*Nanotechnology Perceptions*
*ISSN 1660-6795*
*www.nano-ntp.com*

# Artificial Immune Based Dimension Reduction for Developing Intrusion Detection System

**Nuzhat Parveen[1],Onkar Nath Thakur[2], Rakesh Kumar Tiwari[3], Vikas Gupta[4]**

*1.nuzhat2dec@gmail.com[1], er.onkarthakur@gmail.com[2],rakeshktiwari80@gmail.com[3],*
*2.Department of Computer Science &Engineering[1,2&3],*
*3.vikasgupta.bhopal@gmail.com[4] Department of Electronics & Communication Engineering[4]*
*Technocrats Institute of Technology & Science, Bhopal, India*

***Abstract*-** The increasing reliance on technology for various tasks has led to a surge in computational demands, thereby significantly boosting the use of computer networks in recent decades. This increased need for computation and storage presents a lucrative business opportunity for many companies; however, it also attracts considerable attention from cyber attackers. To counter these threats, numerous researchers have developed various models aimed at detecting and preventing attacks. This paper proposes a novel intrusion detection model that operates in two phases: the first phase involves creating a feature ontology to train a LSTM model, while the second phase focuses on testing the trained LSTM model. For feature selection, the proposed model utilizes an Artificial Immune System-based genetic algorithm, which effectively identifies a robust set of features for classifying Nnetwork session types. The experiments were conducted using a real Nnetwork dataset, and the proposed model demonstrated the capability to detect multiple types of attacks within normal sessions. The results indicate that the proposed model improves accuracy and other performance metrics compared to existing models.

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

## I. INTRODUCTION

Computer networks are one of the most recent IT service developments. The main benefit of computer networks is that it allows access regardless of place or time. Computer networks allow the flexibility of adjusting storage capacities and lowers costs while supporting mobile and collaborative applications/services[1]. Furthermore, Nnetwork services are multisource, allowing end-users to choose from a variety of service providers based on their needs. Computer networks also save on-site storage costs, power consumption, physical space needs, and maintenance costs. As computer network services become more widely available, a growing number of businesses, banks, and governments are embracing the technology[2]. This change also exposed these systems to a wide range of intrusions from hackers and attackers, necessitating the implementation of rigorous security measures. As apps, several Nnetwork service providers offer a variety of security services. The Amazon Web Services (AWS) shop, for example, offers services with limited validity and dates based on the service license duration [3].

Traditional IDSs are not efficient enough to manage such a massive data flow since Nnetwork infrastructure has enormous Nnetwork traffic[4]. The majority of known IDSs are single threaded, but in the computer network context, multi-threaded IDSs are required due to the large amount of data flow. IDS monitors, detects, and alerts the administrative user for Nnetwork activity in a traditional Nnetwork by implementing IDS on important Nnetwork choke points on the user site. However, in a Nnetwork, IDS must be installed on the Nnetwork server

and operated solely by the service provider. If an attacker successfully penetrates and damages or steals a user's data in this case, the Nnetwork user will not be notified directly. The service provider will be the only one who receives the intrusion data, and the user will have to rely[5]. For the sake of his image and repute, the Nnetwork service provider may not want to inform the user about the loss and may choose to keep the information hidden. In this instance, a neutral third-party monitoring provider can provide acceptable Nnetwork user monitoring and alerts. This paper proposed a multi Nnetwork IDS that is administered and monitored by a third-party ID monitoring service that can provide Nnetwork users with alert reports and expert advice, as well as a third-party ID monitoring service that can provide alert reports and expert advice to network service providers [6]. An efficient and reliable distributed Nnetwork IDS approach is developed to address the challenges that traditional IDSs cannot.

## II. RELATED WORK

W. Zhong and N. Yu (2020), conducted an in-depth study on the use of big data-driven deep learning systems for intrusion detection. Their research explores the advancements in deep learning methodologies and the challenges involved in analyzing large-scale data to detect and mitigate intrusions across various network environments. The study provides a thorough examination of how deep learning techniques can be harnessed to improve the detection of sophisticated cyber threats by processing and interpreting vast amounts of data, which traditional methods might struggle to handle effectively [7]. Yi Lu, Menghan Liu, and Jie Zhou (2021), investigated a novel intrusion detection technique that combines adaptive clonal genetic algorithms with backpropagation neural networks. Their research delves into the development and application of this hybrid approach, emphasizing its effectiveness in identifying and mitigating network intrusions. The study also examines the adaptability and performance of this method in real-world scenarios, offering valuable insights into how this innovative approach can contribute to advancing cybersecurity. By integrating genetic algorithms with neural networks, the researchers provide a unique perspective on enhancing the robustness and efficiency of intrusion detection systems, particularly in dynamic and complex network environments [8]. In 2023, C. Park and colleagues developed an AI-based Network Intrusion Detection System (NIDS) to address data imbalance issues and improve system performance. They used Generative Adversarial Networks (GANs) to generate synthetic data for underrepresented attack types, focusing on a GAN architecture based on reconstruction error and Wasserstein distance. By combining this generative model with anomaly detection, the system outperformed previous methods in classification accuracy, enhancing its ability to detect a wider range of network attacks [9]. In another study, J. Liu and collaborators proposed an intrusion detection algorithm based on Particle Swarm Optimization (PSO) combined with the LightGBM framework, known as PSO-LightGBM[10]. This method focuses on optimizing data feature extraction, which is then fed into a one-class Support Vector Machine (OCSVM) for detecting and identifying harmful data. The authors validated their model using the UNSW-NB15 dataset, a widely recognized benchmark in intrusion detection research. However, the use of SVM in this method limits its capability to binary classification, distinguishing only between normal and harmful data. Yue Jin and co-authors proposed a house-level intrusion detection system utilizing WiFi-enabled IoT devices to detect intruders based on signal strength [11]. Their approach centers on the Received Signal Strength Indicator (RSSI) to create a detection router that incorporates an algorithm for identifying intruders and visualizes home security status via IoT. While this system provides an innovative method for home security, it heavily relies on equipment efficiency and a single parameter-signal strength. Additionally, the implementation was designed for static networks, where devices are fixed in place, and does not extend to mobile networks where devices operate as guests, limiting their applicability in more dynamic environments.

## III. METHODOLOGY

The proposed NIDAIFL (Network Intrusion Detection by Artificial Immune Feature Learning) 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 LSTM (Long Short-

Term Memory) network are undertaken. The second module focuses on the testing and evaluation of the trained LSTM 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←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 artificial immune system algorithm. This algorithm operates in two steps for population update: cloning and mutation, which modify the chromosome values in each iteration [13]. The random cloning and mutation steps increase the likelihood of achieving an optimal feature ontology solution, enhancing the model's ability to accurately identify different classes of intrusions.

**Generate Antibodies**

The process of generating antibodies begins with the creation of a random set of features using a Gaussian function, which produces a binary combination of 0s and 1s. In this context, each binary feature set represents an antibody in the genetic algorithm. The algorithm utilizes two flags to indicate the status of each feature: a value of 1 signifies the presence of the feature, while a value of 0 indicates its absence. Additionally, the population of antibodies is defined by a lower bound for the presence of a feature and an upper bound for its absence. Thus, a set of m antibodies population, denoted as AP, constitutes the initial population for the genetic algorithm.

**AP←Generate_Antibody(m, PID)**

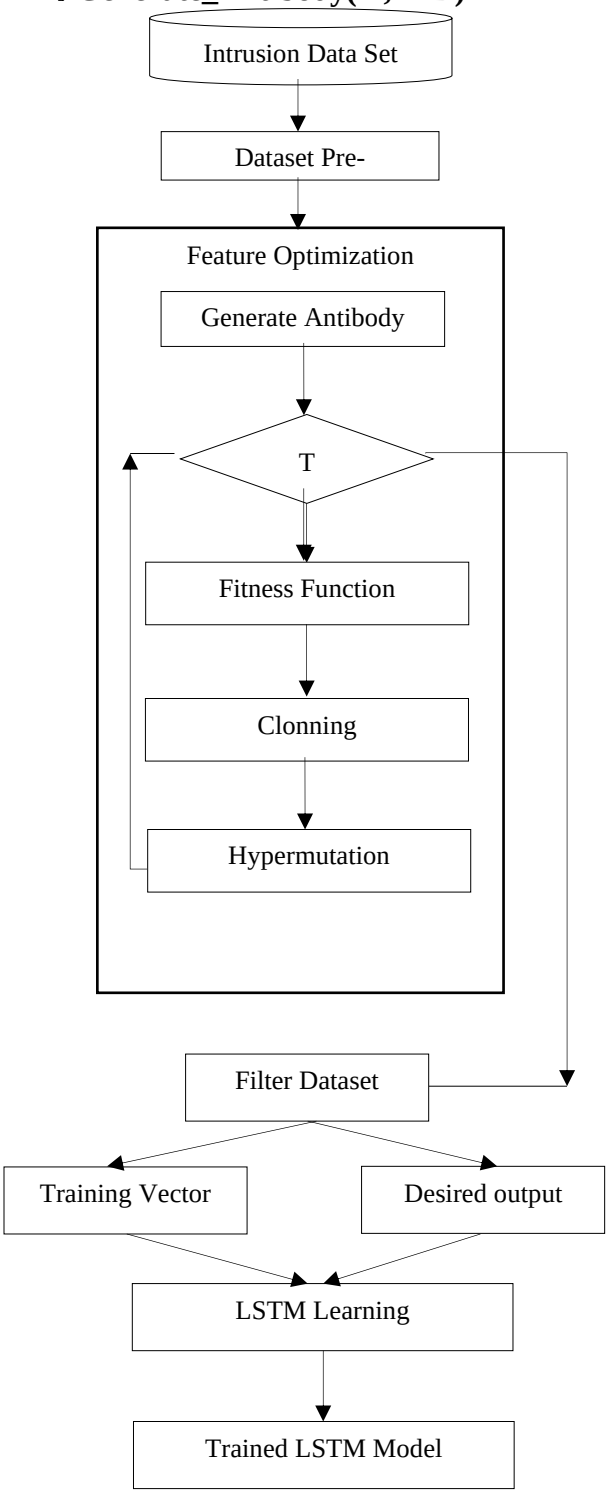


**Figure. 1** Block diagram of NIDAIFL.

**Affinity** The affinity of each antibody within the population is evaluated by constructing a temporary LSTM (Long Short-Term Memory) model based on the present features [14, 15]. The temporary LSTM model is then trained to detect intrusions, and its performance is assessed to determine the accuracy of intrusion class detection. The accuracy achieved by the model in correctly identifying intrusion classes represents the affinity of the

corresponding antibody. The higher the accuracy, the better the affinity of the antibody. The detailed training process for the LSTM model is described under the LSTM section of this paper.

**LSTM**

The LSTM model for intrusion detection is typically composed of several layers [16]:

**Input Layer:**This layer takes the preprocessed network data as input. The input features can include a range of attributes such as packet size, time intervals, protocol type, source and destination IP addresses, etc.

**Embedding Layer (Optional):**If the input data includes categorical features (e.g., protocol types, IP addresses), an embedding layer can be used to transform these features into continuous vectors that are easier for the LSTM to process.

**LSTM Layers:** One or more LSTM layers are used to learn the temporal patterns in the data. The number of LSTM layers and the number of neurons in each layer can be adjusted based on the complexity of the data and the desired level of model expressiveness.

**Dropout Layer:** To prevent overfitting, dropout layers are often added after the LSTM layers. Dropout layers randomly deactivate a fraction of neurons during training, which helps in generalizing the model better to unseen data.

**Cloning:**Based on the affinity values of each antibody in the population, the best solution, denoted as$Ab$, is identified. Once the best antibody $Ab$ is determined, some feature statuses within this set are randomly altered. This alteration involves changing the status of features from present (1) to absent (0) or vice versa. This random modification of the feature set results in the cloning of the model, allowing for the exploration of new potential solutions.

AP←Cloning($A_b$, AP)

**Hypermutation:** Following the cloning process, the generated clones are subjected to a hypermutation procedure. During hypermutation, the clones are mutated in inverse proportion to their affinity values. This means that clones derived from the best-performing antibodies undergo minimal mutation, while those from the worst-performing antibodies are subjected to more extensive mutations. This selective mutation process aims to refine the solution space by making slight adjustments to high-affinity antibodies and more significant changes to low-affinity ones. The type of mutation applied can vary, including uniform, Gaussian, or exponential mutations. After mutation, both the clones and their original antibodies are evaluated, and the top N antibodies are selected for the next iteration.

AP←Hypermutation(AP)

**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.

FID←Filter_Feature(PID, Ab)

Where FID is Filter Intrusion Dataset.

**Proposed Training Algorithm**

Input: ID // Intrusion Dataset
Output: NIDAIFL // LSTM Model

1. PID←Pre_Processing(ID) // PID Preprocessed Intrusion Dataset
2. **AP←Generate_Antibody(m, PID)**
3. Loop 1: its
4. I←Immunity(AP, PID)
5. Ab←Best(AP, I)
6. AP←Cloning($A_b$, AP)
7. AP←Hypermutation(AP)
8. EndLoop
9. I←Immunity(AP, PID)
10. Ab←Best(AP, I)
11. FID←Filter_Feature(PID, Ab)
12. LSTM←Intialize_LSTM()
13. NIDAIFL←Train(LSTM, FID, DO) // DO: Desired Output

## 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**

**Table. 1:** Comparison of precision values of Learning model of Nnetwork intrusion detection.

| Dataset | G-CNN | NIDAIFL |
|---|---|---|
| 4000 | 0.7948 | 0.9418 |
| 6000 | 0.7937 | 0.9419 |
| 8000 | 0.7933 | 0.9421 |
| 10000 | 0.7942 | 0.942 |
| 12000 | 0.7886 | 0.9347 |
| 14000 | 0.7852 | 0.9205 |

**Table. 1** show learning models precision values of intrusion detection, it was found that NIDAIFL has improved the precision value by 15.52%, as compared to G-CNN. Further it was found that use of artificial immune genetic algorithm has efficiently reduces the dataset dimension that increases the learning and detection outcomes.

**Table. 2** Comparison of recall values of Learning model of Nnetwork intrusion detection.

| Dataset | G-CNN | NIDAIFL |
|---|---|---|
| 4000 | 0.7637 | 0.9927 |
| 6000 | 0.5823 | 0.9927 |
| 8000 | 0.5697 | 0.9927 |
| 10000 | 0.5781 | 0.9928 |
| 12000 | 0.9424 | 0.9956 |
| 14000 | 0.8469 | 0.9968 |

**Table. 2** shows that proposed model has increases the recall values of IDS proposed in NIDAIFL. Use of LSTM model with filter feature set has performed well in the model. In the previous model G-CNN generative session increases the confusion and results in lower recall values.

**Table. 3** Comparison of f-measure values of Learning model of Nnetwork intrusion detection.

| Dataset | G-CNN | NIDAIFL |
|---|---|---|
| 4000 | 0.7789 | 0.9666 |
| 6000 | 0.6718 | 0.9666 |
| 8000 | 0.6632 | 0.9667 |
| 10000 | 0.6691 | 0.9667 |
| 12000 | 0.8587 | 0.9642 |
| 14000 | 0.8149 | 0.9572 |

**Table. 3** shows learning models f-measure values of intrusion detection, it was found that NIDAIFL has improved the f-measure value by 23%, as compared to G-CNN. Further it was found that use of artificial immune genetic algorithm has efficiently reduces the dataset dimension that increases the learning and detection outcomes.

**Table. 4** Comparison of accuracy values of Learning model of Nnetwork intrusion detection.

| Dataset | G-CNN | NIDAIFL |
|---|---|---|
| 4000 | 65.87 | 95.07 |
| 6000 | 60.69 | 96.7 |
| 8000 | 69.32 | 97.53 |
| 10000 | 75.71 | 98 |
| 12000 | 88.98 | 97.05 |
| 14000 | 85.75 | 95.82 |

**Table. 4** shows that proposed model has increases the accuracy values of IDS proposed in NIDAIFL. Use of LSTM model with filter feature set has performed well in the model. In the previous model G-CNN generative session increases the confusion and results in lower recall values.

## V. CONCLUSION

This paper has proposed NIDAIFL (Network Intrusion Detection by Artificial Immune Feature Learning) model is comprehensively outlined in this study, detailing its two primary modules: the training module and the testing module. The training module involves the creation of a feature ontology and the training of a Long Short-Term Memory (LSTM) network, while the testing module is dedicated to evaluating the performance of the trained LSTM model. The experimental results show that the NIDAIFL model significantly enhances precision in intrusion detection, achieving a higher precision rate than the G-CNN model. This improvement is attributed to

the use of an artificial immune genetic algorithm, which effectively reduces the dimensionality of the dataset, thereby improving both learning efficiency and detection accuracy. Further proposed model also increases accuracy of detection by 23.07%. In the future scholars can implement the same in under water network environment.

## REFERENCES

[1] F. Lin, Y. Zhou, X. An, I. You, K.-K.R. Choo, Fair resource allocation in an intrusion-detection system for edge computing: Ensuring the security of Internet of Things devices, IEEE Consumer Electronics Magazine, 7 (2018) 45-50.
[2] M.S. Mahdavinejad, M. Rezvan, M. Barekatain, P. Adibi, P. Barnaghi, A.P. Sheth, Machine learning for Internet of Things data analysis: A survey, Digital Communications and Networks, 4 (2018) 161-175.
[3] K.S. Kiran, R.K. Devisetty, N.P. Kalyan, K. Mukundini, R. Karthi, Building a intrusion detection system for IoT environment using machine learning techniques, Procedia Computer Science, 171 (2020) 2372-2379.
[4] B. Mbarek, M. Ge, T. Pitner, Enhanced network intrusion detection system protocol for internet of things, Proceedings of the 35th annual ACM symposium on applied computing, 2020, pp. 1156-1163.
[5] V.P.K. Sistla, V.K.K. Kolli, L.K. Voggu, R. Bhavanam, S. Vallabhasoyula, Predictive Model for Network Intrusion Detection System Using Deep Learning, Rev. d'Intelligence Artif., 34 (2020) 323-330.
[6] B. Hajimirzaei, N.J. Navimipour, Intrusion detection for cloud computing using neural networks and artificial bee colony optimization algorithm, Ict Express, 5 (2019) 56-59.
[7] J. Zhong, Z. Huang, L. Feng, W. Du, Y. Li, A hyper-heuristic framework for lifetime maximization in wireless sensor networks with a mobile sink, IEEE/CAA Journal of Automatica Sinica, 7 (2019) 223-236.
[8] Y. Lu, M. Liu, J. Zhou, Z. Li, [Retracted] Intrusion Detection Method Based on Adaptive Clonal Genetic Algorithm and Backpropagation Neural Network, Security and Communication Networks, 2021 (2021) 9938586.
[9] C. Park, J. Lee, Y. Kim, J.-G. Park, H. Kim, D. Hong, An enhanced AI-based network intrusion detection system using generative adversarial networks, IEEE Internet of Things Journal, 10 (2022) 2330-2345.
[10] J. Liu, D. Yang, M. Lian, M. Li, Research on intrusion detection based on particle swarm optimization in IoT, IEEE Access, 9 (2021) 38254-38268.
[11] Y. Jin, Z. Tian, M. Zhou, Z. Li, Z. Zhang, A whole-home level intrusion detection system using WiFi-enabled IoT, 2018 14th International Wireless Communications & Mobile Computing Conference (IWCMC), IEEE, 2018, pp. 494-499.
[12] K. Jiang, W. Wang, A. Wang, H. Wu, Network intrusion detection combined hybrid sampling with deep hierarchical network, IEEE access, 8 (2020) 32464-32476.
[13] I. Ullah, Q.H. Mahmoud, A scheme for generating a dataset for anomalous activity detection in iot networks, Canadian conference on artificial intelligence, Springer, 2020, pp. 508-520.
[14] Y.K. Al-Douri, V. Pangracious, M. Al-Doori, Artifical immune system using Genetic Algorithm and decision tree, 2016 International Conference on Bio-engineering for Smart Technologies (BioSMART), IEEE, 2016, pp. 1-4.
[15] J.R. Al-Enezi, M.F. Abbod, S. Alsharhan, Artificial immune systems-models, algorithms and applications, (2010).
[16] W. Al Nassan, T. Bonny, K. Obaideen, A.A. Hammal, An lstm 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), IEEE, 2022, pp. 337-342.
[17] https://research.unsw.edu.au/projects/unsw-nb15-dataset.