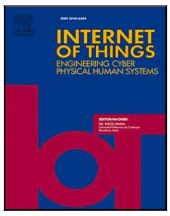
Internet of Things 33 (2025) 101680

Contents lists available at ScienceDirect

Internet of Things

journal homepage: www.elsevier.com/locate/iot



Research article

Novel deep learning-based IoT network attack detection using magnet loss optimization

Chi Duc Luu [a], Viet Hung Nguyen [a,*], Van Quan Nguyen [a], Ngoc-Son Vu [b]

[a] Le Quy Don Technical University, Hanoi, Viet Nam
[b] ETIS - ENSEA, CY Cergy Paris University, CNRS, Cergy, France

ARTICLE INFO

Keywords:
IoT attack detection
Latent representation
Deep learning
Deep Autoencoder
Intrusion detection system

ABSTRACT

The increasing prevalence of Internet of Things (IoT) devices across various industries has raised critical security concerns due to their inherent vulnerabilities and high interconnectivity. While traditional security mechanisms have shown limitations in effectively securing large IoT networks, machine learning (ML) and deep learning (DL) methods have been explored to tackle the attack detection problem in this domain. However, existing approaches still lack optimal regularization and have limited comprehensiveness in validation across different IoT-centric datasets. To address these challenges, this research proposes the extension of the Deep Magnet Autoencoder (DMAE) and introduces a novel approach, the Cascade Deep Magnet Autoencoder (CDMAE), leveraging the Magnet Loss optimization as regularization for better class distinction through local separation in latent space. This enhanced class clustering strengthens attack detection by maximizing inter-class separation while compactly grouping data points of the same class, leading to more precise identification of benign and malicious traffic. Extensive experiments conducted on three contemporary IoT datasets, CIC-BoT–IoT, CIC-ToN–IoT, and CICIoT2023, demonstrate that our proposed models are able to produce meaningful latent representations with powerful discrimination between benign and malicious IoT network data. Empirical insights for fine-tuning the model are also provided through supplementary experiments. Comprehensive results show that the proposed methods significantly boost classification across different IoT datasets with high metric scores, outperforming other approaches.

1. Introduction

As technology continues to revolutionize daily lives, IoT devices have permeated a wide range of industries as well as daily life applications. Due to the extensive use in many different sectors, IoT security is becoming more crucial and has shown a bigger significance [1]. From industries devices, such as healthcare monitors, security cameras, to everyday appliances such as refrigerators and doorbells, IoT has become deeply embedded in the everyday routine of human society.

Compared to traditional computing systems, IoT environments provide a new challenge for cybersecurity. While there have been improvements in both aspects of hardware and software, there still exists a lack of processing power and protection limitations. Moreover, the variety of IoT devices on the market also presents a problem due to varying architectures and behaviors. This is coupled with the scale of IoT networks, which often consist of multiple devices, resulting in high-volume data flows [2]. Furthermore, services provided in IoT environments have definite differences when compared to those running on the Internet as they are dependent on applications while having no true standardization [3], which further emphasizes the vulnerabilities. IoT sensors

* Corresponding author.
E-mail address: hungnv@lqdtu.edu.vn (V.H. Nguyen).

https://doi.org/10.1016/j.iot.2025.101680
Received 14 March 2025; Received in revised form 9 May 2025; Accepted 8 June 2025
Available online 27 June 2025

and devices are also prone to be affected by many different elements such as manufacturing or installation errors, environmental interferences, or random errors, resulting in inconsistencies when mapping data [4]. Additionally, IoT network systems possess the characteristic of having high interconnectivity [5]. For this reason, exploitation on IoT networks can have an impact on a large scale with far-reaching negative consequences.

Therefore, the problem of security in IoT networks is a major growing concern. Classic security implementations such as encryption, authentication, and rule-based methods have proven to be difficult and insufficient for larger networks with many interconnected IoT devices. To address different aspects of limitations in IoT, optimizations approaches such as scalable neurocomputing-assisted cloud–edge system [6] and distributed optimization with event-triggered mechanisms alongside homographic encryption [7] have also been explored to improve resilience and readiness of networks in adverse conditions. Alternatively, machine learning (ML) and deep learning (DL) have been widely explored as novel solutions for security events detection [8]. Traditional ML algorithms, such as Support Vector Machine (SVM) and Random Forest (RF), as well as enhancement techniques, such as the use of data normalization and Principal Component Analysis (PCA), are some of the methods to address attack detection [9,10].

Improving upon the traditional ML approaches, DL models have shown to have strong capabilities in processing large datasets while demonstrating better performance [11]. Various approaches have been proposed, such as CNN-based models [12,13] with positive results. Shi et al. put forward the use of Recurrent Neural Network (RNN) as a mean to better deal with sequential data in network environments [14]. RNN demonstrates stable performance and robustness in a variety of simulations [15]. However, a popular DL-based approach for IoT network attack detection is the employment of Deep Autoencoder (DAE) models [16–18]. With the ability to learn and extract meaningful representation from input data, the DAE approach has proven to be a promising approach, with many models achieving high results, by enhancing detection potential through intermediate abstraction.

While these DL-based approaches have demonstrated strong advancements in IoT network attack detection, several challenges persist, limiting their effective potential. First, many methods lack a strong regularizer for better data separation, indicating room for optimization. Second, within the popular datasets used in previous works, many are dated or serve as stand-ins instead of being data from actual IoT-centric networks. Moreover, extensive research to validate methods across different IoT-centric datasets is still limited. Since each IoT system consists of a range of diverse devices, network flows are inherently different between different datasets. Therefore, there is a gap on the comprehensiveness of IoT attack detection methods. Third, the motivations for specific configurations of models are not yet clearly defined, with many being intuition-based.

To address these limitations, we aim to propose a detection method that can be comprehensive across different up-to-date IoT environments. Specifically, we propose an approach that can help enhance IoT attack detection throughout various datasets with high class separation ability. Our approach extends the proposed Deep Magnet Autoencoder (DMAE) model from [19] and expands upon the design to introduce another novel model called Cascade Deep Magnet Autoencoder (CDMAE). Through the design with combined magnet loss, this approach aims to create a clear and distinct latent representation space that can significantly boost the classifiers' capabilities to detect IoT network attacks. Magnet loss optimization is a loss function developed for metric learning, which has shown prevalence in recent years in domains such as computer vision. For anomaly detection, there have been adaptations of metric learning, such as contrastive learning [17,20] and triplet loss [21,22] to improve latent representation. However, while contrastive loss works on pairs and triplet loss calculates on triplets of samples, they can introduce inconsistencies between different terms and are computationally inefficient due to the formulation of pairs and triplets. On the other hand, magnet loss operates on the entire cluster, which can better capture the context of the neighborhood structure as well as having faster convergence rate with better efficiency [23]. To comprehensively evaluate our approach based on magnet loss, we then conducted extensive experiments on three different IoT datasets: CIC-BoT–IoT, CIC-ToN–IoT, and CICIoT2023. Further additional tests are also carried out to better provide empirical guidance for model configuration.

In summary, the key contributions of this paper are as follows:

- We propose the extension of Deep Magnet Autoencoder with magnet loss optimization for IoT network attack detection. The proposed approach showed enhancements in classification through creating a distinct and meaningful class separation with meaningful latent representation.
- We further introduce a novel Cascade Deep Magnet Autoencoder model for IoT network attack detection. This proposed model also displayed superior capabilities when combining with certain classifiers, and outperformed other methods.
- We conducted extensive experiments on three different datasets, CIC-BoT–IoT, CIC-ToN–IoT, and CICIoT2023, to evaluate the effectiveness of the proposed methods compared with other approaches. Results indicate that the performance of ML classifiers was enhanced through the latent representation produced by our methods.
- We conducted supplementary experiments to provide empirical recommendations for fine-tuning the proposed method.

The paper is organized into the following sections. Section 2 explains other studies related to our research while Section 3 goes into detail about the background knowledge. Section 4 describes our proposed methodology. Section 5 then explains the experiments process and Section 6 shows the corresponding results and discussion. Finally, we provide the conclusion and recommendations for future research.

2. Related works

DL models have emerged as a promising approach for attack detection, capable of automatically learning complex patterns in traffic data. Many DL architectures, including RNN, Convolutional Neural Networks (CNNs), and DAE, have been explored for network intrusion detection. Various surveys were conducted on DL for network attack detection and results indicate that DL approaches are capable in both traditional and IoT network systems, while outperforming classical ML methods [24–26].

A RNN-based detection method was proposed by Yin et al. in [27]. Experiments were conducted for both binary and multiclass classification on the NSL-KDD dataset, showing high accuracy and reduced false positive rate when compared with traditional ML classification models. A method based on LSTM was introduced in [28] where multiple modules are merged into an ensemble of detectors. High accuracy was achieved against a real-world dataset of Modbus network traffic. Different hybrid approaches are also deployed to address the IoT network attack detection. Another variation was presented by combining RNN and Long Short-Term Memory (LSTM) to classify attacks from normal data [29], with promising detection capabilities compared to other approaches.

CNN is another technique for network attack detection, with many proposals showing promising performance for both binary and multiclass classification [30]. A novel approach was proposed in [31] where CNN was used in conjunction with deep neural network as embedding functions in a few-shot learning scenario, showing the capabilities of DL in extracting features from network data. Kan et al. proposed a novel CNN-based approach with Adaptive Particle Swarm Optimization for IoT network intrusion detection [12]. Results show both reliable and accurate performance. Another CNN-based model called Soft Ordering Convolutional Neural Network approach was proposed and tested in [32]. Extensive experiments were conducted on three different datasets, BoT–IoT, CIC-IDS-2017, and CIC-IDS-2018. Results show high F1 Scores and robustness in IoT network environment with strong detection capabilities for Denial of Service (DoS), Distributed DoS (DDoS).

Parra et al. have proposed the use of distributed DL utilizing both CNN and LSTM models to enhance capabilities through different levels of detection in IoT systems [33]. Another hybrid system was proposed where a CNN–LSTM framework was tested on the IoT-23 dataset consisting of device traffic captured from real Raspberry Pi devices [34]. Comprehensive results across different attacks show promising outcomes, outperforming other approaches in accuracy.

Delving into the characteristics of network intrusion data, Zhou et al. introduced a novel framework called Hierarchical Dependency and Class Imbalance for addressing inherent problems of hierarchical dependency omission and decision boundary discontinuity [35]. This framework consists of three components, semantic attribute embedding, oversampling, and classification with Multilayer Perceptron (MLP). The proposed model shows superior performance with understanding of the network attributes hierarchy while addressing data imbalance with high fidelity across different public datasets. Another study by Lin et al. also focused on the characteristics of traffic analysis, specifically from both input and output explainability perspectives [36]. Detection performance measured across MLP, CNN, and LSTM shows that protocol-specific explainable input features which strongly correlate with attack behavior has an enhancing effect on models' performances.

Transformer-based network intrusion detection models have also been leveraged in recent years. Utilizing the long-term characteristics and patterns capturing capability of the transformer, a modular framework proposed by Manocchio et al. demonstrated high detection rate across different datasets, showing potential for transformer-based approaches [37]. A Robust Transformer-based Intrusion Detection System was introduced by Wu et al. [38]. Evaluation on different CICFlowMeter-based datasets indicate superior performance compared to mainstream algorithms while solving overfitting problem and data imbalance. However, the speed of the transformer model remains a challenge. Another robust and privacy preserving approach utilizing unsupervised federated hypernetwork with Series Conversion Normalization Transformer was also proposed for anomaly detection and diagnosis by Hao et al. [39]. Extensive experimental scenarios across nine different datasets shows a strong contender for effective time-series anomaly diagnosis in different servers and monitoring environments.

As a different representation method, graph neural network (GNN) have also been implemented for network intrusion detection. Lo et al. developed E-GraphSAGE, a novel GNN-based detection system capable of capturing both edge features and topological information of IoT network flows [40]. Evaluation across four different datasets demonstrate strong potential for this technique in both binary and multiclass classification. Another approach called GNN-IDS was proposed by Sun et al. [41]. The model construct graphs that are based in both static and dynamic attributes of the network through attack graph and real-time measurements. Evaluation results show strong detection certainty with better demonstration for explainability and robustness. Pujol-Perich et al. introduced a GNN based model which is able to learn host-connection graph representation [42]. This approach helps capture structural network flow characteristics, showing competitive results on the CIC-IDS2017 dataset while having high tolerance against adversarial attacks when compared with state-of-the-art ML models.

For network attack detection, the implementation of DAE is also another popular technique. A method using DAE trained on benign data to detect IoT network attacks, specifically IoT-based botnets, was introduced by Meidan et al. in [43]. The approach was tested on nine commercial IoT devices of the N-BaIoT dataset and shows the promising ability of DAE to accurately detect the botnet attacks. A Stacked Sparse Autoencoder approach was proposed by Yan et al. to improve intrusion detection systems [44]. Results in both binary and multiclass classification indicate strong discriminatory ability and efficiency.

Implementing the Memory-Augmented Deep Autoencoder (MemAE) for network anomaly detection proved to be another effective approach. Min et al. in [45] conducted experiments using the NSL-KDD, UNSW-NB15, and CICIDS2017 datasets with imbalanced data to reflect real world attack environments. Results indicate a method that has good performance while solving the over-generalization problem of the traditional DAE. Leveraging a memory module to enhance pattern storage, the Cognitive Memory-guided Autoencoder introduced in study [46] utilized both feature reconstruction loss and feature sparsity loss. The method promotes better discrimination and diversity, outperforming many other existing approaches.

Table 1
Summary of research gap in related works.

Work	Proposed Approach	Dataset	Research Gap
[12]	Adaptive Swarm Optimization CNN	N-BaIoT	• Lack of latent regularization • Lack of datasets comprehensiveness
[17]	Deep Sparse Contrastive Autoencoder	N-BaIoT	• Lack of datasets comprehensiveness
[18]	Regularized DAE	N-BaIoT	• Lack of datasets comprehensiveness
[19]	DMAE	CIC-IDS-2017, CSE-CIC-IDS-2018	• Not modern IoT-oriented
[27]	RNN	NSL-KDD	• Lack of latent regularization • Lack of datasets comprehensiveness • Not modern IoT-oriented
[28]	Ensemble LSTM	Modbus	• Lack of latent regularization • Lack of datasets comprehensiveness
[29]	LSTM-RNN	NSL-KDD	• Lack of latent regularization • Lack of datasets comprehensiveness • Not modern IoT-oriented
[31]	Few-shot learning DNN/CNN	NSL-KDD, UNSW-NB15	• Not modern IoT-oriented
[32]	Soft Ordering CNN	BoT–IoT, CIC-IDS-2017, CIC-IDS-2018	• Lack of latent regularization • Lack of attacks comprehensiveness
[33]	Distributed DNN and Cloud-based temporal LSTM	N-BaIoT, PhishTank, OpenPhish	• Lack of latent regularization
[34]	CNN–LSTM	IoT-23	• Lack of latent regularization • Lack of datasets comprehensiveness
[35]	Hierarchical Dependency and Class Imbalance with MLP	NSL-KDD, UNSW-NB15, AWID2, CIC-IDS-2017, NF-BoT	• Lack of latent regularization • Not modern IoT-oriented
[36]	MLP, CNN, LSTM with explainable inputs and outputs	DoS/DDoS-MQTT-IoT, Mqttset, WDT	• Lack of latent regularization
[37]	FlowTransformer	NSL-KDD, UNSW-NB15, CSE-CIC-IDS2018	• Lack of latent regularization • Not modern IoT-oriented
[38]	Robust Transformer-based Intrusion Detection System	CIC-IDS-2017, CIC-DDoS2019	• Lack of latent regularization • Not modern IoT-oriented
[39]	Unsupervised federated hypernetwork with SC Nor-Transformer	UCR, NAB, MBA, SWaT, SMD, SMAP, PSM, MSL	• Lack of latent regularization
[40]	GNN-based E-GraphSAGE	BoT–IoT, ToN–IoT, NF-ToT-IoT, NF-BoT–IoT	• Lack of latent regularization
[41]	GNN-IDS	Synthetic datasets	• Lack of latent regularization • Not modern IoT-oriented
[42]	GNN	CIC-IDS-2017	• Lack of latent regularization • Lack of datasets comprehensiveness • Not modern IoT-oriented
[43]	DAE	N-BaIoT	• Lack of latent regularization • Lack of datasets comprehensiveness
[44]	Stacked Sparse Autoencoder	NSL-KDD	• Lack of datasets comprehensiveness • Not modern IoT-oriented
[45]	MemAE	NSL-KDD, UNSW-NB15, CIC-IDS-2017	• Not modern IoT-oriented
[46]	Cognitive Memory-guided Autoencoder	KDDCUP	• Lack of datasets comprehensiveness • Lack of datasets comprehensiveness
[47]	LSTM-Autoencoder	BoT–IoT	• Lack of datasets comprehensiveness • Lack of datasets comprehensiveness
[48]	Improved Conditional Variational Autoencoder-DNN	NSL-KDD, UNSW-NB15	• Not modern IoT-oriented

Popoola et al. introduced a LSTM Autoencoder for latent representation in conjunction with a Deep Bidirectional LSTM classifier [47]. The model demonstrates strong feature dimensionality reduction capabilities while still being able to maintain high metrics. An Improved Conditional Variational Autoencoder model for intrusion detection was proposed by Yang et al. [48], which boasted superior performance compared to many other approaches, even when faced with the imbalanced data problem. Vu et al. proposed the use of regularized DAE to produce latent feature space that can enhance classifiers in [18]. Results show that the proposed regularized DAE has the capability to boost the performance of linear classifiers with the constructed latent representation.

Magnet loss optimization in metric learning was integrated into the DAE model by Tang et al. as a mean to produce better separation between benign and malicious traffic [19]. This creates a strong website attacks detection method through enhancing classic ML classifiers with meaningful latent representation. Another novel approach to the DAE was also introduced with the

implementation of contrastive learning in a combined loss function. The Deep Sparse Contrastive Autoencoder was presented by Luu et al. in [17], which showed promising results for detecting unknown IoT botnet attacks in a zero-shot scenario.

While DL-based methods, especially DAE, have demonstrated capability in network attack detection, several challenges remain. First, many existing approaches lack effective regularization techniques, limiting the ability to learn strong discriminatory latent representations and resulting in false alarms or escaped attack vectors. Optimizing the latent space is the key to improving computational efficiency while enhancing detection. Second, most research still only focuses on a single dataset, which lacks comprehensiveness. In cases where there are multiple dataset scenarios, the datasets are often general network data and not being solely IoT-oriented. Moreover, some benchmark datasets are dated and may not accurately reflect the current IoT landscape. Third, the motivations behind specific model configurations have not yet been explored in depth. In this research, we aim to focus on the IoT network environment and address the attack detection problem through different scenarios with up-to-date benchmark datasets. A summary of research gap in related works are shown in Table 1.

3. Background

3.1. Deep Autoencoder

An Autoencoder is a type of unsupervised neural network variation that aims to reconstruct the input data through a series of encoding and decoding network. The model consists of an encoder, a decoder, and a bottleneck layer or latent layer composed in a symmetrical manner in general. The bottleneck layer is the innermost layer where input data is transformed and compressed into a lower dimensional representation. Through repeated training, the encoder learns to preserve meaningful latent representation in the bottleneck layer so that the output when decoding can be as close as possible to the original input.

$$\hat{\mathbf{X}} = \mathbf{D}(\mathbf{E}(\mathbf{X})) \tag{1}$$

The process is shown in Eq. (1) where $\mathbf{X}$ is the input data, consisting of N samples with D number of features, and $\hat{\mathbf{X}}$ is the reconstructed input, which is also the output of the decoder $\mathbf{D}(\mathbf{E}(\mathbf{X}))$, while $\mathbf{E}(\mathbf{X})$ represents the encoder. When an Autoencoder is composed of multiple layers, it can be referred to as a DAE. Through learning to optimize the loss function and effectively reconstruct the original data, the model can learn the complicated distribution and relationships of the data [49]. Therefore, the bottleneck layer where the latent representation exists is considered a meaningful compressed form of the original input data, which can be used in combination with classifiers to enhance performance. DAE has proven to be effective in many fields including pattern recognition, computer vision, classification. However with the development of hybrid models, performance has shown to be capable of further improvement [50].

3.2. Magnet loss

Magnet loss was first introduced in [23] by Rippel et al. for deep metric learning. This advanced optimization approach aims to improve the separation of different feature classes through local separation in representation space. This technique improves upon other traditional learning approaches such as triplet loss. Instead of focusing on individual points, pairs, or triplets, magnet loss works by penalizing overlaps of an entire local group of nearest clusters at each iteration. Rather than individual samples manipulation, the loss aims to update entire clusters during the training process. This helps to strongly discriminate between different classes while allowing for correct intra-class representation variation identification.

For a training set of N inputs, the loss function is as follows:

$$\mathcal{L}(\Theta) = \frac{1}{N}\sum_{n=1}^{N}\{-log\frac{A}{B}\}_+ \tag{2}$$

where:

$$A = e^{-\frac{1}{2\sigma^2}\|\mathbf{r}_n - \boldsymbol{\mu}(\mathbf{r}_n)\|_2^2 - \alpha} \tag{3}$$

$$B = \sum_{c\neq C(\mathbf{r}_n)}\sum_{k=1}^{K} e^{-\frac{1}{2\sigma^2}\|\mathbf{r}_n - \boldsymbol{\mu}_k^c\|_2^2} \tag{4}$$

In Eq. (2), A is a measure of the closeness of a sample to its assigned cluster and B is a measure of attraction to other classes. $\{.\}_+$ represents the hinge function, which helps optimize training by focusing only on poorly clustered instances. For sample n, $\mathbf{r}_n$ constitutes its representation, with $C(\mathbf{r}_n)$ as the corresponding class, and $\boldsymbol{\mu}(\mathbf{r}_n)$ is its assigned cluster center. α is the desired cluster separation parameter and σ^2 represents the variance of all samples away from their centers which helps ensure distance standardization. The magnet loss function is designed so that as samples move closer to their respective cluster center, the loss is minimized. On the other hand, penalty occurs if a sample is attracted to clusters of other classes. In other words, implementation of the magnet loss helps pull samples of the same class closer together by shortening the distance of said samples to their assigned clusters, while maximizing the distance between them and clusters belonging to other classes. Through this process, clearly separated clusters can be created for different classes while still keeping intra-class variation.

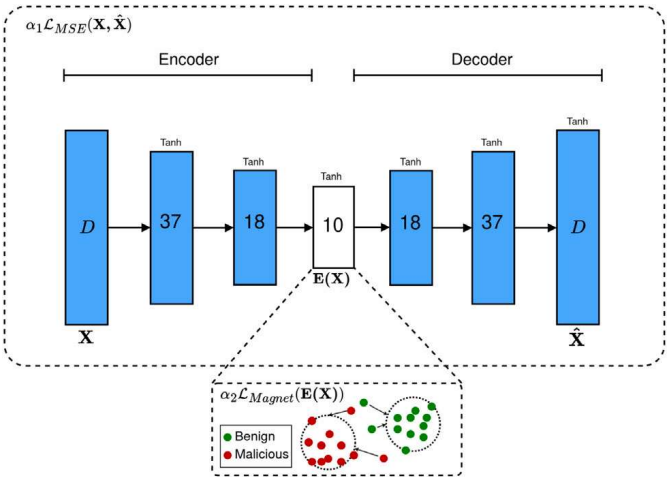


Fig. 1. Proposed Deep Magnet Autoencoder.

4. Proposed methodology

4.1. Deep Magnet Autoencoder

In this section, we will detail the use of the magnet loss optimization with DAE to enhance IoT network attack detection. A method utilizing an implementation of magnet loss in metric learning with DAE, called Deep Magnet Loss Autoencoder (DMAE), was proposed in [19] for website attacks. In this paper, we propose the extension of this model for IoT network attack detection. The model aims to capture a meaningful latent representation that provides powerful features with strong classification enhancement capabilities between benign and malicious data. In order to achieve this, we work under the assumption that IoT network flow data operates under the same principles as websites attack network flow data. This means that normal network data in IoT environments and attack data have a distinction in probability distribution which can be clearly separated.

As can be seen by previous related studies, application of DL has proven to have effectiveness in both general network data as well as IoT network data. Therefore, the extension of the proposed DMAE is hypothesized to be able to address the problem of IoT network attack detection. More specifically, the DMAE model aims to extract powerful features that condense the meaningful characteristics of the original input. This first goal is to create a high-level latent representation that can be used as input for different classifiers to enhance detection capabilities. With the addition of magnet loss, the second goal is to make the latent space be further enhanced through cluster separation based on class. This means that normal network flow will be driven by the model to group together while malicious data points are pushed into different distinct clusters.

Meaningful representation can be extracted by leveraging the ability of the DAE to discover and capture the latent features to distinguish between different data points. Through the implementation of a magnet loss regularizer, the boundaries between benign and malicious classes can be better defined in the latent space by separation into clusters determined by class. Therefore, the objective function of the DMAE model is composed of two components, the reconstruction loss and the magnet loss. For a training set of $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ with N samples and D features, the custom loss function is as follows:

$$\mathcal{L}_{DMAE}(\mathbf{X},\hat{\mathbf{X}}) = \alpha_1\mathcal{L}_{Magnet}(\mathbf{E}(\mathbf{X})) + \alpha_2\mathcal{L}_{MSE}(\mathbf{X},\hat{\mathbf{X}}) \tag{5}$$

Where, $\mathcal{L}_{Magnet}(\mathbf{E}(\mathbf{X}))$ and $\mathcal{L}_{MSE}(\mathbf{X},\hat{\mathbf{X}})$ represents the magnet loss, detailed in Eq. (2), and the reconstruction loss. $\mathbf{E}(\mathbf{X})$ and $\hat{\mathbf{X}}$ are the latent representation and the reconstruction of training input $\mathbf{X}$. The reconstruction loss component helps ensure the preservation of information and maximizing the meaningfulness of the latent layer. Meanwhile, the magnet loss component encourages class separation in latent space by minimizing distance between samples and their assigned centroid, while maximizing distance to other cluster centers. The weight of each loss component is controlled through the values of α_1 and α_2. By balancing the tradeoff between the two losses, the representation in latent space can be affected. For the reconstruction loss, the Mean Squared Error (MSE) Loss was chosen. This is defined as follows:

$$\mathcal{L}_{MSE}(\mathbf{X},\hat{\mathbf{X}}) = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}_i - \hat{\mathbf{x}}_i)^2 \tag{6}$$

The MSE loss is as the name suggests, calculates the average squared difference between the input and its reconstruction from the model. The structure of the model can be seen in Fig. 1.

4.2. Cascade Deep Magnet Autoencoder

To improve upon the existing proposal, we introduce a newer architecture called the Cascade Deep Magnet Loss Autoencoder (CDMAE). The model is composed of a simple DAE component appending to the front of the DMAE model. This is based on the

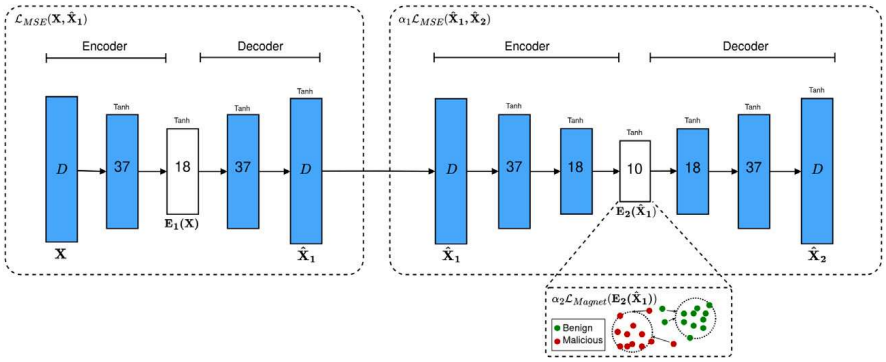


Fig. 2. Proposed Cascade Deep Magnet Autoencoder.

assumption that when capturing real world data, there are inherent irregularities, due to various factors such as device-specific artifacts, environmental factors, or measurement inconsistencies. The captured network data and the subsequent flow aggregation process with tools such as CICFlowMeter or DPKT only results in tabular features that approximate the original network behavior but cannot create an exact one for one copy. Therefore, the input data itself are not pristine representations of network flows but rather an imperfect transformation of the original.

By introducing a DAE before the DMAE, we perform another layer of transformation that attempts to learn the latent characteristics and reconstruct the output to be as close as the input as possible. The reconstructed data can be regarded as a re-encoding of the inherently noisy data. This process can force the model to discard unneeded fluctuations and preserve the most distinguished generalizable patterns, which can potentially be closer to the actual latent behavior of the original traffic. In terms of theory, this approach has a similar intuition to a Denoising Autoencoder [51] where adding noise or corruption to the input data can help capture interesting structures from the distribution. This creates representations that are supposedly more meaningful for downstream learning tasks. Furthermore, by having an additional component of the DAE, the model is further motivated to better capture structural patterns through multi-stage feature learning and forming a higher level of abstraction. Thus, for the CDMAE model, the input $\mathbf{X}$ goes through two transformations as follows:

$$\begin{aligned} \hat{\mathbf{X}}_1 &= \mathbf{D}_1(\mathbf{E}_1(\mathbf{X})) \\ \hat{\mathbf{X}}_2 &= \mathbf{D}_2(\mathbf{E}_2(\hat{\mathbf{X}}_1)) \end{aligned} \tag{7}$$

Where $\hat{\mathbf{X}}_1$ is the reconstructed input of the DAE component while $\hat{\mathbf{X}}_2$ is the reconstruction of $\hat{\mathbf{X}}_1$. The overall objective is calculated as follows:

$$\mathcal{L}_{CDMAE} = \mathcal{L}_{MSE}(\mathbf{X}, \hat{\mathbf{X}}_1) + \alpha_1 \mathcal{L}_{Magnet}(\mathbf{E}_2(\hat{\mathbf{X}}_1)) + \alpha_2 \mathcal{L}_{MSE}(\hat{\mathbf{X}}_1, \hat{\mathbf{X}}_2) \tag{8}$$

Where $\mathcal{L}_{MSE}(\mathbf{X}, \hat{\mathbf{X}}_1)$ is the reconstruction loss of the prepending DAE component. $\mathcal{L}_{Magnet}(\mathbf{E}(\hat{\mathbf{X}}_1))$ is the magnet loss and $\mathcal{L}_{MSE}(\hat{\mathbf{X}}_1, \hat{\mathbf{X}}_2)$ is the reconstruction loss of the DMAE component. The full proposed model is shown in Fig. 2.

4.3. General flow of the proposed approach

The general flow of the approach for detecting IoT network attack is based on the proposed DMAE model and the CDMAE model. The latent representation provided by the models can help boost the detection capabilities of classifiers by extracting meaningful features while strongly separates between normal IoT network data and malicious flows. The combination of reconstruction loss and magnet loss will further enhance clustering and distinction between different classes of data points.

The training process for DMAE consist of two steps. First, the model is trained on both benign and malicious data to learn the structure and discover the latent representation of the provided IoT network flows. Then, for the second step, the trained encoder is extracted to create inputs for training different ML classifiers.

For the CDMAE approach, the general flow is similar. First, the DAE component is trained to reconstruct a set of data that includes both benign and malicious flows. The reconstructed data is then used as training data for the following DMAE component. Finally, a pipeline consisting of both the trained DAE and the encoder of the DMAE is extracted to produce latent representation for ML classifiers training. The general flow of our proposed approach is shown in Fig. 3.

The experimental results are then compared with other approaches, including PCA, DAE, and MemAE [45,52] combined with ML classifiers to evaluate the effectiveness of different latent representation methods based on different metrics.

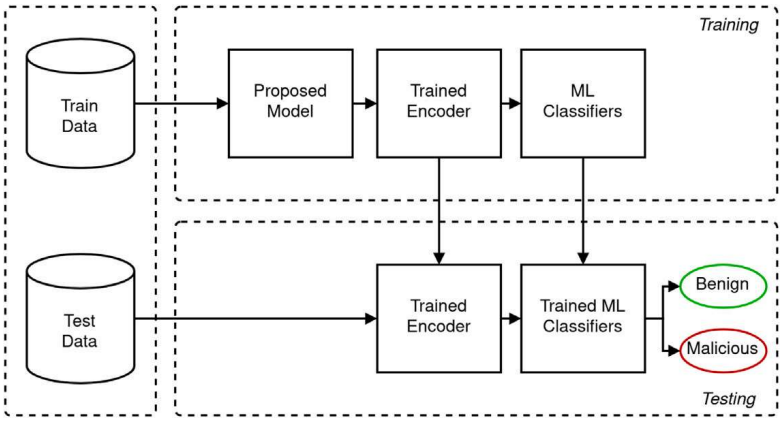


Fig. 3. General flow of the proposed approach.

Table 2
Datasets description.

Dataset	Feature extractor	Feature count	Scenarios	Train volume	Test volume
CIC-BoT–IoT	CICFlowMeter	76	Benign	12 600	5400
			Reconaissance	1050	450
			DDoS	1050	450
			DoS	1050	450
			Theft	1050	450
			Total	16 800	7200
CIC-ToN–IoT	CICFlowMeter	76	Benign	9050	4526
			Backdoor	407	204
			DDoS	134	66
			DoS	97	48
			Injection	407	204
			Man in the Middle	344	171
			Password	407	203
			Ransomware	407	204
			Scanning	407	204
			XSS	407	204
			Total	12 067	6034
CICIoT2023	DPKT	46	Benign	16 800	7200
			DDoS	4200	1800
			DoS	1400	600
			Mirai	1050	450
			Recon	1750	750
			Spoofing	700	300
			Web-based	2100	900
			Brute Force	350	150
			Total	28 350	12 150

5. Experiments

5.1. Datasets and metrics

5.1.1. Datasets

For this research, three different IoT datasets were utilized to evaluate the performance of the proposed approaches. They are the CIC-ToN–IoT dataset, CIC-BoT–IoT dataset [53], and the CICIoT2023 dataset [54]. We designed performance evaluation from a binary classification perspective. Therefore, all samples were labeled with respective binary values for malicious and benign classes. From each dataset, we utilized all scenarios, combining all attack types, then created a smaller dataset with imbalanced data points. We sampled more benign samples than malicious samples to better simulate a real world scenario since benign network flows are more abundant comparing to attack data. To observe the raw performances of the approaches when faced with data imbalance, we then tested these smaller datasets without any prior data balancing technique. A full datasets description can be found at Table 2.

The CIC-BoT–IoT dataset was created from the BoT–IoT dataset [55], which included packet capture files consisting of both normal and botnet traffic. The dataset collects data from a combination of network platforms and simulated IoT devices, which underwent several scenarios, consisting of reconnaissance, DDoS, DoS, and theft. Packet capture files are then converted into network flow features through the use of the CICFlowMeter tool. A total of 83 features were extracted in the original dataset. For our

experiments, we dropped some highly correlated or identifying features to better test the performance of the different approaches. Specifically, 'Flow ID', 'Src IP', 'Src Port', 'Dst IP', 'Dst Port', 'Timestamp', and 'Protocol' were dropped, which resulted in a dataset with 76 network flow features.

The CIC-ToN–IoT dataset is based on the ToN–IoT dataset [56], which includes heterogeneous data sources collected from telemetry data of IoT and IIoT sensors, as well as different operating systems. The dataset contains multiple samples from different attack types including backdoor, DoS, DDoS, injection, Man In The Middle, password, ransomware, scanning, and Cross-Site Scripting. Similar to the CIC-BoT–IoT dataset, it consists of 83 network flow features and was further processed in the same fashion to reduce the number of features to 76 features.

The CICIoT2023 dataset is a real-time benchmark dataset for large scale IoT attack scenarios. Network data was captured on an IoT environment comprising 105 devices in different scenarios. In total, there are seven categories of attacks that were carried out, which are DDoS, DoS, Recon, Web-based, Brute Force, Spoofing, and Mirai. The features extracted for ML were done using the DPKT package, creating a feature set with a total of 46 network features. This results in a different feature space when compared to the CIC-ToN–IoT dataset and the CIC-BoT–IoT dataset.

5.1.2. Metrics

For our research, we utilized Accuracy, Precision, Recall, and F1 Score as metrics to evaluate the effectiveness of the different approaches [11], which was implemented with the Scikit-learn framework [57]. Accuracy measures the proportion of correctly classified traffic sample relative to the total classifications, indicating the effectiveness of an approach in general detection. Precision evaluates the proportion of correctly identified malicious instances among all instances predicted as malicious, with higher precision meaning lower false positives, reflecting better quality classification of attack samples. Recall quantifies the proportion of correctly classified malicious traffic out of all actual malicious instances, highlighting the approach's ability to detect the quantity of attack samples. Lastly, the F1 Score, which is the harmonic mean of Precision and Recall, provides a balanced assessment, especially in the case of imbalanced data. This metric takes into account both false negative and false positive instances to provide a meaningful scale to grading the performance of an approach.

5.2. Experiments setup

5.2.1. Baseline experiments

The baseline experiments consist of training the latent representation approaches on one dataset and then testing its effectiveness on the same dataset when combining with various ML classifiers. We evaluated ten different classifiers, which are Decision Tree (DT), RF, Extra Trees (ET), Bagging Classifier (BC), K-Nearest Neighbors (KNN), SVM, Linear Discriminant Analysis (LDA), Naive Bayes (NB), Logistic Regression (LR), and MLP. For the different latent representation approaches, besides from the original DMAE and the proposed CDMAE, we also trained and tested the PCA technique and a simple DAE approach for comparison. In addition to traditional methods, we included the Memory-Augmented Deep Autoencoder (MemAE) as a modern approach for latent representation from another research for network anomaly detection [45,52].

For the AE model, we created one unified model structure for the simple DAE and DMAE scenarios. The model consists of three fully connected layers on each side with batch normalization and Tanh activation for each layer. The number of neurons for input and output equals the number of network features. For the hidden layers, the sizes were set to 37 and 18 neurons. The latent layer dimension is set according to the rule of thumb proposed in [58], $d = [1 + \sqrt{D}]$, where d is the hidden dimension and D is the number of features. In this case, we set the latent size to 10 neurons, corresponding to the 75 features of the CIC-ToN–IoT and CIC-BoT–IoT datasets. This model utilizes the MSE loss.

The CDMAE approach consists of two different consecutive AE components. The first model responsible for reconstructing the input data from $\mathbf{X}$ to $\hat{\mathbf{X}}_1$. It is a simple DAE consisting of two fully connected layer with Tanh activation for the encoder and decoder. The input and output size corresponds to the number of features and the hidden layers are set to 37, 18, 37. The second AE model is the same model used in the simple DAE and the DMAE scenarios.

In order to keep the feature space consistent, we also set the number of components for PCA to 10 components, equaling the size of the latent layers in the AE models. For the loss weights α_1, α_2 of the proposed DMAE and CDMAE models, we utilized a structured grid search approach combined with empirical monitoring of training loss value convergence to find the appropriate values. In the MemAE approach, we referenced the original experiments in [45]. The full parameter values that were used in the experiments can be referenced in Table 3. All experiments were executed in an Anaconda environment using Pytorch with a Nvidia GeForce RTX 3060 GPU.

5.2.2. Supplementary experiments

In addition to baseline experiments, this research also attempted to measure the robustness of the proposed methods. To achieve this, supplementary experiments with wrong labels were carried out with the CIC-BoT–IoT dataset. Two different scenarios were added where 5% and 10% of the labels are flipped. This means that a certain percentage of data points are sampled randomly, and then benign samples are marked as malicious while malicious samples are changed to benign labels. Other settings are kept the same as the baseline experiments.

To find the optimal size for latent representation, we surveyed the effect of different latent dimensions on the metrics. Using the same model setup while altering only the size of the latent layer, performance was tested with the CIC-ToN–IoT dataset and the CIC-BoT–IoT dataset to determine the influence of the size on the different metrics. We also researched the effect of the loss weights,

Table 3
Parameters description.

Approach		CIC-ToN–IoT				CIC-BoT–IoT				CICIoT2023			
		Learning rate	Epochs	Batch size	Loss weights	Learning rate	Epochs	Batch size	Loss weights	Learning rate	Epochs	Batch size	Loss weights
AE		1	10	256		1	100	256		1	200	256	
DMAE		1	10	256	$\alpha_1 = 0.9$ $\alpha_2 = 0.1$	1	10	256	$\alpha_1 = 0.9$ $\alpha_2 = 0.1$	1	200	256	$\alpha_1 = 1$ $\alpha_2 = 1$
CDMAE	AE	1e−3	200	64		1e−3	200	64		1e−3	200	64	
	DMAE	1	10	256	$\alpha_1 = 1$ $\alpha_2 = 0.1$	1	10	256	$\alpha_1 = 1$ $\alpha_2 = 0.1$	1	200	256	$\alpha_1 = 1$ $\alpha_2 = 1$
MemAE		1e−4	100	64		1e−4	100	64		1e−4	100	64	

Table 4
CIC-BoT–IoT baseline comparison.

Classifier	PCA				AE				MemAE				DMAE				CDMAE			
	Acc	Prec	Rec	F1	Acc	Prec	Rec	F1	Acc	Prec	Rec	F1	Acc	Prec	Rec	F1	Acc	Prec	Rec	F1
DT	0.983	0.979	0.952	0.965	0.982	0.975	0.950	0.963	0.979	0.973	0.944	0.958	0.976	0.967	0.937	0.952	**0.996**	**0.999**	**0.984**	**0.991**
RF	0.983	0.979	0.953	0.966	0.983	0.982	0.951	0.966	0.982	0.978	0.947	0.962	0.98	0.978	0.942	0.96	**0.996**	**0.998**	**0.985**	**0.991**
ET	0.984	0.981	0.953	0.967	0.984	0.982	0.953	0.967	0.983	0.979	0.952	0.965	0.981	0.979	0.945	0.962	**0.996**	**0.999**	**0.984**	**0.991**
BC	0.983	0.983	0.947	0.965	0.983	0.984	0.949	0.966	0.981	0.98	0.942	0.961	0.978	0.976	0.935	0.955	**0.994**	**0.995**	**0.980**	**0.988**
KNN	0.976	0.968	0.933	0.950	0.975	0.969	0.929	0.949	0.980	0.978	0.939	0.958	0.974	0.971	0.923	0.946	**0.980**	**0.970**	**0.948**	**0.959**
SVM	0.900	0.963	0.624	0.757	0.895	0.949	0.613	0.745	0.894	0.947	0.611	0.743	**0.964**	**0.959**	**0.893**	**0.925**	0.957	0.910	0.917	0.914
LDA	0.808	0.824	0.297	0.436	0.78	0.757	0.176	0.286	0.832	0.99	0.329	0.494	**0.965**	**0.940**	**0.918**	**0.929**	0.957	0.919	0.906	0.912
NB	0.729	0.477	0.843	0.609	0.702	0.439	0.690	0.537	0.705	0.443	0.690	0.539	**0.954**	**0.892**	**0.931**	**0.911**	0.894	0.721	0.942	0.817
LR	0.839	0.917	0.391	0.548	0.858	0.908	0.482	0.63	0.829	0.948	0.336	0.496	**0.966**	**0.950**	**0.911**	**0.930**	0.957	0.92	0.908	0.914
MLP	0.909	0.978	0.649	0.78	0.903	0.964	0.635	0.766	0.901	0.959	0.630	0.761	**0.966**	**0.958**	**0.902**	**0.929**	0.958	0.918	0.911	0.915

or the alpha values, of the custom loss function on DMAE's performance with these two datasets. For this scenario, we chose NB as the classifier. The exploration range was set based on the grid search approach conducted prior with the baseline experiments. Specifically, we selected $\alpha_1 \in [0.1, 1.0]$ and $\alpha_2 \in [0.1, 1.0]$ for investigation. These experiments were carried out on an isolated DMAE model to better investigate the relationship of the parameters for the approach.

Finally, in order to better see the impact of our proposed latent representation method, we investigated the distribution of data points in space with different visualization methods. We tested the extracted latent representation from both DMAE and CDMAE with PCA visualization and t-SNE visualization using three different distance metrics, which are Cosine, Manhattan, Chebyshev, on the CIC-BoT–IoT dataset.

6. Results and discussion

In general, results show that the proposed approaches are superior compared to other methods. It can be seen that implementation of the magnet loss has a positive effect on the performance across all metrics. In all cases, the DMAE approach has consistently high metrics when combining with SVM, LDA, NB, and LR, while combination with MLP outperforms in all scenarios except one. With the CDMAE approach, the model excels when using in conjunction with tree-based classifiers, showing strong discriminatory capabilities with DT, RF, and ET. BC also performs extremely well with this proposed approach, which can be due to the fact that the classifier was set to DT estimator. This outcome can be attributed to the characteristics of CDMAE. As a cascade autoencoder, it is able to capture the hierarchical structures in network anomaly patterns through multi-stage feature learning. Tree-based classifiers naturally align with this structure as data are partitioned based on feature thresholds. This results in a synergy that allows the classifiers to exploit the powerful and discriminative representations produced by CDMAE, leading to superior performance compared to non-tree-based models.

6.1. Baseline results

Table 4 indicates improvement across all classifiers with the implementation of magnet loss for latent representation. For this scenario, all classifiers achieved the best metrics when combining with our proposed methods, with five out of ten classifiers each. The CDMAE approach achieves consistently high performance with tree-based classifiers in addition to KNN, with the highest F1 Score overall reaching 0.991. The DMAE approach also shows stable results with every combinations crossing the 0.9 value while also significantly improves upon the weaker classifiers, which are SVM, LDA, NB, LR, and MLP. While other approaches of latent representation also have meaningful results for many classifiers, the proposed approaches show significantly more consistent performance.

With the CIC-ToN–IoT dataset, it can be seen again in Table 5 that in all cases either DMAE or CDMAE has the best performance when assessing individual classifier combination. The proposed CDMAE approach has the highest metrics in all four Accuracy,

Table 5
CIC-ToN–IoT baseline comparison.

Classifier	PCA				AE				MemAE				DMAE				CDMAE			
	Acc	Prec	Rec	F1	Acc	Prec	Rec	F1	Acc	Prec	Rec	F1	Acc	Prec	Rec	F1	Acc	Prec	Rec	F1
DT	0.964	0.924	0.932	0.928	0.964	0.931	0.926	0.929	0.961	0.919	0.924	0.922	0.962	0.917	0.930	0.924	**0.999**	**0.999**	**0.999**	**0.999**
RF	0.976	0.943	0.963	0.953	0.975	0.944	0.955	0.950	0.969	0.934	0.941	0.937	0.970	0.934	0.948	0.941	**0.999**	**0.999**	**0.999**	**0.999**
ET	0.976	0.949	0.957	0.953	0.974	0.945	0.95	0.947	0.967	0.931	0.937	0.934	0.971	0.938	0.948	0.943	**0.999**	**0.999**	**0.999**	**0.999**
BC	0.971	0.943	0.943	0.943	0.971	0.944	0.938	0.941	0.967	0.934	0.932	0.933	0.97	0.941	0.936	0.939	**0.996**	**0.996**	**0.990**	**0.993**
KNN	0.971	0.926	0.961	0.943	0.971	0.93	0.958	0.944	0.965	0.913	0.949	0.930	0.968	0.93	0.942	0.936	**0.974**	**0.942**	**0.954**	**0.948**
SVM	0.924	0.872	0.815	0.843	0.819	0.878	0.318	0.467	0.87	0.847	0.587	0.693	**0.954**	**0.906**	**0.910**	**0.908**	0.938	0.874	0.880	0.877
LDA	0.824	0.663	0.605	0.633	0.804	0.600	0.641	0.620	0.726	0.413	0.234	0.299	**0.950**	**0.897**	**0.905**	**0.901**	0.920	0.841	0.840	0.840
NB	0.781	0.543	0.785	0.642	0.719	0.452	0.595	0.514	0.716	0.416	0.336	0.372	**0.945**	**0.864**	**0.925**	**0.894**	0.917	0.81	0.874	0.841
LR	0.850	0.766	0.574	0.656	0.787	0.604	0.430	0.502	0.751	0.506	0.204	0.291	**0.952**	**0.908**	**0.899**	**0.903**	0.920	0.851	0.825	0.838
MLP	0.951	0.902	0.903	0.903	0.941	0.882	0.883	0.882	0.921	0.820	0.874	0.846	0.958	0.903	0.931	0.917	**0.960**	**0.909**	**0.935**	**0.922**

Table 6
CICIoT2023 baseline comparison.

Classifier	PCA				AE				MemAE				DMAE				CDMAE			
	Acc	Prec	Rec	F1	Acc	Prec	Rec	F1	Acc	Prec	Rec	F1	Acc	Prec	Rec	F1	Acc	Prec	Rec	F1
DT	0.843	0.810	0.802	0.806	0.841	0.807	0.801	0.804	0.979	0.973	0.944	0.958	0.887	0.866	0.855	0.861	**0.999**	**0.999**	**0.999**	**0.999**
RF	0.885	0.918	0.788	0.848	0.876	0.905	0.778	0.837	0.982	0.978	0.947	0.962	0.909	0.912	0.860	0.885	**0.999**	**0.999**	**0.999**	**0.999**
ET	0.881	0.915	0.781	0.843	0.875	0.901	0.779	0.836	0.983	0.979	0.952	0.965	0.907	0.906	0.859	0.882	**0.999**	**0.999**	**0.999**	**0.999**
BC	0.876	0.905	0.776	0.836	0.868	0.890	0.771	0.826	0.981	0.980	0.942	0.961	0.907	0.911	0.855	0.882	**0.989**	**0.996**	**0.978**	**0.987**
KNN	0.858	0.878	0.757	0.813	0.867	0.878	0.782	0.827	0.980	0.978	0.939	0.958	0.906	0.907	0.859	0.882	0.917	0.928	0.864	0.895
SVM	0.853	0.970	0.660	0.786	0.851	0.972	0.652	0.780	0.894	0.947	0.611	0.743	**0.912**	**0.924**	**0.853**	**0.887**	0.894	0.931	0.799	0.860
LDA	0.803	0.783	0.713	0.746	0.800	0.768	0.728	0.747	0.832	0.990	0.329	0.494	**0.911**	**0.920**	**0.855**	**0.886**	0.897	0.915	0.824	0.867
NB	0.808	0.801	0.704	0.749	0.754	0.667	0.791	0.724	0.705	0.443	0.690	0.539	**0.911**	**0.915**	**0.861**	**0.887**	0.888	0.869	0.854	0.861
LR	0.810	0.800	0.711	0.753	0.811	0.786	0.737	0.760	0.829	0.948	0.336	0.496	**0.912**	**0.926**	**0.853**	**0.888**	0.894	0.899	0.833	0.865
MLP	0.864	0.917	0.733	0.814	0.859	0.935	0.701	0.802	0.901	0.959	0.63	0.761	**0.912**	**0.920**	**0.860**	**0.889**	0.897	0.924	0.813	0.865

Precision, Recall, F1 Score when combining with DT, RF, and ET. While the DMAE approach does not reach as high values, the performance is more consistent as most classifier combinations have values above 0.9, with only a few instances falling slightly below. The results also reinforce the notion that implementation of the magnet loss helps create more powerful latent representation since the proposed approaches similarly significantly improves weaker classifiers, such as in the case of SVM, LDA, NB, and LR. While PCA, AE, and MemAE struggles with poor metrics, especially having low recall values due to imbalanced data, DMAE and CDMAE has shown to have the capability to boost this performance, reaching above 0.8 values overall.

The baseline results for the CICIoT2023 dataset shown in Table 6 also have a similar outlook despite having a different feature space when comparing with the CIC-BoT–IoT and CIC-ToN–IoT datasets. In this scenario, except for KNN which worked best with MemAE, nine out of ten classifiers performed the best with either of the proposed two approaches. While general metrics are lower when compared with the CIC-ToN–IoT dataset, instances of DMAE and CDMAE were able to consistently produce metric values above 0.8, without being influenced by the adverse effects of imbalanced data where other approaches suffer in low Precision and Recall for many classifier combinations. Similar to the other datasets, when testing on the CICIoT2023 dataset, the CDMAE model when combined with tree-based classifiers also shows excellent performance while the DMAE approach also displays consistent results with SVM, LDA, NB, LR.

6.2. Supplementary results

In the robustness test scenario, in both cases of having 5% and 10% wrong labels, the proposed approaches still show superior performance. Except for the case of KNN, the same classifier combinations with DMAE and CDMAE show the highest results when compared to other methods. Due to the wrong label information, all approaches suffer performance dip in varying degrees. With a 5% false label limit, the metrics dropped but was still able to keep a relatively good score. It can be seen that DMAE displays better robustness as the performance degradation is generally weaker. While CDMAE retains relative robustness when combining with the tree-based classifiers, performance with other classifiers shows a bigger drop in the face of noisy data (see Table 7).

The classification proficiency of CDMAE suffers even a bigger hit when the noise level is increased to 10%. Recall values show an evident decrease, with classifiers such as LDA and LR dropping below 0.4. However, combinations with DT, RF, ET, and BC still displays a consistently positive results with high F1 Scores. Results with DMAE also shows worse Recall metric when met with noisy labels in the dataset. This shows that while the models were able to have low false positives, there are still many instances where malicious flows escape detection. With other approaches, it is worth noting that while MemAE does not have the highest metrics in individual scenarios, the approach has a relatively high robustness as performance does not suffer as high of a fluctuation when faced with noisy labels in the data (see Table 8).

Figs. 4 and 5 shows that the dimension of the latent layer can have an influence on the performance of our proposed approaches. While values varies between different classifiers and metrics, in general, the size of 10 achieve consistently high value in F1 Score,

Table 7
CIC-BoT–IoT 5% wrong label comparison.

Classifier	PCA				AE				MemAE				DMAE				CDMAE			
	Acc	Prec	Rec	F1	Acc	Prec	Rec	F1	Acc	Prec	Rec	F1	Acc	Prec	Rec	F1	Acc	Prec	Rec	F1
DT	0.957	0.920	0.905	0.913	0.953	0.904	0.908	0.906	0.952	0.903	0.904	0.904	0.952	0.911	0.896	0.903	**0.976**	**0.992**	**0.921**	**0.955**
RF	0.969	0.937	0.939	0.938	0.970	0.94	0.942	0.941	0.968	0.935	0.939	0.937	0.967	0.938	0.929	0.933	**0.976**	**0.988**	**0.923**	**0.955**
ET	0.963	0.924	0.928	0.926	0.963	0.923	0.928	0.925	0.965	0.926	0.933	0.929	0.959	0.917	0.918	0.917	**0.978**	**0.992**	**0.926**	**0.958**
BC	0.967	0.939	0.927	0.933	0.966	0.937	0.927	0.932	0.965	0.927	0.932	0.929	0.963	0.937	0.913	0.925	**0.969**	**0.979**	**0.905**	**0.941**
KNN	0.974	0.963	0.933	0.948	0.974	0.968	0.927	0.947	0.979	0.977	0.939	0.958	0.97	0.958	0.919	0.938	0.932	0.929	0.817	0.870
SVM	0.899	0.964	0.621	0.755	0.827	0.991	0.312	0.475	0.894	0.955	0.607	0.742	**0.959**	**0.952**	**0.878**	**0.914**	0.913	0.880	0.792	0.833
LDA	0.808	0.809	0.302	0.439	0.823	0.987	0.297	0.457	0.801	0.769	0.289	0.42	**0.950**	**0.956**	**0.839**	**0.894**	0.906	0.845	0.805	0.824
NB	0.726	0.474	0.871	0.614	0.677	0.399	0.576	0.471	0.664	0.400	0.690	0.506	**0.943**	**0.935**	**0.830**	**0.879**	0.839	0.665	0.838	0.742
LR	0.831	0.886	0.371	0.523	0.823	0.989	0.297	0.456	0.787	0.695	0.262	0.381	**0.950**	**0.956**	**0.841**	**0.894**	0.911	0.869	0.796	0.831
MLP	0.906	0.960	0.651	0.776	0.887	0.968	0.568	0.716	0.897	0.939	0.631	0.754	**0.962**	**0.957**	**0.889**	**0.922**	0.916	0.89	0.793	0.839

Table 8
CIC-BoT–IoT 10% wrong label comparison.

Classifier	PCA				AE				MemAE				DMAE				CDMAE			
	Acc	Prec	Rec	F1	Acc	Prec	Rec	F1	Acc	Prec	Rec	F1	Acc	Prec	Rec	F1	Acc	Prec	Rec	F1
DT	0.918	0.829	0.848	0.838	0.920	0.835	0.847	0.841	0.920	0.823	0.864	0.843	0.913	0.817	0.84	0.828	**0.960**	**0.988**	**0.878**	**0.930**
RF	0.945	0.872	0.915	0.893	0.946	0.874	0.916	0.894	0.947	0.875	0.921	0.897	0.941	0.863	0.907	0.884	**0.960**	**0.980**	**0.885**	**0.930**
ET	0.932	0.843	0.896	0.868	0.933	0.845	0.897	0.87	0.937	0.851	0.904	0.877	0.93	0.841	0.889	0.864	**0.961**	**0.988**	**0.881**	**0.932**
BC	0.942	0.873	0.901	0.887	0.939	0.869	0.892	0.88	0.942	0.871	0.902	0.886	0.935	0.856	0.892	0.873	**0.949**	**0.966**	**0.860**	**0.910**
KNN	0.967	0.943	0.926	0.934	0.965	0.938	0.922	0.93	0.968	0.944	0.929	0.936	0.96	0.926	0.912	0.919	0.886	0.875	0.724	0.792
SVM	0.898	0.96	0.62	0.753	0.813	0.991	0.256	0.407	0.897	0.954	0.616	0.748	**0.931**	**0.954**	**0.762**	**0.847**	0.83	0.823	0.555	0.663
LDA	0.808	0.825	0.293	0.432	0.786	0.974	0.146	0.254	0.827	0.995	0.308	0.47	**0.926**	**0.946**	**0.748**	**0.836**	0.825	0.866	0.496	0.631
NB	0.723	0.47	0.869	0.61	0.7	0.437	0.691	0.535	0.706	0.443	0.693	0.541	**0.918**	**0.937**	**0.722**	**0.815**	0.824	0.856	0.5	0.631
LR	0.826	0.882	0.353	0.504	0.785	0.977	0.143	0.25	0.829	0.995	0.319	0.484	**0.925**	**0.946**	**0.743**	**0.832**	0.825	0.866	0.497	0.632
MLP	0.904	0.95	0.651	0.773	0.891	0.969	0.584	0.729	0.895	0.927	0.632	0.751	**0.947**	**0.927**	**0.853**	**0.889**	0.866	0.837	0.688	0.755

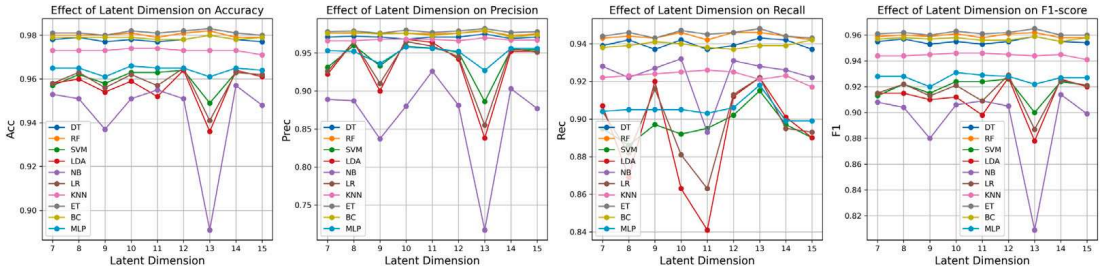


Fig. 4. CIC-BoT–IoT - Effect of latent dimension on DMAE.

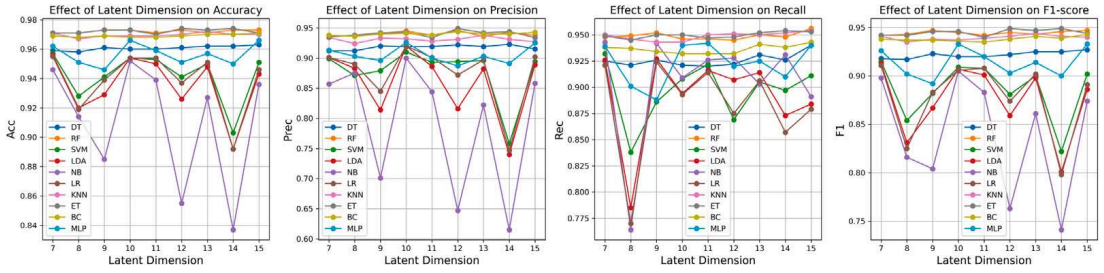


Fig. 5. CIC-ToN–IoT - Effect of latent dimension on DMAE.

showing a balance performance with almost all ML classifiers. Even though there exists some cases where individual metric is better with a different configuration, such as higher Accuracy and Precision with latent dimension of 11 for NB classifier, it comes with a tradeoff of an extreme drop in Recall and less than desirable performance with other classifiers. For DMAE, we propose an optimal latent size of 10. From this empirical finding, we believe that the rule of thumb proposed in [58] is further reinforced through this experiment and serves as a good reference for model design.

Investigation on the different loss weights combinations displayed in Figs. 6 and 7. Results on the CIC-BoT–IoT dataset shows that in general, a higher reliance on magnet loss results in better performance. In cases where MSE loss has higher importance in the custom loss function, it can be seen that all metrics suffer a negative impact, with some even dropping below 0.5. While combinations

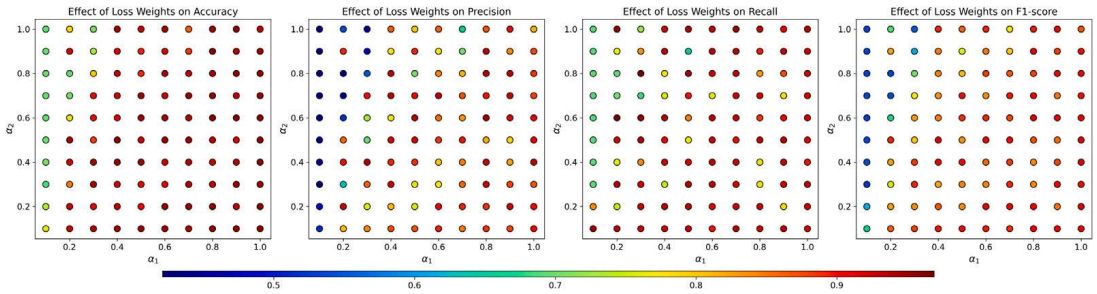


Fig. 6. CIC-BoT–IoT - Effect of loss weights on DMAE.

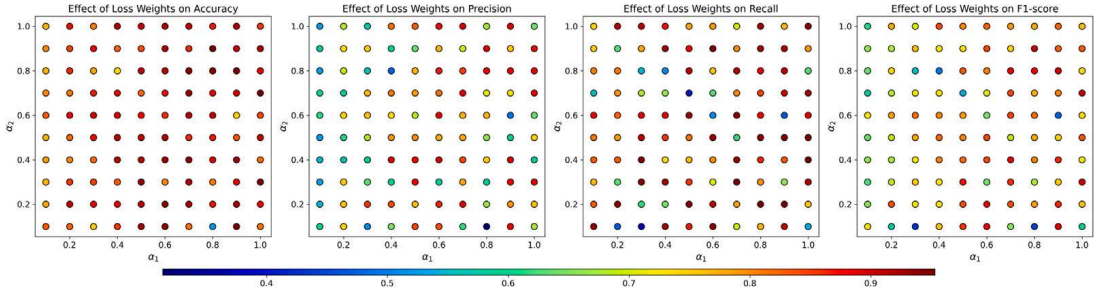


Fig. 7. CIC-ToN–IoT - Effect of loss weights on DMAE.

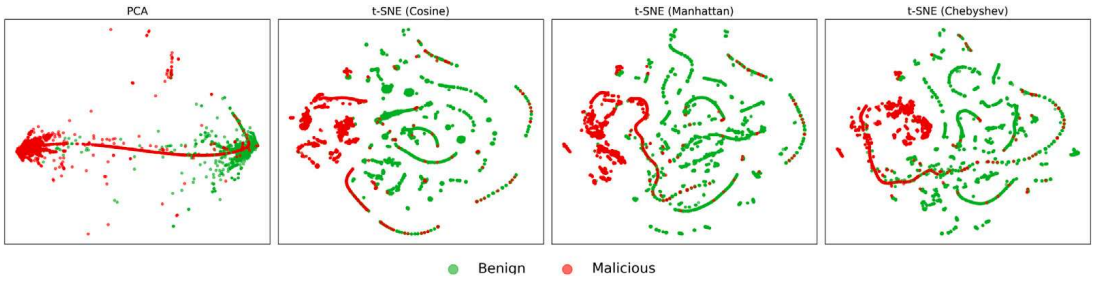


Fig. 8. CIC-BoT–IoT - Visualization of DMAE latent representation.

with higher magnet loss weight have performance deviations, results show superior metrics and consistency compared to the other end of the spectrum. Therefore, bigger alpha values for magnet loss can be a good starting point for tuning the DMAE approach.

With the CIC-ToN–IoT dataset, results are more scattered with variations. However, it can still be seen that a higher weight value for magnet loss generally have better performance, especially in terms of Accuracy. However, in this case, fine-tuning small deviations can result in a large outcome difference, such is in the case of when MSE loss weight is 0.1, changing the magnet loss weight from 0.9 to 1.0 resulted in a big performance drop.

The visualization of latent representation in space is shown in Figs. 8 and 9. Across the four visualization methods, the extracted latent representations from both DMAE and CDMAE exhibit strong separation between benign and malicious data points. In general, the two classes form distinct clusters, with a noticeable split along the vertical axis of each subplot. CDMAE, in general, produces tighter grouping with malicious data points forming larger singular clusters. This aligns with our previous results showing that the latent representation produced from the proposed approaches does indeed produce strong separation for boosting classifiers' performance. However, the remaining existence of overlaps can be attributed to the fact that network flows of many attacks can be very similar to benign flows which results in closer allocation in space.

Through our experiments, it can be seen that the implementation of magnet loss in DAE model has a meaningful impact on the detection capabilities of ML classifiers by creating powerful latent representation. Furthermore, by using a CDMAE approach where a simple DAE is prepended to the DMAE model, another positive outlook was shown. In general, metrics were improved with the proposed methods. However, it is worth noting that the DMAE model requires data specific loss weights which can be costly when tuning this approach. Moreover, for the CDMAE approach, the prepending DAE model is another element to be considered. This is due to the reconstruction of input samples from $\mathbf{X}$ to $\hat{\mathbf{X}}_1$ can show deviations and diversity with various datasets and different configurations.

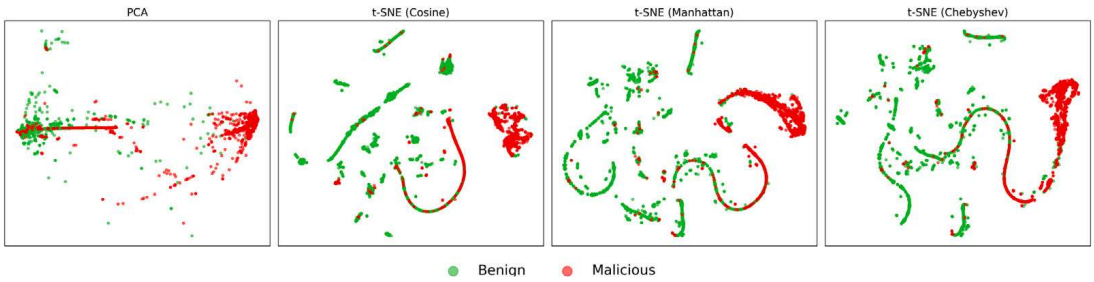


Fig. 9. CIC-BoT–IoT - Visualization of CDMAE latent representation.

7. Conclusion & future work

While the IoT network attack detection problem has been addressed in many studies, the use of regularizer for performance optimization, a comprehensive test across various datasets for different IoT network environments, and clearly defined model configurations are still scarce. This research proposes the implementation of deep magnet loss in AE models for creating meaningful latent representation for IoT network anomaly detection. Specifically, we propose a DMAE approach and a CDMAE approach which were tested against PCA, simple DAE, and MemAE. Extensive results across three different datasets indicate that when combined with our approaches, ML classifiers' performance was improved in most cases and having the highest metrics. Especially, the classification capabilities of weaker classifier were significantly boosted. Even under situations where noise was introduced to the labels of datasets, the proposed models proved to have better results consistently.

We also surveyed the influence of latent dimension and different loss weights configurations on the performance of the model. From the collective results, we came to an empirical verification for the rule of thumb when choosing latent size introduced in [58]. Moreover, through analyzing different loss weights configurations on two different datasets, we found that it is generally better to have a higher magnet loss weights as a starting point when balancing the custom loss function for the DMAE model.

However, our work has some possible limitations. First, the implementation of magnet loss while improving the overall metrics, the fine-tuning and balancing of the loss weights still prove to be a delicate problem when changing datasets, which can be attributed to different data distribution. This can potentially introduce strong knowledge drift on the same data as time passes. Second, the CDMAE model relies on the reconstruction quality of the prepending DAE. We recognize that this element affects the overall approach and variations can result in performance differences. Third, while we conducted robustness experiments with wrong labels, the emergence of adversarial examples to actively evade detection in recent years is another problem that should be addressed. Fourth, while magnet loss optimization also excels at multiclass clustering, the current research settings are limited to binary detection scenarios.

Therefore, we propose the following directions based on the mentioned shortcomings. First, research on methods to dynamically update loss weights based on the input dataset should be conducted. This will possibly ensure a more flexible and less time-consuming approach to fine-tuning the custom loss function. Second, further investigation on the CDMAE model is necessary, especially the effect of the reconstructed input data from the prepending AE on the performance of the following DMAE model. To better evaluate the generalization ability of the CDMAE model from this additional DAE component, a train-on-one, test-on-another scenario should also be analyzed. Third, different adversarial scenarios and defense strategies, such as adversarial training or model hardening, should be considered for a more comprehensive robustness evaluation. Fourth, training and testing the proposed approaches on multiclass scenarios is an essential step for deeper IoT network attack detection.

Aside from model investigation, data quality is another important factor for enhancing performance. It is worth acknowledging that additional techniques for addressing data imbalance, such as SMOTE or data augmentation, could be explored to elevate the learning capabilities of the proposed models. Furthermore, the task of labeling IoT attack traffic is another demanding challenge. Extension of the model to more advanced implementations such as contrastive learning or federated learning could be a potential direction for alleviating this problem of reliance on labeled data. Finally, for real world usage, the time element of training and prediction is also another crucial measurement to be noted. These future directions will help focus on better optimizing and enhancing the proposed approaches.

CRediT authorship contribution statement

Chi Duc Luu: Writing – review & editing, Writing – original draft, Visualization, Software, Methodology, Investigation, Data curation, Conceptualization. **Viet Hung Nguyen:** Writing – review & editing, Writing – original draft, Supervision, Project administration, Methodology, Funding acquisition, Conceptualization. **Van Quan Nguyen:** Writing – review & editing, Visualization, Supervision, Resources, Investigation, Formal analysis, Conceptualization. **Ngoc-Son Vu:** Writing – review & editing, Writing – original draft, Validation, Resources, Investigation.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Viet Hung Nguyen reports financial support was provided by Socialist Republic of Vietnam Ministry of Science and Technology. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors would like to thank the Ministry of Science and Technology of Vietnam for their support provided through the project "Multidimensional detection and automated response using artificial intelligence" (code NDT/CZ/24/01).

Data availability

Data will be made available on request.

References

[1] Q. Gou, L. Yan, Y. Liu, Y. Li, Construction and strategies in IoT security system, in: 2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing, 2013, pp. 1129–1132, http://dx.doi.org/10.1109/GreenCom-iThings-CPSCom.2013.195.

[2] N. Chaabouni, M. Mosbah, A. Zemmari, C. Sauvignac, P. Faruki, Network intrusion detection for IoT security based on learning techniques, IEEE Commun. Surv. Tutorials 21 (3) (2019) 2671–2701, http://dx.doi.org/10.1109/comst.2019.2896380.

[3] S.A. Al-Qaseemi, H.A. Almulhim, M.F. Almulhim, S.R. Chaudhry, IoT architecture challenges and issues: Lack of standardization, in: 2016 Future Technologies Conference, FTC, 2016, pp. 731–738, http://dx.doi.org/10.1109/FTC.2016.7821686.

[4] W. Jiang, B. Zheng, D. Sheng, X. Li, A compensation approach for magnetic encoder error based on improved deep belief network algorithm, Sensors Actuators A: Phys. 366 (2024) 115003, http://dx.doi.org/10.1016/j.sna.2023.115003.

[5] E. Schiller, A. Aidoo, J. Fuhrer, J. Stahl, M. Ziörjen, B. Stiller, Landscape of IoT security, Comput. Sci. Rev. 44 (2022) 100467, http://dx.doi.org/10.1016/j.cosrev.2022.100467.

[6] Q. Li, L. Li, Z. Liu, W. Sun, W. Li, J. Li, W. Zhao, Cloud-edge collaboration for industrial internet of things: Scalable neurocomputing and rolling-horizon optimization, IEEE Internet Things J. (2025) http://dx.doi.org/10.1109/JIOT.2025.3542428, 1–1.

[7] H. Zhang, C. Yu, M. Zeng, T. Ye, D. Yue, C. Dou, X. Xie, G.P. Hancke, Homomorphic encryption-based resilient distributed energy management under cyber-attack of micro-grid with event-triggered mechanism, IEEE Trans. Smart Grid 15 (5) (2024) 5115–5126, http://dx.doi.org/10.1109/TSG.2024.3390108.

[8] M.A. Al-Garadi, A. Mohamed, A.K. Al-Ali, X. Du, I. Ali, M. Guizani, A survey of machine and deep learning methods for internet of things (IoT) security, IEEE Commun. Surv. Tutorials 22 (3) (2020) 1646–1685, http://dx.doi.org/10.1109/comst.2020.2988293.

[9] P. Chaudhary, B.B. Gupta, Ddos detection framework in resource constrained internet of things domain, in: 2019 IEEE 8th Global Conference on Consumer Electronics, GCCE, 2019, pp. 675–678, http://dx.doi.org/10.1109/GCCE46687.2019.9015465.

[10] Y. Kayode Saheed, A. Idris Abiodun, S. Misra, M. Kristiansen Holone, R. Colomo-Palacios, A machine learning-based intrusion detection for detecting internet of things network attacks, Alex. Eng. J. 61 (12) (2022) 9395–9409, http://dx.doi.org/10.1016/j.aej.2022.02.063.

[11] R. Ahmad, I. Alsmadi, Machine learning approaches to IoT security: A systematic literature review, Internet Things 14 (2021) 100365, http://dx.doi.org/10.1016/j.iot.2021.100365.

[12] X. Kan, Y. Fan, Z. Fang, L. Cao, N.N. Xiong, D. Yang, X. Li, A novel IoT network intrusion detection approach based on Adaptive Particle Swarm Optimization Convolutional Neural Network, Inform. Sci. 568 (2021) 147–162, http://dx.doi.org/10.1016/j.ins.2021.03.060.

[13] B.I. Hairab, M. Said Elsayed, A.D. Jurcut, M.A. Azer, Anomaly detection based on CNN and regularization techniques against zero-day attacks in IoT networks, IEEE Access 10 (2022) 98427–98440, http://dx.doi.org/10.1109/ACCESS.2022.3206367.

[14] W.-C. Shi, H.-M. Sun, DeepBot: a time-based botnet detection with deep learning, Soft Comput. 24 (21) (2020) 16605–16616, http://dx.doi.org/10.1007/s00500-020-04963-z.

[15] M. Almiani, A. AbuGhazleh, A. Al-Rahayfeh, S. Atiewi, A. Razaque, Deep recurrent neural network for IoT intrusion detection system, Simul. Model. Pr. Theory 101 (2020) 102031, http://dx.doi.org/10.1016/j.simpat.2019.102031.

[16] V.Q. Nguyen, L.T. Ngo, L.M. Nguyen, V.H. Nguyen, N. Shone, Deep clustering hierarchical latent representation for anomaly-based cyber-attack detection, Knowl.-Based Syst. 301 (2024) 112366, http://dx.doi.org/10.1016/j.knosys.2024.112366.

[17] C.D. Luu, V.Q. Nguyen, T.S. Pham, N.-A. Le-Khac, A zero-shot deep learning approach for unknown IoT botnet attack detection, in: 2023 RIVF International Conference on Computing and Communication Technologies, RIVF, 2023, pp. 278–283, http://dx.doi.org/10.1109/RIVF60135.2023.10471793.

[18] L. Vu, V.L. Cao, Q.U. Nguyen, D.N. Nguyen, D.T. Hoang, E. Dutkiewicz, Learning latent representation for IoT anomaly detection, IEEE Trans. Cybern. 52 (5) (2022) 3769–3782, http://dx.doi.org/10.1109/TCYB.2020.3013416.

[19] D.D. Tang, V.Q. Nguyen, V.H. Nguyen, T.C. Nguyen, N. Shone, A novel deep learning approach with magnet loss optimization for website attack detection, in: 2024 1st International Conference on Cryptography and Information Security, VCRIS, 2024, pp. 1–6, http://dx.doi.org/10.1109/VCRIS63677.2024.10813436.

[20] Y. Qiao, J. Lü, T. Wang, K. Liu, B. Zhang, H. Snoussi, A multihead attention self-supervised representation model for industrial sensors anomaly detection, IEEE Trans. Ind. Inform. 20 (2) (2024) 2190–2199, http://dx.doi.org/10.1109/TII.2023.3280337.

[21] J. Lan, X. Liu, B. Li, J. Zhao, A novel hierarchical attention-based triplet network with unsupervised domain adaptation for network intrusion detection, Appl. Intell. 53 (10) (2022) 11705–11726, http://dx.doi.org/10.1007/s10489-022-04076-0.

[22] G. Andresini, A. Appice, D. Malerba, Autoencoder-based deep metric learning for network intrusion detection, Inform. Sci. 569 (2021) 706–727, http://dx.doi.org/10.1016/j.ins.2021.05.016.

[23] O. Rippel, M. Paluri, P. Dollar, L. Bourdev, Metric learning with adaptive density discrimination, 2015, http://dx.doi.org/10.48550/ARXIV.1511.05939, URL https://arxiv.org/abs/1511.05939.

[24] D. Kwon, H. Kim, J. Kim, S.C. Suh, I. Kim, K.J. Kim, A survey of deep learning-based network anomaly detection, Clust. Comput. 22 (S1) (2017) 949–961, http://dx.doi.org/10.1007/s10586-017-1117-8.

[25] Z. Ahmad, A. Shahid Khan, C. Wai Shiang, J. Abdullah, F. Ahmad, Network intrusion detection system: A systematic study of machine learning and deep learning approaches, Trans. Emerg. Telecommun. Technol. 32 (1) (2020) http://dx.doi.org/10.1002/ett.4150.

[26] S. Sriram, R. Vinayakumar, M. Alazab, S. KP, Network flow based IoT botnet attack detection using deep learning, in: IEEE INFOCOM 2020 - IEEE Conference on Computer Communications Workshops, INFOCOM WKSHPS, IEEE, 2020, http://dx.doi.org/10.1109/infocomwkshps50562.2020.9162668.

[27] C. Yin, Y. Zhu, J. Fei, X. He, A deep learning approach for intrusion detection using recurrent neural networks, IEEE Access 5 (2017) 21954–21961, http://dx.doi.org/10.1109/ACCESS.2017.2762418.

[28] M. Saharkhizan, A. Azmoodeh, A. Dehghantanha, K.-K.R. Choo, R.M. Parizi, An ensemble of deep recurrent neural networks for detecting IoT cyber attacks using network traffic, IEEE Internet Things J. 7 (9) (2020) 8852–8859, http://dx.doi.org/10.1109/JIOT.2020.2996425.

[29] Y. Fu, F. Lou, F. Meng, Z. Tian, H. Zhang, F. Jiang, An intelligent network attack detection method based on RNN, in: 2018 IEEE Third International Conference on Data Science in Cyberspace, DSC, IEEE, 2018, pp. 483–489, http://dx.doi.org/10.1109/dsc.2018.00078.

[30] L. Mohammadpour, T.C. Ling, C.S. Liew, A. Aryanfar, A survey of CNN-based network intrusion detection, Appl. Sci. 12 (16) (2022) 8162, http://dx.doi.org/10.3390/app12168162.

[31] Y. Yu, N. Bian, An intrusion detection method using few-shot learning, IEEE Access 8 (2020) 49730–49740, http://dx.doi.org/10.1109/ACCESS.2020.2980136.

[32] X.-H. Nguyen, K.-H. Le, Robust detection of unknown DoS/DDoS attacks in IoT networks using a hybrid learning model, Internet Things 23 (2023) 100851, http://dx.doi.org/10.1016/j.iot.2023.100851.

[33] G. De La Torre Parra, P. Rad, K.-K.R. Choo, N. Beebe, Detecting internet of things attacks using distributed deep learning, J. Netw. Comput. Appl. 163 (2020) 102662, http://dx.doi.org/10.1016/j.jnca.2020.102662.

[34] A.K. Sahu, S. Sharma, M. Tanveer, R. Raja, Internet of Things attack detection using hybrid Deep Learning Model, Comput. Commun. 176 (2021) 146–154, http://dx.doi.org/10.1016/j.comcom.2021.05.024.

[35] W. Zhou, C. Xia, T. Wang, X. Liang, W. Lin, X. Li, S. Zhang, HIDIM: A novel framework of network intrusion detection for hierarchical dependency and class imbalance, Comput. Secur. 148 (2025) 104155, http://dx.doi.org/10.1016/j.cose.2024.104155.

[36] W. Lin, C. Xia, T. Wang, Y. Zhao, L. Xi, S. Zhang, Input and output matter: Malicious traffic detection with explainability, IEEE Netw. 39 (2) (2025) 259–267, http://dx.doi.org/10.1109/MNET.2024.3481045.

[37] L.D. Manocchio, S. Layeghy, W.W. Lo, G.K. Kulatilleke, M. Sarhan, M. Portmann, FlowTransformer: A transformer framework for flow-based network intrusion detection systems, Expert Syst. Appl. 241 (2024) 122564, http://dx.doi.org/10.1016/j.eswa.2023.122564.

[38] Z. Wu, H. Zhang, P. Wang, Z. Sun, RTIDS: A robust transformer-based approach for intrusion detection system, IEEE Access 10 (2022) 64375–64387, http://dx.doi.org/10.1109/ACCESS.2022.3182333.

[39] J. Hao, P. Chen, J. Chen, X. Li, Effectively detecting and diagnosing distributed multivariate time series anomalies via unsupervised federated hypernetwork, Inf. Process. Manage. 62 (4) (2025) 104107, http://dx.doi.org/10.1016/j.ipm.2025.104107.

[40] W.W. Lo, S. Layeghy, M. Sarhan, M. Gallagher, M. Portmann, E-GraphSAGE: A graph neural network based intrusion detection system for IoT, in: NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium, 2022, pp. 1–9, http://dx.doi.org/10.1109/NOMS54207.2022.9789878.

[41] Z. Sun, A.M. Teixeira, S. Toor, GNN-IDS: Graph neural network based intrusion detection system, in: Proceedings of the 19th International Conference on Availability, Reliability and Security, in: ARES 2024, ACM, 2024, pp. 1–12, http://dx.doi.org/10.1145/3664476.3664515.

[42] D. Pujol-Perich, J. Suarez-Varela, A. Cabellos-Aparicio, P. Barlet-Ros, Unveiling the potential of graph neural networks for robust intrusion detection, ACM SIGMETRICS Perform. Eval. Rev. 49 (4) (2022) 111–117, http://dx.doi.org/10.1145/3543146.3543171.

[43] Y. Meidan, M. Bohadana, Y. Mathov, Y. Mirsky, A. Shabtai, D. Breitenbacher, Y. Elovici, N-BaIoT—Network-based detection of IoT botnet attacks using deep autoencoders, IEEE Pervasive Comput. 17 (3) (2018) 12–22, http://dx.doi.org/10.1109/mprv.2018.03367731.

[44] B. Yan, G. Han, Effective feature extraction via stacked sparse autoencoder to improve intrusion detection system, IEEE Access 6 (2018) 41238–41248, http://dx.doi.org/10.1109/ACCESS.2018.2858277.

[45] B. Min, J. Yoo, S. Kim, D. Shin, D. Shin, Network anomaly detection using memory-augmented deep autoencoder, IEEE Access 9 (2021) 104695–104706, http://dx.doi.org/10.1109/ACCESS.2021.3100087.

[46] H. Lu, T. Wang, X. Xu, T. Wang, Cognitive memory-guided AutoEncoder for effective intrusion detection in internet of things, IEEE Trans. Ind. Inform. 18 (5) (2022) 3358–3366, http://dx.doi.org/10.1109/TII.2021.3102637.

[47] S.I. Popoola, B. Adebisi, M. Hammoudeh, G. Gui, H. Gacanin, Hybrid deep learning for botnet attack detection in the internet-of-things networks, IEEE Internet Things J. 8 (6) (2021) 4944–4956, http://dx.doi.org/10.1109/JIOT.2020.3034156.

[48] Y. Yang, K. Zheng, C. Wu, Y. Yang, Improving the classification effectiveness of intrusion detection by using improved conditional variational AutoEncoder and deep neural network, Sensors 19 (11) (2019) 2528, http://dx.doi.org/10.3390/s19112528.

[49] C. Zhou, R.C. Paffenroth, Anomaly detection with robust deep autoencoders, in: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '17, ACM, 2017, pp. 665–674, http://dx.doi.org/10.1145/3097983.3098052.

[50] S. Chen, W. Guo, Auto-encoders in deep learning—A review with new perspectives, Mathematics 11 (8) (2023) 1777, http://dx.doi.org/10.3390/math11081777.

[51] P. Vincent, H. Larochelle, Y. Bengio, P.-A. Manzagol, Extracting and composing robust features with denoising autoencoders, in: Proceedings of the 25th International Conference on Machine Learning - ICML '08, ICML '08, ACM Press, 2008, pp. 1096–1103, http://dx.doi.org/10.1145/1390156.1390294.

[52] D. Gong, L. Liu, V. Le, B. Saha, M.R. Mansour, S. Venkatesh, A.v.d. Hengel, Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, ICCV, 2019.

[53] M. Sarhan, S. Layeghy, M. Portmann, Evaluating standard feature sets towards increased generalisability and explainability of ML-based network intrusion detection, Big Data Res. 30 (2022) 100359, http://dx.doi.org/10.1016/j.bdr.2022.100359.

[54] E.C.P. Neto, S. Dadkhah, R. Ferreira, A. Zohourian, R. Lu, A.A. Ghorbani, CICIoT2023: A real-time dataset and benchmark for large-scale attacks in IoT environment, Sensors 23 (13) (2023) 5941, http://dx.doi.org/10.3390/s23135941.

[55] N. Koroniotis, N. Moustafa, E. Sitnikova, B. Turnbull, Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-IoT dataset, 2018, arXiv:1811.00701.

[56] A. Alsaedi, N. Moustafa, Z. Tari, A. Mahmood, A. Anwar, TON_IoT telemetry dataset: A new generation dataset of IoT and IIoT for data-driven intrusion detection systems, IEEE Access 8 (2020) 165130–165150, http://dx.doi.org/10.1109/ACCESS.2020.3022862.

[57] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in Python, J. Mach. Learn. Res. 12 (2011) 2825–2830.

[58] V.L. Cao, M. Nicolau, J. McDermott, A hybrid autoencoder and density estimation model for anomaly detection, in: Parallel Problem Solving from Nature, PPSN XIV, Springer International Publishing, 2016, pp. 717–726, http://dx.doi.org/10.1007/978-3-319-45823-6_67.