Enhancing Intrusion Detection using Deep Learning and An Improved Conditional Variational AutoEncoder (ICVAE)

S.Gnanamurthy[1], Santhosh Kumar Chenniappanadar[2]
[1]Department of Computer Science and Engineering, Kuppan Engineering College, Kuppam-517425,
Andra Pradesh, India
[2]Department of Information Technology, Sona College of Technology, Salem-636005, Tamil Nadu, India

Article Info

Article history:

Received Jun 11, 2022
Revised Jul 12, 2022
Accepted Aug 18, 2022

Keywords:

Deep learning
Improved Conditional Variational AutoEncoder (ICVAE)
Intrusion detection
Machine learning

ABSTRACT

In the age of the internet and its vast user base, numerous attacks are launched every day. The requirement for information network security has grown significantly over the past few years due to the enormous expansion of Internet applications. An intrusion detection system is intended to be able to adjust to the constantly shifting threat landscape as the main defense of network infrastructure. A particular branch of machine learning called "deep learning" uses structures resembling neurons to learn new information. By making enormous strides in a variety of fields, including speech processing, computer vision, and natural language processing, to mention a few, deep learning has fundamentally altered the means by which we approach learning tasks. Effective outcomes in intrusion detection systems are demonstrated by machine learning techniques. We present ICVAE-DNN, a novel intrusion detection model that combines an Improved Conditional Variational AutoEncoder (ICVAE) with a deep neural network (DNN). An effective model was trained by optimizing unsupervised SAE. According to the experimental findings, the suggested model outperforms conventional approaches in terms of total rate of detection and low percentage of false positives. The KDDCup99, NSL-KDD, and UNSW-NB15 datasets were used to evaluate the suggested model. Using the UNSW-NB15 dataset, the model achieved a 99% accuracy rate and a 97.99% detection rate.

Corresponding Author:

S.Gnanamurthy,
Department of Computer Science and Engineering,
Kuppan Engineering College, Andra Pradesh, India.
Email: gnanamurthyspec@gmail.com

1. INTRODUCTION

A Hundreds of millions of devices have connections to the Internet of Things (IoT), which has had a boom-like development in recent years due to the explosive growth of cloud computing, IoT, 5G communication, and artificial intelligence technologies [1]. Many consumers' personal information is connected to the Internet via IoT devices, which are often used for data collecting and information transmission. However, cyberattacks on the Internet of Things are on the rise, and IoT systems have become a prime target for cybercriminals due to the fact that many IoT nodes gather and store vast volumes of user personal data. Because current IoT devices have limits with regard to information transfer, performance, and power consumption, these devices have extremely high network vulnerability to hacker attacks [2]. Therefore, it appears that it is crucial to provide efficient and effective strategies to harness the ever-increasing intensity and variety of cybersecurity threats. All malicious software components, including ransomware, Trojan horses, worms, spyware, and computer viruses, are included in the Intrusion detection system (IDS) [3].

In recent years, a growing number of researchers have proposed novel techniques for intrusion detection, such as collaborative shallow learning techniques, deep learning techniques, and anomaly detection techniques. One of the most important parts of the security architecture that can prevent cyberattacks from different kinds of attackers is intrusion detection systems (IDS) [4]. The security literature has offered a variety of IDS techniques. Intrusion detection systems can be either host-based or network-based, depending on the type of invasive activity. Network-based intrusion detection systems (NIDS) are IDS that are used to monitor network activity [5]. It keeps an eye out for unusual or suspicious activity in network traffic and sends out alerts when it is found. AutoEncoder, deep neural network (DNN), deep belief network (DBN), and recurrent neural network (RNN) are examples of deep learning techniques that can automatically extract features and carry out categorization [6].

High-level latent features will be automatically extracted by deep learning without the need for human participation. Many artificial intelligence domains, such as speech processing, computer vision, natural language processing, and others, heavily rely on deep learning. Additionally, network security detection has been done using deep learning. Prior studies have demonstrated that data augmentation can act as a regulator to assist avoid overfitting and enhance performance when data distribution is unbalanced. Variational autoencoders (VAE) are a conventional technique that is frequently utilized as data augmentation when dealing with little data and data restrictions [7]. In addition to being much simpler to train using a gradient-based method, VAE is also better at capturing the unique distribution features of the original sample and creating synthetic samples that look exactly like it. It also has superior convergence properties. Regretfully, VAE is unable to provide particular samples depending on labels. To address these issues, a researcher created the conditional variational auto-encoder (CVAE), an expansion of VAE. In order to rebuild input features, CVAE, a new deep generative network, uses output vectors. It also reduces the network's size and generates fresh attack samples in the selected category [8].

A category of machine learning called deep learning has demonstrated exceptional results across a range of application domains, especially when working with huge datasets. Deep learning techniques operate in an end-to-end manner, producing outcomes after automatically learning from raw data, which is highly useful. Numerous deep learning techniques have demonstrated success with NIDS. Through supervised learning, we employ ICVAE to discover the distribution of intricate traffic and classes [9]. The weight of the DNN hidden layers is initialized using the system parameters of the ICVAE encoder. In order to manage the data used for training and increase the range of training samples, latent variables using Gaussian noise and designated declares are inserted into the constructed ICVAE decoder for creating particular fresh attack information. This improves the detection rate of unknown and minority attacks. The proposed model is evaluated using the UNSW-NB15 and NSL-KDD datasets. The proposed model outperforms the well-known classification techniques in terms of overall precision, false positive rate, and recall. Additionally, it increases the detection rate of minority and unknown attacks [10].

2. LITERATURE REVIEW

[11] proposed that Improving the classification effectiveness of intrusion detection by using improved conditional variational autoencoder and deep neural network. Systems for detecting intrusions are crucial for averting security risks and shielding networks from assaults. This experiment suggests a new intrusion detection model called ICVAE-DNN, which combines a deep neural network (DNN) and an enhanced Conditional Variational AutoEncoder (ICVAE). Potential sparse connections across network information features and categories are learned and investigated using ICVAE. In order to balance the initial training data and boost the diversity of the practice samples, the designed ICVAE decoder creates fresh attack samples based on the designated intrusion categories. This increases the rate of detection of uneven attacks. Furthermore, the experimental findings demonstrate that the suggested ICVAE-DNN outperforms the six well-known classification algorithms—KNN, MultinomialNB, RF, SVM, DNN, and DBN—in terms of detection rates for minority attacks (such as U2R, R2L, shellcode, and worms).

[12] introduced system for Convolutional Neural Networks with LSTM for Intrusion Detection. Deep learning techniques can identify network breaches with high accuracy and a small false alarm rate. This research presents a novel method for enhanced intrusion detection that combines an integrated method of Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM). This study has demonstrated that a combination of long short term memory (LSTM) and convolutional neural networks (CNN) is a highly successful method for network intrusion detection. Using a conventional NSL KDD dataset, unprecedented accuracy was attained without the need for hyperparameter adjustment. This study demonstrates the great potential and efficacy of deep learning techniques for intrusion prevention and anomaly detection.

[13] designed A hybrid intrusion detection system based on sparse autoencoder and deep neural network. Effective outcomes in intrusion detection systems are demonstrated by machine learning techniques. We suggested a hybrid intrusion detection approach that consists of two stages. It is true that

smoothed l1 regularization can learn sparse feature representations. In the second phase, attacks were predicted and categorized using the Deep Neural Network (DNN). On UNSW-NB15, the model's maximum overall accuracy, DR, and F1score were 99.98%, 99.98%, and 99.98%, respectively.

[14] proposed Network intrusion detection system based on conditional variational laplace autoencoder. Synthesizing data in a statistical way that resembles the original attack event-related data is crucial. Thus, in this paper, we present a novel synthesis task paradigm based on Deep Neural Network and Variational Laplace AutoEncoder (VLAE), and the researcher uses the paradigm to create a new intrusion detection model. Attacks can be created and attack samples can be diversified through the use of adversarial learning techniques.

[15] introduced Towards secure intrusion detection systems using deep learning techniques: Comprehensive analysis and review. In light of the ever-increasing harm that security assaults cause to computer systems and networks, a variety of AI-based methods have been employed to present flexible IDS strategies. Deep learning methods have recently become more popular in the field of intrusion detection, and a number of IDS strategies utilizing different deep neural networks are presented in the literature to address security risks and privacy issues. Unbalanced datasets are unavoidable because of the impossibility of collecting an equal number of records for every attack class, which causes issues like over-fitting. The generated data, though, might not be accurate.

[16] Conditional Variational Autoencoder with Inverse Normalization Transformation on Synthetic Data Augmentation in Software Effort Estimation. Making precise estimations with minimal SEE data can be difficult due to the small amount of information in the data. An inversion normalizing transformation (INT) and a conditional variational autoencoder (CVAE) were examined in this study as techniques for creating synthetic data and augmenting existing data. Eleven software engineering data sets, including ISBSG and the PROMISE repository, will be used as case studies. The findings demonstrate that our approach can offer data augmentation strategies for transforming variance data into synthetic data with an identical distribution to the original data.

3. PROPOSED METHODOLOGY

An artificial neural network called an Auto-Encoder (AE) learns how to efficiently compress and encode data before figuring out how to reconstruct it as closely as possible from the original information provided. AE is typically trained without supervision consisting of two components: an encoder and a decoder. While the encoder is frequently employed to decrease dimensionality, the decoder represents a decoding technique that can be used to either generate new data or denoise raw data. The encoder in VAE typically uses a probability distribution to convert input information into a lower dimension. Additionally, the pattern of distribution is regulated, and VAE trains its variational inference during training, in order for the latent space of Z to have an important abstract attribute that can recreate the observed data.

The Variational AutoEncoder (VAE) is a significant generation model made up of the encoder network $q(Z/L)$ and the decoder network $p(Z/L)$. The gradient descent method can be used to train VAE, which is capable of learning approximation inference. Data X is mapped into a continuous latent variable Z by the encoder network with parameters f, which learns an effective compression of the data into this lower-dimensional space. The latent variable is used to create data by the decoder network with parameters q, mapping Z to a reconstructed data ˆX. Here, we build the encoder and decoder with parameters q and f, respectively, using deep neural networks. The new paper's suggested design is an extension of ongoing research that was previously published as the Hybrid Anomaly Detection Model (HADM) [1]. As seen previously, the architecture consists of a feature selection/extraction algorithm and a random forest classifier. The classifier algorithm (Random Forest) uses the best features extracted from the incoming packets by the feature selection (SVMonline)/extraction (CVAE) method to categorize data into three groups (normal, unknown, and attack categories). The chosen features are run through an optimized random forest algorithm for performance evaluation. The parameters and internal structure of the algorithm are described below.

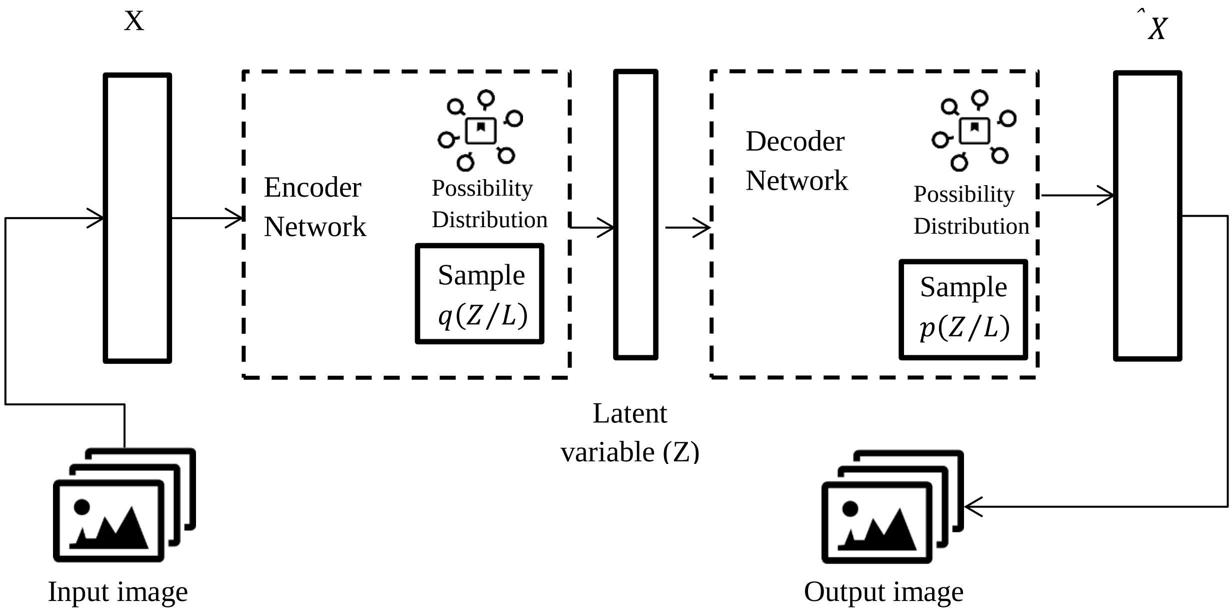

Figure 1. Architecture of Conditional Variational AutoEncoder (CVAE)

As illustrated in Figure 1, CVAE is represented by subjecting the encoder and decoder to class Y. Two variables, X and Y, are now required for the encoder $q(Z/L)$, and two variables, Z and Y, are now required for the decoder $p(Z/L)$. Improved Conditional Variational AutoEncoder is an enhanced variant of CVAE that produces better outcomes. The architecture of ICVAE has an encoding network $q_{\emptyset}(Z/L)$ and the decoder network $p_{\theta}(Z/L)$makes up ICVAE. While the encoded signal is missing the class label Y, the decoder uses it as an additional input, making the decoding probability distribution dependent on the latent value Z and class label Y. New attack examples of the designated class are produced during decoding by connecting a latent variable Z with the name Y, which are then sent into the decoder. The following figure shows the suggested system architecture.

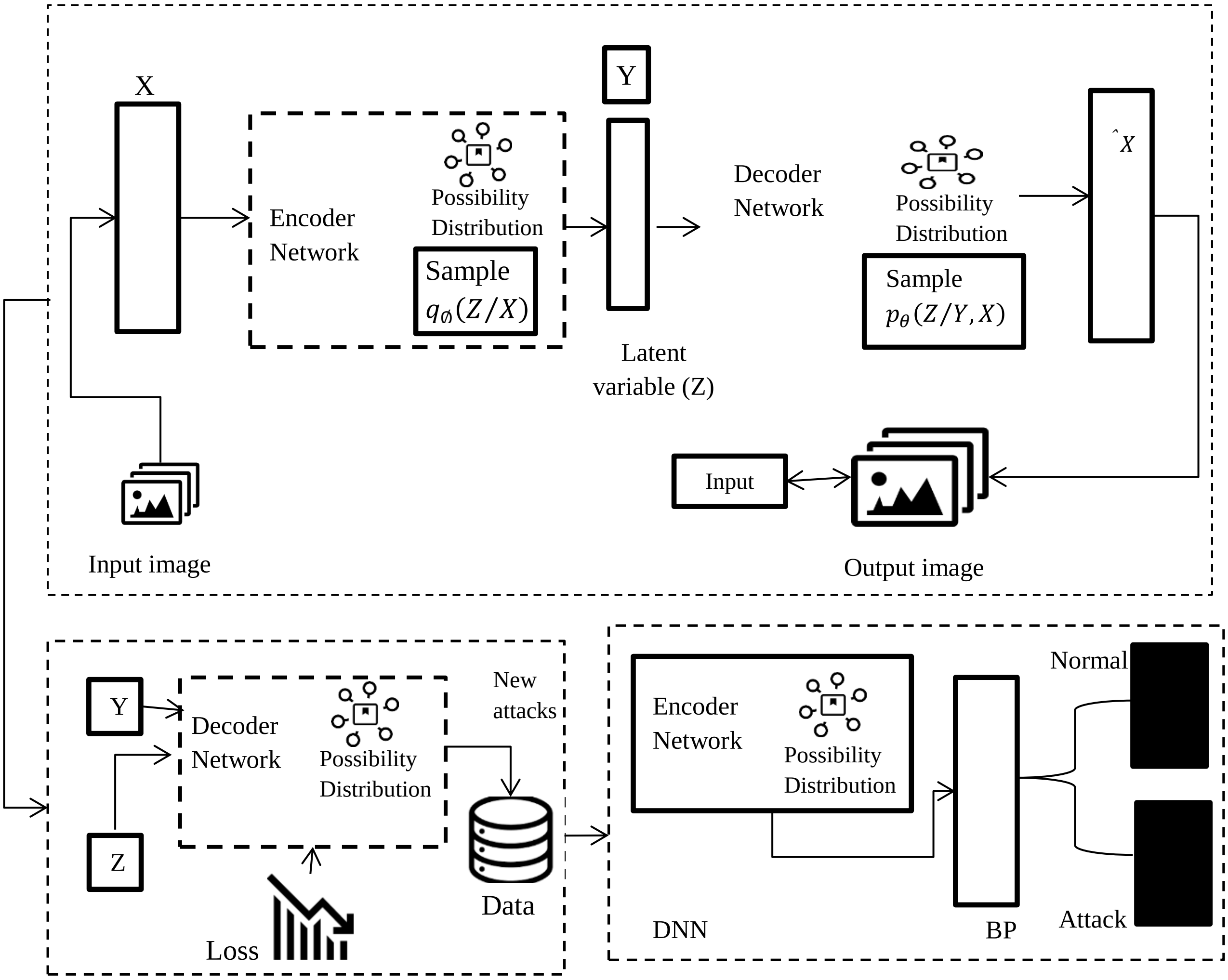

Figure 2. Proposed ICVAE Architecture for Intrusion Detection

Figure 2 displays the proposed ICVAE-DNN framework. ICVAE-DNN is divided into three primary stages: (1) ICVAE training; (2) attack generation; and (3) attack detection. The first stage, in which the ICVAE is trained using the training samples, and each training data sample's reconstruction loss is recorded based on the attack class. The ICVAE decoder generates fresh attack samples based on pre-established classes in the second step. If the criteria for reconstruction loss are met, each newly created attack sample is combined with the initial training data set. The final step involves initializing the amount of weight of the DNN layers that are concealed using the ICVAE decoder, training the DNN classifier using the combined training data set, and using the learned DNN classifier to identify threats on the test data set.

$$\log p(x/y) = \log p(x/y) - \log p\left(X\middle|y\right)\pi\left(X\middle|y\right) \qquad (1)$$

CVAE encoder and decoder conditional probability distributions are associated with class label Y. Using the encoder network component of CVAE, we improve the CVAE architecture by including class label Y only in the decoder network for configuring the network settings for the DNN. In Figure 3, the ICVAE architecture is displayed. The researched system uses a variety of auto-encoder types during the intrusion detection process; this subsection concentrates on the auto-encoder approaches. As seen in Figure 2, the encoder and decoder make up the framework of ICVAE. We employ a Gaussian distribution with multiple variables to represent the $q_\emptyset(Z/X)$. A multivariate Bernoulli distribution is used to fit the decoder $p_\theta(Z/X,Y)$.

$$\beta\,(q,f;\ X,Y) = E\ \log[p_\theta(Z/X\,,Y)]\,;\mathrm{DKL}[q(Z/X) * p(Z/Y)] \qquad (2)$$

Two components make up L (q, f; X,Y) in Equation (2): a log reconstructive probability E $\log[\mathrm{p}_{\theta}(\mathrm{Z/X,Y})]$. A KL split $\mathrm{DKL}[\mathrm{q(Z/X)} * \mathrm{p(Z/Y)}]$ is also present. The conditional probability distribution $\mathrm{p(Z/X,Y)}$is used in the first term to reconstruct X, and the KL divergence metric is used in the second term to describe the encoder distribution $\mathrm{q(Z/X)}$, which approximates the prior distribution $\mathrm{p(Z/Y)}$. The projected probability, or reconstructed data, is the decoder network's output. Reconstruction loss and KL loss make up the ICVAE loss.

$$\mathrm{KL} = \alpha_y^x \left(1 - \alpha_y^x\right)^3 \qquad (3)$$

The decoder output Equation (3), which reads az,y = Decoder(z, y) = ˆ x, is a parameter of the Bernoulli distribution. Next, the log-likelihood that is negative is:

$$-\log \mathrm{L} \;=\; -[\mathrm{x} \cdot \log(\hat{\ }\mathrm{x}) +\; (1 - \mathrm{x})_x \log(1 -\, \hat{\ }\mathrm{x})] \qquad (4)$$

The ICVAE might have a KL-vanishing issue since the KL loss employs the variational estimation method to use the deep neural network $q_{\emptyset}(Z/X)$ to estimate the distribution $p_{\theta}(Z/X, Y)$. We utilize the largest restoration loss for every session as the evaluation criterion after calculating the loss in the reconstruction of all samples of training according to the class.

$$max_{L_n} = h \times max\{l_n(x_n, y_n)\}, y_n \epsilon\ to\ n \qquad (5)$$

The n-th class's maximum reconstruction loss, max_{L_n}, is expressed in Equation (5). In the second phase, we describe the multidimensional regular typical distribution N(0, I) for the decoding $p_{\theta}(Z/X, Y)$ as the preceding distribution $p_{\theta}(Z/Y)$. To create a fresh attack sample(x, y), we may sample an unknown variable z from N(0, I) under a given label ˆ y and input it to the trained decoder. We evaluate the maximum lost max_{L_n} of the associated class j with the reconstruction loss l(x, ˆy). The newly created sample is added to the previous training set R if l(ˆ x, ˆ y) < maxLj; if not, it is discarded.

The last phase, DNN is what we use to identify assaults. A six-layer feedback deep neural network is called a DNN.All of DNN's hidden layers have the activation function ReLU6 [50], and the output layer's activation function is softmax. It employs a softmax algorithm for network traffic categorization and many convolutional layers to capture local aspects of the traffic data. This study showed that a higher accuracy and a reduced false-positive rate could be obtained by utilizing their DNN model. The dataset contains a number of features with large variations between minimum and maximum values. A form of Feed Forward Neural Network (FFN) with a maximum of three layers—one input layer, one output layer, and multiple hidden layers—DNN is a variation of the Multilayer Perceptron (MLP). The DNN hidden layers' network structure is identical to the ICVAE encoder's. Since the ICVAE encoder is capable of autonomously extracting high-level features, the mean weight of the learned ICVAE encoder is employed to establish the weight of the DNN hidden layers. The DNN classifier is then fine-tuned using the combined training data set, and the Adam method is used to optimize the DNN classifier.

4. RESULTS AND DISCUSSION

The performance metrics listed below are often used to illustrate the efficacy and performance of IDS. Accuracy, False Positive (FP), TP (True Positive), precision, FPR (False Positive Rate), TNR (True Negative Rate), and so forth are among the metrics. An assault record that is mistakenly categorized as an ordinary record is denoted by FN.

$$DR = \frac{True\ positive}{true\ Positive - False\ Negative'} \qquad (6)$$

$$Accuracy = \frac{TN + TP}{TN + TP + FP + FN} \qquad (7)$$

$$FRP = \frac{False\ Positive}{False\ Positive + True\ Negative} \qquad (8)$$

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \qquad (9)$$

Equations (6), (7), (8), and (9) above define accuracy, DR, precision, and FPR. The F1-score uses the harmonic mean to calculate recall and precision. The F1-score is more suited for assessing the ability to identify the performance of unbalanced samples than accuracy.

$$F1\, score = \frac{2PR}{2PR + TN + FN} \qquad (10)$$

The F1 score is displayed in equation (10). Since the NSD-KDD dataset removes duplicate and redundant records from the KDD Cup 99 dataset, it is more appropriate for evaluating the efficacy of intrusion detection systems. KDDTrain+_20Percent serves as the training set for the experiments, whereas KDDTest+ and KDDTest-21 serve as the test sets. Table 1 shows the total number of items for each category in the NSL-KDD dataset. Table 1 shows that around half of the unidentified assaults in the experimental dataset were absent from the training dataset.

Table 1. The NSL-KDD dataset's category distribution

Category	**Training dataset**	
	Attack	**Count**
KDD Train(Probe)	Nmap	710
	Ipsweep	320
	portsweep	578
KDD Test(Probe)	Ipsweep	147
	nmap	165
	saint	321
KDD Train(DoS)	Neptune	7654
	Back	453
	Pod	234
KDD Test(DoS)	Teardrop	645
	Apache2	243
	upstrom	645
KDD Train(R2L)	Phf	2
	Spy	1
	Imap	181
KDD Test(R2L)	Named	2
	worm	3
	xlock	2

We completed our experiments to assess the suggested model's performance. Three distinct datasets from the UNSW-NB15 and NSL-KDD databases were used. We contrasted the outcomes of the suggested model with those of other popular detection techniques. The suggested solution was put into practice on a ThinkStation running 64-bit Windows 10 with an Intel E5-2620 CPU and 64 GB of RAM in the TensorFlow environment. The DNN classifier's generalization performance can be enhanced with the right number of hidden layers. To determine the ideal model hyper parameters that produce the most accurate predictions, grid search and by threefold cross-validation trials are conducted. Every set of hyper parameters in a search parameter space is traversed by the grid search. Three-fold cross-validation is performed to assess each set of hyper parameters.

The ideal parameters are those that obtain the highest cross-validation score. On the NSL-KDD and UNSW-NB15 data sets, the suggested model's ideal network architectures are 142-60-90-20-10-3 and 196-120-70-40-21-50, correspondingly. ReLU6 is the triggering unit of every hidden layer in the ICVAE decoding devices, while Sigmoid is the activated value of the output layer. ReLU6 is the activation function of every hidden layer in the DNN, while Softmax is the activation unit of the output layer.

We conducted performance comparisons using the classification approach and the oversampling method. Furthermore, the ICVAE-DNN's detection performance is contrasted with that of other cutting-edge models.

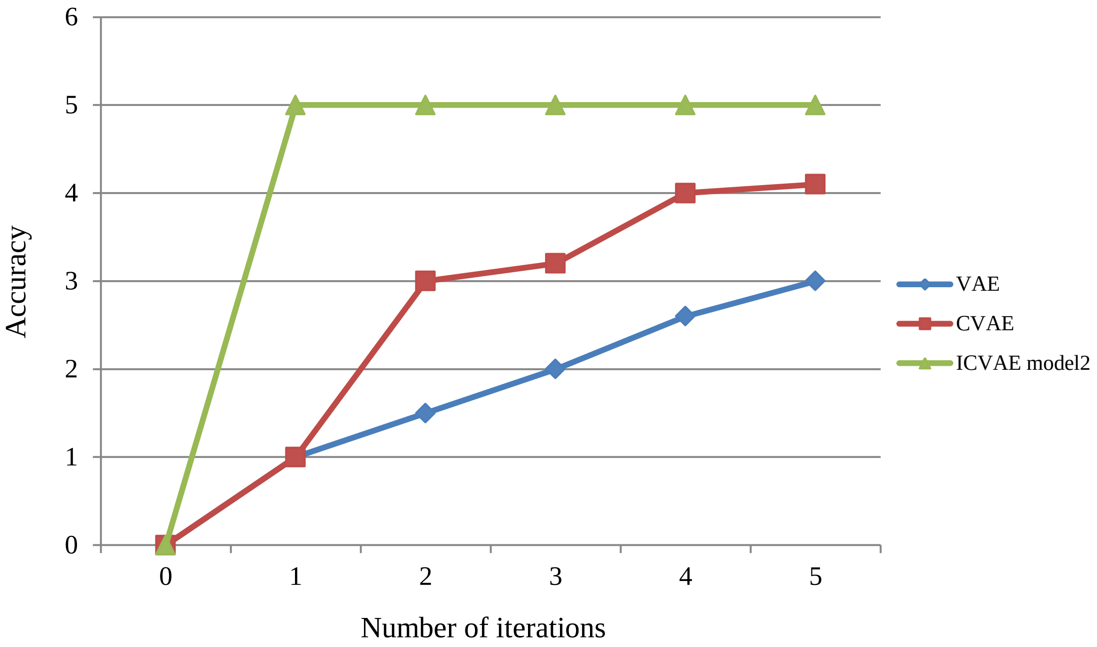

Figure 3. Accuracy analysis

Figure 3 above displays the training conversations on different model dataset. The training samples using the NSL-KDD as well as UNSW-NB15 datasets were unbalanced, as seen in Tables. On the NSL-KDD dataset, the U2R and R2L contain minority records, while on the UNSW-NB15 dataset, worms along with shellcode have minority records. To equalize the training information, we employ the ICVAE decoder to produce multiple entries of the designated category; the outcomes are displayed in the tables below.

Table 2. Sample production in NSL-KDD training set

Category	No. of Original data	No. of new data	Total
Normal	23143	0	23143
Probe	2435	56	2495
DoS	232	342	574
R2L	205	13456	13661

Table 3. sample production in UNSW-NB15 training set

Category	No. of Original data	No. of new data	Total
Normal	52,000	0	52,000
Exploits	33,676	18,324	52,000
DoS	20,263	31,737	52,000
Backdoor	12,875	39,125	52,000
Worms	1,346	50,654	52,000
Shellcode	0	52,000	52,000

In the categories of typical, investigation, analysis, backdoor, shellcode, and worms, ICVAE-DNN achieves the greatest detection rate. In the DoS class, MultinomialNB has the best detection rate (63.23%), suggesting that the characteristics of DoS attacks follow a polynomial distribution. But when it came to minor and significant threats like analysis, backdoors, shellcode, and worms, ICVAE-DNN had the best detection rates. In the worms class, for instance, ICVAE-DNN's detection rate was 77.71% and 85.32% greater than KNN's and SVM's, respectively. Therefore, the majority of analysis and backdoor assaults are incorrectly classified as exploits by classifiers. Table 4 the percentage difference in detection performance between several classification techniques on the UNSW-NB15 dataset.

Table 4. Detection performance on the UNSW-NB15 dataset

Class	KNN	RF	DNN	DBN	ICVAE-DNN
Normal	73.5	76.3	74.87	76.25	91.3

Exploits	67.4	73.6	87.23	87.54	74.6
DoS	46.09	10.3	70.91	8.66	7.06
Backdoor	3.89	56.76	58.33	79.53	20.21
Worms	11.14	4.23	0	59.29	78.4
Shellcode	15.06	60.34	23	0.53	89.61
Accuracy	74.87	87.12	32.54	87.4	41.93
F1-Score	67.3	50.52	8.66	38.72	92.37
FPR	23.9	0.04	82.37	33.87	24.76
Precision	56.76	73.6	46.09	62.7	25.2

5. CONCLUSION

In this article, we propose ICVAE-DNN, an innovative intrusion detection method that combines DNN and ICVAE. Convolutional neural networks, autoencoders, and other deep neural network architectures were used to propose, implement, and train intrusion detection models. IDS solutions use a variety of AI techniques to improve the efficiency of the IDS schemes and boost their efficacy against the latest security threats. One of the contexts that researchers have concentrated on to enhance the function selection/extraction and classification stages of the IDS techniques is deep learning. We use both the original and synthesized datasets to train a CNN classifier. VLAE has more expressive capacity than CVAE since it employs a full-covariance Gaussian as the posterior distribution. In order to achieve this, it first presents background information, illustrating the different kinds of deep neural networks used in the examined IDS techniques. Additionally, this section describes the primary datasets that were used to assess and examine the IDS methods. Using the NSL-KDD (KDDTest+), NSL-KDD (KDDTest-21), and UNSW-NB15 datasets, the classification performance of ICVAE-DNN is assessed and contrasted with six popular classifiers. The suggested ICVAE-DNN achieves higher accuracy, detection rate, and false positive rate when compared to numerous other classifiers. These tests demonstrate that ICVAE-DNN is more suited for identifying network intrusions, particularly those that are unknown or minority attacks. With an accuracy rating of 88.9%, this model outperforms other classifiers.

REFERENCES

[1] Naseer, S., Saleem, Y., Khalid, S., Bashir, M. K., Han, J., Iqbal, M. M., & Han, K. (2018). Enhanced network anomaly detection based on deep neural networks. *IEEE access*, *6*, 48231-48246.

[2] Xu, X., Li, J., Yang, Y., & Shen, F. (2020). Toward effective intrusion detection using log-cosh conditional variational autoencoder. *IEEE Internet of Things Journal*, *8*(8), 6187-6196.

[3] Liu, C., Antypenko, R., Sushko, I., & Zakharchenko, O. (2022). Intrusion detection system after data augmentation schemes based on the VAE and CVAE. *IEEE Transactions on Reliability*, *71*(2), 1000-1010.

[4] Monshizadeh, M., Khatri, V., Gamdou, M., Kantola, R., & Yan, Z. (2021). Improving data generalization with variational autoencoders for network traffic anomaly detection. *IEEE Access*, *9*, 56893-56907.

[5] Ali, M. H., Al Mohammed, B. A. D., Ismail, A., & Zolkipli, M. F. (2018). A new intrusion detection system based on fast learning network and particle swarm optimization. *IEEE Access*, *6*, 20255-20261.

[6] Yang, Y., Zheng, K., Wu, B., Yang, Y., & Wang, X. (2020). Network intrusion detection based on supervised adversarial variational auto-encoder with regularization. *IEEE access*, *8*, 42169-42184.

[7] Sabeel, U., Heydari, S. S., Elgazzar, K., & El-Khatib, K. (2021, December). CVAE-AN: Atypical attack flow detection using incremental adversarial learning. In *2021 IEEE Global Communications Conference (GLOBECOM)* (pp. 1-6). IEEE.

[8] Yu, Q., Kavitha, M. S., & Kurita, T. (2021). Extensive framework based on novel convolutional and variational autoencoder based on maximization of mutual information for anomaly detection. *Neural Computing and Applications*, *33*(20), 13785-13807.

[9] Osada, G., Omote, K., & Nishide, T. (2017). Network intrusion detection based on semi-supervised variational auto-encoder. In *Computer Security–ESORICS 2017: 22nd European Symposium on Research in Computer Security, Oslo, Norway, September 11-15, 2017, Proceedings, Part II 22* (pp. 344-361). Springer International Publishing.

[10] Gurung, S., Ghose, M. K., & Subedi, A. (2019). Deep learning approach on network intrusion detection system using NSL-KDD dataset. *International Journal of Computer Network and Information Security*, *11*(3), 8-14.

[11] Yang, Y., Zheng, K., Wu, C., & Yang, Y. (2019). Improving the classification effectiveness of intrusion detection by using improved conditional variational autoencoder and deep neural network. *Sensors*, *19*(11), 2528.

[12] Ahsan, M., & Nygard, K. E. (2020, March). Convolutional Neural Networks with LSTM for Intrusion Detection. In *CATA* (Vol. 69, pp. 69-79).

[13] Rao, K. N., Rao, K. V., & PVGD, P. R. (2021). A hybrid intrusion detection system based on sparse autoencoder and deep neural network. *Computer Communications*, *180*, 77-88.

[14] Azmin, S., & Islam, A. M. A. A. (2020, December). Network intrusion detection system based on conditional variational laplace autoencoder. In *Proceedings of the 7th International Conference on Networking, Systems and Security* (pp. 82-88).

[15] Lee, S. W., Mohammadi, M., Rashidi, S., Rahmani, A. M., Masdari, M., & Hosseinzadeh, M. (2021). Towards secure intrusion detection systems using deep learning techniques: Comprehensive analysis and review. *Journal of Network and Computer Applications*, *187*, 103111.

[16] Marco, R., Syed Ahmad, S. S., & Ahmad, S. (2022). Conditional Variational Autoencoder with Inverse Normalization Transformation on Synthetic Data Augmentation in Software Effort Estimation. *International Journal of Intelligent Engineering & Systems*, *15*(3).