ISSN 2581 – 3781

AJCSE
ASIAN JOURNAL OF COMPUTER SCIENCE ENGINEERING

Available Online at www.ajcse.info
Asian Journal of Computer Science Engineering 2025;10(4):1-7

RESEARCH ARTICLE

CADAE: Cybercrime Attack Detection using Autoencoders based on Temporal Features

Yerininti Venkata Narayana*, Mooramreddy Sreedevi

Department of Computer Science, Sri Venkateswara University, Tirupati, Andhra Pradesh, India

Received on: 01-10-2025; Revised on: 09-11-2025; Accepted on: 11-12-2025

ABSTRACT

The increase in cybercrime denotes a rising trend of criminal activities conducted through digital means. This flow is fueled by the growing dependence on technology, increased connectivity, and the explosion of online platforms. Addressing this challenge requires enhanced cybersecurity measures to stay ahead of evolving cyber threats. Cybercrime Attack Detection using Autoencoders (CADAE), which relies on temporal features, poses an effective approach in which the unsupervised learning of autoencoders is advantageous in sensitive, complex temporal patterns inherent in cybersecurity data. The CADAE approach used three benchmark datasets: KDDCup99, CICIDS2017, and SIMARGL2022, showcasing admirable performance in discriminating and mitigating cybersecurity threats. The KDDCup99 dataset, a pioneer in this domain, provides a comprehensive set of labeled data, enabling the growth and assessment of intrusion detection systems (IDSs). The CICIDS2017 dataset, designed for the assessment of network IDSs, captures a diverse range of cyber threats, including DoS attacks and malware activity. SIMARGL2022, another significant dataset, focuses on simulated cyber-physical systems, presenting a unique environment for assessing the resilience of critical infrastructure against cyberattacks. The successful utilization of these benchmark datasets underscores their effectiveness in enhancing the capabilities of intrusion detection models.

Key words: Data exfiltration detection, honeypot generation, intrusion detection system, log analysis, malware detection, network anomaly detection, phishing detection, user behavior analysis

INTRODUCTION

The term cybercrime denotes illegal actions conducted through computers, networks, or digital devices that often target digital assets, data, or individuals' personal information, with the intent to cause harm, steal valuable information, or generate illicit profits.[1] Deep learning offers powerful capabilities for detecting and mitigating cybercrime activities by automatically learning complex patterns and features from diverse data sources. Continuous research and development in deep learning algorithms, architectures, and techniques further advance the effectiveness of deep learning-based approaches in cybersecurity.[2] Autoencoders have several potential applications in cybersecurity and combating cybercrimes. It is one of the deep neural architectures consisting of an encoder and a decoder. Their primary purpose is not to engage in malicious activities; they can be used as defensive tools to enhance security and detect anomalies.[3]

Here are some ways autoencoders can be applied in cybersecurity: Intrusion Detection System (IDS), Network Anomaly Detection, Malware Detection, User Behavior Analysis, Phishing Detection, Data Exfiltration Detection, Log Analysis, Honeypot Generation, etc., as shown in Figure 1.

Intrusion Detection

Autoencoders can be employed to create IDSs. They can learn the normal behavior of network traffic and flag any deviations from this behavior as potential intrusions. This helps in identifying known and novel attacks.

Network Anomaly Detection

Autoencoders can analyze network traffic data and identify unusual network behavior, such as distributed denial of service attacks, port scanning,

Address for correspondence:
Yerininti Venkata Narayana
E-mail: naarayanaa808@gmail.com

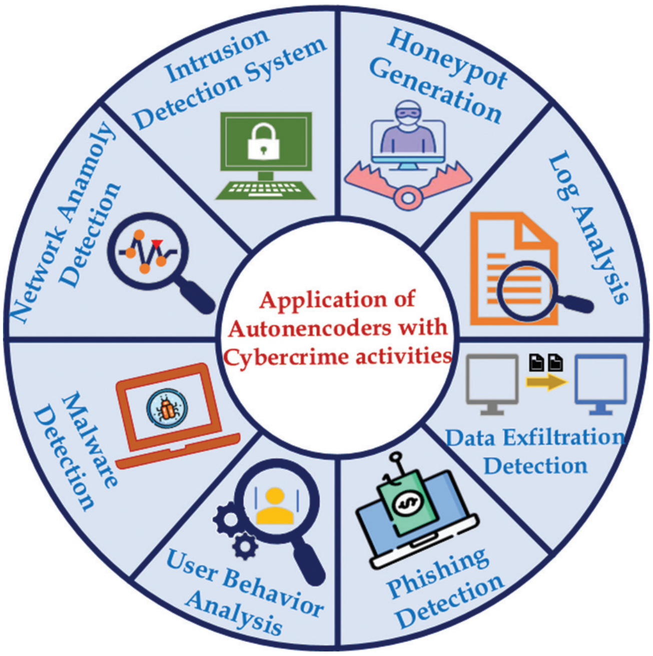
Intrusion Detection System
Honeypot Generation
Log Analysis
Data Exfiltration Detection
Phishing Detection
User Behavior Analysis
Malware Detection
Network Anamoly Detection
Application of Autonencoders with Cybercrime activities

Figure 1: Application of autoencoders

or unusual communication patterns, which are often indicative of cyber threats.

Malware Detection

Autoencoders can be used to detect malware by analyzing the patterns in file structures, system calls, or network communication associated with malicious software. Any deviation from the learned benign patterns can signal the presence of malware.

User Behavior Analysis

Autoencoders can be used to model and analyze user behavior within a network or system. Unusual user activity, such as unauthorized access or privilege escalation, can be detected by comparing it to learned behavior patterns.

Phishing Detection

Autoencoders can assist in identifying phishing attempts by examining email content or URLs. They can recognize patterns and structural elements commonly associated with phishing attacks.

Data Exfiltration Detection

Autoencoders can monitor data leaving a network or system and detect unusual data transfer patterns, which might indicate data exfiltration attempts by malicious actors.

Log Analysis

Autoencoders can be applied to analyze logs generated by various system components. They can help identify irregular log patterns that may suggest a security breach or misconfiguration.

Honeypot Generation

Autoencoders can be used to create realistic honeypots – decoy systems or services designed to attract attackers. By learning and emulating legitimate system behavior, these honeypots can effectively capture and analyze attack attempts.

RELATED WORK

Song *et al.*[4] introduced an autoencoder-based network IDS, a comprehensive study on autoencoders using NSL-KDD, IoTID20, and N-BaIoT benchmark datasets. Using a straightforward autoencoder model, multiple combinations of model structures and latent sizes were evaluated, and the findings highlight the substantial influence of the model's latent size on IDS performance.

Torabi *et al.*[5] presented an efficient autoencoder-based model for detecting anomalies in cloud computing networks using data reconstruction errors to classify anomalies and employed the CIDDS-001 dataset. The presented method outperformed existing approaches, achieving better accuracy, recall, lower false-positive rates, and improved F1-scores.

Xing *et al.*[6] introduced an innovative malware detection model that combines grayscale images of malware with a deep learning autoencoder network. To classify the malware from benign software, it uses reconstruction error and dimensionality reduction, and obtained a high accuracy and a consistent F-score of 96% when tested on Android-side data, outperforming traditional machine learning methods.

Nepal and Joshi[7] focused on detecting insider threat activities and identifying unusual or suspicious behavior using a model that flags instances with high reconstruction errors as anomalies by employing a Gated Recurrent Unit-based Autoencoder. The CERT r4.2 dataset is used for experimentation. The model's performance was assessed at various thresholds, showing strong discrimination with minimal misclassification for both classes, achieving true positive (TP) and true negative rates of 79.81%.

Prabakaran *et al*.[8] introduced an effective phishing detection model that combines variational autoencoders-deep neural networks, and obtained a high accuracy with 97.45% and a fast response time of 1.9 s when tested on approximately 100,000 URLs from publicly available datasets (ISCX-URL-2016 and Kaggle), outperforming other models.

Willems *et al*.[9] created a Network Exfiltration Detection System to spot data exfiltration in ransomware attacks, using aggregated metadata and tested it with real data, and found that aggregation notably enhances the detection of exfiltration, especially in cases such as DNS tunnels, occurring over extended periods.

Farzad and Aaron Gulliver[10] introduced an unsupervised log message anomaly detection model that combines an isolation forest and two deep autoencoder networks. The model achieved high accuracy, with a normal testing accuracy of 99.4%, with BGL, OpenStack, and Thunderbird log datasets, and demonstrated excellent performance in detecting anomalies in log messages.

Siniosoglou *et al*.[11] introduced a novel approach to honeypot systems in modern industrial networks using the Modbus protocol. The NeuralPot honeypot employs two distinct deep neural networks to mimic network Modbus entities, actively confusing potential intruders. The study compares these neural networks and their generated data quantitatively and recommends the GAN architecture due to its closer similarity to real data. The GAN generates 128 values in 0.6969 ms, whereas the autoencoder accomplishes this in 0.4116 ms.

METHODOLOGY

Detecting cybercrime activities using autoencoders is an interesting application of anomaly detection. By leveraging autoencoders, one can train models to recognize typical patterns within data, enabling the identification of deviations from these patterns, which can indicate potential cybercrime or anomalies. Here is a general outline of how you can approach this:

Data Collection and Preprocessing

Gathered a temporal dataset of network traffic, system logs, or any relevant data that captures normal and potentially malicious activities. Preprocessed the data to remove noise, normalize values, and prepared it for feeding into the autoencoder.[12]

Autoencoder Architecture

Defined an autoencoder architecture by adjusting the architecture depending on the complexity of your data. In the encoder, three dense layers were used along with ReLU activation functions, in which the first uses 128 units, the second uses 64 units, and the third uses 32 units. The decoder also uses three dense layers in which the first uses 64 units with ReLU activation function, the second uses 32 units with ReLU activation function, and the third uses n_{features} with linear activation function

Feature Analysis

Since the data are complex, it is necessary to perform feature extraction to represent them in a suitable format for the autoencoder. To understand the underlying structure of the data, the extracted features are analyzed, which involves visualizing the features, clustering them to identify patterns, or using them as input to downstream tasks such as classification or anomaly detection.[13]

Attack Detection

In this, each dataset is separated into a training set containing only normal data and a test set that includes both normal and potentially malicious data. This allowed training the autoencoder on normal patterns and evaluating its performance on both normal and anomalous data. Training the autoencoders using only the normal data from the training set will learn to reconstruct the normal patterns during this phase. After training, the trained autoencoders reconstruct both the normal and anomalous data from the validation/ test set. The distribution of reconstruction errors for normal data is analyzed by setting a threshold to classify instances as normal or anomalous. The instances with reconstruction errors above the threshold are considered potential anomalies.[14]

Performance Evaluation

The cybercrime attack detection using autoencoders (CADAE) approach uses metrics such as Precision, Recall, F1-score, AUROC, and AUPRC to assess the effectiveness of the model.[15]

Figure 2 is the methodology, and below is the algorithm adopted for detecting the cybercrime activities:

```
Algorithm CADAE
Step 1: Encoder:
  • Input: $x$
  • First Dense Layer: 128 units with f(x)
  • Second Dense Layer: 64 units with f(x)
  • Third Dense Layer: 32 units with f(x)
      Here, $f(x) = \max(0, x)$
Mathematically, the encoder can be represented as:
encoded = max(0, max(0, max(0, x * W1 + b1) * W2 + b2) * W3 + b3)
Step 2: Decoder:
  • Input: encoded
  • First Dense Layer: 64 units with f(x)
  • Second Dense Layer: 32 units with f(x)
  • Third Dense Layer: $n_{\text{features}}$ units with linear activation function
      Here, $f(x) = \max(0, x)$
Mathematically, the decoder can be represented as:
decoded = max(0, max(0, max(0, encoded * W4 + b4) * W5 + b5) * W6 + b6)
Note: Wi represents the weights and bi represents the biases for the i-th layer.
Step 3: Overall Model:
  autoencoder($x$)=decoded(encoded($x$))
This equation represents the full autoencoder model, where “x” is the input data.
```

Please note that during the training process, the actual values of weights and biases (*Wi* and *bi*) are learned. The training involves minimizing the reconstruction loss between the input and the output. In Keras, this is typically done by specifying a loss function and an optimizer during the model compilation, and then with our training data, calling the fit method on the model.

RESULTS AND DISCUSSION

To examine the performance of the autoencoders, based on the number of hidden layers, three independent experiments were set up as separate models. KDDCUP99, CICIDS2017, and SIMARGL2022 datasets were used to train and test the models. The dataset KDDCUP99[16] contains 43 features, in which the model is trained with 1,25,973 samples and tested with 22,544 samples with a loss of 0.0290. The dataset CICIDS2017[17] contains 79 features in which the model is trained with 2,89,096 samples and tested with 2,25,711 samples, with a loss of 0.0032. The dataset SIMARGL2022[18] contains 30 temporal features with which the model is trained with 2,50,568 samples and tested with 25,389 samples with a loss of 0.0048. This is collectively illustrated in Table 1.

Performance Analysis

The performance metrics provide valuable insights into different aspects of a classification model’s performance. Precision and recall are particularly

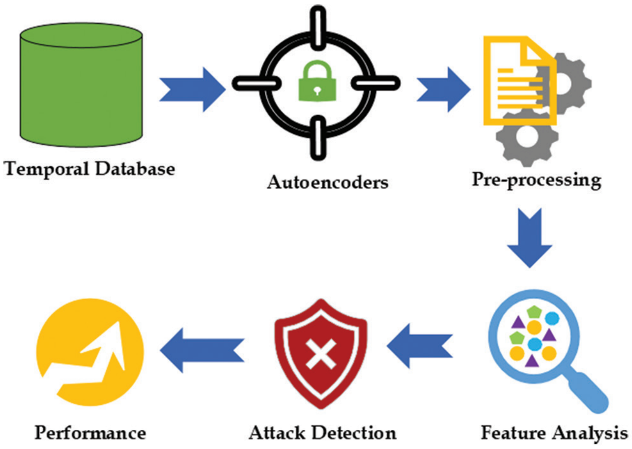

Figure 2: Cybercrime attack detection using autoencoder methodology

Table 1: Dataset’s information

Item	Datasets types		
	KDDCUP99	CICIDS2017	SIMARGL2022
Training samples	1,25,973	2,86,096	2,50,568
Testing samples	22,544	2,25,711	25,389
Features	43	79	30
AUROC	0.10144	0.52068	0.97235
AUPRC	0.97460	0.54828	0.96773
Loss	0.0290	0.0032	0.0048

useful for evaluating models in scenarios where class imbalance is present, as they focus on the correct prediction of positive instances and the ability to capture all positive instances, respectively. F1-measure balances both precision and recall, providing a single metric to assess overall model performance. Accuracy, while intuitive, may not provide a complete picture, especially in imbalanced datasets. Therefore, it is essential to consider multiple metrics to comprehensively evaluate a classification model, which are listed below.

1. Precision: The amount of TP predictions among all instances predicted as positive. Measures the accuracy of positive predictions
2. Recall: The amount of TP predictions among all real positive instances. Measures the ability to capture all positive instances
3. F1-measure: The harmonic mean of Precision and Recall, providing a balanced measure of a model's performance
4. Accuracy: The overall correctness of predictions through the model, measured as the part of correct predictions to the entire number of instances.

The collective information about the performance metrics is given in Table 2 and graphically plotted in Figure 3.

The datasets KDDCUP99, CICIDS2017, and SIMARGL2022 attack samples are plotted in Figures 4-6. The training and loss validation of datasets are shown in Figures 7-9. Furthermore, the loss comparison is plotted in Figure 10.

Table 2: Performance comparisons

Performance metrics	Datasets types		
	KDDCUP99	CICIDS2017	SIMARGL2022
Precision	90.35	56.72	99.22
Recall	95.68	100	100
F-measure	92.94	72.3	99.60
Accuracy	91.73	56.72	99.22

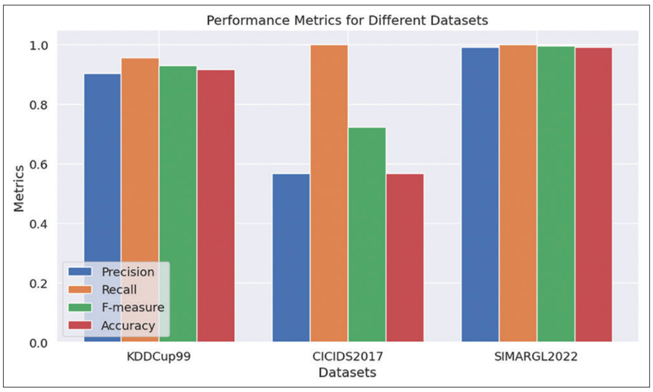

Figure 3: Histogram representation of performance metrics

DISCUSSION

According to the research findings, the loss comparison of the three datasets, the CICIDS2017

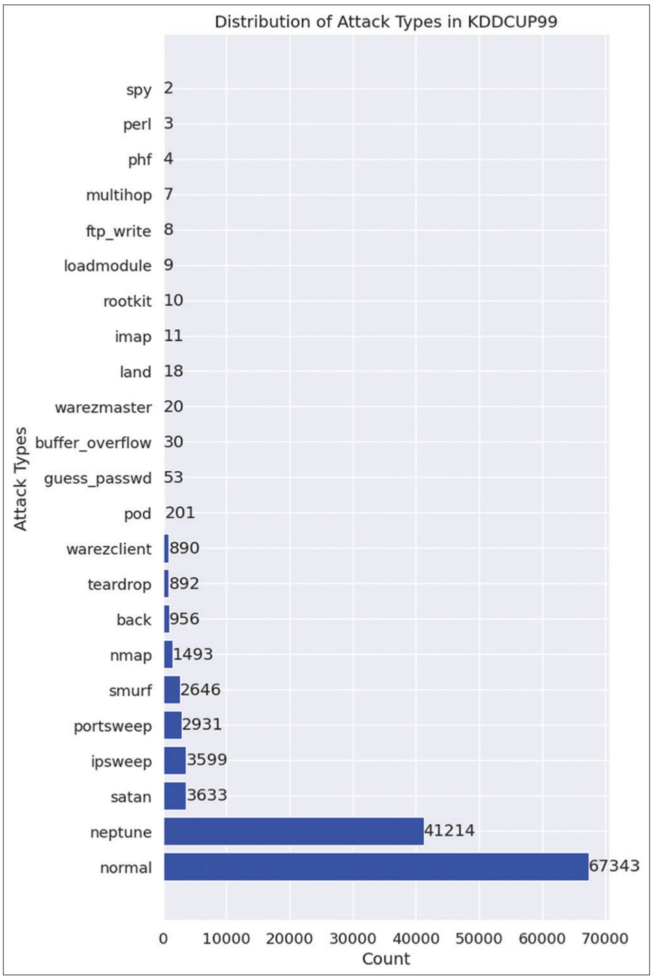

Figure 4: Attack samples in KDDCUP99

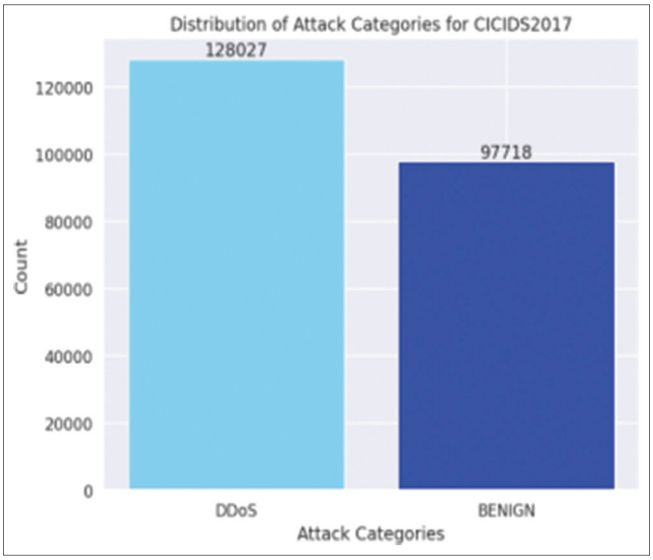

Figure 5: Attack samples in CICIDS2017

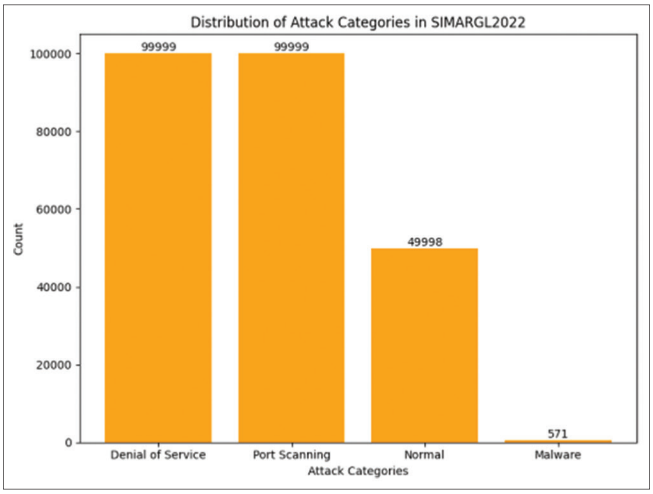

Figure 6: Attack samples in SIMARGL2022

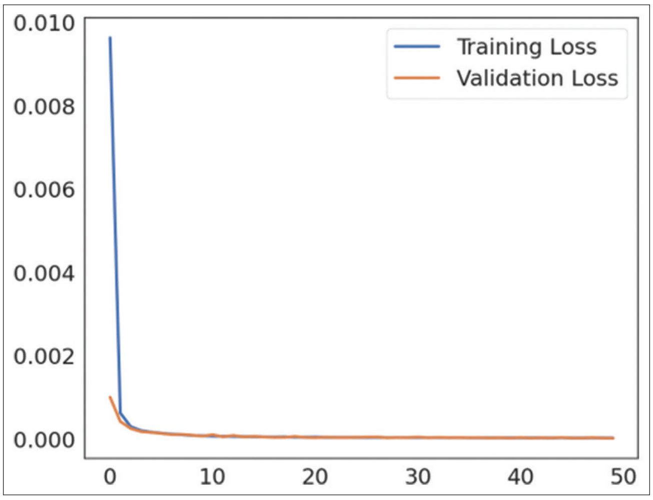

Figure 7: Training validation loss_KDDCUP99

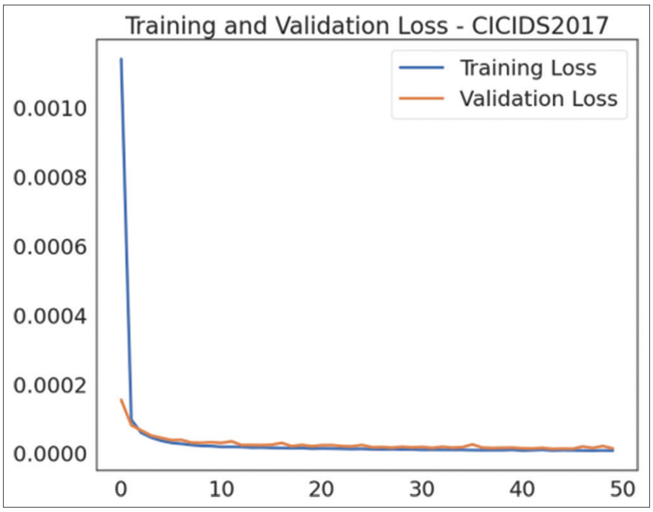

Figure 8: Training validation loss_CICIDS2017

dataset gives less loss when compared to the other datasets. In this, there are three hidden layers used for the model along with ReLU activation functions. The suggested solution, utilizing an autoencoder with KDDCUP99, CICIDS2017, and SIMARGL2022 datasets, outperformed a state-of-the-art system.

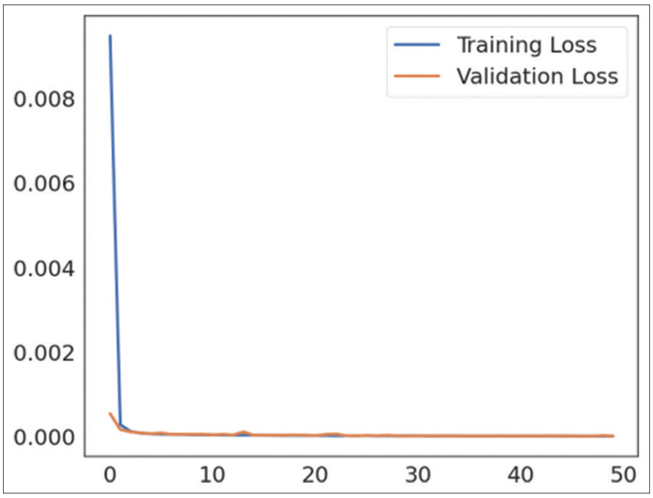

Figure 9: Training_validation_loss_SIMARGL2022

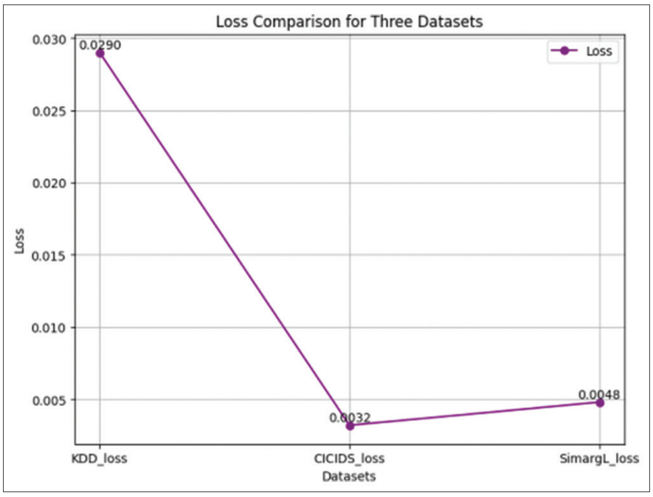

Figure 10: Loss comparison of datasets

Refer to Table 2 for the performance comparison. Furthermore, SIMARGL2022 is the latest dataset compared with the other two datasets, with the latest attack types.

CONCLUSION

Identifying unusual or suspicious behavior within systems or datasets helps to detect potentially harmful activities, such as security breaches, fraud, or system malfunctions, which may otherwise go unnoticed. By promptly identifying anomalies, they can moderate risks, safeguard confidential information, and maintain the honesty and security of their systems and data.

The proposed CADAE methodology aims to detect cybercrime activities such as anomaly detection, with an accuracy – 99.22%, precision – 99.22%, recall – 100% and F-measure – 99.60% for the SIMARGL2022 dataset, which is the highest when compared with KDDCUP99 and

CICIDS2017 datasets. The innovation lies in leveraging the SIMARGL2022 dataset's temporal features, incorporating recent threats such as DOS, Malware, and Port scanning. These features are detected with the proposed model, significantly enhancing the detection rate. In the future, enhancing performance and security measures involves exploring alternative deep learning algorithms or hybrid approaches that bolster the capability to detect and moderate cybercrimes, thereby enhancing user safety.

REFERENCES

1. Narayana YV, Sreedevi M. "DDCATF: Deep Learning Approach for Detection of Cybercrime Activities Based on Temporal Features". In: 2023 International Conference on Self Sustainable Artificial Intelligence Systems (ICSSAS), Erode, India; 2023. p. 462-9.
2. Mandalapu V, Elluri L, Vyas P, Roy N. Crime prediction using machine learning and deep learning: A systematic review and future directions. IEEE Access 2023;11:60153-70.
3. Chen S, Guo W. Auto-encoders in deep learning-a review with new perspectives. Mathematics 2023;11:1777.
4. Song Y, Hyun S, Cheong YG. Analysis of autoencoders for network intrusion detection. Sensors 2021;21:4294.
5. Torabi H, Mirtaheri S, Greco S. Practical autoencoder based anomaly detection by using vector reconstruction error. Cybersecurity 2023;6:1.
6. Xing X, Jin X, Elahi H, Jiang H, Wang G. A malware detection approach using autoencoder in deep learning. IEEE Access 2022;10:25696-706.
7. Nepal S, Joshi B. User Behavior Analytics for Insider Threat Detection using Deep Learning [Conference]; 2022.
8. Prabakaran MK, Sundaram PM, Chandrasekar AD. An enhanced deep learning-based phishing detection mechanism to effectively identify malicious URLs using variational autoencoders. IET Inform Secur 2023;17:423-40.
9. Willems D, Kohls K, Van Der Kamp B, Vranken H. Data exfiltration detection on network metadata with autoencoders. Electronics 2023;12:2584.
10. Farzad A, Aaron Gulliver T. Unsupervised log message anomaly detection. ICT Express 2020;6:229-37.
11. Siniosoglou I, Efstathopoulos G, Pliatsios D, Moscholios ID, Sarigiannidis A, Sakellari G. NeuralPot: An Industrial Honeypot Implementation Based on Deep Neural Networks. 2020 IEEE Symposium on Computers and Communications (ISCC), Rennes, France, 2020. p. 1-7.
12. Imran, Qayyum F, Kim DH, Bong SJ, Chi SY, Choi YH. A survey of datasets, preprocessing, modeling mechanisms, and simulation tools based on AI for material analysis and discovery. Materials (Basel) 2022;15:1428.
13. Haseeb J, Mansoori M, Hirose Y, Al-Sahaf H, Welch I. Autoencoder-based feature construction for IoT attacks clustering. Fut Gener Comput Syst 2022;127:487-502.
14. Abdullahi M, Baashar Y, Alhussian H, Alwadain A, Aziz N, Capretz LF, *et al*. Detecting cybersecurity attacks in internet of things using artificial intelligence methods: A systematic literature review. Electronics 2022;11:198.
15. Alhasan S, Abdul-Salaam G, Asante M, Missah YM, Ganaa E. Analyzing autoencoder-based intrusion detection system performance: Impact of hidden layers. J Inform Secur Cybercrimes Res 2023;6;107-17.
16. Siddique K, Akhtar Z, Khan FA, Kim Y. KDD cup 99 data sets: A perspective on the role of data sets in network intrusion detection research. Computer 2019;52:41-51.
17. Dube R. Use of the CICIDS 2017 Dataset in Information Security Research [Preprint]; 2022.
18. Mihailescu ME, Mihai D, Carabas M, Komisarek M, Pawlicki M, Hołubowicz W, *et al*. The proposition and evaluation of the RoEduNet-SIMARGL2021 network intrusion detection dataset. Sensors 2021;21:4319.