

# Elephant Herd Feature Optimization Based Intrusion Detection System

Shivani Meena<sup>1</sup>, Assistant Professor Rani Kushwaha<sup>2</sup>, Professor Jayshree Boaddh<sup>3</sup>

M.Tech student, Mittal Institute of Technology, Bhopal<sup>1</sup>

CSE Mittal Institute of Technology, Bhopal<sup>2</sup>

HOD, Cse Vaishnavi Institute of technology and Science, Bhopal<sup>3</sup>

**Abstract-** The growing dependence on technology for a wide range of activities has dramatically increased computational demands, driving significant growth in computer network usage over the past few decades. This surge in demand for processing and storage capabilities has opened up business opportunities for companies but has also drawn the attention of cybercriminals. In response to these threats, researchers have developed various attack detection and prevention models. This paper introduces a new intrusion detection model that operates in two phases. The first phase involves building a feature ontology to train a convolutional neural network (CNN), and the second phase tests the trained model. For feature selection, the model uses an Elephant Herd Optimization-based genetic algorithm, which efficiently identifies a strong feature set for classifying network sessions. Experiments on a real-world dataset show that the proposed model can detect various types of attacks within normal sessions. Results demonstrate improved accuracy and performance metrics compared to existing models.

**Index Terms-** Anomaly Detection, ANN, Clustering, Genetic Algorithm, Intrusion Detection.

## I. INTRODUCTION

Computer networks represent one of the latest advancements in IT services, offering the advantage of accessibility regardless of location or time. They provide flexibility in adjusting storage capacities, reduce costs, and support mobile and collaborative applications and services. Additionally, network services are multisource, allowing users to choose from different providers based on their specific needs. By utilizing computer networks, organizations can save on storage costs, power consumption, physical space, and maintenance. As these services become increasingly widespread, businesses, banks, and governments are adopting the technology. However, this expansion has also exposed systems to a range of cyberattacks, prompting the need for robust security measures. Many network service providers offer various security features as part of their service packages. For instance, Amazon Web Services (AWS) provides security services with limited validity based on the duration of the service license.

Traditional intrusion detection systems (IDS) are insufficient for managing the vast data flow within modern network infrastructures, which experience high volumes of traffic. Most conventional IDSs are single-threaded, whereas multi-threaded IDSs are required in cloud environments to handle large-scale data flow. In traditional networks, IDSs monitor,

detect, and alert administrators to suspicious activities by being deployed at critical points on the user's site. However, in a cloud-based network, IDSs must be installed on the cloud server and operated by the service provider. In cases where an attacker breaches and compromises user data, the cloud user may not be directly informed. The service provider receives the intrusion data and may choose not to disclose the incident to the user, potentially to protect their reputation. To address this, a neutral third-party monitoring service can offer unbiased oversight, providing alerts and reports to both cloud users and service providers.

This paper proposes a multi-cloud IDS managed by a third-party monitoring service. This service would deliver alert reports and expert advice to both cloud users and providers, ensuring transparency and reliable intrusion detection. The proposed system introduces an efficient and reliable distributed cloud IDS approach to overcome the limitations of traditional IDSs.

## II. RELATED WORK

Kabir et al. [7] developed an Optimum Allocation Least Square Support Vector Machine (OALSVM) model, where "optimum allocation" refers to selecting specific sessions from a dataset to train the SVM model. The accuracy of the

OALSSVM model in detecting intrusions depends on these selected sessions, leading to improved performance.

Chuanlong Yin [8] explored an intrusion detection system using a recurrent neural network (RNN-IDS) based on deep learning. The study evaluated the model's performance in both binary and multiclass classification, analyzing the impact of factors like the number of neurons and learning rate. Yin compared the RNN-IDS against other machine learning models such as J48, artificial neural networks, random forests, and SVMs using a benchmark dataset.

Kaiyuan et al. [9] proposed a network intrusion detection algorithm that combines hybrid sampling with a deep hierarchical network. Their method applies one-side selection (OSS) to remove noise from majority samples and SMOTE to increase minority samples, creating a balanced dataset. The model uses a CNN to extract spatial features and BiLSTM for temporal features, enhancing detection accuracy and reducing training time.

In another study [10], the authors compared the performance of neural networks, SVMs, and decision trees for anomaly detection. They found that the effectiveness of machine learning algorithms depends heavily on the practical context. Machine learning approaches like neural networks, SVMs, and decision trees are widely used in anomaly detection to improve classification speed and accuracy. Additionally, techniques such as genetic algorithms and information theory further enhance classification by combining theoretical support.

Zina et al. [11] introduced two models for intrusion detection and classification: the Trust-based Intrusion Detection and Classification System (TIDCS) and its accelerated version, TIDCS-A. TIDCS uses advanced feature selection, grouping features to optimize classification. It maintains system integrity by updating trust relationships between nodes, while TIDCS-A enhances this by dynamically determining optimal times for node maintenance, reducing exposure to attacks. Both models base their final classification decisions on a combination of node behavior history and machine learning algorithms, with detected attacks lowering node trustworthiness.

R. Ben et al. [12] proposed a hybrid approach that combines Convolutional Neural Networks (CNN) with bidirectional Long Short-Term Memory (BiLSTM) networks to enhance network intrusion detection capabilities. This approach supports both binary and multiclass classification tasks effectively. The efficacy of this hybrid model was validated using established datasets like UNSW-NB15 and NSL-KDD, demonstrating superior detection performance.

T. Kim et al. [13] developed a method to discern complex packet patterns for distinguishing between network intrusions and benign sessions. They created a new training dataset for a Generative Adversarial Network (GAN) using misclassified data from an original LSTM-DNN-trained dataset. This GAN assesses whether a received packet can be accurately classified by the LSTM-DNN model. If uncertainty arises in classification, the detection process pauses and retries with the next packet. The refined classification algorithm based on LSTM-DNN and the validation model using GAN enables precise real-time network intrusion detection without interrupting sessions or requiring delays to accumulate packets.

### III. METHODOLOGY

The proposed EHDL-IDS (Elephant Herd based Deep Learning for IDS) model is described in detail in this section. The entire process is divided into two main modules: the training module and the testing module. In the first module, the development of a feature ontology and the training of an CNN network are undertaken. The second module focuses on the testing and evaluation of the trained CNN model. Figure 1 illustrates the operational blocks of the proposed model.

**Dataset Cleaning** The dataset cleaning stage involves removing unwanted information from the dataset to improve the quality of the data. The input data contains various attributes, each with its specific relevance to the analysis. For instance, the input dataset used in this study consists of several fields, but some initial feature values, such as session ID, connection type, and transferring protocol, were excluded from the analysis [12]. The cleaned dataset is structured into a matrix format with rows and columns, where each row represents a session and each column represents a feature set associated with that session.

$$PID \leftarrow \text{Pre\_Processing}(ID)$$

Where ID is intrusion dataset and PID is Preprocessed Intrusion Dataset

**Feature Optimization** After preprocessing, the dataset matrix undergoes further analysis to identify the most effective features that contribute directly to the classification of intrusions. To construct this feature ontology, the study employs an elephant herd algorithm.

**Elephant Herd optimization** In this study, a random set of features was developed using a Gaussian function to model the social structure of elephant clans. As a result, each clan is characterized by a unique set of features that yield binary values composed of 0s and 1s. This binary feature set can be likened to the concept of an antibody in genetic algorithms, where the presence of a specific feature is indicated by the

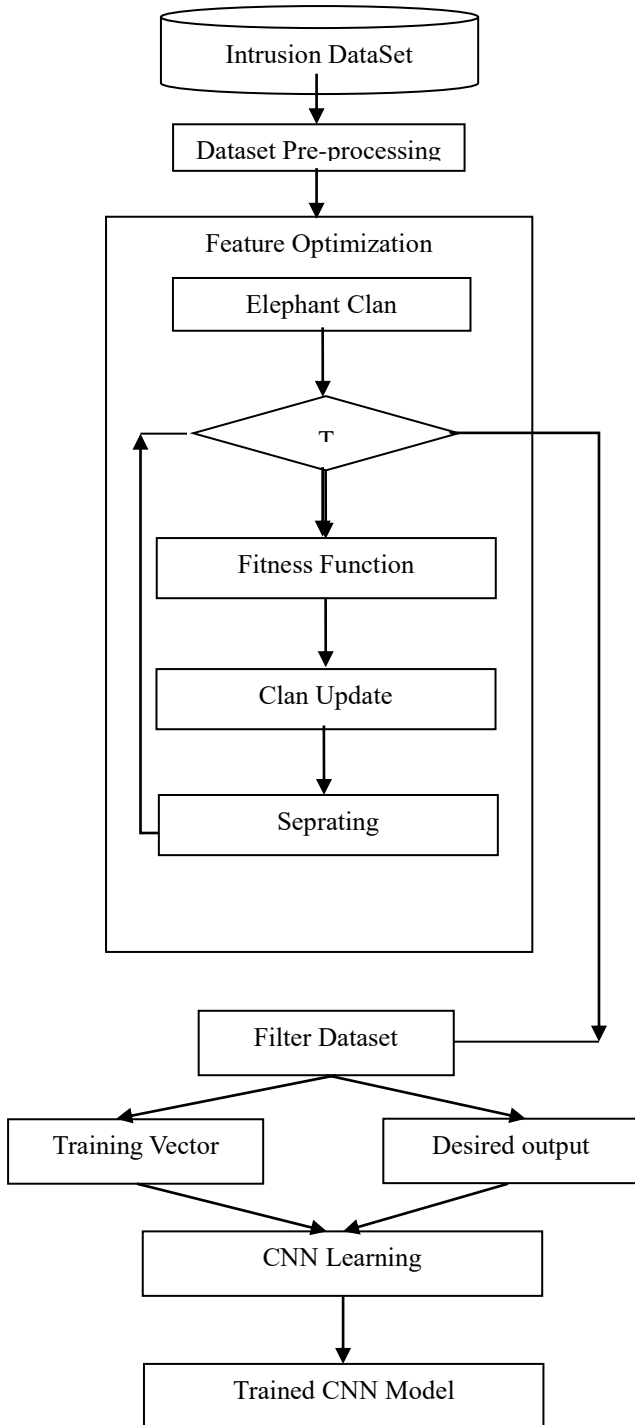


Fig. 1 Block diagram of NIDAIFL.

value 1, and its absence is represented by the value 0. Within this framework, every feature in the population is assigned two distinct flags: a value of 1 signifies that the feature is present, while a value of 0 indicates that the feature is absent. This representation enables a clear and efficient method of

encoding the various characteristics associated with each elephant clan, allowing for a structured approach to feature optimization that mirrors the complex social behaviors observed in actual elephant populations. By leveraging this methodology, the study aims to enhance the detection and classification of software defects through a model inspired by the social dynamics of elephant herds.

$$C \leftarrow \text{Generate\_Herd}(m, n)$$

**Fitness** To effectively evaluate the fitness of each elephant within the context of the study, a comprehensive measure known as fitness is established. This fitness assessment is achieved by constructing a temporary deep learning model CNN that serves as a prototype for evaluating the defect detection capabilities. The model is meticulously trained using the selected features from the dataset, and once training is complete, it is utilized to perform defect detection tasks. During this evaluation process, the accuracy of the model is calculated, which is specifically focused on its ability to correctly identify and classify the defect classes present within the input data. The resulting accuracy value obtained from this analysis directly reflects the fitness of each elephant in the population. Thus, the fitness score serves as a crucial metric, indicating how well each elephant (representing a set of features) contributes to the overall performance of the defect detection model. A higher accuracy score denotes a better ability to detect defects accurately, thereby enhancing the fitness evaluation of that particular elephant within the population. The fitness assessment process not only quantifies the effectiveness of the elephants but also provides insight into the efficacy of the underlying deep learning model used for defect detection.

$$Ef \leftarrow \text{Fitness}(A) \text{ -----Eq. 3}$$

**Clan Update** A best solution matriarch,  $M$  is derived based on the fitness values of each elephant in clan in the population [7]. A number of the statuses were randomly changed based on the best matriarch,  $M$  feature set. The cloning is done by placing the best elephant set page in other elephant of clan.

$$C \leftarrow \text{Clan\_update}(M_b, C)$$

**Separating**

Low fitness elephant were removed from the clan in form of male elephant. This is done after estimating the new clan fitness value.

**Final Feature Set** After  $T$  number of iteration of algorithm final feature set was select. In the second phase of our proposed work, the NIDAIFL input raw dataset underwent processing and filtering using a specific elephant herd algorithm or a predetermined set of features. These extracted

features were then normalized and fed into a spiking deep trained model. This model determined whether the software session belonged to a defective or normal class.

**Convolution Neural Network** The main goal of CNN is to utilize structural information and reuse weight parameters. To achieve this goal, CNNs propose two new operations (i.e., the convo-lution operation and the pooling operation). The main idea of these two operations is to leverage the geometric transformation invariant property of the data [14, 15].

$$\text{Train\_data} \leftarrow \text{Transform\_Matrix}(Ef) \text{-----Eq. 7}$$

$$\text{Trained\_Moel} \leftarrow \text{Train\_CNN}(\text{Train\_data})$$

**Filter Feature**

Once the iterative process is complete, the best antibody is identified from the most recent population update. The features with a value of 1 in the chromosome of this best antibody are considered the selected features for the training vector, while those with a value of 0 are deemed unselected. In this stage, a desired output matrix is also prepared, which serves as the classification target for various session types, including normal, DoS (Denial of Service), U2R (User to Root), and R2L (Remote to Local) attacks. This matrix is used to classify the network sessions and enhance the model's training effectiveness.

$$\text{FID} \leftarrow \text{Filter\_Feature}(\text{PID}, \text{Ab})$$

Where FID is Filter Intrusion Dataset.

**IV. EXPERIMENT AND RESULTS**

The NIDAIFL model, along with the comparative models, was implemented using MATLAB software. The experiments were conducted on a machine equipped with 4 GB of RAM and an Intel i3 6th generation processor. The dataset for the input-output operations was sourced from reference [17]. The performance of the NIDAIFL model was compared against a network malicious session detection model described in reference G-CNN [9].

**Evaluation Parameters**

To evaluate the performance of the models, several metrics were utilized, including Precision, Recall, and F-score. These evaluation parameters are calculated based on the values of True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN).

These metrics provide a comprehensive assessment of the models' effectiveness in accurately detecting and classifying network sessions into the respective categories of intrusion or normal activity.

**Results**

Precision values of different network attacks detection models shown in table 1 . It was found that use of elephant herd optimization algorithm has improved the work efficiency. Use of deep learning model has increases the precision value by 25.52% as compared to existing model.

Table 1 Network attack detection based on Precision parameter.

Dataset	G-CNN	NIDAIFL
3000	0.6237	0.9733
6000	0.6662	0.9766
8000	0.7898	0.9756
9000	0.8114	0.9756
10000	0.8258	0.976
1200	0.6425	0.9764

Table 2 Network attack detection based on Recall parameter

Dataset	G-CNN	NIDAIFL
3000	0.6868	0.9763
6000	0.7561	0.9743
8000	0.8526	0.9702
9000	0.8118	0.9652
10000	0.7757	0.962
1200	0.6416	0.9766

Intrusion detection models recall values shown in table 2. Optimized features by elephant herd algorithm has increases the learning and detection accuracy of deep neural network. Feature reduction by dynamic algorithm has increases the work recall value by 22.31% as compared to G-CNN.

Table 3 Network attack detection based on F-measure parameter.

Dataset	G-CNN	NIDAIFL
3000	0.6537	0.9748
6000	0.7083	0.9754
8000	0.82	0.9729
9000	0.8116	0.9704
10000	0.7999	0.969
1200	0.642	0.9765

F-measure values of different network attacks detection models shown in table 1 . It was found that use of elephant herd optimization algorithm has improved the work efficiency. Use of deep learning model has increases the F-measure value by 24.03% as compared to existing model.

Table 4 Network attack detection based on Accuracy parameter.

Dataset	G-CNN	NIDAIFL
3000	50.58	96.23
6000	69.29	97.51
8000	76.82	97.93
9000	75.06	97.98
10000	73.2	98.07
1200	58.83	98

Intrusion detection models accuracy values shown in table 4. Optimized features by elephant herd algorithm has increases the learning and detection accuracy of deep neural network. Feature reduction by dynamic algorithm has increases the work accuracy value by 31.06% as compared to G-CNN.

### V. CONCLUSION

In conclusion, this study presents a novel intrusion detection model that effectively combines a convolutional neural network (CNN) with an Elephant Herd Optimization-based genetic algorithm for feature selection, demonstrating substantial improvements in accuracy, precision, recall, and F-measure over existing models. The model’s two-phase approach, involving feature ontology construction and training followed by testing, enhances its capability to accurately classify network sessions and detect various attacks. Experimental results across multiple datasets confirm that optimized feature selection using the Elephant Herd Optimization algorithm enhances model efficiency and precision, yielding a 25.52% improvement in precision, a 22.31% increase in recall, a 24.03% rise in F-measure, and a 31.06% boost in accuracy compared to the G-CNN model. These gains emphasize the effectiveness of dynamic feature reduction in enhancing deep learning-based intrusion detection models. Future work could explore additional optimizations and extend this approach to different network environments to further bolster network security and resilience against emerging cyber threats.

### REFERENCES

1. Fuhong Lin, Yutong Zhou, Xingsuo An, Ilsun You, Fair Resource Allocation in an Intrusion Detection System:

Ensuring the Security of Internet of Things Devices, IEEE conference on Consumer electronics Computing Magazine, Volume: 7, Issue: 6, ISSN: 2162-2248, publishedYear 2018.

2. Mohammad Saeid Mahdavejrad, Mohammadreza Rezvan, Mohammadamin Barekatin Peyman Adibi, Payam Barnaghi, Amit P Sheth Machine learning for internet of things data analysis: a survey Digital Communications and Networks Science Direct, Volume:4, Issue 3, Pages: 161- 175, published Year 2018.

3. Sai Kiran, K.V.V.N.L. R.N. Kamakshi Devisetty, N. Pavan Kalyan, K. Mukundini, R. Karthi. "Building a Intrusion Detection System for IoT Environment using Machine Learning Techniques". Procedia Computer Science, Volume 171, 2020, Pages 2372-2379.

4. Bacem Mbarek, Mouzhi Ge, and Tomás Pitner. 2020. Enhanced nnetwork intrusion detection system protocol for internet of things. In Proceedings of the 35th Annual ACM Symposium on Applied Computing (SAC '20). Association for Computing Machinery, New York, NY, USA, 1156–1163.

5. Sstla, V., Kolli, V.K.K., Voggu, L.K., Bhavanam, R., Vallabhasoyula, S. (2020). Predictive model for nnetwork intrusion detection system using deep learning. Revue d'Intelligence Artificielle, Vol. 34, No. 3, pp. 323-330.

6. Bahram Hajimirzaei and Nima Jafari Navimipour. 2019. Intrusion detection for computer network using neural nnetworks and artificial bee colony optimization algorithm. ICT Express 5, 1 (2019), 56–59.

7. E. Kabir, J. Hu, H. Wang, and G. Zhuo, “A novel statistical technique for intrusion detection systems,” Future Gener. Comput. Syst., vol. 79, pp. 303–318, Feb. 2018.

8. ChuanlongYin ,Yuefei Zhu, Jinlong Fei, And Xinzheng He. “A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks” current version November 7, 2017.

9. Kaiyuan Jiang ,Wenya Wang , Aili Wang , And Haibin Wu. "Network Intrusion Detection Combined Hybrid Sampling With Deep Hierarchical Network". IEEE Access February 24, 2020.

10. Khraisat, A., Gondal, I., Vamplew, P. et al. Survey of intrusion detection systems: techniques, datasets and challenges. Cybersecur 2, 20 (2019).

11. Zina Chkribene, AimanErbad, RidhaHamila, Amr Mohamed, Mohsen Guizani, And Mounir Hamdi. "TIDCS: A Dynamic Intrusion Detection and Classification System Based Feature Selection". Digital Object Identifier June 3, 2020.

12. R. Ben Said, Z. Sabir and I. Askerzade, "CNN-BiLSTM: A Hybrid Deep Learning Approach for Network Intrusion Detection System in Software-Defined Networking With

- Hybrid Feature Selection," in IEEE Access, vol. 11, pp. 138732-138747, 2023.
13. T. Kim and W. Pak, "Early Detection of Network Intrusions Using a GAN-Based One-Class Classifier," in IEEE Access, vol. 10, pp. 119357-119367, 2022.
  14. Ullah I., Mahmoud Q.H. (2020) A Scheme for Generating a Dataset for Anomalous Activity Detection in IoT Nnetworks. In: Goutte C., Zhu X. (eds) Advances in Artificial Intelligence. Canadian AI 2020.
  15. Y. K. Al-Douri, V. Pangracious and M. Al-Doori, "Artificial immune system using Genetic Algorithm and decision tree," 2016 International Conference on Bio-engineering for Smart Technologies (BioSMART), Dubai, United Arab Emirates, 2016, pp. 1-4,
  16. Al-Sharhan, Salah. (2010). ARTIFICIAL IMMUNE SYSTEMS – MODELS, ALGORITHMS AND APPLICATIONS. International Journal of Research and Reviews in Applied Sciences. 3.
  17. W. A. Nassan, T. Bonny, K. Obaideen and A. A. Hammal, "AN EHDL-IDS model-based Prediction of Chaotic System: Analyzing the Impact of Training Dataset Precision on the Performance," 2022 International Conference on Electrical and Computing Technologies and Applications (ICECTA), Ras Al Khaimah, United Arab Emirates, 2022, pp. 337-342
  18. <https://research.unsw.edu.au/projects/unsw-nb15-dataset>