applicable only when the attacker is visible. |
| | Vision [12] | Detecting attacks using Visual Odometry technique. | Applicable only when the attacker is visible. |
| Artificial Intelligence-based | Supervised ML [13] | Using SVM to detect time drift in GPS spoofing signal. | No comparison with other techniques. |
| | Supervised ML [14] | Reconstructing the path using embedded sensors and comparing it to the GPS path. | Used Shimmer and Jitter as only features in the benchmark. |
| | Supervised ML [15] | Extracting of multiple features from GPS signal, Evaluating different Tree-based ML models to detect GPS spoofing attacks. | Needed to lower misdetection and false alarm rates. |
| | Supervised ML [16] | Evaluating different instance-based ML models to detect GPS spoofing attacks. | Needed to lower misdetection and false alarm rates. |
| | Supervised ML [17] | Comparing the ensemble models to detect and classify GPS spoofing attacks on UAVs. | Used limited samples. |
| | Supervised ML [18] | Proposing two dynamic selector approaches to select the highest performance model among a group of models. | High computational complexity. |
| | Supervised DL [19] | Applying simple ANN to detect abnormalities in correlator output. | Used a benchmark with 5 features and limited samples. |
| | Supervised DL [20] | Using multi-layer neural network to detect attacks on the network. | Used a dataset with 3 input features. |
| | Supervised DL [21] | Detecting path deviation caused by the attack using LSTM model. | Provided low detection rate, Indicates high detection rate only when the flight trajectory is not complex. |
| | Supervised DL [22] | Proposing a detection technique, using DeepSIM, a satellite imagery matching approach, to detect GPS spoofing attacks on UAVs. | Used the pictures to detect GPS spoofing attacks on UAVs, while there is no guarantee the pictures were reliable and valid. |

Four unsupervised DL models, namely CAE, CRBM, DBNN, and GAN, were applied, and the Adaptive Moment Estimation was used to maximize the performance of these models. These models were evaluated in terms of Gap static, Calinski harabasz score, Silhouette Score, homogeneity, completeness, and V-measure. The last step, output, is to interpret the results and extract useful insights and knowledge from them.

### *A. Data Acquisition and Feature Descriptions*

The corresponding dataset in this study was initially developed and simulated in a previous work [8]. The given dataset consists of 13 features, as shown in Table II. In brief, this dataset contains legitimated and spoofed GPS samples from three types of spoofing attacks, namely simplistic, intermediate, and sophisticated. These GPS spoofing attacks can have a strong impact on features, including Carrier to Noise or Carrier Doppler. The dataset consists of 14,000 samples, 50% spoofed and 50% normal signals. More details about this dataset and the acquisition, tracking, and data preprocessing of such data can be found in [8].

### *B. Data Pre-processing*

In this study, several data preprocessing techniques were performed to improve the quality of data for the training, testing, and validating the unsupervised DL models. The corresponding dataset is balanced; hence, no technique is required to balance the different classes. As previously discussed, unsupervised models can only train and test the data with no labels. Thus, before performing any techniques, the class labels had to be discarded from the given data. After that, the initial step in data preprocessing was to detect correlated and low importance features from the given dataset. Redundant features can have a significant impact on the performance of the models [1]. As a result, two features, TCD and RX, were discarded [8]. Thus, 11 features, namely PRN, DO, PD, TOW, CP, EC, LC, PC, PIP, PQP, and C/N0 were used for training purposes.

The second step, data imputation, and the third step, data transformation, are necessary to guarantee accurate predictions of the results. In data imputation, mode imputation was used to replace the missing value with the value that has the highest frequency for the feature. In data transformation, Min-Max Scaler was applied to subtract the minimum value in the feature and divide it by its range.

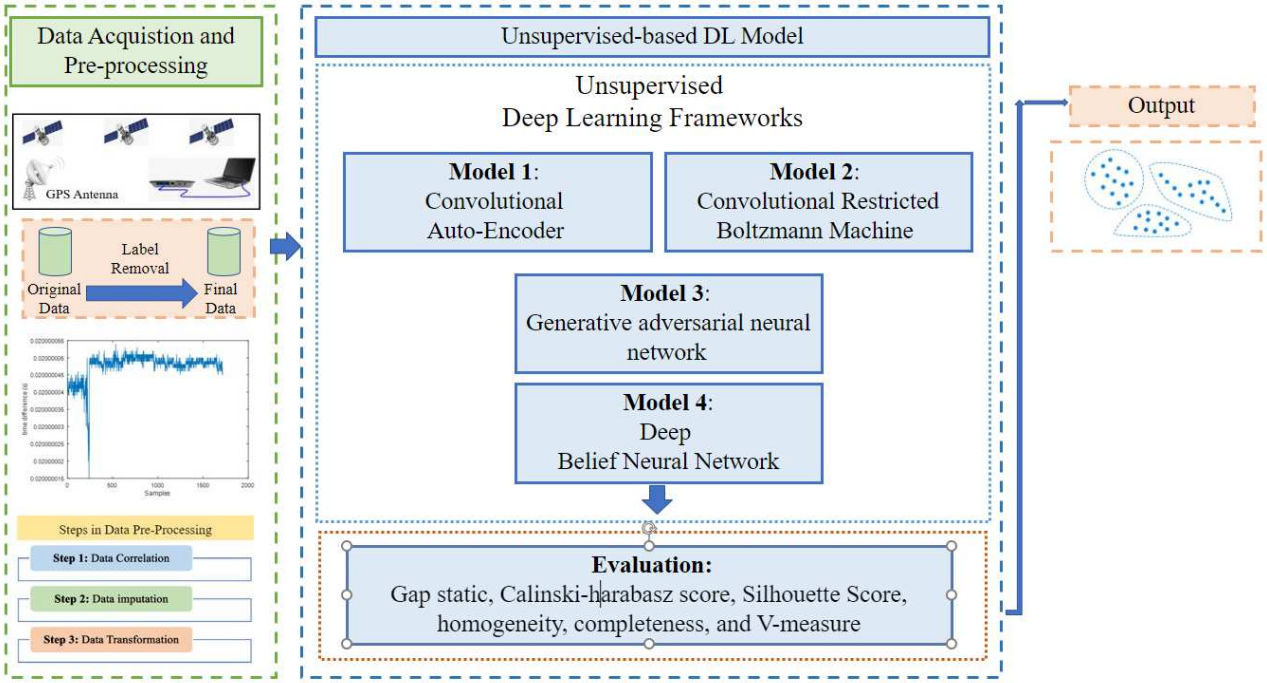


Figure 1. Proposed Unsupervised-based Framework for detecting and classifying GPS spoofing Attacks on UAVs.

TABLE II. LIST OF FEATURES WITH THEIR USED ABBREVIATIONS.

| Feature | Abbreviations |
|---|---|
| Satellite Vehicle Number | PRN |
| The Carrier Doppler | DO |
| Pseudo-Range | PD |
| Receiver Time | RX |
| Time of the week | TOW |
| Carrier Phase Cycles | CP |
| Early Correlator | EC |
| Late Correlator | LC |
| Prompt Correlator | PC |
| Prompt in phase correlator | PIP |
| Prompt Quadrature | PQP |
| Carrier Doppler | TCD |
| Carrier to noise Ratio | C/N0 |

### *C.* Deep Learning Models

In this work, four distinct deep learning approaches were used to detect and classify GPS spoofing attacks. Figure 2 shows an overview of the different categories of unsupervised DL models, namely Auto-Encoder (AE), Restricted Boltzmann Machine (RBM), Belief Neural Network (BNN), and Adversarial Neural Network-(AdNN) based models. From each of these categories we selected one model for this study. Detailed explanations of these categories along with the selected models are provided in the following subsections.

#### *C.1. Convolutional Auto-Encoder*

Auto-Encoder (AE)-based models represent one type of unsupervised DL models that indicates a compressed representation of the knowledge of an input. These models decrease the noise level in the corresponding data. This process can be performed via compression of the input data, encoding, and reconstructing the outputs. In addition, AE-based models can decrease the dimensionality of the data. In this work, we focused on convolutional auto encoder (CAE), which is trained to reproduce its input data in the output layer. The data is passed through the encoder, resulting in a low-dimensional representation of the data. The encoder process can be performed with several pooling and convolutional layers, as shown in Figure 3. The next layer, bottleneck layer, is a dimensional hidden layer that produces the encoding process. It consists of a lower number of nodes, and these nodes show dimensional encoding inputs. The purpose of this layer is to decrease the number of model parameters, resulting in a more efficient model [23].

Furthermore, a decoder takes the output of the encoder to generate the input. The encoder is responsible for interpreting and compressing the input to an internal representation that is defined by the bottleneck layer, while the decoder attempts to create an input data from the encoder. The decoder process is performed via de-convolutional and up-sampling layers. CAE preserves valuable information while minimizing the noise for an inefficient dimensionality reduction.

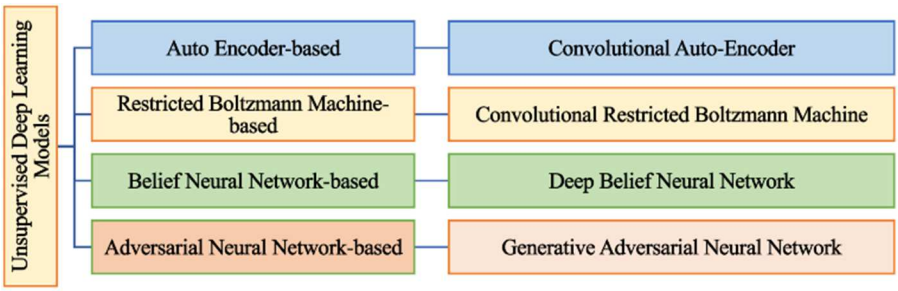


Figure 2. Classification of Unsupervised Deep Learning Models.

Furthermore, a decoder takes the output of the encoder to generate the input. The encoder is responsible for interpreting and compressing the input to an internal representation that is defined by the bottleneck layer, while the decoder attempts to create an input data from the encoder. The decoder process is performed via de-convolutional and up-sampling layers. CAE preserves valuable information while minimizing the noise for an inefficient dimensionality reduction.

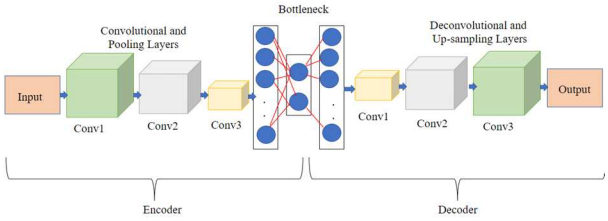


Figure 3. Architecture of CAE.

### *C.2 Convolutional Restricted Boltzmann Machine*

Convolutional Restricted Boltzmann Machine (RBM) models are probabilistic DL models that learn from the probability distribution of their inputs and a hidden representation. These models are energy-based models since they are an integral part of statistical mechanics. They consist of an input layer and a hidden layer without an output layer which gives them a non- deterministic feature. The Convolutional Restricted Boltzmann Machine (CRBM) model is an example of RBM models and is considered as a combination of CNN and RBM model, as shown in Figure 4. The CRBM model uses the weight-sharing method from CNN models. In CRBM models, the connections share the weights in a convolutional pattern with a convolutional filter, which connects filter nodes in feature maps [24].

As clear, the weights of the convolutional filter can mostly be applied to the visible and hidden nodes in visible and hidden layers (refer to Figure 4). It means that each hidden node is connected to the visible nodes. In contrast, the visible nodes share only one bias, and the hidden nodes can share the bias. In addition, the hidden nodes are connected to the pooling layer to detect and predict the output class labels. CRBM models can usually create unwanted border effects when the visible layer is re-generated; hence, the visible nodes are only connected to the few hidden nodes.

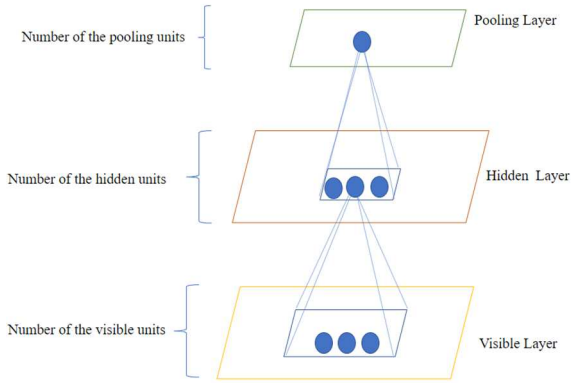


Figure 4. Architecture of CRBM.

### *C.3 Deep Belief Neural Network*

BNN models are one of the types of unsupervised DL models that uses a deep architecture of several stacks of RBM-based models. In these models, RBM models can perform a non-linear transformation on their input vectors and generate the output vectors, which serve as input for the next RBM model in the sequence. One of the mainly used BNN models is deep belief neural network (DBNN), which can learn feature representation effectively from huge amounts of given data and complicated functions [25]. Figure 5 provides an illustration of DBNN architecture with $H$ hidden layers and $V$ visible layers. In DBNN model, the state of the network along with its matrix weight is initialized. Then, the random sample into the model is fed, and the states of the nodes in the first hidden layer are updated. In this model, the state of the visible node is updated with a mean of 0 and variance of 1, while the states of the nodes in the hidden layer are updated after taking the state of the nodes in the visible layers. After that, the random sample is restricted and fed into the model. If the selected sample is already used, a new round of training is performed until a preset number of iterations is achieved or the change in the weight matrix is small.

After that, the output of the first RBM is considered as the input of the second RBM and the next RBM is trained until all of them are trained. As a result, the class labels output can be predicted using the logistic regression function of these RBM

models. In addition, the unique feature of DBN is the use of an unsupervised layer-wise pre-training function, which makes the network highly efficient and accurate. Unlike traditional neural networks, the DBN uses a layer-wise approach to train its weight. Each layer is trained using an RBM to learn the lower-level features of the input data. Once these layers are trained, the weights are then used as the initial values for the next layer, which is trained using backpropagation. This process can lead to faster convergence and higher accuracy than traditional neural networks [26].

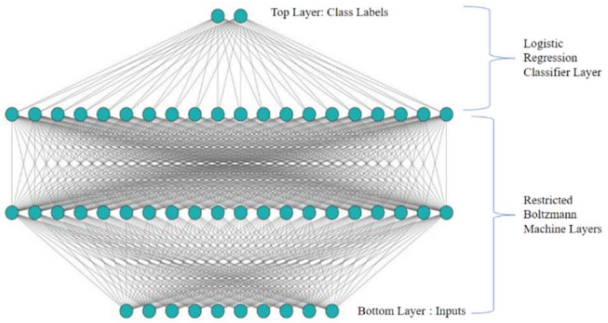


Figure 5. Architecture of DBNN.

*C.4 Adversarial Neural Network*

Generative adversarial neural network (GAN) is an unsupervised DL model performed based on the architecture of AdNN models. The main idea of GAN is based on the Nash equilibrium in game theory. This model consists of two parts, generator, and discriminator. The generator can learn the distribution of the authentic data, whereas the discriminator can determine correctly if the given data is authentic, or it is taken from the generator. To complete the process, the generator and discriminator are required to continuously optimize themselves to improve the generation capacity and the discrimination capacity. The aim of this optimization procedure is to find a Nash equilibrium between the generator and the discriminator [27].

The architecture of GAN is presented in Figure 6. As one can observe, the generator and discriminator functions, D and G, are mainly used as modules. Given the authentic data X and random variable Z, G(Z) is considered as the samples created by generator G, and D(Z) as the samples generated by discriminator D. In generator G, the plausible data is generated, while these generated instances change to fake training samples for the discriminator D.

The discriminator D learns to differentiate the generator's fake data from the real data. It actually penalizes generator G for creating implausible results. Hence, such a process can improve the performance of D and G gradually. In this context, when the discriminator capacity is increased to the highest level and cannot discriminate the data source properly, generator G has achieved the distribution of authentic (real) data.

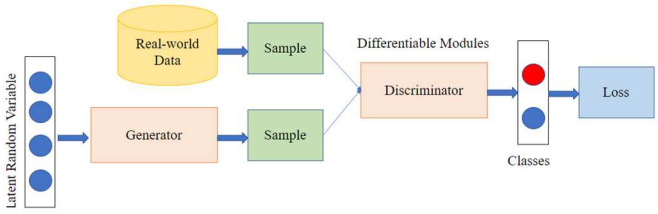


Figure 6. Architecture of GAN.

3.4. Evaluation Metrics

To evaluate the performance of unsupervised DL models, several metrics were used, namely Gap static, Calinski harabasz score, Silhouette Score, homogeneity, completeness, and V-measure. These metrics are briefly defined below:

Gap Statistic: It is a statistical metric used for evaluating the optimal number of clusters in a dataset during the clustering analysis or clustering algorithms like k-means. Clustering involves grouping similar data points together, and determining the right number of clusters which is crucial for the effectiveness of the clustering algorithm. This metric can be computed as follows:

$$Gap(K) = \frac{1}{B}\sum_{i=1}^{B}[\log w_K^*) - \log(W_K)] \quad (5)$$

Where K denotes as the number of the clusters being evaluated, $W_K$ is the within cluster variation for the actual clustering results with K clusters, $w_K^*$ is the within cluster variation for the reference or simulated dataset with K clusters, and B is the number of Monte Carlo Simulations.

Calinski harabasz score: This score, also known as the Variance Ratio Criterion, is a metric used to evaluate the quality of clusters in an unsupervised analysis. It measures the ratio of the between-cluster variance to the within-cluster variance. In other words, it assesses how well-separated the clusters are from each other compared to how compact the data points are within each cluster. This score can be calculated as:

$$Calinski\ harabasz\ score = \frac{tr\,(B_K)}{tr\,(W_K)} * \frac{n_{E-K}}{K-1} \quad (6)$$

Where a set of data E of size $n_E$ clustered into the K clusters, $r\,(B_K)$ is the trace of the between group dispersion matrix, and $tr\,(W_K)$ is the trace of within-cluster dispersion matrix. These traces can be defined as following:

$$W_K = \sum_{q=1}^{K}\sum_{x\in c_q}(x - c_q)(x - c_q)^T \quad (7)$$

$$B_K = \sum_{q=1}^{K} n_q\,(c_q - c_E)(c_q - c_E)^T \quad (8)$$

Where $c_q$ is the set of points in a cluster $q$, $c_E$ is the center of a cluster $E$, $n_q$ is the number of the points in the cluster $q$, and $T$ is the transpose function. A higher Calinski-Harabasz score indicates better separation between clusters, suggesting that the clustering is more effective. This score can be used to help determine the optimal number of clusters for the given data by comparing the scores for different numbers of clusters and selecting the one that maximizes the score.

- *Silhouette Score*: This score is used to evaluate the quality of clusters in a clustering analysis, such as k-means clustering. It provides a measure of how well-separated and distinct the clusters are in the data. The Silhouette Score is based on the idea of how similar a data point is to its own cluster (cohesion) compared to other clusters (separation). This score can be computed, as follows:

$$Silhouette\ Score = \frac{b-a}{\max(a,\ b)} \quad (9)$$

In this equation, $a$ denotes the mean distance between a sample and other sample in the similar class, and $b$ is the mean distance between a sample and other sample in the next nearest cluster. The Silhouette score ranges from -1 to 1; a score close to 1 indicates that the data point is well-clustered and is far from neighboring clusters. In contrast, a score close to 0 suggests that the data point is on or very close to the decision boundary between two neighboring clusters. A score close to -1 indicates that the data point may have been assigned to the wrong cluster. As a result, higher Silhouette scores generally indicate better unsupervised solutions, as they represent better separation and cohesion of clusters.

- *Homogeneity:* It measures the extent to which all elements within a cluster belong to the same class. Mathematically, it is calculated as:

$$H(y,c) = 1 - \frac{H(C|K)}{H(C)} \quad (10)$$

Where $H(C|K)$ is the conditional entropy of the class labels given the cluster assignments and $H(C)$ is the entropy of the class labels. *y* is the true class labels and c is cluster assignments generated by clustering algorithms.

- *Completeness:* It assesses the degree to which all elements that belong to the same class are assigned to the same cluster. It is expressed by:

$$H(y,c) = 1 - \frac{H(K|C)}{H(K)} \quad (11)$$

Where $H(K|C)$ is the conditional entropy of the cluster assignments given the class labels and $H(K)$ is the entropy of the cluster assignments.

- *V-measure:* It combines homogeneity and completeness into a single metric by taking their harmonic mean. V-measure is given by:

$$V(y,c) = \frac{2H(y,c).C(y,c)}{H((y,c))+ C(y,c)} \quad (12)$$

Where *H(.)* is the entropy, and *C(.)* is the conditional entropy. They are used to quantify the uncertainty associated with class labels and cluster assignments.

## IV. Results and Discussions

In this study, four unsupervised neural networks were implemented, CAE, CRBM, DBNN, and GAN. These models were tested using Adaptive Moment Estimation (ADAM) as a loss optimizer, with a learning rate of 0.01. The models were evaluated in terms of Gap static, Calinski harabasz score, Silhouette Score, homogeneity, completeness, and V-measure. Simulations were conducted on an Intel® Xeon® CPU E5-1620 v4 @ 3.50 GHz CPU with 16 GB of memory, TensorFlow 2.0, Python 3.9, and a batch size of 100. Figures 7 through 8 and Table III depict the results of these models.

Figure 7 provides the results of the unsupervised DL models in terms of Gap Static with respect to the number of clusters. As previously mentioned, to obtain the optimal number of the clusters, it is recommended to select the highest Gap Statistic. As one can observe, the highest Gap Statistic belongs to the CAE model with the six clusters with a Gap Statistic of 0.37. Additionally, it is shown that the other models, excluding DBNN, achieve the optimal six clusters with a Gap Statistic rate, ranging between 0.25 and 0.33. For example, the GAN model has an optimal six clusters with a Gap Statistic of 0.33. In contrast, the DBNN model has an optimal four clusters with a Gap Statistic rate of 0.25.

The results corresponding to the Calinski harabasz and Silhouette scores are shown in Figure 8. This figure highlights these scores for the selected DL models with respect to the number of clusters. As previously discussed, a Calinski harabasz score indicates a model with high cluster quality.

As one can observe, the optimal number of the clusters' using CAE is 6 based on Calinski harabasz and Silhouette scores. This model achieves a Calinski harabasz score of 369 and a Silhouette score of 0.96 in cluster 6, which is the highest compared to other models. The other models, GAN and CRBM, have optimal scores in cluster 6; while the worst performance based on these scores belong to DBNN with a Calinski harabasz score of 100 and a Silhouette score of 0.31.

Table III shows the results of the unsupervised models in terms of homogeneity, completeness, and V-measure. As can been seen, the CAE model has the best performance in six clusters in terms of Homogeneity, Completeness, and V-measure among other models, followed by GAN, CRBM, and DBNN. This model has a Homogeneity of 0.13456, a Completeness of 0.0956, and a V-measure of 0.09245. In contrast, the lowest performance belongs to the DBNN model.

For example, this model for four clusters has the better performance compared to the other number of the optimal clusters with a Homogeneity of 0.08732, a Completeness of 0. 06543, and a V-measure of 0.06012. It is worth mentioning that the optimal number of clusters for the unsupervised models, excluding the DBNN is six. For instance, the GAN model has an acceptable result for six clusters with a Homogeneity of 0.1123, a Completeness of 0. 0876, and a V-measure of 0.0812. Same observations can be seen for the CRBM model.

As a result, the key factors of this study can be summarized as follows:

- These unsupervised models, excluding the DBNN, have an optimal six clusters, which can be used as a number of the clusters in process of detecting GPS spoofing attacks on UAVs.
- The CAE model has the best performance among the other unsupervised models.

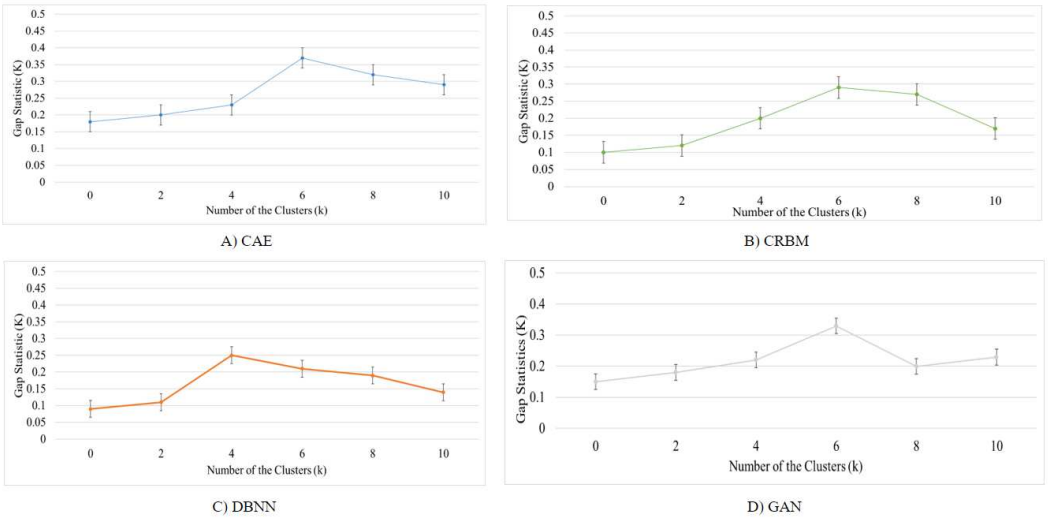


Figure 7. Results of the Unsupervised Models in Terms of Gap Statistic with respect to the Number of the Clusters.

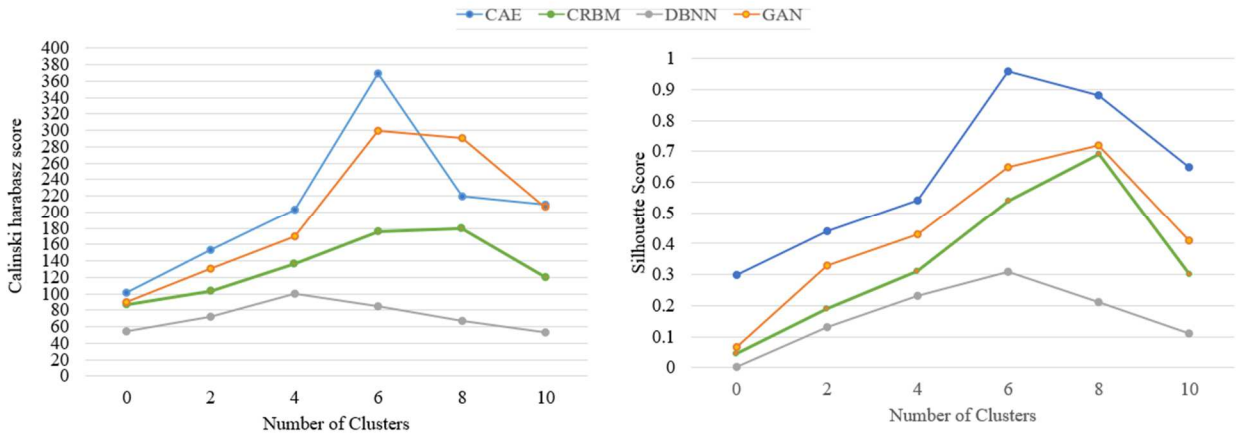


Figure 8. Results of the Calinski harabasz and Silhouette scores with respect to the Number of Clusters.

- The scores of Calinski and Silhouette demonstrated that the better clustering quality and performance in CAE, compared to other models, followed by GAN, CRBM, and DBNN, respectively.
- The DBNN model has the worst performance among the other models.


## ACKNOWLEDGMENT

The authors acknowledge the support of the National. Science Foundation (NSF), Award Number 2006674.


## V. CONCLUSION

This paper investigates the performance of unsupervised models in detecting, classifying, and predicting GPS spoofing attacks on UAVs. The unsupervised models used in this study are Convolutional Auto-Encoder, Convolutional Restricted Boltzmann Machine, Deep Belief Neural Network, and Adversarial Neural Network. The evaluation was performed using seven metrics: Gap static, Calinski harabasz score, Silhouette Score, homogeneity, completeness, and V-measure.The result showed that Convolutional Auto-Encoder outperforms the other models in terms of these metrics, while, these unsupervised models, excluding the DBNN, has an optimal six clusters, which can be used as several of the clusters in process of detecting GPS spoofing attacks on UAVs. In contrast, the DBNN model has the worst performance among other models with an optimal number of four clusters.

TABLE III. RESULTS OF THE MODELS IN TERMS OF HOMOGENEITY, COMPLETENESS, AND V-MEASURE.

| Model | Number of the Clusters (k) | Homogeneity | Completeness | V-measure |
|---|---|---|---|---|
| | 0 | 0.1043 | 0.0812 | 0.0654 |
| | 2 | 0.1099 | 0.08345 | 0.0876 |

| | | | | |
|---|---|---|---|---|
| **CAE** | 4 | 0.1254 | 0.0916 | 0.09023 |
| | **6** | **0.13456** | **0.0956** | **0.09245** |
| | 8 | 0.1287 | 0.08690 | 0.08832 |
| | 10 | 0.1185 | 0.7765 | 0.08976 |
| CRBM | 0 | 0.0854 | 0.06510 | 0.0654 |
| | 2 | 0.0921 | 0.06532 | 0.0432 |
| | 4 | 0.0934 | 0.6439 | 0.0590 |
| | 6 | 0.1098 | 0.0762 | 0.07990 |
| | 8 | 0.1002 | 0.7089 | 0.0632 |
| | 10 | 0.0932 | 0.6990 | 0.05912 |
| DBNN | 0 | 0.07776 | 0.05567 | 0.0465 |
| | 2 | 0.08324 | 0.05345 | 0.0598 |
| | 4 | 0.08732 | 0.06543 | 0.06012 |
| | 6 | 0.07654 | 0.05932 | 0.05562 |
| | 8 | 0.07521 | 0.05034 | 0.05123 |
| | 10 | 0.06543 | 0.04567 | 0.0432 |
| GAN | 0 | 0.1037 | 0.04987 | 0.07912 |
| | 2 | 0.1098 | 0.06590 | 0.07512 |
| | 4 | 0.1102 | 0.0790 | 0.0762 |
| | 6 | 0.1123 | 0.0876 | 0.0812 |
| | 8 | 0.1056 | 0.07231 | 0.0791 |
| | 10 | 0.0976 | 0.0682 | 0.07821 |

## REFERENCES


[1] T. Talaei Khoei, S. Ismail, K. A. Shamaileh, V. K. Devabhaktuni, and N. Kaabouch, “Impact of Dataset and Model Parameters on Machine Learning Performance for the Detection of GPS Spoofing Attacks on Unmanned Aerial Vehicles,” Applied Sciences, vol. 13, no. 1, p. 383, Dec. 2022, doi: 10.3390/app13010383.

[2] T. T. Khoei, A. Gasimova, M. A. Ahajjam, K. A. Shamaileh, V. Devabhaktuni and N. Kaabouch, "A Comparative Analysis of Supervised and Unsupervised Models for Detecting GPS Spoofing Attack on UAVs," 2022 IEEE International Conference on Electro Information Technology (eIT), Mankato, MN, USA, 2022, pp. 279-284, doi: 10.1109/eIT53891.2022.9813826.

[3] J. Zhang, L. Pan, Q. -L. Han, C. Chen, S. Wen and Y. Xiang, "Deep Learning Based Attack Detection for Cyber-Physical System Cybersecurity: A Survey," in IEEE/CAA Journal of Automatica Sinica, vol. 9, no. 3, pp. 377-391, March 2022, doi: 10.1109/JAS.2021.1004261.

[4] K. Kumar, S. Kumar, O. Kaiwartya, A. Sikandar, R. Kharel, and J. L. Mauri, “Internet of Unmanned Aerial Vehicles: QoS Provisioning in Aerial Ad-Hoc Networks,” Sensors, vol. 20, no. 11, p. 3160, 2020, doi: 10.3390/s20113160

[5] B. Pardhasaradhi and L. R. Cenkeramaddi, "GPS Spoofing Detection and Mitigation for Drones Using Distributed Radar Tracking and Fusion," in IEEE Sensors Journal, vol. 22, no. 11, pp. 11122-11134, 1 June1, 2022, doi: 10.1109/JSEN.2022.3168940.

[6] Y. Dang, C. Benzaïd, B. Yang, T. Taleb and Y. Shen, "Deep-Ensemble-Learning-Based GPS Spoofing Detection for Cellular-Connected UAVs," in IEEE Internet of Things Journal, vol. 9, no. 24, pp. 25068-25085, 15 Dec.15, 2022, doi: 10.1109/JIOT.2022.3195320.

[7] Jiang, P., Wu, H. and Xin, C., 2022. DeepPOSE: Detecting GPS spoofing attack via deep recurrent neural network. Digital Communications and Networks, 8(5), pp.791-803.

[8] Feng, Z.; Guan, N.; Lv, M.; Liu, W.; Deng, Q.; Liu, X.; Yi, W. Efficient drone hijacking detection using two-step GA-XGBoost. J. Syst. Archit. 2020, 103, 101694.

[9] Feng, Z.; Guan, N.; Lv, M.; Liu, W.; Deng, Q.; Liu, X.; Yi, W. Efficient drone hijacking detection using onboard motion sensors. In Proceedings of the Design, Automation Test in Europe Conference Exhibition (DATE) 2017, Lausanne, Switzerland, 27–31 March 2017; pp. 1414–1419

[10] Kwon, K.C.; Shim, D.S. Performance analysis of direct GPS spoofing detection method with AHRS/Accelerometer. Sensors 2020, 20, 954.

[11] Qiao, Y.; Zhang, Y.; Du, X. A Vision-Based GPS-Spoofing Detection Method for Small UAVs. In Proceedings of the 13th International Conference on Computational Intelligence and Security, CIS 2017, Hong Kong, China, 15–18 December 2017; pp. 312–316.

[12] Varshosaz, M.; Afary, A.; Mojaradi, B.; Saadatseresht, M.; Parmehr, E.G. Spoofing detection of civilian UAVs using visual odometry. ISPRS Int. J. Geo-Inf. 2019, 9, 6.

[13] S. Semanjski, A. Muls, I. Semanjski, and W. De Wilde, “Use and validation of supervised machine learning approach for detection of GNSS signal spoofing,” in 2019 International Conference on Localization and GNSS (ICL-GNSS). IEEE, 2019, pp. 1–6.

[14] A. Shafique, A. Mehmood and M. Elhadef, "Detecting Signal Spoofing Attack in UAVs Using Machine Learning Models," in IEEE Access, vol. 9, pp. 93803-93815, 2021, doi: 10.1109/ACCESS.2021. 3089847.

[15] G. Aissou, H. O. Slimane, S. Benouadah and N. Kaabouch, "Tree-based Supervised Machine Learning Models For Detecting GPS Spoofing Attacks on UAS," 2021 IEEE 12th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), 2021, pp. 0649-0653, doi: 10.1109/UEMCON53757.2021.9666744.

[16] G. Aissou, S. Benouadah, H. El Alami and N. Kaabouch, "Instance-based Supervised Machine Learning Models for Detecting GPS Spoofing Attacks on UAS," 2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC), 2022, pp. 0208-0214, doi: 10.1109/CCWC54503.2022.9720888.

[17] A. Gasimova, T. T. Khoei and N. Kaabouch, "A Comparative Analysis of the Ensemble Models for Detecting GPS Spoofing attacks on UAVs," 2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 2022, pp. 0310-0315, doi: 10.1109/CCWC54503.2022.9720738.

[18] Talaei Khoei, T., Ismail, S. and Kaabouch, N., 2022. Dynamic selection techniques for detecting GPS spoofing attacks on UAVs. Sensors, 22(2), p.662.

[19] M. R. Manesh, J. Kenney, W. C. Hu, V. [n] K. Devabhaktuni, and N. Kaabouch, “Detection of GPS spoofing attacks on unmanned aerial systems,” in 2019 16th IEEE Annual Consumer Communications & Networking Conference (CCNC). IEEE, 2019, pp. 1–6.

[20] E. Shafiee, M. Mosavi, and M. Moazedi, “Detection of spoofing attack using machine learning based on multi-layer neural network in single frequency gps receivers,” The Journal of Navigation, vol. 71, no. 1, pp. 169–188, 2018.

[21] S. Wang, J. Wang, C. Su and X. Ma, "Intelligent Detection Algorithm Against UAVs' GPS Spoofing Attack," 2020 IEEE 26th International Conference on Parallel and Distributed Systems (ICPADS), Hong Kong, 2020, pp. 382-389, doi: 10.1109/ICPADS51040.2020.00058.

[22] Xue, Nian, Liang Niu, Xianbin Hong, Zhen Li, Larissa Hoffaeller, and Christina Pöpper. "Deepsim: Gps spoofing detection on uavs using satellite imagery matching." In Annual computer security applications conference, pp. 304-319. 2020.

[23] Du, B., Xiong, W., Wu, J., Zhang, L., Zhang, L. and Tao, D., 2016. Stacked convolutional denoising auto-encoders for feature representation. IEEE transactions on cybernetics, 47(4), pp.1017-1027.

[24] Zhang, N., Ding, S., Zhang, J. and Xue, Y., 2018. An overview on restricted Boltzmann machines. Neurocomputing, 275, pp.1186-1199.
[25] Cao, M., Zhang, T., Liu, Y., Wang, Y. and Shi, Z., 2023. A Bayesian optimization hyperband-optimized incremental deep belief network for online battery behaviour modelling for a satellite simulator. Journal of Energy Storage, 58, p.106348.
[26] Zhao, G., Zhang, C. and Zheng, L., 2017, July. Intrusion detection using deep belief network and probabilistic neural network. In 2017 IEEE international conference on computational science and engineering (CSE) and IEEE international conference on embedded and ubiquitous computing (EUC) (Vol. 1, pp. 639-642). IEEE.
[27] Creswell, A., White, T., Dumoulin, V., Arulkumaran, K., Sengupta, B. and Bharath, A.A., 2018. Generative adversarial networks: An overview. IEEE signal processing magazine, 35(1), pp.53-65.