Article

An Effective Method for Detecting Unknown Types of Attacks Based on Log-Cosh Variational Autoencoder

Li Yu * , Liuquan Xu , Xuefeng Jiang

Posted Date: 23 October 2023

doi: 10.20944/preprints202310.1419.v1

Keywords: intrusion detection; variational autoencoder; deep learning attack of unknown type

Article

An Effective Method for Detecting Unknown Types of Attacks Based on Log-Cosh Variational Autoencoder

Li Yu *, Liuquan Xu and Xuefeng Jiang

School of Computer Science and Engineering, Anhui University of Science and Technology, Huainan, 232001, Anhui, China; liyu@aust.edu.cn (L.Y.); liuquanxu@aust.edu.cn (L.X.); 2022200172@aust.edu.cn (X.J.)
* Correspondence: liyu@aust.edu.cn

Abstract: The rising number of unknown-type attacks on the Internet emphasizes the significance of developing efficient intrusion detection systems, even if machine learning-based techniques can detect unknown types of attacks. The necessity for innovative techniques is highlighted by the possibility that traditional machine learning techniques will not be sufficient for identifying these unknown types of attacks. In this research, we address this difficulty by proposing a deep learning-based solution: the log-cosh variational autoencoder (LVAE). When it comes to understanding intricate data distributions and creating freshly reconstructed data, LVAE inherits the strong modeling skills of the variational autoencoder (VAE). To better imitate discrete features of actual attacks and generate unknown types of attacks, this study develops an effective reconstruction loss term employing the logarithmic hyperbolic cosine (log-cosh) function in the log-cosh variational autoencoder (LVAE). When compared to conventional VAEs, LVAE exhibits promising potential for effectively generating data that closely resembles an unknown attack, a crucial capability for increasing the unknown attack detection rate. To categorize the generated unknown-type data, eight feature extraction and classification techniques were used. Using the most recent CICIDS2017 dataset, numerous experiments were carried out, training under varying amounts of real and unknown-type attacks. Our optimal experimental results outperformed several state-of-the-art techniques with accuracy and average F1 scores of 99.89% and 99.83%, respectively. Outstanding results were also shown by the suggested LVAE strategy for producing unknown attack data. In general, our work sets a strong basis for the accurate and efficient identification of unknown types of attacks and contributes to the development of intrusion detection techniques.

Keywords: intrusion detection; variational autoencoder; deep learning attack of unknown type

1. Introduction

1.1. Research Background

The Internet's rapid expansion has permeated every aspect of contemporary life and produced enormous volumes of sensitive data. Unfortunately, hackers now have more opportunity to take advantage of vulnerabilities as a result of the abundance of data. As the Internet continues to develop, so too are the associated assault types, leading to the emergence of ever-more-sophisticated and unknown types of attacks. In addition to consuming precious network resources, these unknown attacks on network traffic have a detrimental effect on the functionality of hosts and devices on the network. Furthermore, by breaching network users' private information security and confidentiality and perhaps endangering national and social security, these attacks represent a serious concern [1]. Since intrusion detection systems (IDS) monitor network traffic and recognize unusual user behavior, they are an essential technology for network security. Unusual conduct or non-traditional data transmission techniques are examples of these activities. When IDS notices these kinds of deviations, it immediately creates alerts and notifies the right staff to take the necessary action. Generally speaking, there are two types of network-based intrusion detection systems: anomaly-based and signature-based. Network traffic patterns are compared to known attack signatures or features by signature-based intrusion detection systems to identify assaults. These methods, however, are unable to identify novel attack variations, unknown attacks, or attacks from related families. By contrast, anomaly-based intrusion detection systems (IDS) identify and label aberrant communications when they detect deviations from a model of typical user behavior. Even though anomaly-based intrusion detection systems (IDSs) are capable of identifying unknown or zero-day attacks, it can be difficult to precisely record network users' ever-changing behavior. As a result, reducing false alarms and attaining high detection accuracy for unknown assaults depend heavily on precisely understanding user behavior. Because of this, research efforts aimed at creating intrusion detection systems that can successfully identify previously unidentified attack types have gradually acquired traction and are currently regarded as a hot topic in the research community.

Many successful approaches have been developed in this field during the last ten years, the majority of which use predetermined rules for classification. Essentially, these techniques make use of complex algorithms based on machine learning. These techniques include those based on support vector machines (SVM) [2] and random forests (RF) [3], which have been used in the past to distinguish different attack classes. These machine learning-based techniques, however, are often not very good at learning large amounts of high-dimensional data characteristics; instead, they are more inclined to focus on learning low-dimensional data features. The following are the primary shortcomings of conventional machine learning techniques: (a) These models don't do well at identifying unknown kinds of attacks since they rely heavily on predetermined traffic features or attributes. And their main focus is on identifying known attacks [4]. (b) Because network architectures are dynamic, traditional intrusion detection algorithms are neither scalable nor adaptable enough to identify unknown types of attacks [5]. (c) Furthermore, these methods base their training on labeled data, which is expensive to compute and easily manipulated using artificial data, resulting in a general degradation in performance [5]. Deep learning techniques have tremendous promise for effectively extracting significant features from large amounts of high-dimensional data [6-9], which helps to solve the issue of constantly evolving network patterns. Pre-training traffic samples and real network traffic data vary significantly, for instance, due to variances in network packet sizes and the relevant communication protocols utilized in the network traffic. In the end, this lowers the accuracy of identifying unknown types of attacks [10].

1.2. Related work

For detecting unknown types of attacks, many novel approaches have emerged [11]. In order to efficiently identify novel and unknown attack types, Singh, A. presented an edge-based hybrid intrusion detection system that integrates three different categorization techniques. Remarkably, their experimental results demonstrate an astounding 93% decrease in false alarms, which greatly increases the overall detection rate for attacks of the unknown sort [12]. Similarly, Zoppi, T. examined the efficacy of 47 distinct algorithms for identifying unidentified attack types in a thorough analysis with a variety of 11 datasets. Notably, compared to other methods already in use, their experimental results demonstrate the superiority of the meta-learning strategy in identifying unknown attack types [13].

A thorough analysis of active learning techniques is provided in [14]. This paper explores the use of k-nearest neighbor techniques in conjunction with deep neural networks to facilitate adaptive incremental detection of unknown types of attacks. Parallel to this, Soltani, M. offers a unique way for identifying zero-day threats using the combination of deep models and clustering techniques. Their technique efficiently clusters and identifies zero-day assaults [15]. Furthermore, Mahdavi, E. presents a method for detecting unknown attacks by combining incremental learning and transfer learning. Results from this method on the KDD99 and CICIDS2017 datasets are encouraging [16]. Moreover, Mananayaka, A.K. employs a combination of four machine learning techniques to concentrate on automatic feature selection. On both datasets, their two-stage hybrid learning technique for classification yields excellent f1 scores and accuracy [17]. Zhou, X. proposed a hierarchical adversarial attack generation technique in conjunction with a hierarchical node selection algorithm to efficiently identify previously unidentified attack types. Their method effectively improves the capacity to identify unknown threats [18].

In a similar vein, Kumar, V. created a two-phase intelligent network technique, especially for identifying zero-day threats. Their technique obtains impressive accuracy rates of over 90% on CICIDS 2018 and real-time datasets by using created signatures [19]. Moreover, Sarhan, M. suggested a zero-sample learning technique to assess how well machine learning-based detection systems perform against unidentified threats. This technique provides insightful information on how well these systems identify and mitigate unknown threats [20]. Sheng, C. created a self-growing attack traffic classification system based on density-based heuristic clustering to improve the detection of unknown forms of attacks. This technique makes it possible to automatically detect unknown attacks in real-time [21]. Likewise, in order to avoid overfitting and identify unknown attacks, Hairab, B.I. used a convolutional neural network and included L1 and L2 regularization algorithms [22].

In contrast, Araujo-Filho, P.F.d. detected zero-day assaults without labeled data by combining temporal convolutional networks, self-attention, and generative adversarial networks [23]. In contrast to earlier techniques, Verkerken, M. studied a multi-level hierarchical approach that combines neural network techniques, autoencoders, random forests, and one-class support vector machines to detect zero-day attacks. This method is able to reliably and effectively detect zero-day attacks with an astounding 96% accuracy [24]. Sohi, S.M. For the first time, it has been demonstrated that the use of recurrent neural networks helps generate unknown types of attacks from malware. And the detection rate has improved by an amazing 16.67% [25]. A distributed anomaly detection technique is developed that employs a mixed Gaussian distribution based on correntropy to detect zero-day assaults instantly. Positive findings are obtained from experiments on the NSL-KDD and UNSW-NB15 datasets [26]. Debicha, I. combines several adversarial classifiers that use migration learning and use their individual judgments to identify attacks [27].

A thorough summary of machine learning-based techniques that have been the subject of much research in the last ten years and have shown remarkable results is given in [28]. Using machine learning approaches, the authors of [29] suggest a three-layer design for tasks related to preprocessing, binary classification, and multi-class classification. Further summarizing current developments in deep learning techniques for identifying unknown attacks, Sabeel, U. highlights a number of strategies that have demonstrated exceptional performance [30]. Rani, S.V.J. presents a revolutionary approach that achieves an astounding accuracy of 99.07% by combining deep hierarchical neural networks with machine learning [31]. A detection technique based on a convolutional neural network and meta-learner is reported in [32], making use of a sizable dataset that was produced by merging five distinct datasets. The experimental findings show how quickly the method may adapt. Shin, G.Y. suggests a novel method that improves accuracy for every kind of attack on the NSL-KDD dataset by training a fuzzy c-mean eigenanalysis model at decision boundary points [33]. Furthermore, Lan, J. presents an unsupervised domain adaptation technique and a hierarchical attention triple network, both of which successfully and accurately identify previously unknown assaults, as the experimental findings show [34]. Zavrak, S. suggested a novel method that combines an autoencoder and a variational autoencoder in order to efficiently detect unknown threats based on stream characteristics. This method is useful in identifying unknown attacks, as experimental findings showed that it outperforms one-class support vector machines and standard autoencoders [35].

The author investigated two deep generation techniques: adversarial autoencoder with conditional denoising and autoencoder combined with the K-nearest neighbor algorithm, in an effort to create an intrusion detection system with strong detection capabilities for unknown threats. The authors assessed each of these three approaches' performance using experiments on four datasets. The outcomes amply demonstrated the potential of the suggested approach for boosting the robustness of intrusion detection systems [36]. Moreover, Long, C. presents an approach that combines autoencoders by selecting the best subset of features using feature selection first. The next step is to integrate many autoencoders to identify unknown threats. The experimental findings demonstrate the method's robustness and efficacy in identifying unknown attacks [37].

To achieve a two-stage detection of unknown forms of attacks, a unique strategy is proposed that integrates extreme value theory with a conditional variational autoencoder. To efficiently learn the distribution of normal data, a benign clustering technique is also used. The suggested approach performs admirably, with a low false alarm rate and a high detection rate, according to experimental evaluations carried out on two datasets [38]. The use of adversarial autoencoder-based and two-way generative adversarial network-based techniques for identifying zero-day attacks is examined in a related study by [39]. Achieving F1 scores over 85% and even 99%, the two-way generative adversarial networks.

To tackle the dynamic nature of attacks, Jin, D. introduces a novel evolvable technique that uses discriminative autoencoders and integrates a federated incremental learning methodology to update the model on a regular basis. With this strategy, the accuracy rate is over 86%, which is excellent [40]. Additionally, Yang, L. presents a real-time approach to detecting unknown threats by using autoencoders to extract features and categorize network traffic. To minimize feature size, the method additionally uses a cluster analysis technique. According to experimental results, applying this strategy improves accuracy by an astounding 19% [41]. Furthermore, Zahoora, U. investigates a novel method that blends heterogeneous voting-based integration with deep compression autoencoders. Experiments demonstrate that the proposed strategy is effective in identifying attacks, with promising outcomes [42]. Boppana, T.K. presents a novel method for detecting unknown threats by combining autoencoders with unsupervised generative adversarial networks. The trial findings show an amazing 97% F1 score, indicating that the approach is effective in correctly recognizing attacks [43].

In a related work, Kim, C. improves the identification of unknown threats by creating a useful technique that combines autoencoders with a one-class classifier. The approach attains a remarkable 97.1% accuracy rate, underscoring its capacity to precisely identify many kinds of assaults [44]. Li, R. presents a technique that combines many LSTM autoencoders in order to handle the problem of identifying zero-day attacks. Surprisingly, tagged attack data is not needed for this approach's training. The efficacy of this approach in identifying various types of network attacks is exhibited by the outcomes of experiments [45]. Furthermore, Li, Z. presents a denoising autoencoder generative adversarial network method that trains a model by generating high-quality data. The method achieves outstanding accuracy rates of 98.6% on the NSL-KDD dataset and 98.5% on the UNSW-NB15 dataset, underscoring its ability to accurately detect attacks [46].

We develop a unique intrusion detection system (IDS), called LVAE, that may effectively identify unknown kinds of attacks in order to overcome the issues mentioned above. We integrate a logarithmic hyperbolic cosine (log-cosh) reconstruction loss function, which successfully optimizes the potential space between input data and reconstructed data, in contrast to conventional variational autoencoders (VAEs). As a result, LVAE makes a significant rise in the quality of generated unknown attacks. Eight methods are used to extract features, and several techniques are applied to identify unknown attacks before the most accurate way is chosen. We demonstrate that LVAE reliably detects unknown types of assaults through extensive testing and comparison, guaranteeing strong and efficient network security.

1.3. The contribution of the work in this paper

The contribution of the work in this paper is as follows:

1. We introduce a novel approach for identifying unknown assaults called LVAE, which we developed by creating an effective reconstruction loss term that makes use of the logarithmic hyperbolic cosine (log-cosh) function. The intricate distribution of actual attack data can be accurately captured by this function, allowing for the simulation of discrete features for modeling that enhance the creation of novel, unknown attacks;
2. We utilize eight different techniques for feature extraction and data classification, choosing the most accurate approach at the end to guarantee outstanding performance in identifying unknowing assaults;

3. We trained our model using the latest CICIDS 2017 dataset, incorporating varying numbers of real and unknown-type attacks. In comparison to several state-of-the-art methods, our LVAE approach demonstrated superior performance, surpassing the most recent advancements and significantly enhancing the detection rate of unknown-type attacks.

2. Materials and Methods

Figure 1 depicts the proposed framework. In this study, the LVAE approach is divided into two primary components: The first component is a generator that creates novel or unknown attacks. The second component is a classifier that combines eight distinct techniques for feature extraction and data categorization.

The former reconstructs the input dataset, which in turn generates data for novel attacks or unknown types of attacks. And the latter classifies the data and extracts features. The first part trains the generator to generate novel attacks or unknown types of attacks efficiently. Then, the original samples are combined with the generated samples of unknown types of attacks and fed into the integrated classifier for training. These two parts are trained independently of each other. The classifier chooses the top-performing classifier to output outcomes after assessing the effectiveness of several approaches.

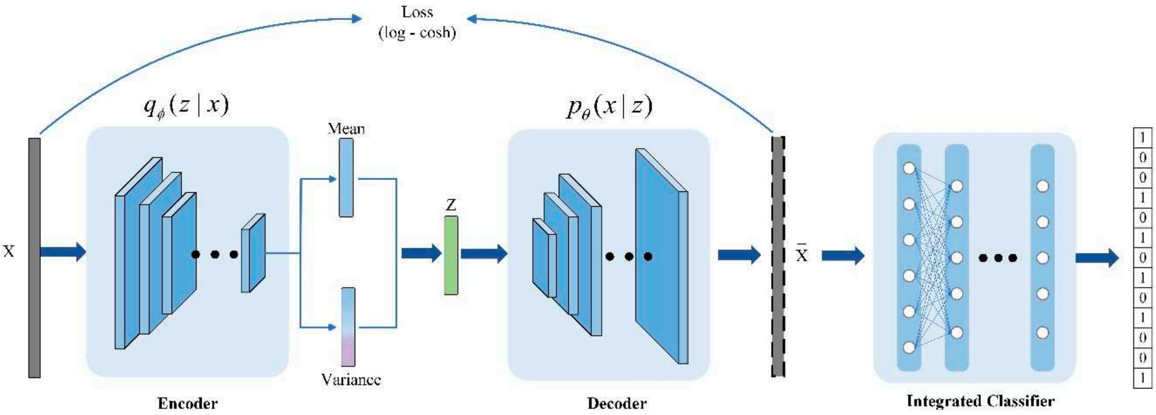


Figure 1. The overall structure of the method proposed in this paper.

2.1. Log-cosh Variational Auto Encoder

This section shows how conventional VAE fits the input data using a multidimensional Gaussian distribution, which subsequently identifies the input data's display distribution. The encoder network $q_\phi(z \mid x)$ makes up the first portion of the typical VAE, with being the model parameters. The decoder network $p_\theta(x \mid z)$ makes up the second section, with θ being the model parameters. In other words, the encoder network maps the input X into the feature variable Z. Its primary function is to effectively compress the input data into a potential low-dimensional space. The decoder's job, on the other hand, is to map the feature variable Z back to the data $\bar{X}$ by reconstructing the feature variable Z. Consequently, the decoder can be used to create new data by reconstructing the latent spatial feature variable Z using the VAE approach. Equation 1 summarizes the conventional VAE loss function formulation.

$$L_{VAE} = \mathrm{E}[\log p(X \mid Z)] - D_{KL}[q(Z \mid X) \,||\, p(Z)] \tag{1}$$

There are two fundamental parts of the VAE loss function in Equation 1. The logarithmic reconstruction loss term, or the probability distribution $\mathrm{E}[\log p(X \mid Z)]$ from which the data is derived, is the initial component. Its goal is to minimize the squared L_2 loss between the reconstructed data $\bar{X}$ and the input data X. The second portion is the Kullback-Leibler (KL) divergence $D_{KL}[q(Z \mid X) \,||\, p(Z)]$, which is the KL divergence measure used to minimize the difference between the learned distribution $q(Z \mid X)$ and the previously defined distribution $p(Z)$. It facilitates the efficient learning of the input data's latent space representation by the VAE model. The VAE loss function encourages the learning distribution to match the predefined distribution while simultaneously optimizing reconstruction accuracy by integrating these two elements. The VAE model can provide excellent reconstructions and successfully capture the underlying structure of the input data because of this dual goal.

The squared L_2 loss function is an essential component of the VAE framework and is represented by Equation 2. In this case, the input vector is represented by X, while the reconstructed vector is represented by $\bar{X}$. The ith sample in X is denoted by the term x_i, while the ith sample in $\bar{X}$ is denoted by the term $\bar{x}_i$. In order to ensure correct reconstruction of the original data, this loss function seeks to minimize the difference between the input vector and its corresponding recreated vector.

$$L_{\text{reconstruction}} = \sqrt{\sum_{i=1}^{n} (x_i - \bar{x}_i)^2}^{\,2} = \sum_{i=1}^{n} | x_i - \bar{x}_i |^2 = L_2^{\,2} \tag{2}$$

The primary purpose of the conventional VAE is data generation; nonetheless, it has significant limitations. Reconstructing data using a low-dimensional latent spatial feature variable Z is one such difficulty. Dealing with intrusion detection datasets, like the CICIDS2017 dataset, which frequently contains discrete and high-dimensional data, makes this extremely difficult. As a result, the efficiency of the created data is decreased since there is typically a large reconstruction loss between the generated and input data.

Our goal in solving this problem is to balance the weights given to the reconstructed data and the latent space in Equation 1. Nevertheless, the estimated reconstruction loss is typically quite tiny, leading to a limited error margin and a negligible penalty for the reconstruction loss component. Consequently, the KL divergence, represented by the second portion of Equation 1, becomes the dominant part of the loss function.

An efficient way to address this issue and take different loss regions into consideration is to make the reconstruction loss weight heavier when the loss is tiny and to keep the reconstruction loss from increasing too much when the reconstruction error rises linearly. Therefore, we use the log-cosh function in place of the logarithmic reconstruction loss term in Equation 1 (i.e., the first portion). The reconstruction loss is concentrated closer to the origin when the logarithmic hyperbolic cosine function is used, avoiding excessive penalization when the reconstruction error is high. Equation 3 provides a clear formula for the logarithmic hyperbolic cosine function.

$$f(x) = \log(\cosh(x)) = \log \frac{e^{x} + e^{-x}}{2} \tag{3}$$

In this work, Equation 4 illustrates the log-cosh loss function that we employ. X and $\bar{X}$ are the input vector and reconstruction vector, respectively.

$$L_{\log-\cosh} = \sum_{i=1}^{n} \log(\cosh(X_i - \bar{X}_i)) = \sum_{i=1}^{n} \log \frac{e^{X_i - \bar{X}_i} + e^{-(X_i - \bar{X}_i)}}{2} \tag{4}$$

Briefly stated, the logarithmic reconstruction loss term (i.e., the first part) of Equation 1 is replaced by the log-cosh function in our LVAE technique, which reconstructs the loss. As demonstrated by Equation 5, we finally create the loss function.

$$Loss = \sum_{i=1}^{n} \log(\cosh(X_i - \bar{X}_i)) - D_{KL}[q(Z \mid X) \,||\, p(Z)] \tag{5}$$

Algorithm 1 Train LVAE to generate new or unknown types of attack data

```
Input: data set X.
Output: new or unknown type of attack dataset $\overline{X}$ .

Data preprocessing: removing redundant information, filling in missing values and normalization.
Iteration
    for number of epochs learned do
        for mini-batch quantities do
            The data $x_i$ is input to the encoder to obtain the feature variable Z.
            The feature variable Z is input to the decoder to obtain the reconstructed data $\overline{x}_i$ .
            Backpropagation calculates the loss values and gradients for Equation 5.
            Gradient descent.
        end for
    end for
Until Equation 5 converges.
Output new or unknown type of attack dataset $\overline{X}$ .
```

In order to provide novel and unknown kinds of attack data, we present an enhanced VAE model in this work that uses log-cosh as the reconstruction loss function. Next, the generated data is used to assess how well the model detects unknown assaults. In contrast to conventional VAEs, we use Equation 5 as the data generation phase's loss function. The input data is first mapped to the latent space feature variable Z using the encoder, which is then fed into the trained decoder to reconstruct the data. Lastly, the classification stage is used to classify the newly generated, unknown sorts of attack data. Algorithm 1 outlines the training process for the LVAE method to generate new or unknown types of attack data, where x_i is the ith sample of the input dataset and $\overline{x}_i$ represents the ith sample of the output dataset.

2.2. Classification stage

Many studies have proven that machine learning [28] and deep learning [30] techniques are useful for identifying unknown kinds of attacks. By combining eight distinct techniques—including a mixture of deep learning and machine learning methods—we hope to significantly improve the detection rate for unknown threats in this study. These methods include multilayer perceptron, naive bayes, decision trees, random forests, support vector machines, logistic regression, gradient boosting, and gated recurrent units.

To guarantee optimal performance, we preprocess the data before training the integrated classifier. During this preprocessing phase, the data is normalized, superfluous information is eliminated, and missing values are filled in. We start by deleting any duplicate data from the dataset. We then ensure that the data is complete for additional analysis by adding 0 to any missing values.

We use Min-Max normalization to address the problem of features with a large range of values dominating the effect. According to Equation 6, this normalization method scales each feature to a value between 0 and 1. Through this approach, we guarantee that every feature makes a proportionate contribution to the overall classification process. and the ith characteristic is indicated by x_i .

$$x_i = \frac{x_i - (x_i)_{\min}}{(x_i)_{\max} - (x_i)_{\min}} \tag{6}$$

Algorithm 2 Training the Classifier

Input: data set $\overline{X}$.

Output: Classification results.

```
Data preprocessing: removing redundant information, filling in missing values and
    normalization.
Iteration
    for c=1 in rang(1,9)
        Input data $\overline{x}_i$ into the classifier to get the predicted value $h_\theta(\overline{x})_i$ .
        Backpropagation calculates Equation 6 losses and gradients.
        Gradient descent.
    end for
Until Equation 6 converges.
Output the classification results.
```

The preprocessed data is utilized to train the integrated classifier after the data preprocessing step. Equation 7 defines the loss function for the classification stage. The true value of the ith sample is denoted by y_i in this equation, the predicted value of the ith sample is denoted by $h_\theta(x)_i$, and the number of samples is represented by n. Algorithm 2 describes the training procedure for the classification stage. The number of classifiers in the integrated classifier is denoted by C in this instance. In the rebuilt dataset, $\overline{x}_i$ stands for the ith sample, and $h_\theta(\overline{x})_i$ stands for the corresponding anticipated value.

$$L_{\text{Classification}} = -\frac{\sum_{i=1}^{n} y_i \log h_\theta(x)_i + (1-y_i)\log(1-h_\theta(x)_i)}{n} \tag{7}$$

3. Experiments

To conduct the experiments in our experimental setup, we used a personal laptop. The laptop that we utilized for the study had the following specifications: it had an AMD Ryzen TM 7 6800H CPU that ran at 3.20 GHz and Windows 10 installed on it. To meet the studies' computing demands, the laptop was outfitted with 16 GB of RAM. In addition, we used a laptop GPU with a GeForce RTX 3060 to speed up the calculations.

We used Python 3.7 programming and TensorFlow version 2.1 as the deep learning framework to carry out our tests. These resources gave us a reliable and effective setting in which to carry out our research and evaluate the outcomes.

3.1. Description of the dataset

The CICIDS2017 dataset [47], which includes both traditional and cutting-edge attacks, is what we use in this section. Introduced in 2018, the goal of this dataset is to imitate abstract network activities and inject them into attack scenarios in order to simulate actual network attacks. Expert-created attack profiles and weekly network activity comprise the dataset.

The CICIDS2017 dataset includes data samples with 80 characteristics, including the number of forward packets, the minimum length of the flow, and the maximum time between two flows. It is noteworthy that the distribution of the dataset is extremely unbalanced. For example, whereas there are over two million data samples from normal traffic, there are only 11 instances of Heartbleed assaults. Because of this feature, the CICIDS2017 dataset is more indicative of cyberattacks that occur in the real world. Table 1 shows the distribution of the dataset in detail.

Our LVAE model aims to efficiently identify unknown sorts of attacks, in contrast to traditional methods. In order to do this, we create the training set by choosing an equal number of attack samples and normal samples at random, which together account for 50% of the dataset. In the same way, we select the same number of samples for the test set as we did for the training set. This guarantees that we can precisely assess our method's detection ability for unknown threats. Table 2 provides a detailed description of the dataset that we used for our research.

3.2. Assessment of indicators

The following metrics are commonly used in IDS to assess the performance of the method: True positives (TP) represent samples that correctly predicted the attack class, while true negatives (TN) represent samples that correctly predicted the normal class. False positives (FP) indicate samples where the normal class was predicted to be the attack class, and on the contrary, false negatives (FN) represent samples where the attack class was predicted to be the normal class.

Table 1. Detailed distribution of CICIDS2017 dataset.

Traffic Class	Label	Numbers	Ratio
Benign	Benign	2273,097	80.30%
DDoS	DDoS	128,027	4.52%
DoS	DoS Hulk	231,073	8.16%
	DoS GoldenEye	10,293	0.36%
	DoS slowloris	5,796	0.20%
	DoS Slowhttptest	5,499	0.19%
Port Scan	Port Scan	158,930	5.61%
Botnet	Bot	1,966	0.07%
Brute Force	FTP-Patator	7,938	0.28%
	SSH-Patator	5,897	0.20%
Web Attack	Web Attack – Brute Force	1,507	0.05%
	Web Attack – Sql Injection	21	0.001%
	Web Attack – XSS	652	0.02%
Infiltration	Infiltration	36	0.002%
Heartbleed	Heartbleed	11	0.001%
Total	N	2830,743	100%

Table 2. We use a detailed distribution of the dataset.

Training set distribution	Training dataset	Test set distribution	Test dataset
Benign (50%)	10,000	Benign 100	2,000+8,000(generated)
	20,000		2,000+18,000(generated)
	30,000		2,000+28,000(generated)
	40,000		2,000+38,000(generated)
	50,000		2,000+48,000(generated)
	60,000		2,000+58,000 generated)
Attack (50%)	70,000	Attack 1,900	2,000+68,000(generated)
	80,000		2,000+78,000(generated)
	90,000		2,000+88,000(generated)

Accuracy is the ratio of correctly classified samples to all samples, as shown in Equation 8.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{8}$$

Precision is the ratio of samples correctly predicted as attacks to the total number of positively predicted samples, as shown in Equation 9.

$$Precision = \frac{TP}{TP + FP} \tag{9}$$

Recall is the ratio of samples correctly predicted as attacks to the total number of negatively predicted samples, as shown in Equation 10.

$$Recall = \frac{TP}{TP + FN} \tag{10}$$

The F1 score is the metric used to balance precision and recall and is the most important metric for evaluating a method. P means precision, and R means recall, as shown in Equation 11.

$$F1 = 2 \times \frac{PR}{P + R} \tag{11}$$

3.3. Setting of model hyperparameters

The stage of creating a novel or unknown kind of attack in our suggested LVAE method is divided into two sections: the encoder and the decoder. The ReLu activation function is used in the remaining hidden levels of the encoder and decoder, whereas the Sigmoid activation function is used in the last layer. The loss function is optimized by the Adam optimizer after the model has been trained for 100 epochs with a batch size of 10.

For the integrated classifier section, our method combines eight distinct approaches to detect new or unknown types of attacks. The key parameters for each method are as follows:

1. Multilayer Perceptron (MLP): The hidden layer consists of 80 nodes with ReLu activation. The model is trained for 5 epochs with a batch size of 5, and the Adam optimizer is used for loss optimization;
2. Gaussian-based Naive Bayes (GaussianNB): No specific priorities are set;
3. Decision Tree (DT): The criterion is set to entropy, and the maximum number of tree layers is set to 4. The remaining parameters use default values;

4. Random Forest (RF): The number of estimators is set to 100, while the other parameters use default values;
5. Support Vector Machine (SVM): The gamma parameter is set to scale, and C is set to 1;
6. Logistic Regression (LR): The penalty is set to L2, C is set to 1, and the maximum number of iterations is set to 1200,000;
7. Gradient Boost (GB): The random_state is set to 0;
8. Gated Recurrent Unit (GRU): The hidden layer consists of 80 nodes with a sigmoid activation function. Dropout is set to 0.2. The output layer utilizes the softmax activation function. The model is trained for 5 epochs with a batch size of 10, and the Adam optimizer is used for loss optimization.

Our LVAE approach accurately detects unknown sorts of attacks by carefully choosing and fine-tuning the model parameters, as previously mentioned. The choice of ideal hyperparameters is critical to the model's overall performance, and we have carried out a number of tests to identify these ideal values. In the face of new dangers, this guarantees that our approach will continue to be successful in identifying unknown attacks.

4. Results and discussion

In this segment, we showcase our experiment's outcomes and offer an in-depth analysis and conversation on the acquired results. We also talk about the computational cost of each LVAE approach component.

1. Experimental Outcomes and Analysis: In order to assess the effectiveness of our LVAE technique, we performed a number of experiments. The results demonstrate that our approach is able to accurately detect attacks of unknown types. We make sure the model is successful against new threats by carefully choosing and fine-tuning its parameters. Our method's overall performance was further improved by carefully determining the ideal hyperparameters through experimentation;
2. Computational Cost Analysis: To identify novel or unknown attacks, we integrate eight different techniques in the integrated classifier section. Every technique has a set of carefully chosen parameters that are used to maximize performance. Each approach has a different computing cost, but we have taken steps to guarantee efficiency without sacrificing accuracy.

All things considered, our LVAE approach shows promise in identifying unknown attacks while taking each component's computing cost into account. This makes it a workable option for practical applications where efficacy and accuracy are essential considerations.

4.1. Analysis of experimental results

Unlike conventional techniques, our method generates a new dataset by mixing the created unknown-attacks dataset with the original dataset. This new dataset is only utilized as a test set to assess LVAE's capacity to detect unknown threats. This novel testing method provides valuable insights into the performance of our approach.

Samples are chosen at random from the original dataset to train the model during the training phase. As indicated in Tables 3–5, the detection performance of various techniques is assessed using varying numbers of samples during the LVAE classification phase.

Table 3 illustrates how well various techniques detect unknown assaults with numbers of samples ranging from 10,000 to 30,000. The greatest training and testing accuracies attained were 99.95% and 99.01%, respectively, for a sample size of 10,000. Moreover, the highest possible F1 score of 98.52% was attained. The best results for testing and training were obtained with sample sizes of 20,000, 99.98%, and 99.50%, respectively. Furthermore, a 99.25% F1 score was attained. With an F1 score of 99.50%, the highest training and testing accuracies for a sample size of 30,000 were 99.93% and 99.67%, respectively.

These outcomes show how well our LVAE technique works in identifying unknown assaults with varying sample sizes. The high F1 scores and accuracy attained show the reliability and robustness of our method for recognizing and categorizing unidentified assault types.

These experimental outcomes show how well LVAE works in identifying unknown attacks. Under different amounts, it can still recognize unknown attacks with more than 99% accuracy and a more than 98% F1 score. It also fully demonstrates the superiority of our method.

For samples ranging from 40,000 to 60,000, we examine the effectiveness of various techniques in identifying unknown attacks in Table 4. We find that our model achieves a stunning 99.97% training accuracy, 99.75% testing accuracy, and 99.63% F1 score at 40,000 samples. Test accuracy is 99.80%, training accuracy is 99.94%, and the F1 score is 99.70% when the sample size is increased to 50,000. With a training accuracy of 99.99%, a testing accuracy of 99.83%, and an F1 score of 99.75%, even better performance is achieved when the sample size is increased further to 60,000.

As the number of samples rises, we see steady progress in all metrics, suggesting that our model successfully learns the underlying data properties. This demonstrates even more how well our suggested LVAE performs, especially when it comes to creating unknown attacks.

Table 5 offers an overview of the efficiency with which various techniques work in identifying unknown attacks for sample sizes ranging from 70,000 to 90,000. Our model obtains 99.98% training accuracy, 99.86% testing accuracy, and a 99.81% F1 score at a sample size of 70,000. Interestingly, the training accuracy rises to an astounding 99.99% when the sample size grows to 80,000 and 90,000, while the testing accuracy rises from 99.88% to 99.89% and the F1 score rises from 99.81% to 99.84%.

To sum up, our suggested LVAE approach improves the reconstruction loss component by utilizing the log-cosh loss function, which leads to better performance in identifying unknown attacks. We have shown through considerable experimentation that LVAE consistently and reliably detects unknown assaults.

It is nonetheless noteworthy that, at a sample size of 30,000, the Gaussian-based naive Bayes algorithm exhibits subpar performance in identifying unknown types of attacks. This can be partially explained by the significant mistakes in the test data, which have a negative effect on the prediction power of the model.

In order to verify the efficacy of our methodology, we carried out a thorough comparative study, taking into account various techniques. Notably, every study in our review has used a variety of techniques to detect unknown threats, with encouraging results.

Table 6 demonstrates the significant gains in accuracy and F1 score that our methodology achieves, demonstrating the effectiveness of our LVAE method in identifying unknown kinds of attacks.

Table 3. Performance of various methods of detection between 10,000 and 30,000 samples.

Number of samples	Methods	Train (accuracy)	Test (accuracy)	F1
Train (10,000) Test (10,000)	MLP	95.44%	89.65%	93.67%
	NB	80.84%	97.82%	97.98%
	DT	88.29%	98.95%	98.49%
	RF	**99.95%**	98.87%	98.45%
	SVM	93.36%	94.07%	96.03%
	LR	85.09%	96.40%	97.21%
	GB	99.65%	98.88%	98.46%
	GRU	80.97%	**99.01%**	**98.52%**
Train (20,000) Test (20,000)	MLP	95.19%	98.63%	98.83%
	NB	63.60%	94.37%	96.68%
	DT	91.57%	97.56%	98.31%
	RF	**99.98%**	99.47%	99.24%
	SVM	92.95%	97.12%	98.07%

	LR	83.35%	97.23%	98.12%
	GB	99.24%	97.55%	98.27%
	GRU	87.30%	**99.50%**	**99.25%**
Train (30,000) Test (30,000)	MLP	96.13%	99.02%	99.26%
	NB	79.27%	3.88%	7.46%
	DT	91.13%	98.37%	98.87%
	RF	**99.93%**	99.66%	**99.50%**
	SVM	93.62%	98.08%	98.71%
	LR	84.39%	98.17%	98.76%
	GB	99.09%	98.91%	99.13%
	GRU	88.04%	**99.67%**	**99.50%**

Table 4. Performance of various methods of detection between 40,000 and 60,000 samples.

Number of samples	Methods	Train (accuracy)	Test (accuracy)	F1
Train (40,000) Test (40,000)	MLP	96.90%	99.64%	99.61%
	NB	60.49%	98.65%	99.10%
	DT	91.73%	98.77%	99.15%
	RF	**99.97%**	99.73%	99.62%
	SVM	94.44%	98.98%	99.25%
	LR	86.85%	98.63%	99.07%
	GB	99.21%	99.74%	99.62%
	GRU	91.13%	**99.75%**	**99.63%**
Train (50,000) Test (50,000)	MLP	97.33%	98.62%	99.15%
	NB	77.73%	98.94%	99.29%
	DT	92.07%	99.03%	99.33%
	RF	**99.94%**	**99.80%**	**99.70%**
	SVM	94.94%	98.19%	98.90%
	LR	88.29%	98.90%	99.25%
	GB	99.17%	99.78%	99.69%
	GRU	93.26%	**99.80%**	**99.70%**
Train (60,000) Test (60,000)	MLP	97.41%	99.06%	99.04%
	NB	77.71%	99.13%	99.42%
	DT	92.56%	99.18%	99.43%
	RF	**99.99%**	**99.83%**	**99.75%**
	SVM	94.28%	98.97%	99.32%
	LR	88.52%	99.09%	99.38%
	GB	99.21%	**99.83%**	**99.75%**
	GRU	93.54%	99.82%	99.74%

Table 5. Performance of various methods of detection between 70,000 and 90,000 samples.

Number of samples	Methods	Train (accuracy)	Test (accuracy)	F1
Train (70,000) Test (70,000)	MLP	97.28%	99.85%	**99.81%**
	NB	59.99%	99.85%	99.79%
	DT	91.84%	99.29%	99.52%
	RF	**99.98%**	**99.86%**	99.79%
	SVM	92.07%	97.71%	98.77%
	LR	87.63%	99.21%	99.47%
	GB	99.16%	99.84%	99.79%
	GRU	91.12%	**99.86%**	99.79%
Train (80,000) Test (80,000)	MLP	97.68%	**99.88%**	**99.81%**
	NB	63.57%	99.38%	99.39%
	DT	92.67%	99.37%	99.57%
	RF	**99.99%**	99.87%	**99.81%**
	SVM	95.52%	98.85%	99.31%
	LR	89.10%	99.31%	99.54%
	GB	99.09%	99.86%	**99.81%**
	GRU	92.97%	99.87%	**99.81%**
Train (90,000) Test (90,000)	MLP	97.71%	99.32%	99.55%
	NB	65.17%	99.45%	99.63%
	DT	92.85%	99.44%	99.62%
	RF	**99.99%**	99.87%	99.83%
	SVM	95.80%	99.31%	99.55%
	LR	89.75%	99.38%	99.59%
	GB	99.07%	99.88%	**99.84%**
	GRU	95.32%	**99.89%**	99.83%

Table 6. Comparative study.

Research	Approach	Dataset	Accuracy	F1
[10]	AE, DNN	CICIDS2017	97.28%	72.81%
[38]	CVAE, Extreme value theory.	CICIDS2017	92.10%	97.15%
[24]	AE, one-class SVM, RF, neural network.	CICIDS2017	98.77%	98.97%
[48]	CNN.	CICIDS2017	94.64%	99.62%
proposed	Improved VAE, A.	CICIDS2017	**99.89%**	**99.83%**

A represents the integration of classifiers, i.e., the eight methods of MLP, GaussianNB, DT, RF, SVM, LR, GB and GRU.

4.2. Calculated cost analysis of the various components of the LVAE methodology

In this experiment, we used a personal laptop with a GeForce RTX 3060 laptop GPU to thoroughly evaluate the LVAE approach. After thorough analysis, The time requirements of each

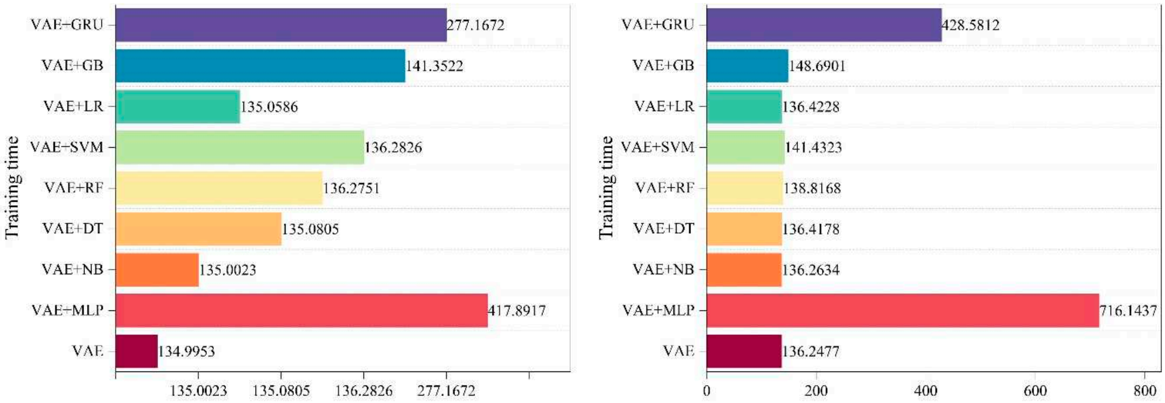
doi:10.20944/preprints202310.1419.v1

step in the LVAE approach for various training data quantities are shown in Figures 2–5. Interestingly, the VAE phase consistently needed the least amount of time for the creation of unknown attacks. This demonstrates the efficiency of our method for generating unknown attacks. The other parts stand for the total amount of time needed for the training process. It is noteworthy to notice that the combination of VAE and MLP produced the largest time consumption, with the combination of GRU coming in second. The third-highest time consumption was attributed to the combination with SVM; the time needs for the other components were comparable.

When the number of training samples rises, the total amount of time consumed stays relatively low, according to the examination of time consumption for each LVAE component. This discovery demonstrates how well LVAE performs at little computing cost when it comes to effectively identifying unknown attacks.

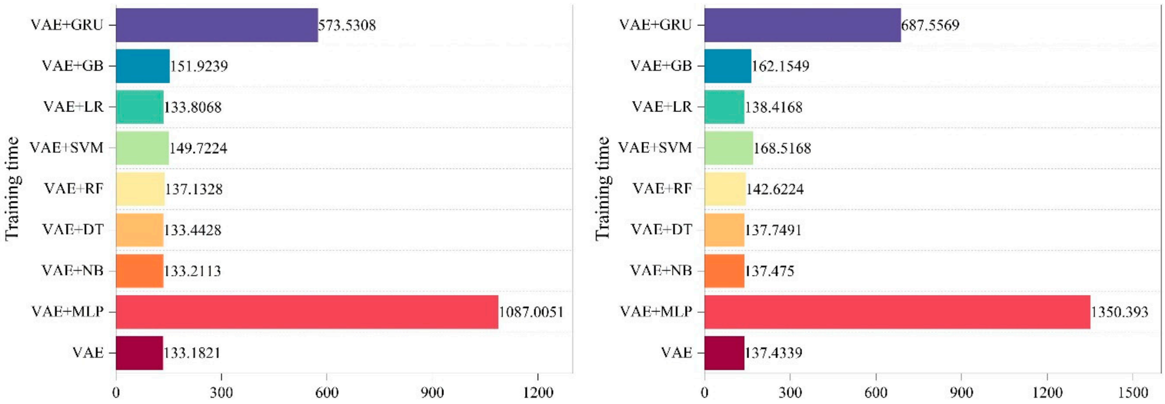
For sample sizes between 10,000 and 40,000, Figures 2 and 3 show the time consumption at different stages in the LVAE process. The VAE component, in particular, shows the least amount of time required to generate unknown assaults, indicating how effective the system is at producing them. While the other components require comparable amounts of time, MLP and GRU demand a substantial amount of time during the classification step.

The time spent at each stage of the LVAE approach for sample sizes ranging from 50,000 to 90,000 is shown in Figures 4 and 5, which supports the earlier findings even more. The enhanced VAE efficiently produces unknown attacks, which are categorized by means of a classifier that incorporates eight techniques. The time analysis shown in Figures 2–5 supports the idea that our approach effectively generates unknown attack data and learns data features quickly.



(a) Model training time (seconds) at 10,000 samples (b) Model training time (seconds) at 20,000 samples

Figure 2. Time consumed in sections between 10,000 and 20,000.

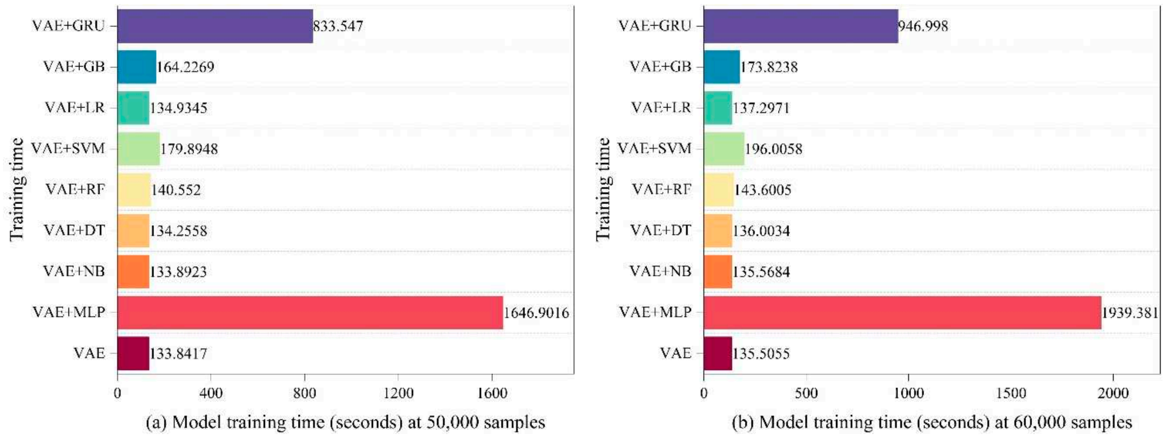


(a) Model training time (seconds) at 30,000 samples (b) Model training time (seconds) at 40,000 samples

Figure 3. Time consumed in sections between 30,000 and 40,000.

Additionally, Tables 3-5 show that LVAE was able to accurately detect unknown threats. These tables show how well our approach performs in identifying unknown attacks. The efficacy of LVAE in identifying unknown attacks is demonstrated by its persistent ability to outperform alternative approaches and achieve outstanding accuracy rates when combined with different categorization algorithms.

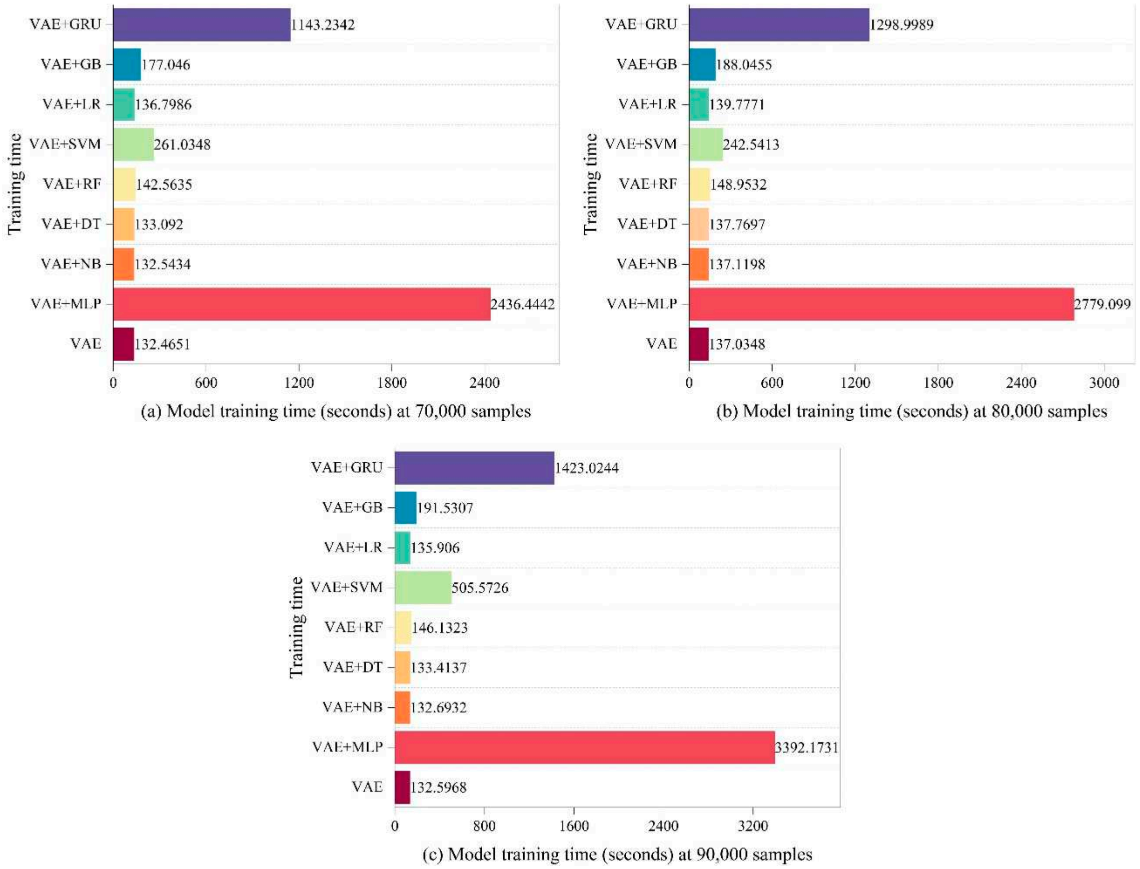
To sum up, the performance of identifying unknown attacks is greatly enhanced by our LVAE approach. The effectiveness of our technique is enhanced by the correct classification of these attacks and the efficient production of unknown attacks. Our LVAE method's efficiency and dependability are confirmed by the testing results and the time consumption analysis.



(a) Model training time (seconds) at 50,000 samples

(b) Model training time (seconds) at 60,000 samples

Figure 4. Time consumed in sections between 50,000 and 60,000.



(a) Model training time (seconds) at 70,000 samples

(b) Model training time (seconds) at 80,000 samples

(c) Model training time (seconds) at 90,000 samples

Figure 5. Time consumed in sections between 70,000 and 90,000.

Additionally, as Figure 6 shows, we evaluated the loss in our LVAE approach during the production of unknown attacks. It is clear that before the 20th epoch, the loss rapidly declines and converges to its minimum value. It then keeps up a steady state of convergence. This observation, along with the outcomes shown in Tables 3–5, adds more proof of our LVAE approach's efficacy. In just a few training epochs, it shows that our approach obtains a high detection rate and quickly converges to optimal performance. These results highlight our LVAE method's effectiveness and reliability in identifying unknown attacks.

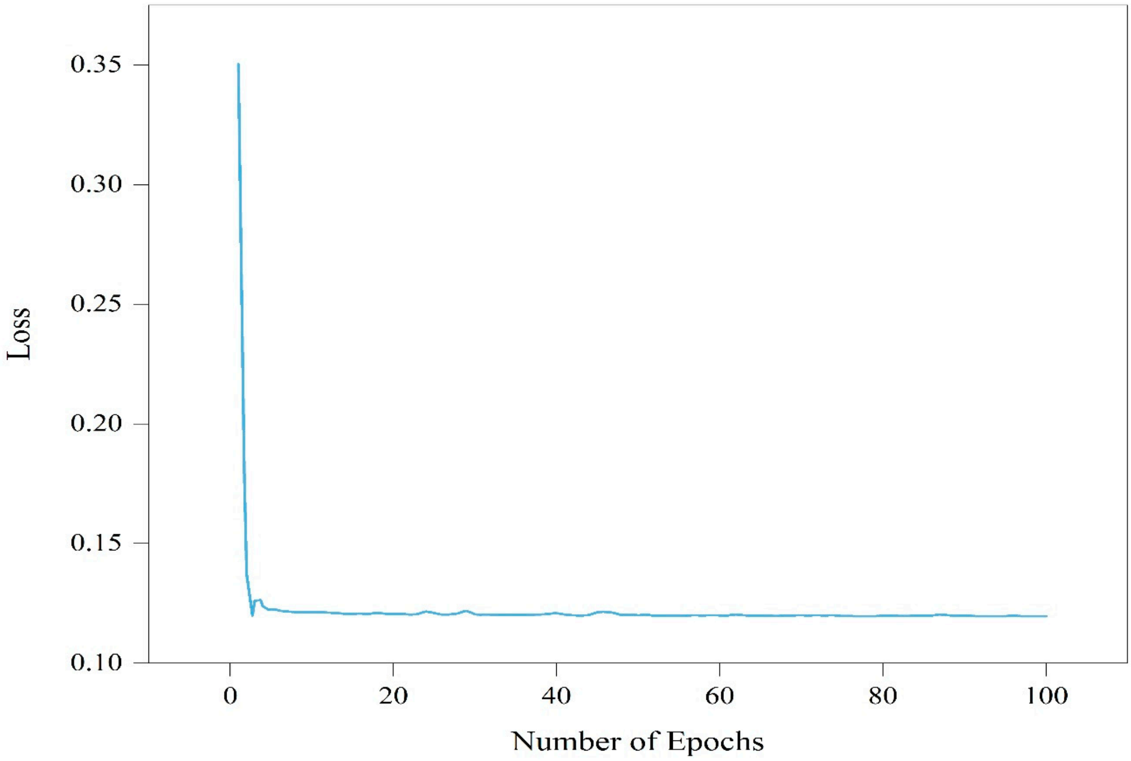


Figure 6. As the number of epochs increases, the loss of the model changes.

5. Conclusion

In this research, we present a novel LVAE technique that takes advantage of VAE's power to learn intricate data distributions and produce data reconstructions. Unknown-type attacks can be effectively generated by utilizing the log-cosh function as a reconstruction loss term in our method. Furthermore, our method employs eight classifiers to combine and efficiently understand the unique characteristics of these unknown threats.

To evaluate the performance of our LVAE method, we conduct experiments on the CICIDS2017 dataset, which contains a variety of modern attacks across different quantities. The results clearly demonstrate that our LVAE method surpasses several state-of-the-art methods in terms of detection accuracy. This highlights the effectiveness of our designed LVAE method in accurately detecting unknown attacks and improving the overall detection rate.

In future research, we aim to explore even more effective methods for detecting unknown attacks and apply them in real-world scenarios to detect unknown attacks in real-time. By continuously refining and enhancing our approach, we strive to contribute to the field of intrusion detection and enhance the security of systems and networks.

Author Contributions: Conceptualization, methodology, validation, writing, L.Y.; Conceptualization, Data curation, Formal analysis, L.X. and X.J.; Supervision, funding acquisition and review, L.Y., L.X. and X.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Project of Key Research and Development Program of Anhui Province, grant number 202104d07020010 and the China National Natural Science Foundation, Grant Number 61572034.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All datasets are available in the mentioned references.

Acknowledgments: Our thanks to the above projects for their support.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Dong, S.; Xia, Y.; Peng, T. Network Abnormal Traffic Detection Model Based on Semi-Supervised Deep Reinforcement Learning. *IEEE Transactions on Network and Service Management* **2021**, *18*, 4197-4212. [CrossRef]
2. Tian, Y.; Mirzabagheri, M.; Bamakan, S.M.H.; Wang, H.; Qu, Q. Ramp loss one-class support vector machine; A robust and effective approach to anomaly detection problems. *Neurocomputing* **2018**, *310*, 223-235. [CrossRef]
3. Kamarudin, M.H.; Maple, C.; Watson, T.; Safa, N.S. A LogitBoost-Based Algorithm for Detecting Known and Unknown Web Attacks. *IEEE Access* **2017**, *5*, 26190-26200. [CrossRef]
4. Ahmad, R.; Alsmadi, I.; Alhamdani, W.; Tawalbeh, L.a. A Deep Learning Ensemble Approach to Detecting Unknown Network Attacks. *Journal of Information Security and Applications* **2022**, *67*, 103196. [CrossRef]
5. Liu, Y.; Chen, K.; Liao, X.; Zhang, W. A genetic clustering method for intrusion detection. *Pattern Recognition* **2004**, *37*, 927-942. [CrossRef]
6. Xu, X.; Shen, F.; Yang, Y.; Shen, H.T.; Li, X. Learning Discriminative Binary Codes for Large-scale Cross-modal Retrieval. *IEEE Transactions on Image Processing* **2017**, *26*, 2494-2507. [CrossRef]
7. Luo, Y.; Yang, Y.; Shen, F.; Huang, Z.; Zhou, P.; Shen, H.T. Robust discrete code modeling for supervised hashing. *Pattern Recognition* **2018**, *75*, 128-135. [CrossRef]
8. Hu, M.; Yang, Y.; Shen, F.; Xie, N.; Shen, H.T. Hashing with Angular Reconstructive Embeddings. *IEEE Transactions on Image Processing* **2018**, *27*, 545-555. [CrossRef]
9. Xu, X.; Lu, H.; Song, J.; Yang, Y.; Shen, H.T.; Li, X. Ternary Adversarial Networks With Self-Supervision for Zero-Shot Cross-Modal Retrieval. *IEEE Transactions on Cybernetics* **2020**, *50*, 2400-2413. [CrossRef]
10. Lee, J.-S.; Chen, Y.-C.; Chew, C.-J.; Chen, C.-L.; Huynh, T.-N.; Kuo, C.-W. CoNN-IDS: Intrusion detection system based on collaborative neural networks and agile training. *Computers & Security* **2022**, *122*, 102908. [CrossRef]
11. Lopez-Martin, M.; Sanchez-Esguevillas, A.; Arribas, J.I.; Carro, B. Contrastive Learning Over Random Fourier Features for IoT Network Intrusion Detection. *IEEE Internet of Things Journal* **2023**, *10*, 8505-8513. [CrossRef]
12. Singh, A.; Chatterjee, K.; Satapathy, S.C. An edge based hybrid intrusion detection framework for mobile edge computing. *Complex & Intelligent Systems* **2022**, *8*, 3719-3746. [CrossRef]
13. Zoppi, T.; Ceccarelli, A.; Puccetti, T.; Bondavalli, A. Which algorithm can detect unknown attacks? Comparison of supervised, unsupervised and meta-learning algorithms for intrusion detection. *Computers & Security* **2023**, *127*, 103107. [CrossRef]
14. Boukela, L.; Zhang, G.; Yacoub, M.; Bouzefrane, S. A near-autonomous and incremental intrusion detection system through active learning of known and unknown attacks. In Proceedings of the 2021 International Conference on Security, Pattern Analysis, and Cybernetics（SPAC), 18-20 June 2021, 2021; pp. 374-379. [CrossRef]
15. Soltani, M.; Ousat, B.; Jafari Siavoshani, M.; Jahangir, A.H. An adaptable deep learning-based intrusion detection system to zero-day attacks. *Journal of Information Security and Applications* **2023**, *76*, 103516. [CrossRef]
16. Mahdavi, E.; Fanian, A.; Mirzaei, A.; Taghiyarrenani, Z. ITL-IDS: Incremental Transfer Learning for Intrusion Detection Systems. *Knowledge-Based Systems* **2022**, *253*, 109542. [CrossRef]
17. Mananayaka, A.K.; Chung, S.S. Network Intrusion Detection with Two-Phased Hybrid Ensemble Learning and Automatic Feature Selection. *IEEE Access* **2023**, *11*, 45154-45167. [CrossRef]
18. Zhou, X.; Liang, W.; Li, W.; Yan, K.; Shimizu, S.; Wang, K.I.K. Hierarchical Adversarial Attacks Against Graph-Neural-Network-Based IoT Network Intrusion Detection System. *IEEE Internet of Things Journal* **2022**, *9*, 9310-9319. [CrossRef]
19. Kumar, V.; Sinha, D. A robust intelligent zero-day cyber-attack detection technique. *Complex & Intelligent Systems* **2021**, *7*, 2211-2234. [CrossRef]
20. Sarhan, M.; Layeghy, S.; Gallagher, M.; Portmann, M. From zero-shot machine learning to zero-day attack detection. *International Journal of Information Security* **2023**, *22*, 947-959. [CrossRef]
21. Sheng, C.; Yao, Y.; Li, W.; Yang, W.; Liu, Y. Unknown Attack Traffic Classification in SCADA Network Using Heuristic Clustering Technique. *IEEE Transactions on Network and Service Management* **2023**, 1-1. [CrossRef]
22. Hairab, B.I.; Elsayed, M.S.; Jurcut, A.D.; Azer, M.A. Anomaly Detection Based on CNN and Regularization Techniques Against Zero-Day Attacks in IoT Networks. *IEEE Access* **2022**, *10*, 98427-98440. [CrossRef]

23. Araujo-Filho, P.F.d.; Naili, M.; Kaddoum, G.; Fapi, E.T.; Zhu, Z. Unsupervised GAN-Based Intrusion Detection System Using Temporal Convolutional Networks and Self-Attention. *IEEE Transactions on Network and Service Management* **2023**, 1-1. [CrossRef]
24. Verkerken, M.; D'hooge, L.; Sudyana, D.; Lin, Y.D.; Wauters, T.; Volckaert, B.; Turck, F.D. A Novel Multi-Stage Approach for Hierarchical Intrusion Detection. *IEEE Transactions on Network and Service Management* **2023**, 1-1. [CrossRef]
25. Sohi, S.M.; Seifert, J.-P.; Ganji, F. RNNIDS: Enhancing network intrusion detection systems through deep learning. *Computers & Security* **2021**, *102*, 102151. [CrossRef]
26. Moustafa, N.; Keshk, M.; Choo, K.-K.R.; Lynar, T.; Camtepe, S.; Whitty, M. DAD: A Distributed Anomaly Detection system using ensemble one-class statistical learning in edge networks. *Future Generation Computer Systems* **2021**, *118*, 240-251. [CrossRef]
27. Debicha, I.; Bauwens, R.; Debatty, T.; Dricot, J.-M.; Kenaza, T.; Mees, W. TAD: Transfer learning-based multi-adversarial detection of evasion attacks against network intrusion detection systems. *Future Generation Computer Systems* **2023**, *138*, 185-197. [CrossRef]
28. Dina, A.S.; Manivannan, D. Intrusion detection based on Machine Learning techniques in computer networks. *Internet of Things* **2021**, *16*, 100462. [CrossRef]
29. Lai, Y.C.; Sudyana, D.; Lin, Y.D.; Verkerken, M.; D'hooge, L.; Wauters, T.; Volckaert, B.; Turck, F.D. Task Assignment and Capacity Allocation for ML-Based Intrusion Detection as a Service in a Multi-Tier Architecture. *IEEE Transactions on Network and Service Management* **2023**, *20*, 672-683. [CrossRef]
30. Sabeel, U.; Heydari, S.S.; El-Khatib, K.; Elgazzar, K. Unknown, Atypical and Polymorphic Network Intrusion Detection: A Systematic Survey. *IEEE Transactions on Network and Service Management* **2023**, 1-1. [CrossRef]
31. Rani, S.V.J.; Ioannou, I.; Nagaradjane, P.; Christophorou, C.; Vassiliou, V.; Yarramsetti, H.; Shridhar, S.; Balaji, L.M.; Pitsillides, A. A Novel Deep Hierarchical Machine Learning Approach for Identification of Known and Unknown Multiple Security Attacks in a D2D Communications Network. *IEEE Access* **2023**, 1-1. [CrossRef]
32. Lu, C.; Wang, X.; Yang, A.; Liu, Y.; Dong, Z. A Few-shot Based Model-Agnostic Meta-Learning for Intrusion Detection in Security of Internet of Things. *IEEE Internet of Things Journal* **2023**, 1-1. [CrossRef]
33. Shin, G.Y.; Kim, D.W.; Han, M.M. Data Discretization and Decision Boundary Data Point Analysis for Unknown Attack Detection. *IEEE Access* **2022**, *10*, 114008-114015. [CrossRef]
34. Lan, J.; Liu, X.; Li, B.; Zhao, J. A novel hierarchical attention-based triplet network with unsupervised domain adaptation for network intrusion detection. *Applied Intelligence* **2023**, *53*, 11705-11726. [CrossRef]
35. Zavrak, S.; İskefiyeli, M. Anomaly-Based Intrusion Detection From Network Flow Features Using Variational Autoencoder. *IEEE Access* **2020**, *8*, 108346-108358. [CrossRef]
36. Vu, L.; Nguyen, Q.U.; Nguyen, D.N.; Hoang, D.T.; Dutkiewicz, E. Deep Generative Learning Models for Cloud Intrusion Detection Systems. *IEEE Transactions on Cybernetics* **2023**, *53*, 565-577. [CrossRef]
37. Long, C.; Xiao, J.; Wei, J.; Zhao, J.; Wan, W.; Du, G. Autoencoder ensembles for network intrusion detection. In Proceedings of the 2022 24th International Conference on Advanced Communication Technology (ICACT), 13-16 Feb. 2022, 2022; pp. 323-333. [CrossRef]
38. Yang, J.; Chen, X.; Chen, S.; Jiang, X.; Tan, X. Conditional Variational Auto-Encoder and Extreme Value Theory Aided Two-Stage Learning Approach for Intelligent Fine-Grained Known/Unknown Intrusion Detection. *IEEE Transactions on Information Forensics and Security* **2021**, *16*, 3538-3553. [CrossRef]
39. Abdalgawad, N.; Sajun, A.; Kaddoura, Y.; Zualkernan, I.A.; Aloul, F. Generative Deep Learning to Detect Cyberattacks for the IoT-23 Dataset. *IEEE Access* **2022**, *10*, 6430-6441. [CrossRef]
40. Jin, D.; Chen, S.; He, H.; Jiang, X.; Cheng, S.; Yang, J. Federated Incremental Learning based Evolvable Intrusion Detection System for Zero-Day Attacks. *IEEE Network* **2023**, *37*, 125-132. [CrossRef]
41. Yang, L.; Song, Y.; Gao, S.; Hu, A.; Xiao, B. Griffin: Real-Time Network Intrusion Detection System via Ensemble of Autoencoder in SDN. *IEEE Transactions on Network and Service Management* **2022**, *19*, 2269-2281. [CrossRef]
42. Zahoora, U.; Rajarajan, M.; Pan, Z.; Khan, A. Zero-day Ransomware Attack Detection using Deep Contractive Autoencoder and Voting based Ensemble Classifier. *Applied Intelligence* **2022**, *52*, 13941-13960. [CrossRef]
43. Boppana, T.K.; Bagade, P. GAN-AE: An unsupervised intrusion detection system for MQTT networks. *Engineering Applications of Artificial Intelligence* **2023**, *119*, 105805. [CrossRef]
44. Kim, C.; Chang, S.Y.; Kim, J.; Lee, D.; Kim, J. Automated, Reliable Zero-day Malware Detection based on Autoencoding Architecture. *IEEE Transactions on Network and Service Management* **2023**, 1-1. [CrossRef]
45. Li, R.; Li, Q.; Zhou, J.; Jiang, Y. ADRIoT: An Edge-Assisted Anomaly Detection Framework Against IoT-Based Network Attacks. *IEEE Internet of Things Journal* **2022**, *9*, 10576-10587. [CrossRef]
46. Li, Z.; Chen, S.; Dai, H.; Xu, D.; Chu, C.K.; Xiao, B. Abnormal Traffic Detection: Traffic Feature Extraction and DAE-GAN With Efficient Data Augmentation. *IEEE Transactions on Reliability* **2023**, *72*, 498-510. [CrossRef]

47. Sharafaldin, I.; Lashkari, A.H.; Ghorbani, A.A. Toward generating a new intrusion detection dataset and intrusion traffic characterization. *ICISSp* **2018**, *1*, 108-116. [CrossRef]
48. Xu, C.; Shen, J.; Du, X. A Method of Few-Shot Network Intrusion Detection Based on Meta-Learning Framework. *IEEE Transactions on Information Forensics and Security* **2020**, *15*, 3540-3552. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.