Jestr

Journal of Engineering Science and Technology Review 18 (2) (2025) 21-27

Research Article

JOURNAL OF
Engineering Science and Technology Review

www.jestr.org

Artificial Hummingbird Algorithm with Deep Variational Autoencoder Driven Intrusion Detection in Big Data Analytics Environment

V. S. Thiyagarajan[1,*], K. S. Shashikala[2], M. Pavithra[3] and R. M. Parivarthan[2]

[1]*Department of CSE, Karpaga Vinayaga College of Engineering and Technology, Tamilnadu, India*
[2]*Department of AI&ML, New Horizon college of Engineering, Bangalore, India*
[3]*Department of CS&BS, Panimalar Engineering College, Chennai, India*

Received 9 January 2025; Accepted 23 February 2025

Abstract

The exponential growth of data has significantly increased the importance of data safety and advanced analysis techniques in Big Data (BD) environments. Intrusion Detection Systems (IDS) play a critical role in monitoring and analyzing data to identify intrusions within networks or systems. However, the high volume, variability, and velocity of data generated in modern networks present challenges for traditional methods, leading to inefficiencies and complexities. To address these issues, BD techniques have been integrated with IDS to enhance efficiency and accuracy. This manuscript presents the Artificial Hummingbird Algorithm with Artificial Intelligence-Driven Intrusion Detection in Big Data Environment (AHAAI-IDBDE)**.** The AHAAI-IDBDE framework employs Feature Selection (FS) with hyperparameter tuning to optimize intrusion detection. The MapReduce framework is used to handle BD processing, and Z-score normalization scales the input data. To select an optimal set of features, the Binary Volleyball Premier League (BVPL) algorithm is utilized, while intrusions are detected using the Deep Variational Autoencoder (DVAE) model. Hyperparameter tuning is conducted using the Artificial Hummingbird Algorithm (AHA)**.** The performance of the AHAAI-IDBDE method has been evaluated using a benchmark IDS dataset. Experimental results demonstrate that the proposed method achieves superior accuracy and efficiency compared to existing systems, as evidenced by extensive comparative analysis.

Keywords: Big Data, IDS, Binary Volleyball Premier League, MapReduce, Artificial Intelligence, Hyperparameter Tuning

1. Introduction

Big Data (BD) presents significant challenges in terms of management, storage, and analysis when employing conventional software and database techniques [1]. BD is characterized by its high velocity, volume, and variety of data, necessitating innovative methodologies for effective management. An Intrusion Detection System (IDS) serves as a software or hardware monitoring tool that scrutinizes data to detect any potential attacks targeting a system. Conventional IDS models tend to complicate networks and diminish effectiveness in handling BD due to the intricate and time-consuming nature of their analysis processes [2]. Consequently, the system remains susceptible to threats for prolonged durations prior to receiving any notifications [3]. Thus, leveraging BD methodologies for the evaluation and storage of data within IDS can significantly mitigate training and computational durations [4].

The IDS comprises two principal detection methodologies: Signature-based IDS, which identifies malicious activities through known signatures stored in a database, and anomaly-based IDS, which discerns atypical behavior within the network [5]. Signature-based IDS have been deemed ineffective in contemporary contexts for two fundamental reasons. Firstly, they necessitate prior knowledge of attacks, rendering them incapable of recognizing novel threats [6]. Secondly, the upkeep of attack signatures within databases and the execution of computations on IoT systems possessing limited storage and processing capacity proves to be inefficient [7]. Conversely, anomaly-based IDS excel in identifying unusual behaviors, thereby demonstrating proficiency in detecting novel attacks that deviate from established patterns [8]. With the recent advancements in Machine Learning (ML) and Deep Learning (DL) methodologies, these techniques can be integrated into anomaly-based IDS to surmount existing limitations. Numerous researchers have formulated ML methodologies aimed at minimizing false positive rates and establishing precise IDS [9]. Nonetheless, when engaging with BD, ML models necessitate extensive training durations for data classification. The integration of BD and ML techniques within IDS can alleviate challenges concerning computational inefficiency while enhancing both detection accuracy and speed [10].

This manuscript introduces an Artificial Hummingbird Algorithm with Artificial Intelligence-Driven Intrusion Detection in Big Data Environment (AHAAI-IDBDE) technique. The objective of the AHAAI-IDBDE technique is to harness Feature Selection (FS) in conjunction with a hyperparameter selection strategy for the purpose of intrusion detection. To manage BD effectively, the MapReduce framework is employed. In the AHAAI-IDBDE technique, Z-score normalization is utilized to standardize the input data. For the selection of an optimal feature set, the Binary Volleyball Premier League (BVPL) algorithm is implemented. The AHAAI-IDBDE technique employs a Deep Variational Autoencoder (DVAE) model to facilitate intrusion detection. The Artificial Hummingbird Algorithm (AHA) is utilized in the hyperparameter selection process. The efficacy of the AHAAI-IDBDE methodology has been empirically evaluated using a benchmark IDS dataset. The

*E-mail address: thiyagarajan@kveg.in
ISSN: 1791-2377
doi:10.25103/jestr.182.03

principal contributions of this paper are summarized as follows:

- Develop an automated AHAAI-IDBDE technique for intrusion detection in the big data environment by the use of FS with a hyperparameter selection strategy
- Employ BVPL technique for electing an optimal set of features. This approach can enhance the efficiency and relevance of the features selected for intrusion detection.
- Apply DVAE model for detecting intrusions. This indicates a utilization of advanced neural network architectures for more accurate and sophisticated intrusion detection.
- Utilize AHA for the hyperparameter selection process. This suggests an innovative approach to fine-tune the parameters of the intrusion detection system.

2. Literature

Krishna and Thirumuru [11] introduce an efficient ensemble DL-based IDS. The data preprocessing includes transforming qualitative data into numeric data employing the One-Hot Encoding method. Then, the process of normalization was executed and Manta-Ray Foraging Optimization was recommended to elect the best feature subsets. Afterwards, SMOTE oversampling develops a novel minority instance for balancing the processed database. Lastly, CNN–SVM method was developed for classifying the types of attacks. Mahdavisharif et al. [12] implement a BD-aware DL technique to develop a proficient and efficacious IDS model. A particular framework of Long Short-Term Memory (LSTM) could be created, and the architecture will recognize complex networks and extended dependencies among received traffic packets. Furthermore, employing BD analytic methods should increase the speed of DL methods in this study. Pustokhina et al. [13] introduced an innovative DL based hyperparameter search (HPS) CNN with BiLSTM (CBL) method named HPS-CBL for IDS in BD platform. The developed method employs improved-GA (IGA) for hyperparameter tuning.

Salama and Ragab [14] developed an innovative BC with Explainable AI Driven Intrusion Detection for IoT Driven Ubiquitous Computing System (BXAI-IDCUCS) technique. Moreover, the DNN method was utilized to recognize and categorize intrusions. Finally, BC technology must be implemented to protect inter-cluster data transmission methods. Kumar [15] introduced a Hybrid Meta-heuristic Optimization based Subset of FS (HMOFS) with Optimum Wavelet KELM (OWKELM) based Classification method named HMOFS-OWKELM system. The Hadoop Ecosystem was employed for handling BD. The HC notion was integrated into the MFO technique. Further, OWEKM method was implemented for classification method in which the optimum parameter setting in the WKELM was executed through the rat swarm optimizer (RSO).

Ponmalar and Dhanakoti [16] provided an innovative method for increasing the intrusion detection method by managing the essential BD difficulties related to various categories of heterogeneous security information. In order to accomplish the previous objective, the ensemble SVM has been incorporated with the Chaos Game Optimization (CGO) model. This technique enhances the classification accuracy of intrusion and likewise recognizes 9 various kinds of attacks existing in the database. Sheeba et al. [17] devise BD Analytics with the IoTs based Intrusion Detection utilizing Modified Buffalo Optimizer Algorithm with DL (IDMBOA-DL) method. The Hadoop MapReduce tool was implemented to handle BD. The MBOA technique was exploited for developing the optimum features by choosing the best group of feature subsets. Lastly, the SCA with CAE system should be employed for recognizing and categorizing the intrusions in the IoT network.

3. The Proposed Method

In this manuscript, we focus on design and development of an AHAAI-IDBDE approach. The purpose of the AHAAI-IDBDE technique exploit the FS with hyperparameter selection strategy for intrusion detection. Fig. 1 illustrates the entire process of AHAAI-IDBDE approach.

3.1. MapReduce

The Hadoop ecosystem provides platforms to access, store, and analyze vast amounts of data efficiently [18]. The MapReduce framework is a core component of Hadoop, designed for data parallel processing. It operates as a parallel processing programming model that simplifies distributed computation. A typical MapReduce instance comprises two stages, the Mapper and the Reducer. The output of the Mapper stage serves as the input for the Reducer stage. In the Mapper stage, the extracted data is transformed into key-value pairs, which are then passed to the MapReduce framework. The Mapper stage's primary function is to process input data by extracting specific field values and identifying any values that are no longer part of the dataset. This stage generates key-value pairs by processing documents both sequentially and in parallel, as represented in Eq. (1):

$$Map_{Phase}(key_1, value_1) \rightarrow list(key_2, value_2) \quad (1)$$

In the Reducer stage, the intermediate key-value pairs produced by the Mapper are aggregated and processed to produce the final output. The Reducer combines all intermediate values associated with the same key, resulting in the consolidated output, as described in Eq. (2):

$$Reduce\_Phase(key2, list(value2)) \rightarrow list(key\_3, value\_3) \quad (2)$$

After deduction by MapReduce, the huge dataset is small and acts as a requirement for further process.

3.2. Z-score normalization

The AHAAI-IDBDE methodology employs Z-score normalization to appropriately scale the input variables. Z-score normalization, commonly known as standardization, constitutes a statistical approach utilized to convert numerical data into a standardized metric characterized by a mean of 0 and a standard deviation (SD) of 1 [19]. In the realm of intrusion detection, Z-score normalization guarantees that all features within a dataset adhere to a uniform scale, thus facilitating precise interpretation and comparison of their respective values. This procedure entails subtracting the mean of each feature from the corresponding data points and subsequently dividing the resultant by the standard deviation. The resultant Z-scores signify the extent to which a data point deviates from the mean in terms of standard deviations. Z-score normalization is of particular significance in machine learning contexts, encompassing intrusion detection, as it promotes efficient model convergence and mitigates the risk

of features with larger scales disproportionately affecting model performance.

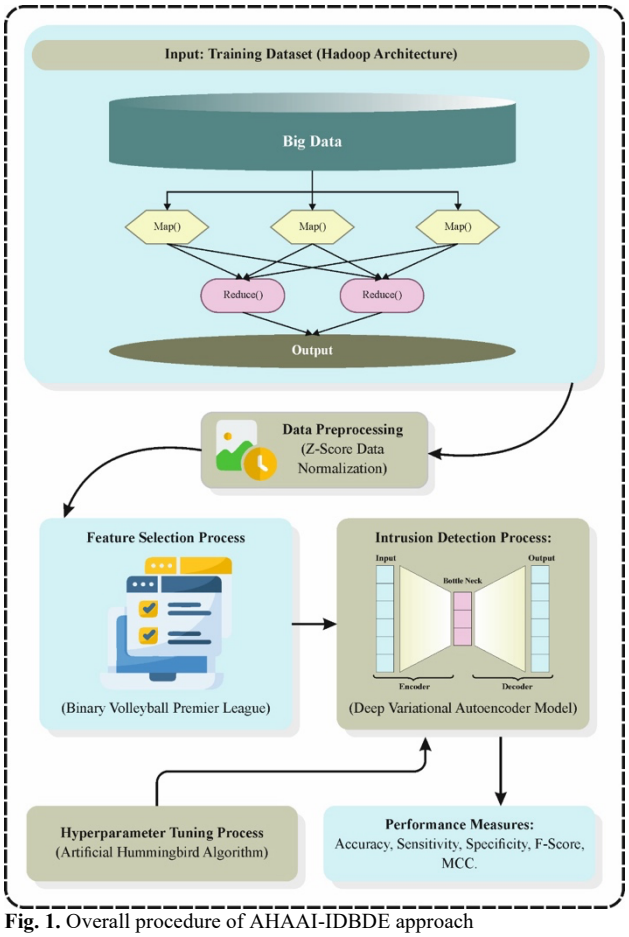

Fig. 1. Overall procedure of AHAAI-IDBDE approach

3.3. Feature selection

For the purpose of determining an optimal set of features, the Binary Volleyball Premier League (BVPL) methodology has been employed. This innovative approach, originally conceptualized by Moghdani and Salimifard [20], draws its inspiration from the organizational structure of a volleyball league. Within the BVPL paradigm, "active players" are indicative of features present in the dataset, whereas "passive players" denote alternative features that have the potential to enhance the overall performance metrics. The procedure comprises multiple phases. Initially, two matrices—substitution and formation—are assigned values in a randomized manner. These matrices correspond to the quantity of features and the dimensions of the team, respectively. Subsequent phases entail the formulation of a schedule for the feature selection procedure and the categorization of features into distinct groups predicated on their relevance to the designated task. Teams (features) are subsequently assessed through four methodologies: substitution, knowledge exchange, repositioning, and a tactical approach aimed at preserving the functionality of high-performing features. The learning phase entails the modification of features based on those demonstrating superior performance. Ultimately, the technique culminates in additional phases aimed at enhancing performance: relegation (elimination of less pertinent features), upgrades (integration of new features), and season transfers (modifications predicated on feature relevance). The BVPL approach is executed in a binary format through the selection of an appropriate transfer function tailored to each dataset. A cost function is utilized to evaluate the quality of the feature set, grounded in the accuracy metrics derived from the KNN machine learning algorithm, which serves to quantify the efficacy of the selected features within classification endeavors.

Algorithm 1: BVPL Algorithm
Input: $t = 0$, parameters, cost function
Output: mean, and SD of fitness, average of elected features, average accuracy
Initialization
For n runs = 1 to n runs
$t = 1$;
While $t <$ max_iteration
Create a league schedule
For $i = 1: (N - 1)$
Best team=Choose best team based on *costfunction*
For (all the matches in schedule week table *i*)
Execute Competition process among team *A*, and *B*
Define losing and winning teams
Implement distinct approaches for losing and winning teams
Upgrade Best team
Execute learning stage
End For
$i = i + 1$
End For
Execute for promotion and relegation procedure
Implement season transfer method
$t = t + 1$
End While
End For

The fitness function (FF) used in the BVPL system is designed to balance the number of selected features (lower count) and the classification accuracy (higher accuracy) achieved by deploying these selected features. Equation (3) represents the FF used to evaluate performance:

$$Fitness = \alpha\gamma_R(D) + \beta\frac{|R|}{|C|} \tag{3}$$

Here, γR(D) represents the error rate of the classifier, |R| denotes the cardinality of the selected feature subset, and |C| indicates the total number of features in the dataset. The parameters α and β correspond to the influence of classification accuracy and subset size, respectively, where α, β ∈ [0,1] and β -1−α

3.4. DVAE based classification

To detect intrusions, the AHAAI-IDBDE technique employs the DVAE model, which is a variant of the autoencoder (AE) [21]. The primary difference between AE and DVAE lies in the hidden representation (z) of the VAE, which is assumed to follow a Gaussian distribution parameterized by standard deviation (σ) and mean (μ). This distribution is encoded by qϕ(z|x) and decoded by pθ(x|z). Thus, the loss function for a data point $x^{(i)}$ in a VAE consists of two terms, as shown in Eq. (4):

$$\ell VAE\left(x^{(i)}, \theta, \phi\right) = D_{KL}\left(q_\phi\left(z|x^{(i)}\right)\middle|p_\theta(z)\right) - E_{q\phi\left(z|x^{(i)}\right)}\left[\log p_\theta\left(x^{(i)}|z\right)\right] \tag{4}$$

The 1st term in Eq. (4) is KL divergence among the estimated posterior $\left(q_\phi\left(z|x^{(i)}\right)\right)$ and the previous

distribution $\left(p_\theta(z)\right)$. This divergence processes that near the latter is to previous. The 2nd term $-E_{q\phi\left(z|x^{(i)}\right)}\left[logp_\theta\left(x^{(i)}|z\right)\right]$ is reconstruction error (RE) of DVAE. This term compels the decoded to engage in the learning to recreate the input data.

Since it can be challenging to directly sample from qϕ(z|x), a reparameterization trick is employed to handle high variance during the Monte Carlo process. Instead of directly sampling from the distribution, the reparameterization trick generates a sample $z^{(i)}$ from a standard normal distribution using the following equation:

$$z^{(i,k)} = \mu^{(i)} + \sigma^{(i)} \times \epsilon^{(i,k)}; \epsilon^{(i,k)} \sim N(0, I), \quad (5)$$

Whereas, $\sigma^{(i)}$ and $\mu^{(i)}$ refers to the SD and mean of the Gaussian distribution of individual hidden variable $z^{(i)}$, correspondingly. The values of $\mu^{(i)}$ and $\sigma^{(i)}$ are acquired through the encoded by utilizing functions $\mu^{(i)} = f_1(x^i, \phi)$ and $\sigma^{(i)} = f_2\left(x^{(i)}, \phi\right)$.

3.5. Hyperparameter tuning AHA model

The hyperparameter selection process is performed using the Adaptive Hummingbird Algorithm (AHA), a novel metaheuristic method inspired by the natural foraging behavior of hummingbirds [22]. The AHA model employs three distinct foraging mechanisms: omnidirectional, axial, and diagonal flight patterns. The steps involved in the AHA process are illustrated in Fig. 2**.**

In the first step, the initial population of N hummingbirds, represented by the vector X, is randomly initialized as follows:

$$X_j = r \times (U - L) + L, j = 1,2, \dots, N \quad (6)$$

Where U and L are the limits of each value of X_j and $r \in [0,1]$ indicates the random integers. Meanwhile, the visit table related to the X_b is shown below:

$$VT_{ji} = \begin{cases} 0 & if\ j \neq i \\ null & j = i \end{cases} \quad i, j = 1, \dots, N \quad (7)$$

If $i = j$, then $VT_{ji} = null$ indicates the amount of food hummingbirds found at a certain position. The j^{th} hummingbirds for visiting the i^{th} food source is represented as $VT_{ji} = 0$.

Guided foraging

During the foraging process, the hummingbird discovers the food source with the highest visiting rate and selects the agent that has the maximum nectar-refilling rate from the population X, representing the best agent for guided foraging. The three types of flight behaviors involved in this process are axial, omnidirectional, and diagonal. The description of axial flight $(D_i, i = 1, \dots, d)$ is shown below:

$$D_i = \begin{cases} 1 & if\ i = R \\ 0 & else \end{cases} \quad (8)$$

Where R∈[1,d] represents a randomly chosen dimension. Furthermore, the diagonal flight is represented as follows:

$$D_i = \begin{cases} 1\ if\ i = P(j), j \in [1, k],\ P = randperm\ (k) \\ 0 \qquad\qquad else \end{cases} \quad (9)$$

Where $randperm\ (k) \in [1, k]$ denotes the random integer and $k \in [2, \lceil r_1(d-2) \rceil + 1]$. Moreover, the formula of omnidirectional flight is represented as:

$$D_i = 1, i = 1, \dots, d \quad (10)$$

Where $R \in [1, d]$ shows the random number and $r_1 \in [0,1]$ refers to random integers. Moreover, based on guided foraging, the solution is updated using Eq. (11):

$$V_i(t+1) = X_i(t) + a \times D \times \left(X_i(t) - X_i(t)\right) \quad (11)$$

Where the i^{th} values at t^{th} iteration of X is denoted as $X_i(t)$. $a \in N(O, 1)$ represents random value. The desired solution explored by X_i is denoted as $X_i(t)$:

$$X_i(t+1) = \begin{cases} X_i(t),\ if\ f\left(X_i(t)\right) \leq f\left(V_i(t+1)\right) \\ V_i(t+1), \quad otherwise, \end{cases} \quad (12)$$

Where f indicates the fitness value.

Territorial foraging

A hummingbird searches for food instead of visiting other flowers, after consuming nectar from the flower. Thus, the bird makes a rapid migration towards the region closer to its territory where the new food source might be placed rather than the older one. The local foraging of hummingbirds and a potential solution can be expressed as follows:

$$V_i(t+1) = b \times D \times X_i(t) + X_i(t), b \in N(0, 1) \quad (13)$$

Migration foraging

The hummingbird migrates towards the new spot away from its preferred feeding position and is in need of food. The hummingbird step migrates to a location with one of the worst rates of nectar refill chosen at random from the search area when the generation number surpasses the predefined coefficient of migration. Thus, VT is improved as this hummingbird shifts from using the older to the newer solutions. The hummingbird migrates to a new nectar source established randomly from the source with a low nectar replenishment rate.

$$X_w(t+1) = r \times (U - L) + L. \quad (14)$$

Where X_w indicates the worst fitness value. The AHA technique derives an FF to realize better classifier result. It explains a positive integer to refer the good result of candidate outcomes. During this case, the decreasing the classifier rate of errors have been supposed to be FF, as demonstrated in Eq. (15).

$$fitness(x_i) = ClassifierErrorRate(x_i) = \frac{No.of\ misclassified\ instances}{Total\ no.of\ instances} * 100 \quad (15)$$

Where x_i represents the i^{th} candidate solution,

4. Result Analysis

In this section, the simulation outcome of the AHAAI-IDBDE methodology has been tested utilizing the IDS database [23], encompassing 125973 instances as illustrated in Table 1.

Table 1. Details on database

Attack Type	No. of Instances
Dos	45927
R2l	995
Probe	11656
U2r	52
Normal	67343
Total No. of Instances	**125973**

Table 2 and Fig. 3 illustrate the feature selection (FS) analysis of the AHAAI-IDBDE algorithm in terms of Best Cost (BC) and the selected features. The results indicate that the BSO-FS, WOA-FS, and GA-FS methods exhibit suboptimal performance, reflected by an increased BC. In contrast, the QBSO-FS model achieves a noteworthy BC of 0.000665. However, the AHAAI-IDBDE technique demonstrates superior performance, achieving a BC of 0.000427, which represents the best result among the evaluated methods.

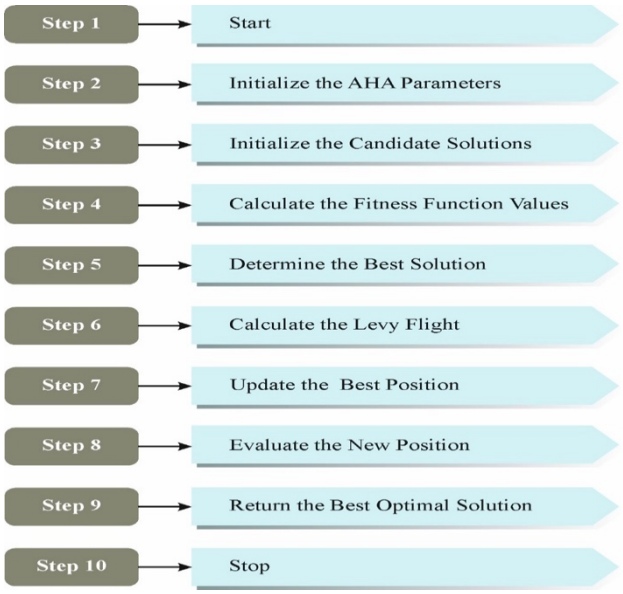

Fig. 2. Steps involved in AHA

Table 2. FS outcome of AHAAI-IDBDE technique under BC

Methods	Best Cost	Selected Features
AHAAI-IDBDE	0.000427	1,3,4,6,9,11,15,17,20,22,30,38
QBSO-FS	0.000665	2,3,5,6,7,8,9,11,14,16,18,32,36,39
BSO-FS	0.000673	2,4,5,6,7,9,11,13,15,16,17,19,20,21,23,38,38,40
WOA-FS	0.000840	3,5,8,13,18,20,21,22,23,25,26,28,30,32,33,34,36,38,40
GA-FS	0.001091	21,7,27,32,25,34,1,24,40,28,26,10,5,33,14,16,12,36,23,30,38,22,15

Fig. 4 represents the classifier performances of the AHAAI-IDBDE algorithm on test database. Figs. 4a-4b signifies the confusion matrices achieved by the AHAAI-IDBDE method on 70:30 of TRPH/TSPH. The simulation value referred that the AHAAI-IDBDE technique has detection and classified all 5 classes. Then, Fig. 4c exposes the PR study of the AHAAI-IDBDE methodology. The experimental value inferred that the AHAAI-IDBDE approach has gained better value of PR at 5 class. However, Fig. 4d exposes the ROC outcome of the AHAAI-IDBDE technique. The experimental value described that the AHAAI-IDBDE methodology has managed to capable performances with better values of ROC at 5 class.

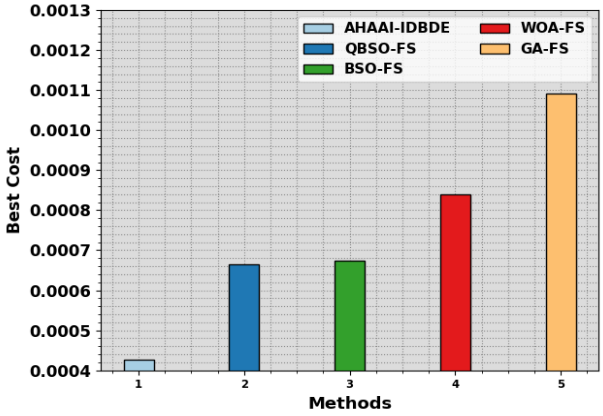

Fig. 3. FS outcome of AHAAI-IDBDE technique under BC

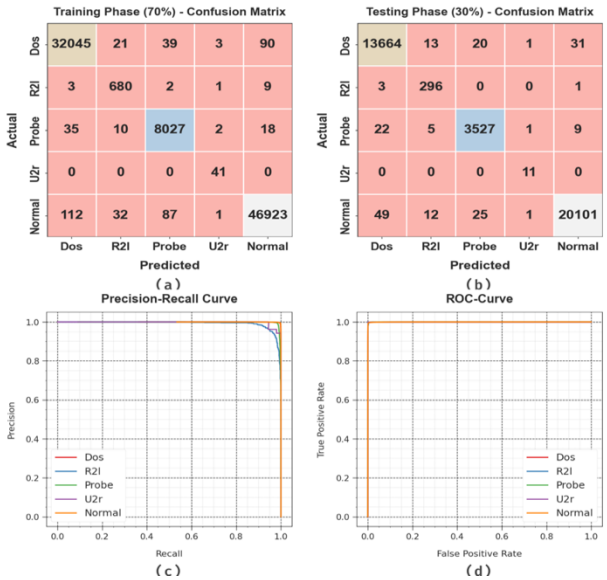

Fig. 4. Classifier outcome of (a-b) Confusion matrices and (c-d) PR and ROC curves

The IDS outcome of the AHAAI-IDBDE technique are displayed in Table 3 and Fig. 5. The results emphasized that the AHAAI-IDBDE methodology recognized five classes. With 70% of TRPH, the AHAAI-IDBDE method provides average $accu_y$ of 99.79%, $sens_y$ of 99.21%, $spec_y$ of 99.84%, F_{score} of 96.94%, and MCC of 96.83%. Additionally, with 30% of TSPH, the AHAAI-IDBDE system gains average $accu_y$ of 99.80%, $sens_y$ of 99.34%, $spec_y$ of 99.85%, F_{score} of 96.12%, and MCC of 96.10%.

The $accu_y$ curves for TRaining (TR) and VaLidation (VL) exposed in Fig. 6 for the AHAAI-IDBDE methodology suggest appreciated perceptions into its solution at distinct epochs. Specifically, there is a constant enhancement in both TR and TS $accu_y$ with enhanced epochs, implying the model's ability in learning and identifying designs in the both datasets. The increasing trend in TS $accu_y$ emphasizes the model's efficiency to the TR data and its capability to create correct estimates on unobserved data, importance robust generalization abilities.

Table 3. IDS outcome of AHAAI-IDBDE technique on 70:30 of TRPH/TSPH

Classes	$Accu_y$	$Sens_y$	$Spec_y$	F_{Score}	MCC
TRPH (70%)					
Dos	99.66	99.52	99.73	99.53	99.26
R2l	99.91	97.84	99.93	94.58	94.58
Probe	99.78	99.20	99.84	98.81	98.69

U2r	99.99	100.00	99.99	92.13	92.42
Normal	99.60	99.51	99.71	99.63	99.21
Average	**99.79**	**99.21**	**99.84**	**96.94**	**96.83**
TSPH (30%)					
Dos	99.63	99.53	99.69	99.49	99.21
R2l	99.91	98.67	99.92	94.57	94.61
Probe	99.78	98.96	99.87	98.85	98.73
U2r	99.99	100.00	99.99	88.00	88.64
Normal	99.66	99.57	99.77	99.68	99.32
Average	**99.80**	**99.34**	**99.85**	**96.12**	**96.10**

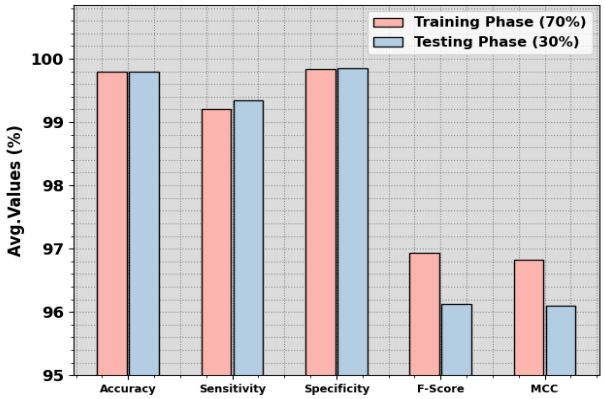

Fig. 5. Average of AHAAI-IDBDE technique on 70:30 of TRPH/TSPH

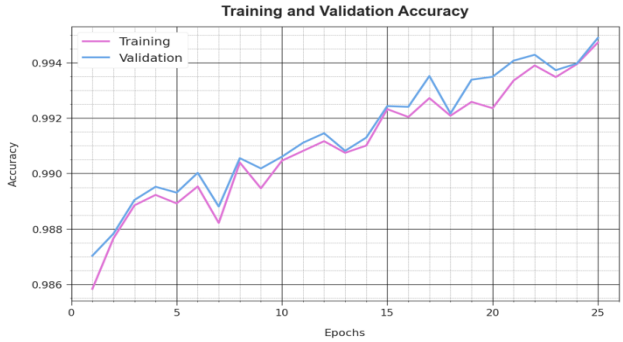

Fig. 6. $Accu_y$ curve of the AHAAI-IDBDE technique

Fig. 7 offers a wide-ranging analysis of the TR and TS loss values for the AHAAI-IDBDE system through several epochs. The TR loss constantly diminishes as the model upgrades its weights to decrease classifier errors on either TR or TS data. The loss curves evidently expose the model's position with TR data, underscoring its capability to capture outlines efficiently in both TR and TS data. Notable is the constant refinement of parameters in the AHAAI-IDBDE methodology, aimed at decreasing discrepancies among calculations and actual TR labels.

Table 4 demonstrates a brief comparative outcome of the AHAAI-IDBDE technique in terms of distinct metrics [24]. In Fig. 8, the AHAAI-IDBDE algorithm is compared with existing systems in terms of $sens_y$ and $spec_y$. Based on $sens_y$, the AHAAI-IDBDE method reaches higher $sens_y$ of 99.34% while the QBSO-FDNN, RBFNetwork, LR, RF, RT, and DT methods provide decreased $sens_y$ of 98.89%, 93.40%, 97.26%, 92.39%, 95.68%, and 95.68%, correspondingly. Eventually, based on $spec_y$, the AHAAI-IDBDE algorithm obtains maximum $spec_y$ of 99.85% while the QBSO-FDNN, RBFNetwork, LR, RF, RT, and DT methodologies provide lesser $spec_y$ of 99.42%, 92.38%, 96.92%, 93.83%, 95.39%, and 95.37%, correspondingly.

Fig. 7. Loss curve of the AHAAI-IDBDE technique

Table 4. Comparative analysis of AHAAI-IDBDE technique with other systems

Methods	$Sens_y$	$Spec_y$	$Accu_y$
AHAAI-IDBDE	99.34	99.85	99.80
QBSO-FDNN	98.89	99.42	98.90
RBFNetwork	93.40	92.38	92.93
Logistic Regression	97.26	96.92	97.10
Random Forest	92.39	93.83	93.04
Random Tree	95.68	95.39	95.55
Decision Tree (J48)	95.68	95.37	95.53

In Fig. 9, the AHAAI-IDBDE technique is compared with existing approaches in terms of $accu_y$. The results imply that the AHAAI-IDBDE technique reaches enhanced $accu_y$ of 99.80% while the QBSO-FDNN, RBFNetwork, LR, RF, RT, and DT models provide decreased $accu_y$ of 98.90%, 92.93%, 97.10%, 93.04%, 95.55%, and 95.53%, respectively.

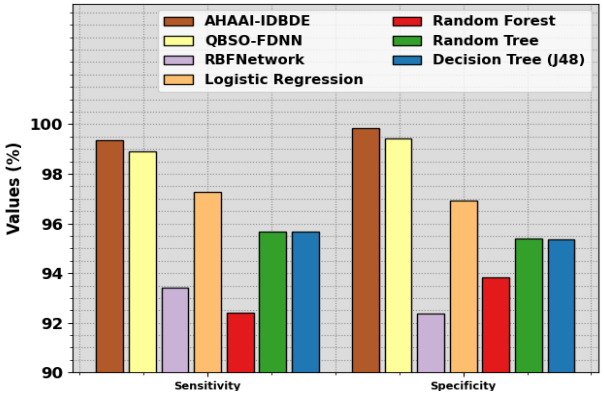

Fig. 8. $Sens_y$ and $spec_y$ analysis of AHAAI-IDBDE methodology with other models

Therefore, the AHAAI-IDBDE methodology has been applied for accurate classification of intrusions in the BD environment.

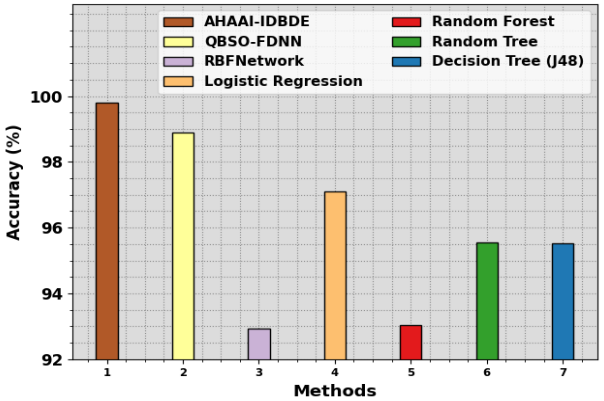

Fig. 9. $Accu_y$ outcome of AHAAI-IDBDE methodology with other models

5. Conclusion

In this manuscript, we present the design and development of the AHAAI-IDBDE approach for intrusion detection. The AHAAI-IDBDE technique combines feature selection (FS) with a hyperparameter selection strategy to enhance the effectiveness of intrusion detection systems (IDS). For efficient handling of big data (BD), the MapReduce framework is employed. Z-score normalization is utilized to scale the input data, ensuring that all features contribute equally to the detection process. To select the optimal set of features, the BVPL (Binary Variable Processing Layer) method is employed. Intrusion detection is carried out using the DVAE (Denoising Variational Autoencoder) model, which has demonstrated its capability in detecting anomalies in complex network traffic. The AHA (Adaptive Hyperparameter Algorithm) is applied for hyperparameter tuning, optimizing the performance of the model. The effectiveness of the AHAAI-IDBDE approach is evaluated through extensive experiments using benchmark IDS datasets. Comparative analysis with existing algorithms highlights the superiority of the AHAAI-IDBDE technique, demonstrating improved accuracy and efficiency in detecting intrusions. The results validate the potential of AHAAI-IDBDE for real-time, large-scale intrusion detection applications.

References

[1] G. Donkal and G. K. Verma, “A multimodal fusion based framework to reinforce IDS for securing Big Data environment using Spark,” *J. Inform. Secur. Applic.*, vol. 43, pp. 1–11, Dec. 2018, doi: https://doi.org/10.1016/j.jisa.2018.10.001.

[2] F. Karataş and S. A. Korkmaz, “Big Data: Controlling Fraud by Using Machine Learning Libraries on Spark,” *Int. J. Appl. Mathem. Electron. Comp.*, vol. 6, no. 1, pp. 1–5, Mar. 2018, doi: https://doi.org/10.18100/ijamec.2018138629.

[3] D. S. Terzi, R. Terzi, and S. Sagiroglu, “Big data analytics for network anomaly detection from netflow data,” in *2017 Int. Conf. Comp. Sci. Eng. (UBMK), Antalya, Turkey,* 2017, pp. 592–597. doi: 10.1109/UBMK.2017.8093473.

[4] K. Peng, V. C. M. Leung, and Q. Huang, “Clustering Approach Based on Mini Batch Kmeans for Intrusion Detection System over Big Data,” *IEEE Acc.*, vol. 6, pp. 11897–11906, 2018, doi: https://doi.org/10.1109/access.2018.2810267.

[5] D. Dietrich, B. Heller, B. Yang, and EMC Education Services, Eds., “*Data science & big data analytics: discovering, analyzing, visualizing and presenting data.”*, Indianapolis, IN: Wiley, 2015.

[6] K. Vimalkumar and N. Radhika, "A big data framework for intrusion detection in smart grids using apache spark,” in *2017 Int. Conf. Adv. Comp., Commun. Inform. (ICACCI), Udupi, India*, pp. 198–204.doi: 10.1109/ICACCI.2017.8125840.

[7] J. Á. González Ordiano *et al.*, “Concept and benchmark results for Big Data energy forecasting based on Apache Spark,” *J. Big Data*, vol. 5, no. 1, pp. 1 – 11, Mar. 2018, doi: https://doi.org/10.1186/s40537-018-0119-6.

[8] J. Moreno, M. Serrano, and E. Fernández-Medina, “Main Issues in Big Data Security,” *Future Intern.*, vol. 8, no. 3, Sep. 2016, Art. no. 44, doi: https://doi.org/10.3390/fi8030044.

[9] L. Wang and R. Jones, “Data analytics for network intrusion detection,” *J. Cyber Secur. Techn.*, vol. 4, no. 2, pp. 1–18, Dec. 2019, doi: https://doi.org/10.1080/23742917.2019.1703525.

[10] M. A. Manzoor and Y. Morgan, “Real-time Support Vector Machine based Network Intrusion Detection system using Apache Storm”, *2016 IEEE 7th Annual Inform. Techn., Electron. Mobile Commun. Conf. (IEMCON)*, Vancouver, BC, Canada, 2016, pp. 1-5.doi: 10.1109/IEMCON.2016.7746264.

[11] K. P. R. Krishna and R. Thirumuru,"A balanced intrusion detection system for wireless sensor networks in a big data environment using CNN-SVM model”, *Inform. Automat.*, vol. 22, no. 6, pp. 1296–1322, Mar. 2023.

[12] M. Mahdavisharif, S. Jamali, and R. Fotohi, “Big Data-Aware Intrusion Detection System in Communication Networks: a Deep Learning Approach”, *J. Grid Comp.*, vol. 19, no. 4, pp. 1 – 28, Oct. 2021, doi: https://doi.org/10.1007/s10723-021-09581-z.

[13] I.V. Pustokhina, D.A. Pustokhin, E.L. Lydia, P. Garg, and A. Kadian et al., "Hyperparameter search based convolution neural network with Bi-LSTM model for intrusion detection system in multimedia big data environment”, *Multim. Tools Applicat.*, vol. 81, no. 24, pp. 34951–34968, Mar. 2022.

[14] R. Salama and M. Ragab, “Blockchain with Explainable Artificial Intelligence Driven Intrusion Detection for Clustered IoT Driven Ubiquitous Computing System”, *Comput. Sys. Sci. Eng.,* vol. 46, no. 3, pp. 2917 – 2932, Apr. 2023.

[15] Kumar, B. V., & Mohan, S, “Hybrid metaheuristic optimization based feature subset selection with classification model for intrusion detection in big data environment”, *Turkish J. Comp. Mathem. Educ.*, vol. 12, no. 12, pp. 2297–2308, Apr. 2021.

[16] A. Ponmalar and V. Dhanakoti, “An intrusion detection approach using ensemble Support Vector Machine based Chaos Game Optimization algorithm in big data platform”, *Appl. Soft Comput.*, vol. 116, Dec. 2021, Art. no. 108295, doi: https://doi.org/10.1016/j.asoc.2021.108295.

[17] R. Sheeba, R. Sharmila, A. Alkhayyat, and R.Q. Malik, “Modified Buffalo Optimization with Big Data Analytics Assisted Intrusion Detection Model”, *Comput. Sys. Sci. Eng.,* vol. 46, no. 2, pp. 1415-1429, Aug. 2023.

[18] G. Arun and C. N. Marimuthu, “Diabetes classification using MapReduce-based capsule network”, *Automatika*, vol. 65, no. 1, pp. 73–81, Nov. 2023, doi: https://doi.org/10.1080/00051144.2023.2284031.

[19] J. Ahn, S.H. Ji, S.J. Ahn, M. Park, H. S. and Lee et al, “Performance evaluation of normalization-based CBR models for improving construction cost estimation”, *Automat. Constr.*, vol. 119, May 2020, Art. no. 103329.

[20] E. Naka, “A Competitive Parkinson-Based Binary Volleyball Premier League Metaheuristic Algorithm for Feature Selection”, *Cybernetic., Informat. Techn.,* vol. 23, no. 4, pp. 91-109, Apr. 2023.

[21] P. V. Dinh, Q. U. Nguyen, D. T. Hoang, D. N. Nguyen, S. P. Bao, and E. Dutkiewicz, “Constrained Twin Variational Auto-Encoder for Intrusion Detection in IoT Systems,” *arXiv.org*, Dec. 04, 2023. https://arxiv.org/abs/2312.02490.

[22] I. Attiya, M.A. Al-qaness, M. Abd Elaziz, and A.O. Aseeri, “Boosting task scheduling in IoT environments using an improved golden jackal optimization and artificial hummingbird algorithm”, *AIMS Mathemat.*, vol. 9, no. 1, no. 1, pp. 847-867, Sep. 2024.

[23] M. Tavallaee, E. Bagheri, W. Lu, and A. Ghorbani, “A Detailed Analysis of the KDD CUP 99 Data Set.” Second IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA), 2009. [Online]. Available: https://www.unb.ca/cic/datasets/nsl.html

[24] Vijayakumar B. and Dr. Mohan. S., “A Novel Feature Selection with Fuzzy Deep Neural Network for Attack Detection in Big Data Environment,” *Indian J. Comp. Sci. Engin.*, vol. 12, no. 3, pp. 539–550, Jun. 2021, doi: https://doi.org/10.21817/indjcse/2021/v12i3/211203009.