Robust Anomaly Detection in IoT Networks using Deep SVDD and Contractive Autoencoder

Sharmin Aktar
Department of Computer Science
University of New Orleans
New Orleans, LA, USA 70148
Email: saktar@uno.edu

Abdullah Yasin Nur
Department of Computer Science
University of New Orleans
New Orleans, LA, USA 70148
Email: ayn@cs.uno.edu

***Abstract*—Securing Internet of Things (IoT) devices against threats is crucial due to their significant impact on cyber-physical systems. Traditional intrusion detection systems often fall short in protecting the vast and diverse array of IoT devices. One key limitation is their lack of an anomaly detection objective, which is essential for identifying sophisticated threats that do not match known patterns. To address this critical gap, we have introduced a unique approach that utilizes an objective-based anomaly detection model. Our model, integrating a Deep Support Vector Data Description (DSVDD) with a Contractive Autoencoder (CAE), named DSVDD-CAE, enhances the relevance of latent representations for anomaly detection and thereby improves accuracy. This innovative combination has significantly outperformed popular anomaly detection algorithms like KMeans, OCSVM, and Isolation Forest. On the ToN-IoT dataset, our method achieved a precision of 98.77%, a recall of 99.74%, an F1-score of 99.25%, and an accuracy of 99.57%. Similarly, on the IoTID20 dataset, it reached a precision of 98.25%, a recall of 99.80%, an F1-score of 99.01%, and an accuracy of 99.64%. These results demonstrate that our model excels in accurately detecting both known and novel IoT attacks, thereby significantly advancing the field of IoT security and providing a more resilient cyber-physical ecosystem.**

***Index Terms*—Internet of Things, IoT, Anomaly Detection, Deep Learning, Autoencoder, Contractive Autoencoder, Support Vector Data Description, DSVDD-CAE, IoT Security, Intrusion Detection**

I. Introduction

The Internet of Things (IoT) represents a network of interconnected devices that enable continuous communication between physical devices [1]. The rapid evolution of these devices has led to the development of smart ecosystems. However, this has attracted the attention of cybercriminals, making IoT vulnerable to potential malicious attacks. Among the most notable threats are the denial of service (DoS) and distributed denial of service (DDoS) attacks, which aim to overload the capacity of the network gateway [24]. For example, the Mirai botnet attack in 2016 [8] and the Reaper botnet discovered in 2017 [9], both targeted IoT devices and caused significant disruptions, highlighting the urgent need for effective IoT intrusion detection systems (IDS) that can detect both known and zero-day attacks [1]. Traditional IDSes mostly rely on predefined rules or signatures making them effective against known threats, but they struggle to detect novel or complex attacks [10]. Furthermore, the high dimensionality and heterogeneity of IoT data present significant challenges for these traditional detection methods, often resulting in a high number of false alarms.

To address these challenges, researchers have adopted machine learning techniques for developing IDS. However, most of the traditional machine learning models follow shallow learning methods, failing to detect the attack from an enormous dataset. Recently, deep learning based IDS methods have grabbed significant attention since they are beneficial for large and complex networks due to their ability to extract more sophisticated features. Among these, the autoencoders based techniques have emerged as a promising approach owing to their capability to learn a compact representation of the data, which can be used to detect anomalies or attacks [12].

In this paper, we propose a novel deep learning model for IoT attack detection based on contractive autoencoders and deep support vector data description (DSVDD), whose objective function is based on anomaly detection. The objective function guides the learning process of the model, encouraging it to bring similar data points close together in a sphere-like region while keeping them distinct in the underlying space. This helps in effectively distinguishing between normal and unusual data points [21]. Our model leverages the ability of autoencoders to learn a compact representation of the data and the robustness of DSVDD to outliers, making it effective at detecting both known and novel attacks. By combining this anomaly detection-based objective function with techniques that learn a compact representation of the data and are robust to outliers, our model becomes effective at detecting both known and new types of attacks. We evaluate our model on benchmark IoT datasets and show that it outperforms existing methods in terms of state-of-the-art evaluation metrics.

The key contributions of this paper are as follows:

- We propose a novel DSVDD-CAE model for detecting attacks in IoT networks, combining the robustness of contractive autoencoders and the anomaly detection capabilities of Deep SVDD.
- We employ a semi-supervised training approach that uses only normal data. This approach enables the model to identify anomalies based on their deviation from the normal representation.

- We implement a stochastic approach for optimal threshold selection, enhancing the model's accuracy in anomaly detection.
- We validate effectiveness of our model using two different datasets: the ToN-IoT dataset [19] and the IoTID20 dataset [23]. The results demonstrate superior performance compared to other state-of-the-art machine learning models on both datasets.

The rest of the paper is arranged as follows. In Section II, various IoT attack detection methods have been discussed. Section III provides our methodology. Our experimental results have been demonstrated in Section IV. Lastly, Section V concludes the paper.

II. Related Work

In today's digital age, the Internet of Things (IoT) has created a new ecosystem of interconnected devices, which is frequently being utilized by many organizations to enhance their performance and make more informed decisions [15]. However, the surge in Internet of Things (IoT) devices and applications introduces security vulnerabilities, necessitating the development of advanced anomaly detection methods. Over recent years, machine learning and artificial intelligence have emerged as promising tools for enhancing security in IoT environments.

In the early stages of this field, Kolias et al. [7] conducted research on the Mirai botnet, one of the initial botnets to utilize Internet of Things (IoT) technology. They emphasized some of the unique difficulties associated with IoT devices, including their heterogeneity and the significant amount of network traffic that is generated. Their innovative work marked the beginning of using machine learning techniques to detect IoT-originated attacks. Building on earlier work, Booij et al. [16] highlighted the importance of diversity within the intrusion detection datasets in the IoT domain. They introduced the ToN-IoT dataset, which has a variety of features and attack types. They also developed a new method to compare datasets using cross-training classifiers, showing the necessity of diverse configuration requirements for detecting network intrusions in IoT.

Meidan et al. [14] introduced a network-based detection technique called N-BaIoT, which uses deep autoencoders to detect IoT botnet attacks. The authors infected nine commercial IoT devices in their lab with two of the most widely known IoT-based botnets, Mirai and BASHLITE1. Their method extracts behavior snapshots of the network and trains deep autoencoders to learn the usual behaviors of IoT devices. When the autoencoder fails to reconstruct a sample, anomalies are detected by indicating a deviation from the normal behavior. This approach has demonstrated a significant contribution to the IoT intrusion detection field.

Moustafa et al. [13] developed a realistic botnet dataset named Bot-IoT, which includes both legitimate and simulated IoT network traffic along with various types of attacks. They also presented a realistic testbed environment to address the limitations of existing datasets. The reliability of their dataset was evaluated using statistical and machine learning methods for forensic purposes, providing a baseline for identifying botnets in IoT-specific networks. The authors emphasized the significance of developing intelligent Intrusion Detection Systems (IDS) for IoT devices, considering the expanding attack surface and the rising frequency of attacks on IoT platforms.

Khraisat et al. proposed a novel ensemble Hybrid Intrusion Detection System (HIDS), which combines the benefits of both the Signature Intrusion Detection System (SIDS) and the Anomaly-based Intrusion Detection System (AIDS), to improve IoT device security [17]. The system's performance was assessed using the Bot-IoT dataset, where it showed higher detection rates and fewer false positives compared to conventional IDS methods.

Even with significant advancements in IoT security, the necessity for a model that can accurately detect both known and unknown attacks continues to be a critical requirement. A key limitation in many existing techniques is the lack of an anomaly detection objective. This paper proposes a novel model that addresses this gap by incorporating an anomaly detection objective, combining the strengths of Deep Support Vector Data Description (DSVDD) and Contractive Autoencoder (CAE). Employing a semi-supervised approach, our model identifies anomalies by analyzing the reconstruction error and the distance to the center of a hypersphere in the latent space. This methodology enables our model to accurately detect both known and novel attacks, enhancing IoT security by addressing the limitations of existing methods, which often go unnoticed by traditional AI tools.

III. Proposed Methodology

In this section, we present the core components of our intrusion detection model. This model, named the Deep Support Vector Data Description based on Contractive Autoencoder (DSVDD-CAE), combines the characteristics of contractive autoencoders with Deep Support Vector Data Description.

A. Contractive Autoencoder

An Autoencoder is a type of neural network designed to encode inputs into a compressed form and then decode it to reconstruct the original data. The objective is to extract the essential features of the data in the compacted form. The autoencoder consists of two parts: an encoder that maps the input data into a low-dimensional latent space, and a decoder that rebuilds the input data from this latent representation [2].

An autoencoder is trained to minimize the difference between the original input and its reconstruction, which is called the reconstruction error. This is typically measured using a loss function such as the mean squared error. By minimizing this loss, the autoencoder learns useful features of the input data through the encoding process, enabling it to reconstruct the original input accurately.

The objective function of the autoencoder can be represented as:

$$J_{AE}(\theta) = \frac{1}{n}\sum_{i=1}^{n} ||x_i - x_i'||^2, \quad (1)$$

where n is the number of data points, x_i is the i-th input data point, and x_i' is the i-th reconstructed data point.

While a basic autoencoder focuses on reconstructing the input by keeping the reconstruction loss as less as possible, a contractive autoencoder is another type of autoencoder that is designed to learn a robust representation of the input data. It introduces a penalty term to the standard reconstruction loss function of the autoencoder, which is called the Frobenius norm of the Jacobian matrix of the encoder activations with respect to the input. This encourages the model to learn a representation of the input data that is invariant to small changes in the input [3].

The objective function of a contractive autoencoder can be represented as:

$$J_{CAE}(\theta) = \frac{1}{n}\sum_{i=1}^{n} ||x_i - x_i'||^2 + \lambda||J_f(x)||_F^2, \quad (2)$$

where n is the number of data points, x_i is the i-th input data point, x_i' is the i-th reconstructed data point, $||J_f(x)||_F^2$ is the squared Frobenius norm of the Jacobian matrix of the encoder activations with respect to the input, and λ is a hyperparameter that controls the trade-off between the reconstruction error and the robustness of the representation.

B. Support Vector Data Description (SVDD)

Support Vector Data Description, typically referred to as SVDD, is a one-class classification method that aims to distinguish data by placing it within a hypersphere in a designated feature space [20]. The purpose is to have regular data points inside this hypersphere and any unusual or outlier points outside of it. The primary objective of SVDD is to find the most compact or smallest hypersphere, characterized by its center $c \in F_k$ and radius R, that encompasses the majority of the usual data in the feature space F_k.

Given a dataset $D_n = \{x_1, x_2, \ldots, x_n\}$, the mathematical representation of SVDD's objective is to minimize [20]:

$$\min_{c,R,\xi} R^2 + \sum_{i=1}^{n} \gamma\xi_i \quad (3)$$

Subject to the constraint:

$$\|u(x_i) - c\|^2 \leq R^2 + \xi_i, \forall i \quad (4)$$

In the above formulations:

- The notation $\|\cdot\|$ denotes the Euclidean norm which measures the distance between two points in the feature space.
- The variables ξ_i are known as slack variables, introduced to provide flexibility to the hypersphere boundary and handle potential outliers or anomalies in the data.
- The parameter γ is a regularization term that controls the trade-off between the minimization of the hypersphere radius and the penalty incurred by the data points that lie outside the hypersphere.

Fig. 1 provides a visual representation of the SVDD method. The normal samples presented in green reside within the depicted hypersphere, defined by radius R. In contrast, anomalies are marked in red, positioned outside the hypersphere boundary, indicating their deviation from the usual data distribution.

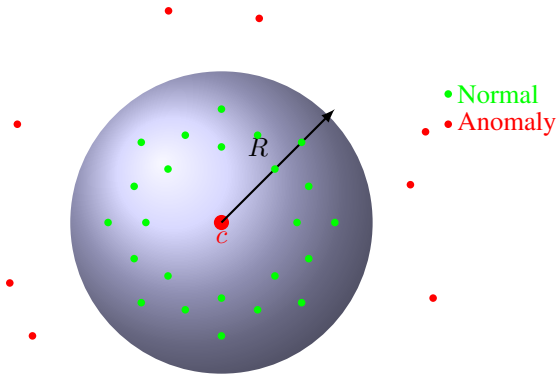


Fig. 1: Visualization of SVDD hypersphere with normal and anomalous data points.

C. Deep SVDD with Contractive Autoencoding (DSVDD-CAE)

In this work, we have integrated the principles of Contractive Autoencoders (CAE) with Deep Support Vector Data Description (DSVDD) to introduce a novel approach, DSVDD-CAE, for anomaly detection. This model is designed to learn a compact and robust representation of the input data in a low-dimensional latent space and to detect anomalies based on the distance of the data points to the center of a hypersphere in this latent space.

The DSVDD-CAE model leverages the strengths of both the contractive autoencoder and Deep SVDD. The contractive autoencoder part of the model learns a robust representation of the input data that is invariant to small changes in the input, which is achieved by adding a regularization term to the loss function of the standard autoencoder. This encourages the autoencoder model to generate robust representations by mapping similar inputs to similar points in the latent space.

The Deep SVDD aspect of the model, unlike traditional SVDD that operates in the original feature space, introduces a hypersphere in the latent space and employs a deep neural network to map the data into a low-dimensional latent space, making it more suitable for handling high-dimensional data [4]. The center and radius of this hypersphere are learned during the training process. The goal of Deep SVDD is to map the data into a low-dimensional latent space and capture the majority of the data within this hypersphere. This can be formulated as:

$$J_{DSVDD}(\theta) = R^2 + \frac{1}{\nu n}\sum_{i=1}^{n} \max(0, ||f(x_i) - c||^2 - R^2), \quad (5)$$

where $f(x_i)$ is the i-th encoded input, c and R represent the center and radius of the hypersphere. This function encourages the model to map the usual data points close to the center of the hypersphere in the latent space. The integration of Deep SVDD aids in effectively dealing with high-dimensional data and ensuring that the model captures the underlying structure of the data in a compact latent space, which in turn facilitates more accurate anomaly detection.

The objective function of our DSVDD-CAE model can be represented as:

$$J_{DSVDD-CAE}(\theta) = J_{CAE}(\theta) + \alpha J_{DSVDD}(\theta), \quad (6)$$

where α is a hyperparameter that controls the trade-off between the two objectives, contractive autoencoder and Deep SVDD.

We have trained our DSVDD-CAE model using a semi-supervised learning approach using only normal data. The model jointly learns to reconstruct the normal data with a low reconstruction error and to map the normal data points close to the center of the hypersphere in the latent space. Since the model has not utilized any anomalous instance during training, it generates a high reconstruction error and maps those points far from the center of the hypersphere. This combined difference of reconstruction error and distance to the center of the hypersphere is used to classify data points as normal or anomalous. We evaluated the model's performance using a separate test set containing a mixture of both normal and anomalous samples. The model's performance has been measured using metrics such as accuracy, precision, recall, and F1-score.

D. Anomaly Detection using DSVDD-CAE

The proposed anomaly detection model utilizes a Deep SVDD model with a Contractive AutoEncoder (DSVDD-CAE). The model operates in three main steps as outlined in Algorithm 1:

1. **Anomaly Score Calculation**: The first stage, outlined in lines 17-23 of Algorithm 1, involves the calculation of an anomaly score. This score measures how much a data point deviates from it's normal behavior. It is calculated using two factors: the distance from the center of the hypersphere in the latent space, and the reconstruction error. The formula for this computation is:

$$s(x_i) = ||z_i - c||^2 + \alpha \times \text{MSE}(x_i, \hat{x}_i), \quad (7)$$

Here, z_i represents the encoded form of x_i as seen in line 18 of Algorithm 1, c denotes the center of the hypersphere, $\hat{x}_i$ is the reconstructed data point as computed in line 20 of Algorithm 1, and α is a hyperparameter that controls the balance between the two terms. The MSE term, computed in line 21 of Algorithm 1, represents the mean squared error between the original and reconstructed data points, termed as the reconstruction error.

2. **Threshold Selection**: As outlined in lines 7-16 of Algorithm 1, this step focuses on finding an optimal threshold.

Algorithm 1 Anomaly Detection using DSVDD-CAE

1: **function** PREDICT(xtrain, xtest, xval, yval)
2: score-normal ← ComputeAnomalyScore(xtrain)
3: threshold ← GetOptimalThreshold(xval, yval, score-normal)
4: anomaly score ← ComputeAnomalyScore(xtest)
5: is anomaly ← anomaly score ≥ threshold
6: **return** is anomaly
7: **function** GETOPTIMALTHRESHOLD(xval, yval, score-normal)
8: Define a range of potential_thresholds from min(score-normal) to max(score-normal)
9: best_f1 ← 0
10: best_threshold ← potential_thresholds[0]
11: **for** each threshold in potential_thresholds **do**
12: Predict anomalies for xval based on current threshold to get predicted_yval
13: Compute F1 score using actual yval and predicted_yval
14: **if** current F1 score > best_f1 **then**
15: Update best_f1 and best_threshold with current F1 score and threshold
16: **return** best_threshold
17: **function** COMPUTEANOMALYSCORE(x)
18: z ← EncodeInput(x)
19: distance ← ComputeDistance(z)
20: x recon ← DecodeInput(z)
21: recon error ← ComputeReconstructionError(x recon, x)
22: anomaly score ← CombineScores(distance, recon error)
23: **return** anomaly score

The selection of this threshold is crucial as it decides if a data point is an anomaly based on its anomaly score. This step is performed using validation data, which has both normal and anomalous samples. Anomaly scores are calculated for each data point in this validation set.

The optimal threshold is derived by analyzing the F1-score across a range of potential thresholds, which is established from the minimum to the maximum score observed in normal data (score-normal), as demonstrated in line 8 of Algorithm 1. The F1-score, combining precision and recall, is essential in binary classification analysis. Precision is the ratio of true positive results among all classified as positive, while recall is the proportion of true positive results among all actual positive instances (see Section IV-C for more details). For each threshold within this range, the F1-score is calculated on the validation set, which is a mix of normal and anomalous samples. By iterating through this range, the algorithm identifies the threshold that maximizes the F1-score, ensuring a balanced trade-off between precision and recall, which is crucial for the effective detection of anomalies.

3. **Anomaly Detection**: Lines 1-6 of Algorithm 1 present

the anomaly detection method, which employs the trained model and chosen threshold to detect anomalies. For any given data point, the model initially encodes it into a lower-dimensional space using DSVDD-CAE, followed by decoding to reconstruct the original input. Subsequent computations determine the reconstruction error and the distance to the center of the hypersphere. A data point is categorized as an anomaly if its calculated anomaly score meets or exceeds the threshold; otherwise, it is classified as normal.

This procedure establishes a robust method for anomaly detection in data, utilizing both distance-based and reconstruction-based approaches.

IV. Experiments

In this section, we summarize the dataset utilized for our experiments, explain the evaluation metrics of our proposed model's performance, detail our experimental setup, and present the experimental results.

A. Dataset

We conducted our experiments using two datasets: the TON-IoT dataset [19] and the IoTID20 dataset [23]. These datasets are designed to simulate realistic network traffic patterns in IoT environment, including both normal and anomalous behaviors.

The TON-IoT dataset [19] comprises network traffic data gathered from various IoT and IIoT sensors, along with system trace datasets from Linux and Windows systems. We specifically utilize the Windows 10 subset in this paper, which was gathered using the Performance Monitor Tool on Windows 10 systems. This subset records a variety of activities, including desk, processor, memory, process, and network activities from the Windows 10 systems, comprising a total of 124 features with seven distinct types of attack.

The IoTID20 dataset, proposed by Ullah and Mahmoud [23], is another comprehensive dataset for evaluating intrusion detection systems in IoT networks. It includes a significant set of features with corresponding weights and provides a foundation for developing new intrusion detection techniques. The dataset was generated using a well-designed testbed architecture that mimics the complexity and scalability of industrial IoT networks. It includes several normal and cyber-attack events from network traffic, making it an ideal choice for our experiments.

In the following sections, we will explain how we utilized these datasets in our experimental setup and discuss the results obtained.

B. Experimental Setup

We have utilized PyTorch as the deep learning framework for our model. The experiment was conducted on a system equipped with an 11th Gen Intel(R) Core(TM) i7-1185G7 @ 3.00 GHz 1.80 GHz and 16 GB RAM. During the preprocessing phase, both datasets underwent several key transformations, including the removal of duplicate records, normalization of feature values, and elimination of redundant features.

TABLE I: Hyperparameters used in the DSVDD_CAE model

Hyperparameter	**Value**
Input Dimension[1]	124/75
Hidden Dimension	64
Latent Dimension	32
Batch Size	32
Learning Rate	0.001
Epochs	10
Optimizer	Adam
Lambda (Contractive loss coefficient)	0.0001
Alpha (Sphere-Reconstruction loss weighting)	0.1
Momentum (for updating c)	0.1

[1] The Input Dimension is 124 for ToN-IoT and 75 for IoTID20 dataset.

The dataset was then partitioned into three sections: training, validation, and testing subsets. The training subset consists solely of normal samples (labeled as 0). For the validation and testing subsets, we ensured a balance with 90% normal samples (labeled as 0) and 10% anomalous samples (labeled as 1) [22]. The DSVDD-CAE model was configured with an input dimension aligning with the number of features in the respective dataset (124 features for TON-IoT and 75 features for IoTID20), a hidden dimension of 64, and a latent dimension of 32. The detailed hyperparameters are presented in Table I.

In the training phase of the DSVDD-CAE model, the center c and radius R of the hypersphere are initialized at the onset and updated after each epoch. Initially, the center c is set to the mean of the latent representations of the training data, and the radius R is established to encompass all data points in the training set by setting it to the maximum distance from c to any data point [4].

During the training, the center c is updated iteratively to better adapt to the data distribution. Specifically, after each epoch, a new center c_{new} is computed as the mean of the latent representations of the data. The updating formula for the center is given by:

$$c_{\text{new}} = (1-\text{momentum})c_{\text{old}}+\text{momentum}\cdot\text{mean}(\text{latent representations})$$

where the momentum hyperparameter (set to 0.1 as per Table I controls the rate of updating c, providing a balance between stability and adaptability to new data distributions [4].

The model's learning process is further refined by the Learning Rate, Epochs, and Optimizer hyperparameters, set to 0.001, 10, and Adam respectively in Table I. These parameters control the speed of learning, the number of iterations, and the optimization algorithm used to minimize the loss function.

The Contractive loss coefficient and Sphere-Reconstruction loss weighting, represented by Lambda and Alpha hyperparameters respectively in Table I, are crucial for the computation of the overall loss during training [3]. These values, 0.0001 for Lambda and 0.1 for Alpha, help in balancing the contribution of the contractive loss and the reconstruction loss, ensuring the

model learns a robust representation of the data. The radius (R) is set to the maximum distance of any data point from the initialized center (c), ensuring that all training data points in the latent space are surrounded.

C. Evaluation Metrics

In our research, we have adopted several evaluation metrics to assess the performance of our anomaly detection models. Given the nature of anomaly detection, where the number of anomalous samples is typically smaller compared to the normal samples, traditional metrics such as accuracy may not provide an ideal evaluation [22]. Therefore, we have chosen to use the following metrics:

- **Area Under the Receiver Operating Characteristics curve (AUC)**: The AUC measures the entire two-dimensional area underneath the entire ROC curve from (0,0) to (1,1). AUC provides an aggregate measure of performance across all possible classification thresholds. It is particularly useful when we need to compare different models.
- **Precision**: Precision measures the proportion of predicted positives that are actually positive. It is calculated as $Precision = \frac{TP}{TP+FP}$. High precision indicates a low rate of false alarms.
- **Recall**: Also known as sensitivity or TPR, recall measures the proportion of actual positives that are correctly identified. It is calculated as $Recall = \frac{TP}{TP+FN}$.
- **Accuracy**: Despite its limitations in this context, accuracy can still provide some insights. It measures the proportion of total predictions that are correct. It is calculated as $Accuracy = \frac{TP+TN}{TP+FP+TN+FN}$.
- **F1-score**: The F1-score combines precision and recall into one metric by using their harmonic mean. It provides a balanced measure that equally weighs the significance of both precision and recall, offering a comprehensive view of model performance. It is calculated as $F1-score = \frac{2*Precision*Recall}{Precision+Recall}$.

These metrics together provide a comprehensive evaluation of our anomaly detection model's performance.

Method	Precision	Recall	F1-score	Accuracy
KMeans	84.77%	72.19%	76.28%	88.92%
OCSVM	91.34%	76.38%	81.35%	91.29%
Isolation Forest	86.86%	83.52%	77.89%	89.68%
Proposed Method	**98.77%**	**99.74%**	**99.25%**	**99.57%**

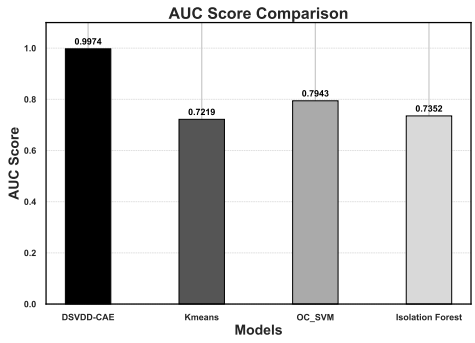
TABLE II: Comparison of evaluation metrics for different methods on the TON-IoT dataset.

D. Results and Discussion

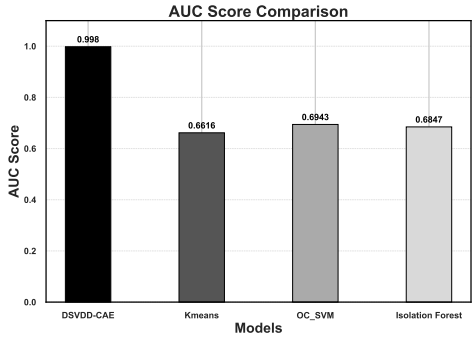
We evaluated our DSVDD-CAE model on two distinct datasets, TON-IoT and IoTID20, to determine its effectiveness in detecting anomalies across different IoT networks. The evaluation metrics—accuracy, precision, recall, and F1-score—highlight the model's robust performance on both datasets. Specifically, on the TON-IoT dataset, our model achieved an accuracy of 99.57%, precision of 98.77%, recall of 99.74%, and F1-score of 99.25%, as tabulated in Table II. Similarly, on the IoTID20 dataset, our model exhibited an accuracy of 99.64%, precision of 98.25%, recall of 99.80%, and F1-score of 99.01%, as detailed in Table III.

Method	Precision	Recall	F1-score	Accuracy
KMeans	66.13%	66.16%	66.15%	87.81%
OCSVM	69.40%	69.43%	69.41%	88.98%
Isolation Forest	70.51%	68.95%	69.68%	89.50%
Proposed Method	**98.25%**	**99.80%**	**99.01%**	**99.64%**

TABLE III: Comparison of evaluation metrics for different methods on the IoTID20 dataset



(a) AUC Score Comparison (TON-IoT dataset)

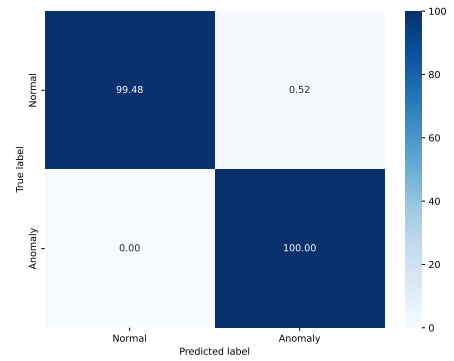


(b) AUC Score Comparison (IoTID20 Dataset)

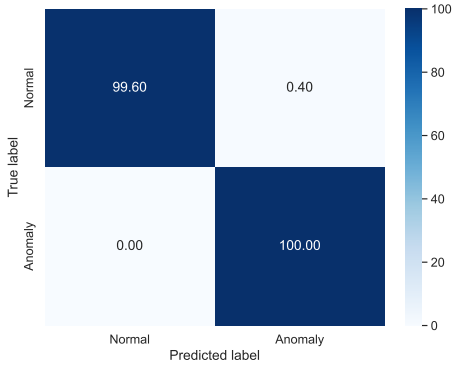
Fig. 2: AUC Score Comparisons for both datasets

In addition to these primary metrics, we calculated the AUC score for both datasets. Given that our evaluation dataset contains only 10% anomalies, the AUC score effectively measures our model's performance, minimizing the influence of class imbalance between normal and anomalous samples.

We conducted a comparative analysis against commonly used anomaly detection methods like KMeans [25], Isolation Forest [26], and OC-SVM [27]. Our DSVDD-CAE model significantly outperformed these methods, achieving higher F1-scores of 99.25% and 99.01% on the TON-IoT and IoTID20 datasets respectively, as shown in Tables II and III. The superiority of our model is further illustrated in Figure 2, which compares the AUC scores among different methods. On the IoTID20 dataset, our model's performance was approximately



(a) Confusion Matrix for ToN-IoT Dataset



(b) Confusion Matrix for IoTID20 Dataset

Fig. 3: Confusion Matrices for Anomaly Detection on ToN-IoT and IoTID20 Datasets

43.75% ahead of the next best method, OCSVM. Similarly, for the TON-IoT dataset, our model showcased an improvement of roughly 25.58% over OCSVM.

Furthermore, we evaluated our model using confusion matrices for both datasets, illustrated in Figures 3a and 3b. For the TON-IoT dataset, the confusion matrix reveals a true positive rate of 100%, a true negative rate of 99.48%, and a slight false positive rate of 0.52%. Conversely, for the IoTID20 dataset, our model achieved a true positive rate of 100% and a true negative rate of 99.60%, with a false positive rate of 0.40%. These results highlight our model's capability in accurately identifying both normal and anomalous instances across different IoT datasets, further substantiating the earlier discussed precision, recall, F1-score, and accuracy values.

V. Conclusion

In this paper, we present a novel Deep Learning-based Intrusion Detection model based on an anomaly detection objective to detect anomalies in IoT networks. Our proposed framework, DSVDD-CAE, is a combination of Deep Support Vector Data Description and Contractive Autoencoder which can efficiently model normal data and detect anomalies deviating from the usual pattern. The experimental results demonstrate the effectiveness of our approach on two different datasets: the ToN-IoT and IoTID20. On the ToN-IoT dataset, our model achieved a precision of 98.77%, recall of 99.74%, F1-score of 99.25%, and an accuracy of 99.57%. Similarly, on the IoTID20 dataset, the proposed model reached a precision of 98.25%, recall of 99.80%, F1-score of 99.01%, and an accuracy of 99.64%. The comparison among other baseline models ensures the superiority of our model's performance, showcasing its potential in significantly advancing IoT security through proficient anomaly detection. Moreover, our model exhibits robustness by maintaining superior performance across different datasets, whereas traditional methods do not show the same level of robustness, emphasizing the advantages of our DSVDD-CAE model in diverse IoT security scenarios.

References

[1] Khraisat, A. and Gondal, I. and Vamplew, P. and Kamruzzaman, J. and Alazab, A., *A Novel Ensemble of Hybrid Intrusion Detection System for Detecting Internet of Things Attacks*, Electronics, Volume 8, Number 11, Page 1210, 2019. https://doi.org/10.3390/electronics8111210

[2] Goodfellow, Ian and Bengio, Yoshua and Courville, Aaron, *Deep Learning*, MIT Press, 2016.

[3] Rifai, Salah and Vincent, Pascal and Muller, Xavier and Glorot, Xavier and Bengio, Yoshua, *Contractive auto-encoders: Explicit invariance during feature extraction*, Proceedings of the 28th International Conference on Machine Learning (ICML-11), pages 833–840, 2011.

[4] Ruff, Lukas and Vandermeulen, Robert and Goernitz, Nico and Deecke, Lucas and Siddiqui, Shoaib A and Binder, Alexander and Müller, Emmanuel and Kloft, Marius, *Deep one-class classification*, International Conference on Machine Learning, pages 4393–4402, 2018.

[5] Chalapathy, Raghavendra and Chawla, Sanjay, *Deep learning for anomaly detection: A survey*, arXiv preprint arXiv:1901.03407, 2019.

[6] Atzori, Luigi and Iera, Antonio and Morabito, Giacomo, *The Internet of Things: A survey*, Computer Networks, 2010.

[7] Kolias, Constantinos and Kambourakis, Georgios and Stavrou, Angelos and Voas, Jeffrey, *DDoS in the IoT: Mirai and other botnets*, Computer, 2017.

[8] Antonakakis, Manos and April, Tim and Bailey, Michael and Bernhard, Matt and Bursztein, Elie and Cochran, Jaime and Durumeric, Zakir and Halderman, J. Alex and Invernizzi, Luca and Kallitsis, Michalis and others, *Understanding the Mirai Botnet*, 26th USENIX Security Symposium (USENIX Security 17), 2017.

[9] Chen, Yu-An and Hwang, Ren-Hung, *IoT security: ongoing challenges and research opportunities*, Service Oriented System Engineering (SOSE), 2018 IEEE Symposium on, 2018.

[10] Sommer, Robin and Paxson, Vern, *Outside the closed world: On using machine learning for network intrusion detection*, IEEE Symposium on Security and Privacy, 2010.

[11] Vinayakumar, R and Soman, KP and Poornachandran, Prabaharan and Sachin Kumar, S and Akarsh, S and Alazab, Mamoun, *Applying convolutional neural network for network intrusion detection*, International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2019.

[12] Aktar, Sharmin and Nur, Abdullah Yasin, *Towards DDoS attack detection using deep learning approach*, Computers & Security, 2023.

[13] Koroniotis, Nickolaos and Moustafa, Nour and Sitnikova, Elena and Turnbull, Benjamin, *Towards the Development of Realistic Botnet Dataset in the Internet of Things for Network Forensic Analytics: Bot-IoT Dataset*, University of New South Wales Canberra, Australia, 2019.

[14] Meidan, Yair and Bohadana, Michael and Mathov, Yael and Mirsky, Yisroel and Breitenbacher, Dominik and Shabtai, Asaf and Elovici, Yuval, *N-BaIoT: Network-based Detection of IoT Botnet Attacks Using Deep Autoencoders*, IEEE Pervasive Computing, volume 17, number 3, pages 12–22,

[15] Catillo, M and Pecchia, A and Villano, U, *A Deep Learning Method for Lightweight and Cross-Device IoT Botnet Detection*, Applied Sciences, volume 13, number 2, pages 837, 2023. https://doi.org/10.3390/app13020837

[16] Booij, Tim and Chiscop, Irina and Meeuwissen, Erik and Moustafa, Nour, *ToN IoT: The Role of Heterogeneity and the Need for Standardization of Features and Attack Types in IoT Network Intrusion Datasets*, IEEE Internet of Things Journal, DOI: 10.1109/JIOT.2021.3085194.

[17] Khraisat, Ansam and Gondal, Iqbal and Vamplew, Peter and Kamruzzaman, Joarder, *Hybrid Intrusion Detection System for Enhanced Security of IoT Devices*, Electronics, volume 8, number 12, pages 1210, 2019.

[18] Vu, Ly and Cao, Van Loi and Nguyen, Quang Uy and Nguyen, Diep N. and Hoang, Dinh Thai, *Learning Latent Representation for IoT Anomaly Detection*, IEEE Transactions on Systems, Man, and Cybernetics, 2020.

[19] N. Moustafa. (2020). TON-IoT Dataset. [Online]. Available:https://cloudstor.aarnet.edu.au/plus/s/ds5zW91vdgjEj9i

[20] Tax, D.M., Duin, R.P. Support Vector Data Description. Machine Learning 54, 45–66 (2004). https://doi.org/10.1023/B:MACH.0000008084.60811.49

[21] Zhou, Yu and Liang, Xiaomin and Zhang, Wei and Zhang, Linrang and Song, Xing, *VAE-based Deep SVDD for anomaly detection*, Neurocomputing, Elsevier, 2021.

[22] Kale, R. and Lu, Z. and Fok, K. W. and Thing, V. L. L., *A Hybrid Deep Learning Anomaly Detection Framework for Intrusion Detection*, 2022 IEEE 8th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing, (HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS), Jinan, China, pages 137-142, 2022. https://doi.org/10.1109/BigDataSecurityHPSCIDS54978.2022.00034

[23] Ullah, Imtiaz and Mahmoud, Qusay H., *A scheme for generating a dataset for anomalous activity detection in iot networks*, Canadian Conference on Artificial Intelligence, Springer International Publishing, 2020.

[24] Aktar, S. and Nur, A. Y., *Hash Based AS Traceback against DoS Attack*, 2021 4th International Conference on Advanced Communication Technologies and Networking (CommNet), Rabat, Morocco, 2021, pp. 1-7.

[25] MacQueen, J., *Some methods for classification and analysis of multivariate observations*, Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, Volume 1, Number 14, Page 281-297, 1967. https://projecteuclid.org/euclid.bsmsp/1200512992

[26] Liu, F. T. and Ting, K. M. and Zhou, Z., *Isolation Forest*, Proceedings of the 2008 Eighth IEEE International Conference on Data Mining, Page 413-422, 2008. https://doi.org/10.1109/ICDM.2008.17

[27] Schölkopf, B. and Platt, J. C. and Shawe-Taylor, J. and Smola, A. J. and Williamson, R. C., *Estimating the Support of a High-Dimensional Distribution*, Neural Computation, Volume 13, Number 7, Page 1443–1471, 2001. https://doi.org/10.1162/089976601750264965