NATURENGS
E-ISSN 2717-8013
https://dergipark.org.tr/en/pub/naturengs

# Enhancing Cybersecurity through GAN-Augmented and Hybrid Feature Selection Machine Learning Models: A Case Study on EVSE Data

Hayriye TANYILDIZ [a,*], Canan BATUR ŞAHİN [a], Özlem BATUR DİNLER [b]

[a] Malatya Turgut Ozal University, Faculty of Engineering and Natural Sciences,Malatya 44210, Türkiye
[b] Siirt University, Faculty of Engineering, Departmant of Computer Engineering, Siirt, 56000, Turkey

*Corresponding author

ARTICLE INFO

Received 04.06.2024
Accepted 28.06.2024

Doi: 10.46572/naturengs.1495489

ABSTRACT

Electric Vehicle Charging Stations (EVCSs) are critical components of modern transportation infrastructure. However, smart grid integrations create new security vulnerabilities. Cyber attacks can damage EVCSs, cause financial losses, and compromise user security. Traditional security measures are not enough. By analyzing the large data sets produced by EVCSs, machine learning (ML) can detect anomalies and provide predictive maintenance. The increasing importance of EVCSs in smart grid infrastructure necessitates taking advanced security measures. Machine learning-based intrusion detection systems represent a promising solution to the dynamic and complex cyber threats facing these critical infrastructures. Through continuous learning and adaptation, ML can provide a robust defense mechanism that ensures the security and reliability of EVCSs in the face of evolving cyber threats.
The primary aim of this study is to develop preventive solutions against cyber attacks for newly emerging electric charging systems. The proposed GAN-Augmented and Hybrid Feature Selection Machine Learning Models achieved an accuracy rate of 97.56%, which is promising in terms of the model's usability and security. The methods or approaches used provide a new framework that can be utilized in other studies.

***Keywords:*** *Electric Vehicle Charging Stations*, GAN, Anaomaly Detection, Cyber Attack

## 1. Introduction

Electric Vehicle Charging Stations (EVCSs) serve as the backbone of the EV infrastructure, providing the power necessary to keep electric vehicles running. With the rapid adoption of electric vehicles worldwide, the number and complexity of EVCSs has also increased. These charging stations are an integral part of not only individual EV users, but also fleet operators and public transportation systems, making them a critical component of modern transportation infrastructure.

However, the integration of EVCSs into the broader smart grid introduces new vulnerabilities and attack vectors [3]. Unlike traditional standalone systems, smart grids are interconnected networks that facilitate two-way communication between suppliers and consumers. While this interconnectedness is beneficial for efficiency and management, it also opens the door to multiple entry points for cyber attacks. Cyber attacks on EVCSs can cause significant disruptions, financial losses, and compromise user security and privacy. For example, an attacker could potentially change charging rates, disrupt service availability, or steal sensitive user data.

The types of cyber threats that EVCSs may face are diverse. There can be relatively simple attacks, such as unauthorized access and data theft, or more complex attacks, such as Distributed Denial of Service (DDoS) attacks, which can overload the system and render it inoperable.There is also the threat of ransomware, where attackers lock the system and demand payment to restore functionality.The consequences of such attacks are serious, not only directly affecting users, but also causing cascading effects on overall grid stability and public trust in EV infrastructure. Unlike traditional methods, ML-based systems can adapt to new threats by learning from historical data and constantly improving their detection capabilities.This adaptability is crucial to

* Corresponding author. e-mail address: 36223626008@ozal.edu.tr
ORCID : 0000-0002-6300-9016

maintaining robust security in an ever-changing threat landscape.

The integration of EVCSs into the broader smart grid introduces new vulnerabilities and attack vectors [1]. Unlike traditional standalone systems, smart grids are interconnected networks that facilitate two-way communication between suppliers and consumers. While this interconnectedness is beneficial for efficiency and management, it also opens the door to multiple entry points for cyber attacks. Cyber attacks on EVCSs can cause significant disruptions, financial losses, and compromise user security and privacy. For example, an attacker could potentially change charging rates, disrupt service availability, or steal sensitive user data. The types of cyber threats that EVCSs may face are diverse. There can be relatively simple attacks, such as unauthorized access and data theft, or more complex attacks, such as Distributed Denial of Service (DDoS) attacks, which can overload the system and render it inoperable. There is also the threat of ransomware, where attackers lock the system and demand payment to restore functionality. The consequences of such attacks are serious, not only directly affecting users, but also causing cascading effects on overall grid stability and public trust in EV infrastructure. Unlike traditional methods, ML-based systems can adapt to new threats by learning from historical data and constantly improving their detection capabilities. This adaptability is crucial to maintaining robust security in an ever-changing threat landscape.

In this work, we investigate the application of Generative Adversarial Networks (GANs) to augment cybersecurity datasets and improve the performance of various machine learning models. We use a dataset from Electric Vehicle Supply Equipment (EVSE) systems, preprocess it, and generate synthetic data using a GAN. We then train multiple machine learning models with and without applying a hybrid feature selection method on the augmented dataset to evaluate the impact on model performance. Our results show that data augmented with GAN-augmented feature selection can improve the accuracy and robustness of machine learning classifiers in cybersecurity applications.

Bu makale Giriş bölümünde, EVCS siber saldırı problemine ve önlenmesinin önemine genel bir bakış sunar ve ardından Bölüm 2'de ilgili literatürün gözden geçirilmesi gelir. Bölüm 3, tasarım, veri toplama ve analiz yöntemleri de dahil olmak üzere araştırma metodolojisini özetlemektedir. Çalışmanın sonuçları Bölüm 4'te ve bulguların ayrıntılı bir tartışması ve yorumlanması sunulmaktadır. Makale, Bölüm 5'da gelecekteki araştırmalar için önerilerle sonuçlanmaktadır

### *1.1. Literature Review*

In study [2], the authors proposed an intrusion detection system using deep belief network (DBN). DBN is an algorithm for increasing the number of different unsupervised networks grouped together to serve as input for the next layer. This is achieved using autoencoders, specifically restricted Boltzmann machines (RBMs). They implemented the model using TensorFlow. The results showed that the accuracy of this model reached 86% and the F1 score reached 84%.

In their study, Arsalan et al. [3] proposed a model predictive control (MPC)-based machine learning (ML) network integrated with training data pre-processing. The superior performance of the proposed approach is validated using different case study scenarios of training datasets.

Kem et al. [4] considered both classification- and recency-based models for anomaly detection, as well as an ensemble method to combine both models. They conducted evaluations based on real-world EV charging session data with simulated attacks. The results showed that regression-based prediction provides a significant increase in detection performance of attacks affecting individual reports during a charging session. Therefore, they stated that the proposed solution could make a positive contribution to EV charging safety, durability and reliability.

Malik et al. [15] proposed an intrusion detection system using Deep Belief Networks (DBN). DBN is an algorithm designed to increase the number of different unsupervised networks that are stacked together to serve as input for the next layer. This is achieved using autoencoders, particularly Restricted Boltzmann Machines (RBMs). The results showed that the accuracy of this model reached 86%, and its F1 score was 84%.

Basnet et al. [16] proposed a deep learning-based IDS to detect DDoS attacks within EVCSs. They implemented Deep Neural Networks (DNN) and Long Short-Term Memory (LSTM) algorithms, demonstrating that the LSTM model was superior in terms of precision and recall.

## 2. Materials and Methods

### *2.1. Data Sets*

The Canadian Cyber Security Institute's EVSE Dataset 2024 [5] focuses on the security of electric vehicle charging stations. Reconnaissance includes data from a variety of scenarios that include both benign and attack conditions, such as Denial of Service (DoS), Cryptojacking, and Backdoor attacks. The dataset includes EVSE's power consumption data, network traffic logs, and host activities. It aims to support research in anomaly detection and behavioral profiling using machine learning. The dataset is organized into three main directories: Network Traffic, Host Activities, and Power Consumption. Data set details are as in Table 1.

**Table 1.** Dataset Column Description

| Column ID | Description |
|---|---|
| Time | Data reading date, |
| State | Idle state for EVSE-B, Represents the situation where there is no V2G communication (i.e. no connection to EVCC or ISO15118 communication). Conversely, state of charge refers to the state in which the EVCC is connected and actively communicating. |
| **Scenario** | Includes attack scenarios (Recon, DoS, Cryptojacking, Backdoor, Benign) |
| Attack | It refers to attack types (Cryptojacking, Backdoor, None (ie. Benign), tcp-port-scan, service-version-detection, os-fingerpriting, aggressive-scan, syn-stealth-scan, vulnerability-scan, slowloris-scan, upd-flood, icmp-flood, pshack-flood, icmp-fragmentation, tcp-flood, syn-flood, synonymousIP-flood) |
| Interface | It refers to the interface of VSE-B that is targeted by the malicious actor during network attacks. |
| Label | Attack Status |

## 2.2. Feature Selection

Selecting features in data sets is an important step to increase the performance and interpretability of the model [6]. The main purpose of feature extraction is a widely used method to extract information from relevant data [7]. In this context, two common methods, the filter method using correlation matrix and the Recursive Feature Elimination (RFE) method, will be discussed.

### 2.2.1. Filter Method with Correlation Matrix

Correlation matrix is a method that measures the linear relationship between features. Especially features with high correlation (usually greater than 0.95) may cause the model to overfit. In this case, it may be useful to remove some of the highly correlated features.

The filter method using correlation matrix follows these steps:

1. Calculating the Correlation Matrix: Calculates the correlation coefficients between all features.
2. Creating the Upper Triangle Matrix: Repetitive calculations are avoided by focusing only on the correlation values remaining in the upper triangle.
3. Identification of Highly Correlated Features: Features whose correlation value is above a certain threshold value (for example 0.95) are detected.
4. Extraction of These Features: Detected highly correlated features are removed from the data set.

In addition to being simple and fast, this method also provides advantages in terms of interpretability. However, it can ignore non-linear relationships.

### 2.2.2. Wrapper Method: Recursive Feature Elimination (RFE)

Recursive Feature Elimination (RFE) is a wrapper method that evaluates the performance of a given model and selects the features that have the greatest impact.

The steps of RFE are:

1. Training the Model: The model is trained with all features.
2. Calculating the Importance Levels of Features: The importance of each feature in the model is calculated.
3. Least Important Feature Removal: The least important feature is removed and the model is retrained.
4. Repeat of this Process: This process is repeated until the desired number of features is reached.

## 2.3. GAN Architecture

GAN (Generative Adversarial Network) is an artificial intelligence and machine learning model developed by Ian Goodfellow and his team [8] in 2014. GAN generates new and realistic data samples from datasets using competitive learning between two neural networks.

These two neural networks:

1. Generator: This network takes random noise or latent space as input and produces data. Its aim is to produce data that is as close to real data as possible.
2. Discriminator: It tries to distinguish whether the samples produced are real data or fake. This network takes real data and Generator generated data as input and tries to separate the two.

These two networks compete with each other. While the generator tries to fool the discriminator, the discriminator tries to recognize the fake data. In this process, both networks constantly improve themselves [9].

## 2.4. Classifiers

**Random Forest**

Random Forest is an algorithm used in classification and regression problems. It creates an ensemble feature by combining multiple decision trees. While one tree has

limited prediction ability, a combination of many trees has more stable prediction ability [10].

**Gradient Boosting**

Gradient Boosting Machines (GBM) is a powerful algorithm that creates a powerful classifier by iteratively training a set of decision tree classifiers and optimizing them over time [11]. GBM aims to achieve high accuracy by improving the performance of the model with each iteration. This is done by combining multiple weak learners into a single strong learner.

**Support Machine Learning**

It is an algorithm that aims to separate data belonging to two different classes in a linear or non-linear way. It is especially preferred in large data sets and it is possible to get fast results [12].

**Deccission Tree**

It is one of the hierarchical supervised learning models [14]. One of the main hierarchical models is the decision tree. It has two categories: classification tree and regression tree.

# 3. Results and Discussion

We evaluated the performance of various machine learning models for classification tasks, focusing on their accuracy, specificity, sensitivity, precision, and F1 score. Models compared include Logistic Regression, Random Forest, Decision Tree, Support Vector Machine, Gradient Boosting, and our Proposed Model.

In order to reduce the computational cost and increase the performance, a hybrid feature selection algorithm was used on the dataset using the Correlation matrix and RFE methods intertwined. GAN was applied to the data set with the selected features.

## *3.1. Data Augmentation*

In this study, a GAN-based data augmentation method was used to expand our data set and increase the performance of our model. It is modeled with 3 types of methods: GAN, DCGAN and WGAN. GANs are powerful deep learning models that can generate new and realistic data using a dataset.

### *3.1.1 GAN*

The table below summarizes the hyperparameters used in our GAN model.

**Table 2.** List of GAN Hyperparameter

| Hyperparameter | Description | Value |
|---|---|---|
| **Noise Dimension** | Dimension of the random noise vector for the generator input | 100 |
| **Data Dimension** | Dimension of the input data for the discriminator | X_rfe.shape[1] |
| **Discriminator** | | |
| - Dense Layer 1 | Number of units | 512 |
| - Dense Layer 2 | Number of units | 256 |
| - Dense Layer 3 | Number of units | 128 |
| - Activation Function | Activation function for intermediate layers | LeakyReLU (alpha=0.01) |
| - Output Activation | Activation function for the output layer | Sigmoid |
| - Loss Function | Loss function used | Binary Crossentropy |
| - Optimizer | Optimizer used | Adam |
| - Metrics | Evaluation metric | Accuracy |
| **Generator** | | |
| - Dense Layer 1 | Number of units | 128 |
| - Dense Layer 2 | Number of units | 256 |
| - Dense Layer 3 | Number of units | 512 |
| - Output Layer | Number of units | data_dim (data dimension) |
| - Activation Function | Activation function for intermediate layers | LeakyReLU (alpha=0.01) |
| - Output Activation | Activation function for the output layer | Tanh |
| **GAN** | | |
| - Loss Function | Loss function used | Binary Crossentropy |
| - Optimizer | Optimizer used | Adam |
| **Training** | | |
| - Epochs | Number of epochs for training | 100 |
| - Batch Size | Size of each batch during training | 64 |
| - Print Interval | Frequency of training status updates | 100 |

In our GAN application, we used the noise size of 100 for the generator input, which is standard in many GAN models. The discriminator was designed with three dense layers of alternating units (512, 256, and 128) and used LeakyReLU activation functions in the intermediate layers and a sigmoid activation function in the output layer. The generator was similarly designed with three dense layers (128, 256, and 512 units) and used a tanh activation function in the output layer to generate data that matched the actual data distribution. Accuracy optimization was achieved by compiling the discriminator with a binary cross-entropy loss function and an Adam optimizer. The generator and the GAN itself also used binary cross-entropy and Adam for loss optimization. The GAN model was trained for 100 epochs with a batch size of 64. The model's performance and training status were monitored at regular intervals to provide information about the training progress and model effectiveness.

### *3.1.2. DCGAN*

The table 3 below summarizes the hyperparameters used in our DCGAN model.

**Table 3.** List of DCGAN Hyperparameter

| Hyperparameter | Description | Value |
|---|---|---|
| Noise Dimension | Size of the random noise vector fed into the generator. | 100 |
| Learning Rate (lr) | Step size used by the optimizer during gradient descent. | 0.0002 |
| Batch Size (batch_size) | Number of samples processed before updating internal parameters of the model. | 64 |
| Number of Epochs (epochs) | Number of complete passes through the training dataset. | 100 |
| Generator Dropout Rate | Dropout rate for input units in the generator layers. | 0.3 |
| Leaky ReLU Slope | Slope of the Leaky ReLU activation function in the negative region. | 0.2 |
| Adam Optimizer Beta1 | Beta1 value for momentum of the Adam optimizer used for the generator. | 0.5 |
| Adam Optimizer Beta2 | Beta2 value for momentum of the Adam optimizer used for the generator. | 0.999 |

### *3.1.3. WGAN*

The table 4 below summarizes the hyperparameters used in our WGAN model.

**Tablo 4.** List of WGAN Hyperparameter

| Hyperparameter | Description | Value |
|---|---|---|
| Noise Dimension (noise_dim) | Size of the random noise vector fed into the generator. | 100 |
| Learning Rate (lr) | Step size used by the optimizer during gradient descent. | 0.00005 |
| Batch Size (batch_size) | Number of samples processed before updating internal parameters of the model. | 64 |
| Number of Epochs (epochs) | Number of complete passes through the training dataset. | 10000 |
| Number of Critic Iterations (n_critic) | Number of times the critic is updated per generator update. | 5 |
| Weight Clipping Parameter (clip_value) | Range within which the critic's weights are clipped to enforce Lipschitz continuity. | 0.01 |
| Dropout Rate | Dropout rate for input units in the critic layers. | 0.3 |

## *3.2. Result of Machine Learning Models*

### *3.2.1 Result of Machine Learning Models with GAN*

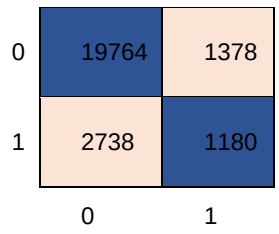


**Figure 1.** Confusion Matrix of Logistic Regresyon

When the matrix in Figure 1 is examined, 20944 out of a total of 25060 data in the data set were classified correctly, while 4116 were classified incorrectly. Of the non-attack data, 19764 were classified correctly and 1378 were classified incorrectly. 1180 of the attacked data were classified correctly and 2738 were classified incorrectly.

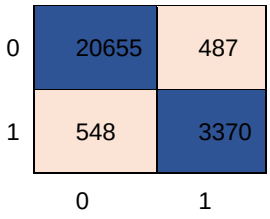


**Figure 2.** Confusion Matrix of Decission Tree

When the matrix in Figure 3 is examined, 24025 of the total 25060 data in the data set were classified correctly, while 1035 were classified incorrectly. Of the non-attack data, 20655 were classified correctly and 487 were classified incorrectly. Of the attacked data, 3370 were classified correctly and 548 were classified incorrectly.

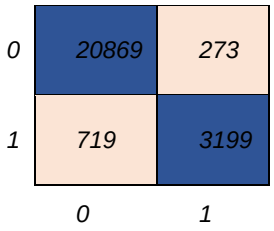


**Figure 3.** Confusion Matrix of Random Forest

When the matrix in Figure 3 is examined, 24088 out of a total of 25060 data in the data set were classified correctly, while 992 were classified incorrectly. Of the non-attack data, 20869 were classified correctly and 273 were classified incorrectly. 3199 of the attacked data were classified correctly and 719 were classified incorrectly.

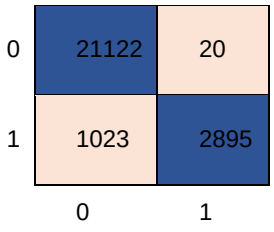


**Figure 4.** Confusion Matrix of SVM

When the matrix in Figure 4 is examined, 24027 of the total 25060 data in the data set were classified correctly, while 1043 were classified incorrectly. Of the non-attack data, 21142 were classified correctly and 20 were classified incorrectly. Of the attacked data, 2895 were classified correctly and 1023 were classified incorrectly.

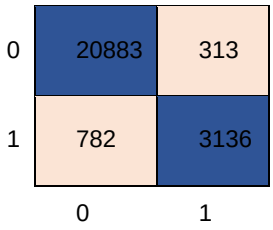


**Figure 5.** Confusion Matrix of Gradient Boosting

When the matrix in Figure 5 is examined, 24019 out of a total of 25060 data in the data set were classified correctly, while 1095 were classified incorrectly. Of the non-attack data, 20883 were classified correctly and 313 were classified incorrectly. 3136 of the attacked data were classified correctly and 782 were classified incorrectly.

### *3.2.2 Result of Machine Learning Models with DCGAN*

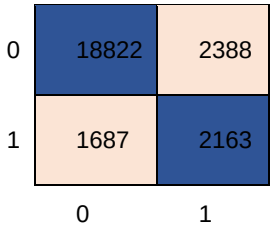


**Figure 6.** Confusion Matrix of Logistic Regresyon with DCGAN

When the matrix in Figure 6 is examined, 20985 out of a total of 25060 data in the data set were classified correctly, while 4075 were classified incorrectly. Of the non-attack data, 18822 were classified correctly and 2388 were classified incorrectly. 2163 of the attacked data were classified correctly and 1687 were classified incorrectly.

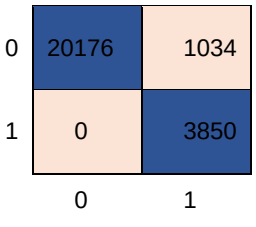


**Figure 7.** Confusion Matrix of Decission Tree with DCGAN

When the matrix in Figure 7 is examined, 24026 of the total 25060 data in the data set were classified correctly, while 1034 were classified incorrectly. Of the non-attack data, 20176 were classified correctly and 1034 were classified incorrectly. All of the attacked data classified correctly.

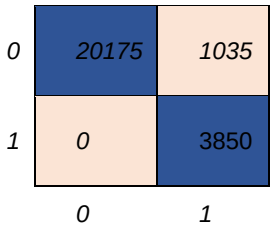


**Figure 8.** Confusion Matrix of Random Forest with DCGAN

When the matrix in Figure 8 is examined, 24025 out of a total of 25060 data in the data set were classified correctly, while 1035 were classified incorrectly. Of the non-attack data, 20175 were classified correctly and 1035 were classified incorrectly. All of the attacked data classified correctly.

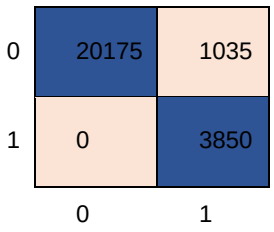


**Figure 9.** Confusion Matrix of SVM with DCGAN

When the matrix in Figure 9 is examined, 24025 out of a total of 25060 data in the data set were classified correctly, while 1035 were classified incorrectly. Of the non-attack data, 20175 were classified correctly and 1035 were classified incorrectly. All of the attacked data classified correctly.

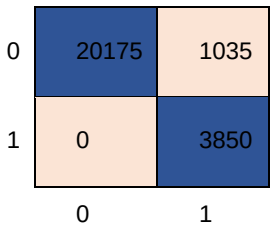


**Figure 10.** Confusion Matrix of Gradient Boosting with DCGAN

When the matrix in Figure 8 is examined, 24025 out of a total of 25060 data in the data set were classified correctly, while 1035 were classified incorrectly. Of the non-attack data, 20175 were classified correctly and 1035 were classified incorrectly. All of the attacked data classified correctly.

### *3.2.3 Result of Machine Learning Models with WGAN*

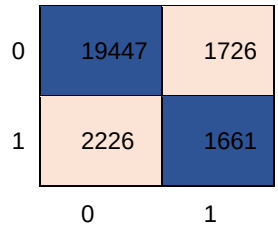


**Figure 11.** Confusion Matrix of Logistic Regresyon with WGAN

When the matrix in Figure 11 is examined, 21108 out of a total of 25060 data in the data set were classified correctly, while 3952 were classified incorrectly. Of the non-attack data, 19447 were classified correctly and 1726 were classified incorrectly. 1661 of the attacked data were classified correctly and 2226 were classified incorrectly.

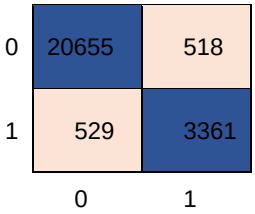


**Figure 12.** Confusion Matrix of Decission Tree with WGAN

When the matrix in Figure 12 is examined, 24016 of the total 25060 data in the data set were classified correctly, while 1047 were classified incorrectly. Of the non-attack data, 20655 were classified correctly and 518 were classified incorrectly. Of the attacked data, 3361 were classified correctly and 529 were classified incorrectly.

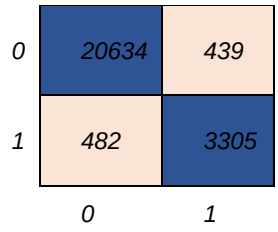


**Figure 13.** Confusion Matrix of Random Forest with WGAN

When the matrix in Figure 13 is examined, 23365 out of a total of 25060 data in the data set were classified correctly, while 921 were classified incorrectly. Of the non-attack data, *20634* were classified correctly and 429 were classified incorrectly. 3305 of the attacked data were classified correctly and 482 were classified incorrectly.

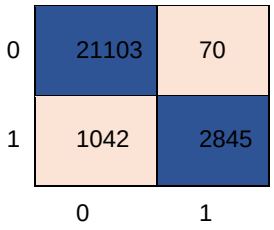


**Figure 14.** Confusion Matrix of SVM with WGAN

When the matrix in Figure 14 is examined, 23948 of the total 25060 data in the data set were classified correctly, while 1112 were classified incorrectly. Of the non-attack data, 21103 were classified correctly and 70 were classified incorrectly. Of the attacked data, 2845 were classified correctly and 1042 were classified incorrectly.

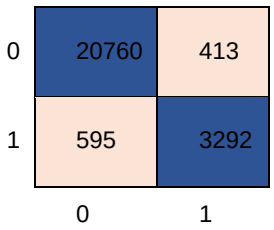


**Figure 15.** Confusion Matrix of Gradient Boosting with WGAN

When the matrix in Figure 15 is examined, 24052 out of a total of 25060 data in the data set were classified correctly, while 1008 were classified incorrectly. Of the non-attack data, 20760 were classified correctly and 413 were classified incorrectly. 3292 of the attacked data were classified correctly and 595 were classified incorrectly

## *3.3 Result of Proposed Model*

In order to further increase the performance of the Random Forest Model, which has the highest accuracy rate among the results obtained with machine learning, machine learning models were run with the newly formed data set by applying binary feature selection to the GAN applied dataset in the Proposed Model.

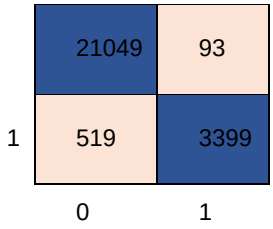


**Figure 16.** Confusion Matrix of Proposed Model

When the matrix in Figure 6 is examined, 24448 out of a total of 25060 data in the data set were classified correctly, while 612 were classified incorrectly.

Of the non-attack data, 21049 were classified correctly and 93 were classified incorrectly. 519 of the attacked data were classified correctly and 93 were classified incorrectly.

**Table 4.** Accuracy Values of All Models

| | **Acc. (%)** | **Spec. (%)** | **Sens. (%)** | **Pre. (%)** | **F1 (%)** |
|---|---|---|---|---|---|
| Logistic Regresyon+GAN | 83.58 | 46.13 | 87.83 | 93.48 | 90.57 |
| Random Forest+GAN | 96.04 | 92.14 | 96.67 | 98.71 | 97.68 |
| Decission Tree+GAN | 95.87 | 87.37 | 97.42 | 97.70 | 97.56 |
| SVM+GAN | 95.84 | 99.31 | 95.38 | 99.91 | 97.59 |
| Gradient Boosting+GAN | 95.64 | 90.92 | 96.39 | 98.52 | 97.45 |
| Logistic Regresyon+DCGAN | 83.74 | 4753 | 9177 | 8874 | 90.23 |
| Random Forest+DCGAN | **95.87** | **78.81** | 100 | 95.12 | 97.50 |
| Decission Tree+DCGAN | 95.87 | 78.83 | 100 | 95.12 | 97.50 |
| SVM+DCGAN | 95.87 | 78.81 | 100 | 95.12 | 97.50 |
| Gradient Boosting+DCGAN | 95.87 | 78.81 | 100 | 95.12 | 97.50 |

| | | | | | |
|---|---|---|---|---|---|
| Logistic Regresyon+WCGAN | 84.23 | 49.04 | 89.73 | 91.85 | 90.78 |
| Random Forest+WCGAN | 95.91 | 85.98 | 97.72 | 9745 | 97.59 |
| Decission Tree+WCGAN | 95.82 | 86.65 | 97.50 | 97.55 | 97.53 |
| SVM+DCGAN | 95.56 | 9760 | 95.29 | 99.67 | 97.43 |
| Gradient Boosting+WCGAN | 95.98 | 88.85 | 97.21 | 98.05 | 97.63 |
| **Proposed Model** | **97.56** | **97.34** | **97.59** | **99.56** | **98.57** |

Performance measurements show that our Proposed Model outperforms all other models on many evaluation criteria. Specifically, the Proposed Model achieved the highest accuracy of 97.56%, significantly outperforming Logistic Regression (83.58%) and other advanced models such as Random Forest (96.04%) and SVM (95.84%). showed improvements.

In terms of specificity, SVM showed the highest performance at 99.31%; this demonstrates his strong ability to accurately identify negative examples. However, our Proposed Model also showed a high specificity of 97.34%; This is commendable considering its balanced performance across all metrics.

Sensitivity is another important metric where our Proposed Model excels; Scoring 97.59%, it is higher than Logistic Regression sensitivity (87.83%) and comparable to top-performing models such as Random Forest (96.67%) and Decision Tree (97.42%).This demonstrates the robustness of the Proposed Model in identifying positive examples.

Sensitivity and F1 score are critical to understanding the balance between sensitivity and positive predictive value. The Proposed Model achieved the highest sensitivity of 99.56% and the highest F1 score of 98.57%. These measurements highlight the model's efficiency in providing reliable and consistent performance by maintaining high recall while minimizing false positives.

# 4. Conclusion

It clearly shows that our Proposed Model provides superior performance compared to traditional models and other advanced machine learning techniques. Having the highest scores in terms of accuracy, precision and F1 score, the Proposed Model proves to be highly effective and reliable for classification tasks.

The outstanding performance of our Proposed Model can be attributed to its advanced algorithmic design and robust training process, which allows it to process complex datasets with higher precision and recall. These findings show that the Proposed Model can be used effectively in practical applications where high accuracy and reliability are very important.

Overall, our Proposed Model sets a new benchmark in classification performance, offering significant improvements over existing methodologies and paving the way for more accurate and reliable machine learning applications.

# 5. Limitation and Future Work

Bu konunun incelenmesinde araştırma veri setinin toplanması çeşitliliği ilgili verilerin gizliliğinden dolayı veri kısıtı bulnmaktadır.

In the future, it is planned to conduct in-depth research on specific communication technologies vulnerabilities with more advanced data sets, focusing on types of cyber attacks.

# Acknowledgement

The present paper does not include any research with human participants conducted by any of the authors.

Ethical Approval

This manuscript does not contain any studies with human participants carried out by any of the authors.

Conflict of Interest

There is no conflict of interest for authorship.

# References

**[1] Acharya**, **S.,** Y. Dvorkin, H. Pandžić and R. Karri, (2020), Cybersecurity of Smart Electric Vehicle Charging: A Power Grid Perspective, IEEE Access, vol. 8, pp. 214434-214453,

**[2] ElKashlan M**, Elsayed MS, Jurcut AD, Azer M. A, (2023), Machine Learning-Based Intrusion Detection System for IoT Electric Vehicle Charging Stations (EVCSs). Electronics., 12(4):1044.

**[3**] **Ali Arsalan**, Laxman Timilsina, Behnaz Papari, Grace Muriithi, Gokhan Ozkan, Phani Kumar, Christopher S. Edrington, (2023), Cyber Attack Detection and Classification for Integrated On-board Electric Vehicle Chargers subject to Stochastic Charging Coordination, Transportation Research Procedia, Volume 70, Pages 44-51, ISSN 2352-146

**[4] Dustin Kern**, Christoph Krauß, and Matthias Hollick. (2023), Detection of Anomalies in Electric Vehicle Charging Sessions. In Proceedings of the 39th Annual Computer Security Applications Conference (ACSAC '23). Association for Computing Machinery, New York, NY, USA, 298–309.

**[5]** https://www.unb.ca/cic/datasets/evse-dataset-2024.html

**[6] Güven, Z. A**. (2024). Ses analizi ile müzik türlerinin sınıflandırılmasına yönelik kapsamlı bir çalışma. Niğde Ömer Halisdemir Üniversitesi Mühendislik Bilimleri Dergisi, 13(1), 325-333. https://doi.org/10.28948/ngumuh.1344605

**[7] Parhi, K. K**., and Ayinala, M. (2013). Low-complexity Welch power spectral density computation. IEEE Transactions on Circuits and Systems I: Regular Papers, 61(1), 172-182.

**[8] Goodfellow, I**., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial nets. Advances in neural information processing systems, 27.

**[9]** https://www.geeksforgeeks.org/generative-adversarial-**network-gan/**

**[10] Tanyıldız, H**., Şahin, C. B., & Dinler, Ö. B. Effective Cyber Attack Detection Based on Augmented Genetic Algorithm with Naive Bayes. NATURENGS, 4(2), 30-35.

**[11] Natekin, A**., & Knoll, A. (2013). Gradient boosting machines, a tutorial. Frontiers in neurorobotics, 7, 21**.**

**[12] Khan, S. N.,** Khan, S. U., Aznaoui, H., Şahin, C. B., & Dinler, Ö. B. (2023). Generalization of linear and non-linear support vector machine in multiple fields: a review. Computer Science and Information Technologies, 4(3), 226-239.

**[13] https://www.unb.ca/cic/datasets/evse-dataset-2024.html**

**[14] Suthaharan, S**., & Suthaharan, S. (2016). Decision tree learning. Machine Learning Models and Algorithms for Big Data Classification: Thinking with Examples for Effective Learning, 237-269.

**[15] Malik, R**., Singh, Y., Sheikh, Z. A., Anand, P., Singh, P. K., & Workneh, T. C. (2022). [Retracted] An Improved Deep Belief Network IDS on IoT-Based Network for Traffic Systems. Journal of Advanced Transportation, 2022(1), 7892130.

**[16] Basnet, M**., & Ali, M. H. (2020, September). Deep learning-based intrusion detection system for electric vehicle charging station. In 2020 2nd International Conference on Smart Power & Internet Energy Systems (SPIES) (pp. 408-413). IEEE.