International Journal of Data Science and Artificial Intelligence (IJDSAI)
Volume 3, Issue 5, September – October (2025)

RESEARCH ARTICLE

MOUNTAIN GAZELLE OPTIMIZED DEEP LEARNING MODEL FOR INTRUSION DETECTION AND MITIGATION IN CLOUD

P. William [1,*], and Abdulatif Alabdultif [2]

[1] School of Engineering and Technology, Sanjivani University, Kopargaon, Maharashtra, India.
[2] Department of Computer Science, College of Computer, Qassim University, Buraydah 52571, Saudi Arabia.
[3]Dept. of Electrical and Electronic Engineering, University, City, Country.

[*]*Corresponding e-mail: william160891@gmail.com*

Abstract – Cloud Computing is a rapidly expanding technology that provides a range of online services, such as software, computing resources, and databases. However, its shared, distributed, and virtualized nature also makes it highly vulnerable to security threats, including data breaches, unauthorized access, and various types of cyberattacks that can compromise service availability and data integrity. To overcome these issues, a novel MOUNTain gazelle optimized Deep Learning framework (MOUNT-DL) has been proposed for effective intrusion detection and mitigation. The proposed model integrates variational autoencoder-based feature extraction, Mountain Gazelle Optimization (MGO)-based feature selection, and a Temporal Convolutional Network-based Bidirectional Gated Recurrent Unit (TCN-BiGRU) network for classifying the intrusions accurately. The model classifies four attack types and benign traffic using the CICIDS 2017 dataset. The efficacy of the MOUNT-DL approach is assessed utilizing f1-score, precision, accuracy, and recall. Experimental findings demonstrate that the MOUNT-DL approach achieves an accuracy of 98.02% compared to existing methods such as CNN, Deep-IDS, and EOS-IDS.

***Keywords** – Cloud Computing, Mitigation, Mountain Gazelle Optimization, Intrusion Detection, Deep Learning.*

1. INTRODUCTION

Cloud computing has become a leading model in modern computing, offering scalable, on-demand resources and services over the internet. It allows organizations to reduce infrastructure costs, enhance accessibility, and dynamically allocate resources as per workload demands [1-3]. With the rapid adoption of cloud-based systems across domains such as healthcare, finance, and IoT, the security and integrity of cloud environments have become paramount. Despite its benefits, the shared and distributed nature of cloud infrastructure makes it highly susceptible to various forms of cyber threats and vulnerabilities [4].

Intrusion attacks in cloud environments, including Man-In-The-Middle (MITM), Probe, Denial-of-Service (DoS), and User-to-Root (U2R), pose significant threats to data confidentiality, availability, and integrity. Attackers often exploit vulnerabilities in virtual machines, APIs, or resource-sharing layers to breach cloud systems [5]. Traditional perimeter-based security mechanisms are insufficient in detecting sophisticated, multi-stage, or zero-day attacks in dynamic cloud settings [6,7]. Furthermore, the complexity of cloud traffic patterns and the high volume of data make manual monitoring impractical, necessitating intelligent and automated intrusion detection systems (IDS) [8,9].

Numerous IDS approaches have been proposed employing deep learning and machine learning techniques, including Long Short-Term Memory (LSTM), Convolutional Neural Networks (CNN), Support Vector Machines (SVM), and autoencoder-based models [10,11]. While these models have demonstrated notable success, they often suffer from challenges such as overfitting, high computational cost, poor adaptability to imbalanced datasets, and insufficient feature optimization [12-14]. Moreover, many existing systems lack interpretability, delay response time, or fail to generalize to evolving attack vectors [15]. To overcome these issues, a novel MOUNT-DL approach has been proposed for effective intrusion detection and mitigation. The primary contributions of the proposed work are as follows,

- The key goal of this work is to develop an effective approach for intrusion detection in the cloud with high accuracy and to improve the security level.
- A variational autoencoder is utilized to extract essential statistical patterns from network traffic, which improves anomaly detection rate and enhances the DL approach's ability to detect malicious activities with greater precision.
- The proposed technique uses the MGO technique to select ideal features, reducing dataset dimensionality while keeping crucial data. This technique improves the efficacy of the

intrusion detection model by concentrating on the most important features.

- The proposed TCN-BiGRU method is utilized for precise classification of Attack and Non-Attack categories, also increasing classification performance with low computing complexity.
- The efficacy of the proposed approach is assessed utilizing variables, namely accuracy, computational overhead, security rate, PR, RC, and F1 score.

The remaining portions of this work are as follows. The related work is detailed in Section II. In Section III, the MOUNT-DL model is described, and Section IV presents the experimental results. Finally, Section V concludes the study.

2. LITERATURE SURVEY

In 2022, Mayuranathan, M., et al., [14] proposed a highly efficient security solution for detecting intrusions (EOS-IDS) within a cloud computing setting by utilizing a combined deep learning approach. To demonstrate its effectiveness, the suggested EOS-IDS method is tested with two standard datasets, DARPA IDS and CSE-CIC-IDS2018, and the results are contrasted with various current IDS methods.

In 2023, Yang, R., et al. [15] presented an intelligent ID approach based on cloud-edge collaboration. Experiments showed that the method reduced training time, memory needs, and storage for the model training procedure by more than 50%. Meanwhile, the detection accuracy was comparable to centrally trained approaches. The algorithm trained on cloud-edge collaboration can detect threats that local edge devices are not aware of.

In 2023, Salvakkam, D.B., et al. [16] recommended an Ensemble Intrusion Detection Model for Cloud Computing Utilizing DL (EICDL) to identify threats with greater accuracy. The EICDL model's performance is compared to that of modern ML approaches and current IDS, and the simulation results show that the EICDL ensemble method enhances identification and has a recall rate of 92.14% for identifying potential intrusions or assaults. The suggested EICDL ensemble approach greatly enhances the precision and efficiency of threat detection.

In 2024, Racherla, S., et al., [17] presented Deep-IDS, a novel and simple-to-deploy IDS based on DL. It was trained employing the CIC-IDS2017 dataset and utilizes a LSTM approach with 64 LSTM units. Deep-IDS has 97.67%, 98.17%, and 97.91% precision, recall, and F1 scores, respectively. Deep-IDS blocks malicious traffic by detecting and neutralizing intrusion attempts in an average of 1.49 seconds.

In 2024, Samriya, J.K., et al., [18] suggested a Network IDS (NIDS) that uses two ML models, namely eXtreme Gradient Boosting (XGBoost) and SVM approaches. The UNR-IDD and NSL-KDD datasets are employed to calculate the performance of the developed approach. The experiment's outcomes indicate that it outperforms baselines and has the potential to be applied to contemporary NIDS.

In 2024, Jayasankar, T., et al., [19] presented a dynamic search fireworks optimization-combined feature selection with optimum deep recurrent neural network (DFWAFS-ODRNN) approach for attack detection in an IoT environment. The two steps of the DFWAFS-ODRNN approach are intrusion categorization and feature selection. The accuracy of the DFWAFS-ODRNN approach intrusion detection is 96.11%.

In 2024, Aljuaid, W.A.H., and Alshamrani, S.S. [20] suggested a DL framework to effectively identify cyberattacks in the cloud environment by utilizing a sophisticated convolutional neural network (CNN)-based model. The suggested model has proven to be quite successful in defending cloud networks against a range of possible threats, according to experiments. The CNN approach has demonstrated its capacity to identify and categorize network attacks with over 98.67% accuracy, recall, and precision.

3. PROPOSED METHODOLOGY

In this part, a novel MOUNT-DL technique has been proposed. Initially, the dataset will be pre-processed and feature extracted using a Variational autoencoder. After feature extraction, the features are selected using the MGO. After feature selection, the attacks are classified into 5 classes, namely MITM, Probe, DoS U2R, and normal, by using the TCN-BiGRU model. Figure 1 demonstrates the overall block diagram of the proposed approach.

3.1 Data Preprocessing

Preprocessing refers to the operations performed on unprocessed data before training a DL approach. The model's accuracy and efficiency can only be improved by altering and preparing the data to suit the learning process. The preprocessing phase usually includes the tasks listed below:

3.1.1 Data Cleaning

The initial phase in the preprocessing process is cleaning the data, which often involves removing rows that have invalid or absent information, eliminating columns with a single value (for example, columns where every entry is zero), and discarding features that are already understood to be unrelated to the classification task.

3.1.2 Data Normalization

The suggested cyber threat detection model utilizes z-score normalization for normalizing the dataset features. It converts the data, and every feature has a mean (0) and a standard deviation (1). It is expressed as:

$$n = \frac{y-\mu}{\sigma} \tag{1}$$

where, n, y, μ and σ are the normalized feature, original feature, mean, and standard deviation.

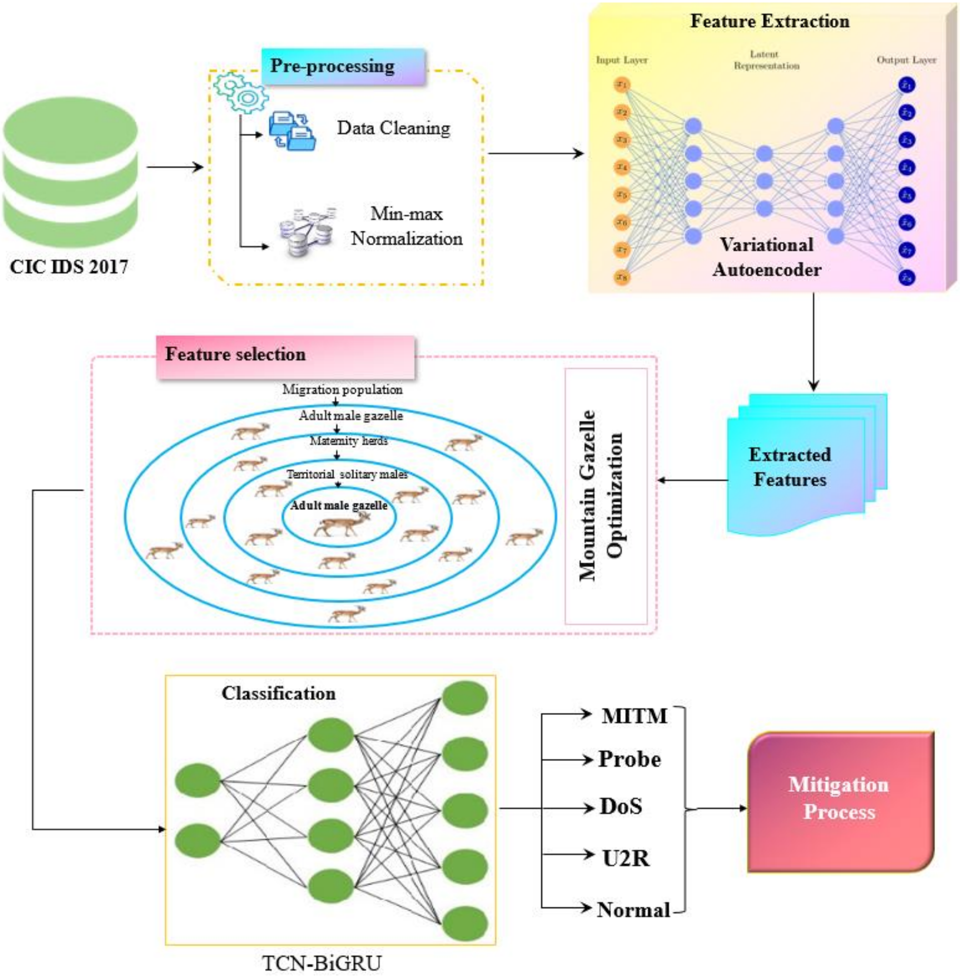


Figure 1. Workflow of the proposed Methodology

3.2 Feature Extraction via Variational Autoencoder (VAE)

Feature extraction is performed using the VAE model to identify key features relevant to intrusion classification. The encoded information is saved in the latent space, and it allows the VAE to capture the important features. Another neural network called the decoder attempts to reconstruct the original input from a point sampled from the latent space. The latent representation is mapped back into the data space using the decoder. Then, the reconstruction error is measured, and the overfitting is avoided by the regularization. As given in Figure 2, VAE has two elements, like an encoder $q_\varphi(x|y)$ which serves as an approximate side, and the decoder $p_\varphi(x|y)$ which serves as a data likelihood $y$with respect to the latent parameter$x$. In this process, the encoding function as a variation inference model and it maps the inputs to the approximate side with respect to the x. Then, the decoding is considered as a generative model and maps latent variables $x$back to distribution. For achieving this, it is considered that inputs are tested from a latent variable's Gaussian distribution. During the process of training, the model is optimized by minimizing two losses, like reconstruction loss$L$ and Kullback–Leibler divergence E_{KL}.

The fitness term for the VAE is derived from Equation (2), and it is the variation lower bound of the margin data likelihood. This margin data likelihood is the total with respect to the margin own data point likelihood$(y^{(j)})$ is give as $log\, p_\varphi\,(y) = \sum_{j=1}^{m} log\, p_\varphi\,(y^{(j)})$.Then, this term is again written for $(y^{(j)})$ as:

$$log\, p_\varphi\,(y^{(j)}) = E_{KL}(q_\varphi(x|y^{(j)})||p_\varphi(x|y^{(j)})) + L(\theta, \phi; y^{(j)}) \quad (2)$$

Since the $E_{KL}$is always non-negative, and by using Bayes' process, the variation lower limit $log\, p_\varphi\,(y^{(j)})$ is given as:

$$-E_{KL}(q_\varphi(x|y^{(j)})||p_\varphi(x|y^{(j)})) + D_{q_\varphi(x|y^{(j)})}[log\, p_\varphi\,(y^{(j)}|x) \quad (3)$$

The standard normal distribution is presented for indicating the latent variables x and it is given as:

$$x = \alpha + \beta\varepsilon \quad (4)$$

where, $\alpha, \beta$is the mean, standard deviation and $\varepsilon$is random parameter. The extracted features are utilized as input for the feature selection approach.

3.3 Feature Selection via Mountain Gazelle Optimization

The MGO technique is applied for feature selection. This algorithm mimics the social and migratory performance of gazelles, focusing on selecting the utmost related features for intrusion detection. By iteratively optimizing the feature set, this technique enhances the efficiency and accuracy of the classification process.

The MGO algorithm takes inspiration from the social interactions and lifestyle patterns of mountain gazelles. It makes use of a mathematical model based on the core ideas of gazelle social dynamics. Four key elements that are observed in the lives of mountain gazelles are incorporated into the MGO algorithm: migration to seek food (MSF), bachelor male herd (BMH), Territorial Solitary Male (TSM), and Maternity Herds (MH). Every gazelle in the MGO algorithm during optimization is a member of one of three herds: solitary territorial males, bachelor male herds, or maternity herds. The adult male gazelle living inside a herd's area is the global best answer in MGO.

MGO uses processes that have been theoretically stated and described to carry out optimization activities. These processes allow for the simultaneous investigation and exploitation of possibilities, enabling solutions to explore different choices and get closer to the greatest potential results. In conclusion, the MGO algorithm creates a workable optimization strategy by drawing on the social behaviour and life cycles of mountain gazelles. The MGO algorithm effectively balances exploration and exploitation by considering the unique characteristics of various herds, leading to improved optimization outcomes.

A) Territory Solitary Males (TSM)

As male gazelles meet adulthood, they can defend themselves and create separate, isolated areas that are separated from one another. Male adults fight for dominance over female territory. While juvenile males attempt to seize control of these regions or the females themselves, mature men work to shield their territory. The territory of an adult male is shown below.

$$TSM = \mathrm{male}_{\mathrm{gazelle}} - |\propto_1 \cdot \mathrm{BH} - \propto_2 \cdot \mathrm{N(v)} \cdot \mathrm{G}| \cdot \mathrm{cof}_{\mathrm{s}} \tag{5}$$

$$\mathrm{BH} = \mathrm{N}_{\mathrm{sm}} \cdot \lfloor \mathrm{s}_1 \rfloor + \mathrm{U}_{\mathrm{ps}} \cdot \lceil \mathrm{s}_2 \rceil, \mathrm{sm} = \{\lceil \mathrm{N}/3 \rceil \ldots \mathrm{N}\} \tag{6}$$

$$\mathrm{G} = \mathrm{T}_1(\mathrm{D}) \cdot \exp\left(2 - \mathrm{it}\left(\frac{2}{\mathrm{Maxit}}\right)\right) \tag{7}$$

$$\mathrm{Cof}_{\mathrm{s}} = \begin{cases} ((\mathrm{m}+1) + \mathrm{s}_3 \\ (\mathrm{m} \times \mathrm{T}_2(\mathrm{D})) \\ (\mathrm{s}_4(\mathrm{D})) \\ (\mathrm{T}_3(\mathrm{D}) \times \mathrm{T}_4(\mathrm{D})^2 \times \cos(2\mathrm{s}_4)\mathrm{T}_3(\mathrm{D}) \end{cases} \tag{8}$$

$$\mathrm{m} = -1 + \mathrm{it} \times \frac{-1}{\mathrm{Maxit}} \tag{9}$$

The symbol $\mathrm{male}_{\mathrm{gazelle}}$ indicates the position vector. The gazelle's preliminary position is shown by N(v). Arbitrary numbers having values are denoted by $\propto_1$ and $\propto_2$. BH stands for the young male herd's coefficient vector. $\mathrm{Cof}_{\mathrm{s}}$ is an additional randomly generated coefficient vector that is modified at the end of each iteration to improve the search region's efficiency. U_{ps} represents the average number of populations, rounded to the nearest integer, chosen randomly from T. N_{sm} presents a young man within the range of sm. T represent the total number of gazelles, while the values s_1, s_2, s_3, and s_4 are arbitrary numbers between 0 and 1. While T_2, T_3, and T_4 represent arbitrary integers within the usual range and problem dimensions, T_1 is a random number taken from a standard distribution.

B) Maternity herds (MH)

Maternity herds are essential to the survival of mountain gazelles because they guarantee the birth of healthy male progeny. Male gazelles help with both childbirth and the assistance of other males attempting to mate with the female. The following behavior is described:

$$\mathrm{MH} = (\mathrm{BH} + \mathrm{Cof}_{1,\mathrm{s}}) + (\propto_3 \cdot \mathrm{male}_{\mathrm{gazelle}} - \propto_4 \cdot \mathrm{N}_{\mathrm{rand}}) \cdot \mathrm{Cof}_{2,\mathrm{s}} \tag{10}$$

The variables $\propto_3$ and $\propto_4$ here represent random numbers 1 or 2. One gazelle is randomly selected, and $\mathrm{N}_{\mathrm{rand}}$ is the position vector of that gazelle. Random selection is used to create the coefficient vectors $\mathrm{Cof}_{1,\mathrm{s}}$ and $\mathrm{Cof}_{2,\mathrm{s}}$.

C) Bachelor male herd (BMH)

Male gazelles frequently mark their territories as they become older to show that they are superior to females. Then, in a competition for dominance over females, young male gazelles engage in violent altercations with adult males. Numerical expressions for this behaviour are as follows:

$$BMH = (N(v) - D) + (\propto_5 \cdot male_{gazelle} - \propto_6 \cdot BMH) \cdot Cofr \tag{11}$$

$$D = (|N(v)| + |male_{gazelle}|) \times (2 \times s_6 - 1) \tag{12}$$

Here, the gazelle in the current iteration is denoted by N(v). $\propto_5$ and $\propto_6$ are randomly selected numbers, each between 1 and 2. s_6 is a number from 0 to 1.

D) Migration in search of food (MSF)

Mountain gazelles explore their favourite verdant meadows by wandering. Equation (13) represents this random movement.

$$\mathrm{MSF} = \mathrm{Y} + (\mathrm{K} - \mathrm{Y}) \times \mathrm{s}_7 \tag{13}$$

Here, K and Y represent the upper and lower bounds. s_7 indicates a random number between 0 and 1. The selected subset after convergence is passed to the TCN-BiGRU classifier for final intrusion detection.

3.4 Classification via TCN-BiGRU

The selected features from the MGO are fed into the TCN-BiGRU approach, which introduces a new combination of BiGRU and TCN to enhance the prediction of attacks in the cloud, categorizing the data as either Attack or Normal, and thereby improving the overall accuracy of detection in cloud environments.

Temporal Convolutional Network (TCN)

The input to the TCN layer is the extracted features. These features are typically converted into dense vectors using word embeddings. The three primary components of a TCN, a unique 1-D full CNN, are the residual block, the dilated convolution, and the causal convolution. The value at time t in the higher layer is solely based on the values at t and earlier in the lower layer to causal convolution. The standard 1-D casual convolutional layer for a 1-D input $h \in \mathbb{R}^S$ and a filter $c: \{0, \ldots, k-1\} \rightarrow \mathbb{R}$ is defined as follows (14) and (15):

$$C(h_s) = (h * c)(s) = \sum_{x=0}^{k-1} c_x h_{s-x} \tag{16}$$

$$\widetilde{os} = \left(C(h_1), C(h_2), \ldots, C(h_S)\right) \tag{17}$$

where C (·) is a convolutional operation, k is the convolutional kernel size, and os is an output sequence. The TCN uses a hyperparameter to skip a portion of the input, allowing the filter to act on a range that is longer than the filter's length. In particular, when infused with the causal convolution, the i^{th} layer DC can be written as (18).

$$C(h_s) = (h * g_i c)(s) = \sum_{x=0}^{k-1} c_x h_{s-g_i x} \tag{18}$$

$$\widetilde{os} = \left(C(h_1), C(h_2), \ldots, C(h_S)\right) \tag{19}$$

where g_i which can be adjusted to 2^{i-1} is the i^{th} layer's dilation factor. The past direction is indicated by $s - g_i x$. A TCN layer is represented by equation (19), and TCN is created by stacking several TCN layers.

BiGRU Layer

The output feature map from the TCN layer becomes the input to the BiGRU layer. The BiGRU receives a sequence of features, retaining the temporal order, but now enriched with local context from the TCN. The BiGRU takes the special feature vectors like edges, shapes, and textures are produced by TCN as its input. GRU neural systems are a subset of RNNs. To overcome the issue that typical RNNs rewrite their memory in unit steps and suffer from gradient dispersion, use RNN. GRU is a simple LSTM that can be simply determined while preserving the effectiveness of LSTM neural networks. Formulas 20, 21, 22, and 23 are used to compute h_s.

$$i_s = \varphi(G_q y_s + V_q h_{s-1} + n_q) \tag{20}$$

$$a_s = \varphi(G_a y_s + V_a h_{s-1} + n_a) \tag{21}$$

$$\tilde{h}_s = \tanh\left(G_f y_s + V_h (h_{s-1} \otimes i_s) + n_s\right) \tag{22}$$

$$h_s = (1 - a_s) \otimes h_{s-1} + a_s \otimes \tilde{h}_s \tag{23}$$

Here, y_s denotes the input vector, and h_s provides the output vector of the GRU. At time s, the input vector y_s and the hidden state h_{s-1} are fed as inputs to the GRU network, which generates the output h_s. The Sigmoid function is represented by the symbol φ to aid GRU neural networks in remembering or storing information. The reset and update gates are i_s and a_s, respectively, and the elementwise production is ⊗. Furthermore, the candidate's assumed state at the time s is represented by $\tilde{h}_s$. The forward and backward hidden layers make up the BiGRU structure. Two symmetric hidden-layer state vectors are created by feeding each data pattern into both the forward and reverse GRU networks. After a symmetrical merger, an overall coded representation of the input text can be obtained using both of those state vectors, as shown in equation (24):

$$H_s = [\overrightarrow{H_s} \oplus \overleftarrow{H_s}] \tag{24}$$

The data from the network module is then fed into the dense layer and the softmax activation, which classifies the attack into attacks and normal. By enhancing the framework's ability to identify long-term dependencies in the input sequence, the proposed TCN-BiGRU technique raises the security level and enhances classification efficiency in identifying attacks.

3.5 Attack mitigation

The proposed cloud intrusion detection and mitigation system integrates an intelligent mitigation mechanism that activates immediately after an intrusion is detected by the TCN-BiGRU classifier. Utilizing an event-condition-action (ECA) strategy, the system communicates with the Software-Defined Networking (SDN) controller to dynamically enforce countermeasures such as dropping malicious packets or blocking suspicious IPs. To ensure the continuity of legitimate cloud operations, a dynamic safe list of verified IP addresses and ports is maintained, preventing false positives from affecting normal traffic. This safe list is periodically updated through an automated scanning mechanism that adapts to changing cloud behaviour. By combining deep learning-based detection with SDN-driven mitigation, the framework ensures real-time, accurate, and adaptive protection for cloud environments.

4. RESULT AND DISCUSSION

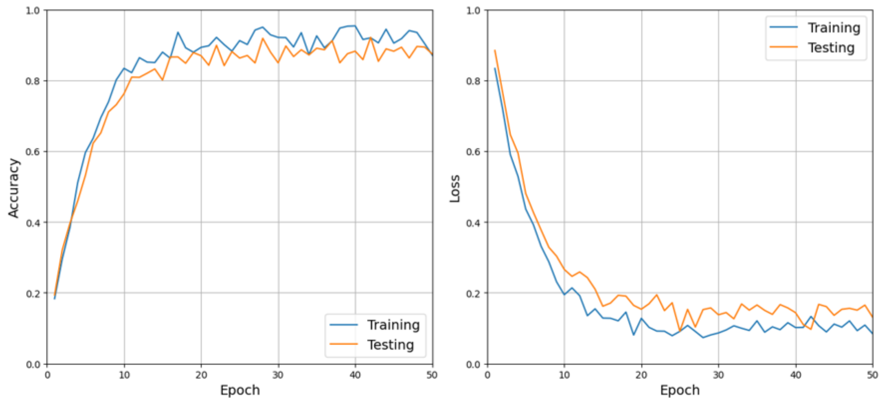
This section analyzes the experimental findings of the proposed approach. The PC requirements for this simulation are 12 GB of RAM, an Intel Core i9-9820X 3.30 GHz CPU, and Ubuntu 20.04.1 LTS. Jupyter Notebook Anaconda was used to generate Python scripts. The proposed approach's efficacy is compared to existing methodologies in terms of f1score, recall, accuracy, and precision

4.1 Dataset Description

CIC IDS 2017 Dataset: Among the available IDS datasets, we selected the CIC IDS 2017 dataset, which includes contemporary DoS attacks and benign network flows to mimic real-world scenarios. The experimental setup involved physical machines and tools like curl for generating normal traffic, using a realistic network topology with protocols such as HTTP, HTTPS, SSH, FTP, SMTP, IMAP, and POP3. We used Friday's data, which includes DoS attacks and benign traffic. The dataset, in labeled .csv format, spans several GBs with 85 features, making it ideal for evaluating feature selection on five machine learning classifiers.

4.2 Comparison Analysis

This section's simulation results assess how well the proposed method identifies and classifies intrusions. The proposed framework is contrasted with the current EOS-IDS, Deep-IDS, and CNN approaches. The approach is evaluated using f1score, FAR, accuracy, recall, and precision.



(a) (b)

Figure 2. Accuracy and Loss curve of the CIC IDS 2017 dataset

The suggested approach attained an accuracy of 98.02% on the CIC IDS 2017 dataset. Figures 2(a) and 2(b) exhibit the classification of training and testing using accuracy and loss plots of the suggested approaches. These charts demonstrate the proposed approach's effectiveness at identifying attacks. Furthermore, low loss values suggest successful learning with less overfitting during the training period.

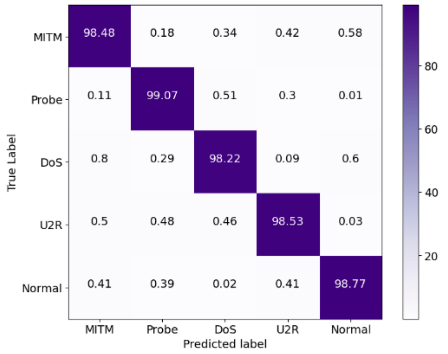


Figure 3. Confusion matrix for CICIDS2017

Figure 3 demonstrates the classification confusion matrix of the proposed model across the CICIDS2017 datasets. From this confusion matrix, the MOUNT-DL approach has a lower error rate with high classification accuracy in detecting intrusions. According to the results, the proposed approach on the CICIDS2017 Dataset accurately classifies 98.48% for MITM, 99.07% for Probe, 98.22% for DoS, 98.53% for U2R, and 98.77% for Normal class.

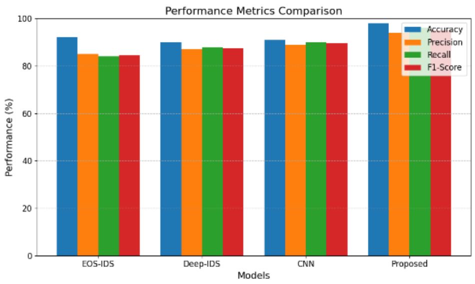


Figure 4. Performance Comparison for the proposed method

The performance evaluation of the suggested and current EOS-IDS, Deep-IDS, and CNN techniques across the CICIDS2017 datasets regarding precision, f1score, recall, and accuracy is demonstrated in Figure 4. On the CICIDS2017 datasets, the proposed framework performs better than the other methods, with an accuracy of 98.02%. These findings indicate that the MOUNT-DL approach outperforms earlier methods in every metric that was assessed.

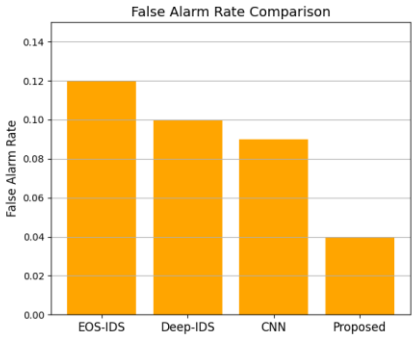


Figure 5. FAR Comparison of the proposed method

Figures 5 compare the FAR of the MOUNT-DL approach with existing EOS-IDS, Deep-IDS, and CNN approaches across the datasets CICIDS2017. The suggested method performs better at reducing false alarms than the current EOS-IDS, Deep-IDS, and CNN models. The proposed approach attains a lower FAR on CICIDS2017 datasets.

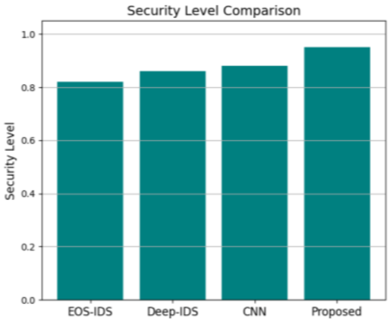


Figure 6. Security Level Comparison

The security performance comparison between the suggested technique and current approaches is shown in Figure 6. By utilizing the TCN-BiGRU model, the suggested framework attains the highest level of security. the proposed method reached 97%, whereas CNN, Deep-IDS, and EOS-IDS scored 88%, 86%, and 82% respectively, proving its robustness in handling cloud-based threats.

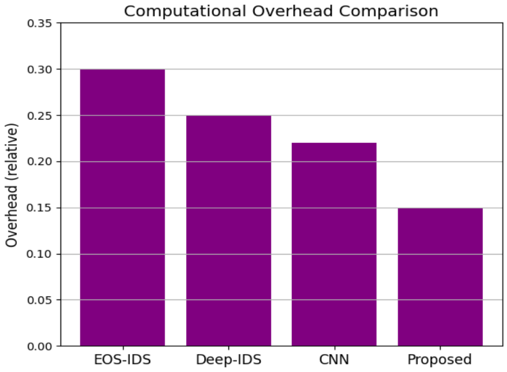


Figure 7. Computational Overhead Comparison

Figure 7 illustrates the computational overhead over time comparison for the proposed and existing CNN, Deep-IDS, and EOS-IDS approaches. The computational overhead of the proposed approach was minimal at 0.15, outperforming CNN (0.22), Deep-IDS (0.25), and EOS-IDS (0.30), making it efficient for cloud environments.

5. CONCLUSION

In this research, a new MOUNT-DL approach has been developed for intrusion detection and mitigation in cloud computing environments. The integration of a Variational Autoencoder for feature extraction, MGO for optimal feature selection, and the hybrid TCN-BiGRU model for accurate classification significantly enhances the overall detection performance. The MOUNT-DL system achieves a high accuracy of 98.02% and effectively classifies multiple intrusion types with minimal false alarms. The incorporation of SDN with an ECA-based mitigation mechanism further strengthens the real-time defensive capability of the model, ensuring secure cloud operations. Comparative analysis confirms that the MOUNT-DL method outperforms existing approaches in terms of accuracy, security level, false alarm reduction, and computational efficiency, making it highly suitable for practical deployment in modern cloud infrastructures. In the future, this work can be improved by incorporating federated learning to enhance collaborative intrusion detection across multi-cloud environments.

CONFLICTS OF INTEREST

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

FUNDING STATEMENT

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors

ACKNOWLEDGEMENTS

The author would like to express his heartfelt gratitude to the supervisor for his guidance and unwavering support during this research for his guidance and support.

REFERENCES

[1] T. A. Devi and A. Jain, “Enhancing Cloud Security with Deep Learning-Based Intrusion Detection in Cloud Computing Environments,” *2024 2nd International Conference on Advancement in Computation & Computer Technologies (InCACCT)*, pp. 541–546, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[2] K. Sivaprasad Yerneni, A. Ravi Teja, K. Sri Harsha, and Y. Naresh Kiran Kumar Reddy, “Towards Proactive Cloud Security: A Survey on ML and Deep Learning-Based Intrusion Detection Systems,” *J. Contemp. Edu. Theo. Artific. Intel.*, JCETAI-116, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[3] D. M. A. A. Afraji, J. Lloret, and L. Peñalver, “Deep learning-driven defense strategies for mitigating DDoS attacks in cloud computing environments,” *Cyber Security and Applications*, p. 100085, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[4] P. V. Sivarambabu, R. Agrawal, A. Tirumala, S. M. Subani, V. Parisae, and S. S. Nukala, “Enhancing Cloud Security Through AI-Driven Intrusion Detection Utilizing Deep Learning Methods and Autoencoder Technology,” *Generative Artificial Intelligence: Concepts and Applications*, pp. 249–264, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[5] H. Waqas and T. Henry, “Machine Learning-Powered Intrusion Detection Systems for IoT and Cloud Environments,” 2025. [CrossRef] [Google Scholar] [Publisher Link]

[6] S. S. Qureshi, J. He, S. U. Qureshi, N. Zhu, A. Wajahat, A. Nazir, F. Ullah, and A. Wadud, “Advanced AI-driven intrusion detection for securing cloud-based industrial IoT,” *Egyptian Informatics Journal*, vol. 30, p. 100644, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[7] S. Berríos, S. Garcia, P. Hermosilla, and H. Allende-Cid, “A Machine-Learning-Based Approach for the Detection and Mitigation of Distributed Denial-of-Service Attacks in Internet of Things Environments,” *Applied Sciences*, vol. 15, no. 11, p. 6012, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[8] V. Govindrajan, “Machine learning based approach for handling imbalanced data for intrusion detection in the cloud environment,” *2025 3rd International Conference on Disruptive Technologies (ICDT)*, pp. 810–815, Mar. 2025. [CrossRef] [Google Scholar] [Publisher Link]

[9] H. G. A. Umar, I. Yasmeen, M. Aoun, T. Mazhar, M. A. Khan, I. H. Jaghdam, and H. Hamam, “Energy-efficient deep learning-based intrusion detection system for edge computing: a novel DNN-KDQ model,” *Journal of Cloud Computing*, vol. 14, no. 1, p. 32, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[10] A. Alabdulatif, “A novel ensemble of deep learning approach for cybersecurity intrusion detection with explainable artificial intelligence,” *Applied Sciences*, vol. 15, no. 14, p. 7984, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[11] P. Chitte and S. Chaudhari, “Artificial immune based intrusion detection and mitigation system using entropy fluctuation method and deep maxout classifier,” *International Journal of Machine Learning and Cybernetics*, pp. 1–24, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[12] H. Bhat and A. P. Rodrigues, "IoT Intrusion: Detection Methods and Mitigation Strategies," *2025 International Conference on Artificial Intelligence and Data Engineering (AIDE)*, pp. 901–906, Feb. 2025. [CrossRef] [Google Scholar] [Publisher Link]

[13] P. Sarumathy, S. Rajasree, and A. Chandrasekar, "An AI-Based Intrusion Prevention System to Enhance Cloud Security," *2025 4th International Conference on Sentiment Analysis and Deep Learning (ICSADL)*, pp. 313–320, Feb. 2025. [CrossRef] [Google Scholar] [Publisher Link]

[14] M. Mayuranathan, S. K. Saravanan, B. Muthusenthil, and A. Samydurai, "An efficient optimal security system for intrusion detection in cloud computing environment using hybrid deep learning technique," *Advances in Engineering Software*, vol. 173, p. 103236, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[15] R. Yang, H. He, Y. Xu, B. Xin, Y. Wang, Y. Qu, and W. Zhang, "Efficient intrusion detection toward IoT networks using cloud–edge collaboration," *Computer Networks*, vol. 228, p. 109724, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[16] D. B. Salvakkam, V. Saravanan, P. K. Jain, and R. Pamula, "Enhanced quantum-secure ensemble intrusion detection techniques for cloud based on deep learning," *Cognitive Computation*, vol. 15, no. 5, pp. 1593–1612, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[17] S. Racherla, P. Sripathi, N. Faruqui, M. A. Kabir, M. Whaiduzzaman, and S. A. Shah, "Deep-IDS: A Real-Time Intrusion Detector for IoT Nodes Using Deep Learning," *IEEE Access*, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[18] J. K. Samriya, S. Kumar, M. Kumar, H. Wu, and S. S. Gill, "Machine learning based network intrusion detection optimization for cloud computing environments," *IEEE Transactions on Consumer Electronics*, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[19] T. Jayasankar, R. Kiruba Buri, and P. Maheswaravenkatesh, "Intrusion detection system using metaheuristic fireworks optimization-based feature selection with deep learning on Internet of Things environment," *Journal of Forecasting*, vol. 43, no. 2, pp. 415–428, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[20] W. A. H. Aljuaid and S. S. Alshamrani, "A deep learning approach for intrusion detection systems in cloud computing environments," *Applied Sciences*, vol. 14, no. 13, p. 5381, 2024. [CrossRef] [Google Scholar] [Publisher Link]

AUTHORS

P. William is working as Director (Research) at Sanjivani University, Kopargaon. He is the Post Doctoral Fellow from Amity University Dubai, UAE and Adjunct faculty of Victorian Institute of Technology, Australia. He is recognized in World Top 2% Scientist list by Stanford University and Elsevier. He is a member of IEEE, QCFI, ISTE and various other professional bodies. His research includes innovation and development of cutting-edge solutions in the fields of natural language processing, artificial intelligence, deep learning, machine learning, soft computing, cybersecurity, and cloud computing. He has published 225+ papers in Scopus indexed journals and Conferences. He has 30+ patents published with grants in his credit. He has authored and edited 20+ books with renowned publishers of global recognition. He has been associated with numerous Multi-National Companies and various Educational Groups for his expertise in research, corporate training and consulting where he has contributed to the advancement of knowledge and practice in his domain. A focused professional with experience of consulting in Research, Innovation and Development. He served as a Chairperson and Auditor in multiple committees of national recognition. Delivered Keynote speeches and chaired many sessions in International Conferences. He was appointed as Series Editor, Guest Editor and Reviewer in Scopus/ Web of Science indexed journals.

Abdulatif Alabdulatif is an associate professor at the School of Computer Science & IT, Qassim University, Saudi Arabia. He completed his Ph.D. degree in Computer Science from RMIT University, Australia in 2018. He received his B.Sc. degree in Computer Science from Qassim University, Saudi Arabia in 2008 and his M.Sc. degree in Computer Science from RMIT University, Australia in 2013. He has published more than 70 academic papers in prominent journals. His research interests include applied cryptography, cloud computing, and E-health.

Arrived: 19.09.2025
Accepted: 22.10.2025