Wireless Personal Communications (2020) 111:2065–2089
https://doi.org/10.1007/s11277-019-06972-0

Crow-AFL: Crow Based Adaptive Fractional Lion Optimization Approach for the Intrusion Detection

R. Ganeshan[1] · Paul Rodrigues[2]

Published online: 27 November 2019

Abstract
Intrusion detection has played a major role in ensuring the cybersecurity in various networks. Literature works deal with several cyber attacks in the data through designing various supervised approaches, but have not considered the size of the database during the optimization. Since, the data increases in size exponentially, it is necessary to cluster the database before detecting the presence of an intruder in the system. This work has considered these challenges and thus, has introduced a Crow based Adaptive Fractional Lion (Crow-AFL) optimization approach. The proposed intrusion detection system clusters the database into several groups with the Crow-AFL and detects the presence of intrusion in the clusters with the use of the HSDT classifier. Then, the compact data is provided to the deep belief network trained with Crow-AFL for identifying the presence of intrusion in the entire database. The simulation of the proposed Crow-AFL algorithm is done with the DARPA's KDD cup dataset 1999. The metrics, accuracy, TPR, and TNR, measure the performance of the proposed Crow-AFL algorithm, and it has shown better performance with the value of 96%, 95%, and 96%, respectively.

Keywords Intrusion detection · Cyber security · Clusters · DBN · DARPA's KDD cup dataset 1999

1 Introduction

Cybersecurity has emerged as the major concern nowadays since large number intruders try to access the information present in the networks [1]. Various researchers have opted IDS for detecting the presence of the cyber attacks in the networks [2]. The private and the government firms have the cyber analysts to analyze the passage of the information

✉ R. Ganeshan
ganeshramasamy111@kluniversity.in

Paul Rodrigues
Prigues@kku.edu.sa

[1] Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, AP, India

[2] Department of Computer Engineering, College of Computer Science, King Khalid University, Abha, Saudi Arabia

over the network and to identify the intruder, but the main challenge arises during the differentiation of the normal activity from the intrusion [3]. Nowadays IDS have gained popularity among the companies for detecting the presence of the intrusion in the network [4]. Various applications, such as optical networks [5], Cyber-Physical Systems (CPS) [6], and Mobile Cyber-Physical System (MCPS) [7], have used the IDS for detecting the presence of the cyber attacks [8]. The IDS is usually implemented in the dynamic and high dimensional environment, and hence, it should provide better robustness [9, 10]. Literature has reviewed the intrusion detection in two categories, namely anomaly detection, and signature-based detection systems [11]. In the anomaly detection model, the presence of the intrusion in the network is identified through the alarm based system and thus, provides continuous learning and less maintenance, whereas the signature-based IDS performs the intrusion detection through the signature matching schemes [12]. The intrusions in the network can be categorized as Denial of Services (DoS), Remote to Local, User to Root, and Probe [13, 14].

Detection of the intrusion discussed in the literature falls under two major categories, and they are 1) supervised classification, and 2) unsupervised classification. The supervised models perform the classification of the intrusion through the training of the labeled database, while the unsupervised classification approaches perform the training of the unlabelled data [13]. Literature has introduced various machine learning, like Artificial Neural Networks (ANN), decision trees, fuzzy logic [15], Principle Component Analysis (PCA) [16], Bayesian networks, K-Nearest Neighbor (KNN) [17, 18], multimodal classifier [19], for the intrusion detection.

Boulaiche and Adi [4] presented the IDS by automating the process of signature generation, and for this purpose, they have utilized the honeypot traffic data analysis. They identified the presence of intrusion in the data by building the intrusion database. One of the major advantages of this scheme is the high detection rates and small false positive rates. Acharya and Singh [2] presented the Intelligent Water Drops (IWD) algorithm for identifying the suitable features for the intrusion detection. Along with the SVM classifier, the IWD algorithm performed the suitable selection of intrusion in the data in the network, and it provided low misclassification rate. Besides, the usage of the SVM based optimization approach had increased the computation time. Wu et al. [20] proposed the machine learning based schemes for the intrusion detection and for ensuring the cybersecurity. Raman et al. [9] presented the adaptive scheme for the intrusion detection with the Hypergraph based Genetic Algorithm (HG-GA) model. The authors have also utilized the SVM based approach for the feature selection purpose. Even though the model has various advantages, such as robustness and adaptive, the scheme required high run time.

Folino and Pisani [8] proposed the distributed Genetic Programming (GP) framework, which can be categorized under the ensemble algorithm for the intrusion detection. The proposed framework has the parallel system for fast processing of the intrusion detection. Pajouh et al. [13] presented the naive Bayes and the k-NN based model for the intrusion detection, and was also opted for the dimensionality reduction. The algorithm has tried to overcome the presence of rare cyber attacks in the network through better feature selection schemes, but rather fails in identifying less dangerous attacks. Bamakan et al. [21] proposed the Time-Varying chaos PSO (TVCPSO) through the modification of the PSO. The model had considered the detection rate and false alarm rate, for the feature selection. Devi et al. [22] presented the Adaptive Neuro-Fuzzy Inference System for the intrusion detection in the KDD cup 99 data set.

Analysis of various techniques suggests that the Support Vector Machine (SVM) has good performance based on the efficiency and robustness since the SVM has a minimal structural risk, high generalization ability, etc. [23]. However, SVM classifiers have poor performance, while performing the feature subset selection, parameter optimization and imbalanced dataset [9, 24]. Other literatures have suggested the Fuzzy min–max neural network and Particle Swarm Optimization (PSO) [25], decision tree [26], Deep learning [27] based schemes for identifying the nature of the cyber attacks on the network.

The contributions of this research work for detecting the intrusions in the networks is briefed as follows:

- Firstly, the paper proposes the Crow based Adaptive Fractional Lion algorithm (Crow-AFL) algorithm by modifying the Adaptive Dynamic Directive Operative Fractional Lion algorithm (ADDOFL) algorithm with the Crow Search Algorithm (CSA). The proposed Crow-AFL algorithm acts as a clustering algorithm for dividing the database into several groups.
- Secondly, the proposed Crow-AFL algorithm is used for training the weights of the DBN network for providing the information about the presence of intrusion in the data.

The rest of this paper is organized as follows: Sect. 1 confers introduction to the IDS for ensuring the network from the cyber attacks and reviews the various techniques and analyzes the pros and cons for ensuring the cybersecurity. Section 2 provides the brief explanation of the proposed Crow-AFL algorithm designed primarily for detecting the intrusions in the network affected from the cyber-attacks. The simulation results achieved by the proposed Crow-AFL algorithm are presented in Sect. 3, and Sect. 4 concludes the research work.

2 Proposed Method: Crow-AFL Algorithm for the Intrusion Detection in the Networks

This section presents the proposed IDS for identifying the presence of the intrusion in the system, and also, the system guarantees the cybersecurity for the network users. Figure 1 illustrates the proposed IDS model integrated with the Crow-AFL algorithm for finding the presence of the intruder in the system. The network contains numerous users, and they communicate with the other users or the servers with the use of the authentication key provided by the network server. The outsiders or the intruders enter the network and try to steal/hack the information present in the database. The proposed IDS system finds the presence of the intruder in the network by developing the DBN model and the training the weights within the DBN through the optimization approach. Initially, the database is subjected to the clustering with the help of the proposed Crow-AFL algorithm, since the database present in the networks has a large size. Then, the model uses the Hyperbolic Secant-based Decision Tree (HSDT) classifier for detecting the presence of the intrusions within each cluster, and the intrusion information from each cluster is collected together to form the compact data. Here, the DBN is used to provide the final information/intrusion class in the compact data, and this is done by training the weights of the DBN with the proposed Crow-AFL algorithm.

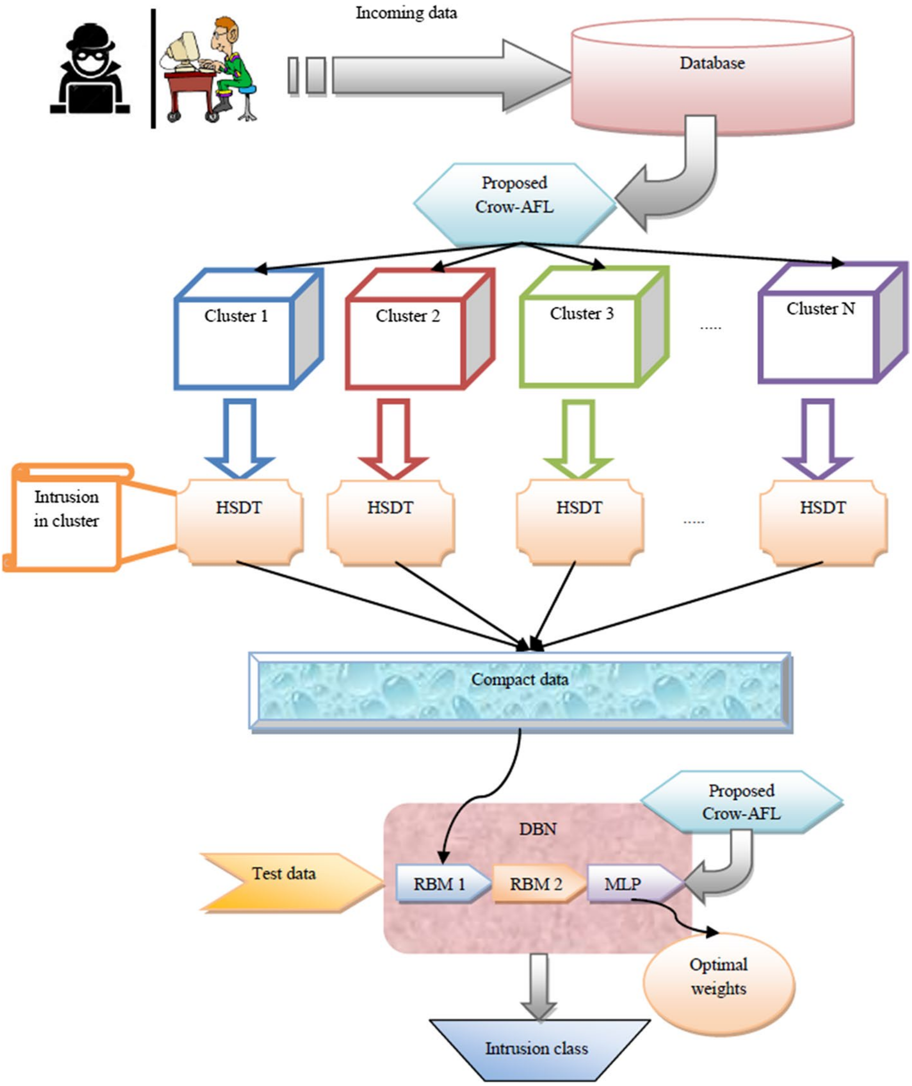

Fig. 1 Intrusion detection system based on the proposed Crow-AFL algorithm

2.1 Clustering the Data Using the Proposed Crow-AFL Algorithm

Consider the network has Ynumber of users and each user pass over Q data samples in the network. From, the data collected from each user, the database I is constructed, and thus, the database I has the size of $Y \times Q$. The data carried over the network has large size, and hence, the detection of the intrusion in the large database I is a complex process. Thus, this work clusters the database into N number of clusters with the use of the proposed Crow-AFL algorithm for reducing the complexity of the IDS. In this section, the proposed Crow-AFL algorithm acts as a clustering algorithm for grouping the database into N clusters, each of size $1 \times Y$ and the expression for the clusters from the Crow-AFL is described as follows,

$$I = \{X_1, X_2, \ldots, .X_i, \ldots, X_N\} \tag{1}$$

where X_i refers to the data in the ith cluster

2.1.1 Solution Encoding for Finding the Optimal Cluster Centroids with the Proposed Crow-AFL Algorithm

The proposed Crow-AFL algorithm tries to cluster the database into Q clusters, and thus, the database tries to find the Q cluster centroids for clustering the database. The database I subjected to clustering process is classified into N clusters, and the Crow-AFL algorithm aims to identify optimal cluster center points from the database for the clustering. The solution for the Crow-AFL algorithm for clustering the database is expressed as follows,

$$f = \left\{f_1, f_2, \ldots, f_i, \ldots, f_N\right\} \tag{2}$$

where f_i is the ith optimal centroid for the clustering.

2.1.2 Algorithmic Description of the Proposed Crow-AFL Algorithm

The proposed Crow-AFL algorithm is an optimization algorithm, which finds the solution cluster centroid point required for clustering the database. The proposed Crow-AFL algorithm is the integration of the CSA [28] and the ADDOFL algorithm [29]. The ADDOFL algorithm has used the kernel based functions for clustering the database, and hence, the algorithm provides better accuracy for clustering the data. The CSA algorithm has the improved convergence rate when compared with the other optimization algorithms, such as PSO, and GA. Since the use of the fractional calculus in the existing ADDOFL algorithm

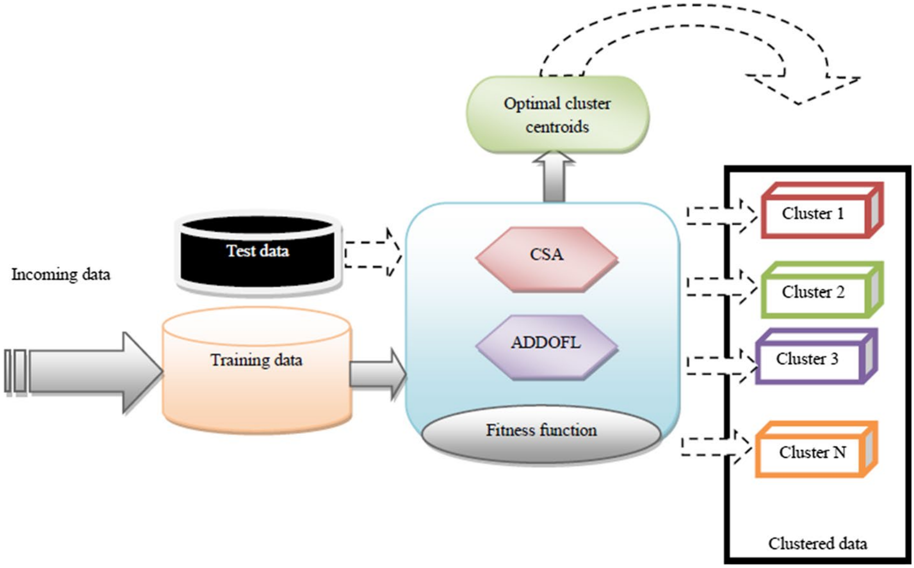

Fig. 2 Clustering the database using the proposed Crow-AFL algorithm

increases the complexity of the process, the incorporation of the CSA makes the algorithm to achieve faster convergence. Figure 2 presents the architecture of the proposed Crow-AFL algorithm for clustering the database.

The steps involved in the proposed Crow-AFL algorithm is described as follows,

Initialization of the population of the Crow-AFL for the clustering The proposed Crow-AFL algorithm finds the optimal solution through the behavior of the lion, and hence, the optimization process considers three types of lions in the population randomly. The proposed Crow-AFL algorithm considers the male lion, female lion, and the nomadic lion same as the ADDOFL algorithm, and it is expressed as follows,

$$Solution = \left\{P^{Male}, P^{Female}, P^{Nomad}\right\} \tag{3}$$

where $P^{Male}, P^{Female}, and\, P^{Nomad}$ indicate the population of the male lion, female lion, and the nomad lion, respectively.

Evaluation of fitness for optimization The optimal cluster centroid obtained through the proposed Crow-AFL algorithm need to satisfy some of the criteria, and the fitness function defines them. The fitness function defined for the proposed Crow-AFL algorithm is calculated based on the four kernel functions, and thus, the fitness for each cluster centroid is defined as the ratio of the fuzzy compactness and the fitness measure.

$$f = \frac{aP^{Male}}{2 \times aP_{\alpha}} \tag{4}$$

where aP^{Male} and aP_{α} refer to the required fuzzy compactness and the separation distance, respectively.

Calculation of the fertility measure of the male and the female lion Here, the fertility rate of both the male and the female lion is evaluated based on the measure of the fertility rate. For finding the fertility rate of the male lion, factors, such as laggardness rate, and the sterility rate are utilized, and based on that measure, the presence of the unfertile lion in the solution gets eliminated.

For the evaluation of the fertility rate of the female lion, the Crow-AFL algorithm uses the Dynamic Directive Operative Search (DDOS) algorithm. The expression for evaluating the fertility of the female lion is expressed in Eq. (5).

$$P_m^{Female} = \begin{cases} P_n^{Female};\ if\ m = n \\ P_m^{Female};\ otherwise \end{cases} \tag{5}$$

where the terms P_m^{Female} and P_n^{Female} are the vectors representing the fertility rate of the female lion. The DDOS angle considers the factors, namely pursuit angle, pursuit distance, and pursuit height, for calculating the required fertility rate of the female lion, and the expression for the fertility rate of the female lion is described in the following expressions,

$$P_{m\_angle}^{Female} = P_g^{Female} + T.k.K_g(\theta_g) \tag{6}$$

$$P_{m\_r}^{Female} = P_g^{Female} + T.k.K_g\left(\theta_g + K^{*}.\frac{\phi\max}{2}\right) \tag{7}$$

$$P_{m\_r}^{Female} = P_{g}^{Female} + T.k.K_{g}\left(\theta_{g} - K^{*}.\frac{\phi\mathrm{max}}{2}\right) \tag{8}$$

where K and K^{*} represent the dynamic parameter and the random sequence involved in the DDOS for calculating the pursuit angle and the distance, respectively. θ_g is the pursuit angle at the direction g. k and $\phi_{\max}$ are the maximum limit of the pursuit distance and the angle. The term K_g in Eq. (8) is the distance vector at the direction g.

Mating The updated fertility of the male and the female lion provides the expression for the mating. The mating process yields new solution based on the fertility rate. Moreover, the growth rate for each solution is also identified in this step.

Crossover The new solution obtained in the mating process is subjected to the crossover operation, and the crossover operation yields new solution represented as follows,

$$P_{c}(w) = \left(W_{w}\right) \times P^{Male} + \left(\overline{W_{w}}\right) \times P^{Female} \tag{9}$$

where $P_c(w)$ is the new solution and W_w is the mask used for the crossover operation. The operation of the crossover yields the maximum of four solutions.

Mutation The solution from the crossover is subjected to the mutation process and hence, results in new solution. For each solution from the mutation process, the fitness value is computed, and the solution with the better fitness is retained throughout the optimization process.

Growth function for the solution The Crow-AFL algorithm uses the growth function to find the better solution and this process yields better mutation rate.

Update based on the fractional calculus and the CSA This work utilizes the first order ($\eta = 1$) fractional calculus and the CSA for formulating the growth of the male lion in the solution. The solution based on the male lion gets updated based on the fractional calculus is represented as follows,

$$P_{w+1}^{Male} = \eta P_{w}^{Male} + \frac{1}{2}\eta P_{w-1}^{Male} \tag{10}$$

where the term η refers to the fractional order of the fractional calculus. Here, the position update equation of the CSA optimization is also utilized. The CSA algorithm updates the position by considering the position of the other solution. Now, based on the CSA algorithm, the position update can be briefed as,

$$P_{w+1}^{Male} = P_{w}^{Male} + h.J_{w} * (o_{w} - P_{w}^{Male}) \tag{11}$$

where h is the random value, J_{w}^{Male} is the flight length of the wth solution, and o_w is the memory. Solving Eq. (11),

$$P_{w+1}^{Male} = P_{w}^{Male}\left[1 - h.J_{w}^{Male}\right] + j.J_{w}^{Male} * (o_{w}) \tag{12}$$

From the above equation, the value of the P_{w}^{Male} is found, and it is represented as follows,

$$P_{w}^{Male} = \frac{1}{\left[1 - h.J_{w}^{Male}\right]}\left[P_{w+1}^{Male} - h.J_{w}^{Male} * (o_{w})\right] \tag{13}$$

Now substitute the value of the P_{w}^{Male} in the Eq. (10), for obtaining the required solution update. The solution update based on the proposed Crow-AFL algorithm is expressed as,

$$P_{w+1}^{Male} = \eta \ \frac{1}{\left[1 - h.J_{w}^{Male}\right]} \left[P_{w+1}^{Male} - h.J_{w}^{Male} * (o_{w})\right] + \frac{1}{2}\eta \ P_{w-1}^{Male} \tag{14}$$

Rearranging the above equation, the final expression for the update of the male lion is obtained, and it is expressed as follows,

$$P_{w+1}^{Male} = \frac{1 - h.J_{w}^{Male}}{\left[1 - h.J_{w}^{Male} - \eta\right]} \left[\frac{1}{2}\eta \ P_{w-1}^{Male} - \frac{\eta * h.J_{w}^{Male} * (o_{w})}{1 - h.J_{w}^{Male}}\right] \tag{15}$$

Territorial defense and the takeover As the solution contains the male lion and the nomad lion, the fitness of both the lions is computed. The male lion tries to defend its territory from the nomad lion. If the fitness of the nomad lion is compared with the fitness of the male lion, the solution with the better fitness is replaced. Finally, when the fitness of the male cub lion is better than the male lion, the cub lion replaces the male lion.

Termination The optimization procedure is repeated until the end of the iteration, and the proposed Crow-AFL algorithm provides the optimal cluster centroids.

2.2 Decision Tree Construction Using HSDT Classifier

This work utilized the HSDT classifier for identifying the presence of the intrusion in each cluster identified to form the Crow-AFL algorithm. The HSDT classifier uses the secant entropy function for modifying the functional tangent probability utilized in work [30]. The entropy function based on the hyperbolic secant function is expressed as follows,

$$fn\left(prob^{i}\right) = \frac{1}{2}\left[\log\left(prob^{i}\right) - 2aSech\left(prob^{i}\right)\right] \tag{16}$$

where the term $aSech(.)$ refers to the hyperbolic secant function for the HSDT classifier. Based on the values of the entropy function defined in the Eq. (16), the decision tree is built for each cluster data. This yields the required intrusion information in each cluster.

2.3 Generation of the Compact Data

The HSDT classifier provides the information about the presence of intrusion in each cluster and based on that the compact data is constructed. The compact data comprises of the output of each HSDT classifier present in the cluster. The expression for the compact data is expressed as follows,

$$F = \left\{F_{1}, F_{2}, \dots, F_{i}, \dots, F_{N}\right\}; 1 \leq i \leq N \tag{17}$$

where F_i is to the information about the presence of intrusion in the cluster i as provided by the HSDT classifier, and the term N refer to the total number of clusters.

2.4 Detection of the Intrusion: Incorporating the Proposed Crow-AFL with the DBN

This section presents the architecture of the DBN network along with the proposed Crow-AFL algorithm for identifying the presence of the intrusion within the database. Figure 3 presents the architecture of the DBN along with the proposed Crow-AFL algorithm for the intrusion detection. The DBN used in this work comprises of the two RBM layers and one MLP layers. The compact data obtained from the HSDT classifier is given as the input to the DBN architecture. The RBM layer present in the DBN has the input and the visible layer, in which the inputs are fed to the input layer, and then, the compact data multiplied with the weights are provided to the hidden layer of the RBM1. The output of the RBM1 layer acts as the input to the second RBM layer.

The input layer present in the RBM layers is also represented as the visible layer, and the compact data is provided to the visible layer of the RBM layer1.

$$D^1 = \left\{D^1_1, D^1_2, \ldots, D^1_i, \ldots, D^1_N\right\}; 1 \leq i \leq N \tag{18}$$

The hidden layer present in the RBM layer one is expressed as follows,

$$A^1 = \left\{A^1_1, A^1_2, \ldots, A^1_j, \ldots, A^1_b\right\}; 1 \leq j \leq b \tag{19}$$

where D^1_i refers to the *i*th neuron present in the first RBM layer of the DBN and A^1_j is the *j*th neuron of the hidden layer. Both the input and the hidden layers in the RBM layer contains the bias, and they are represented as p and q. For each neuron present in the RBM layer one the weights are present, and this value is used in the computation of the output. The expression (20) depicts the weights present in the RBM layer 1.

$$Z^1 = \left\{Z^1_{ij}\right\}; 1 \leq i \leq N; 1 \leq j \leq b \tag{20}$$

where the term Z^1_{ij} represents the weights between the *i*th visible neuron and the *j*th hidden neuron in the first RBM layer. Based on the bias present in the hidden layer q and the

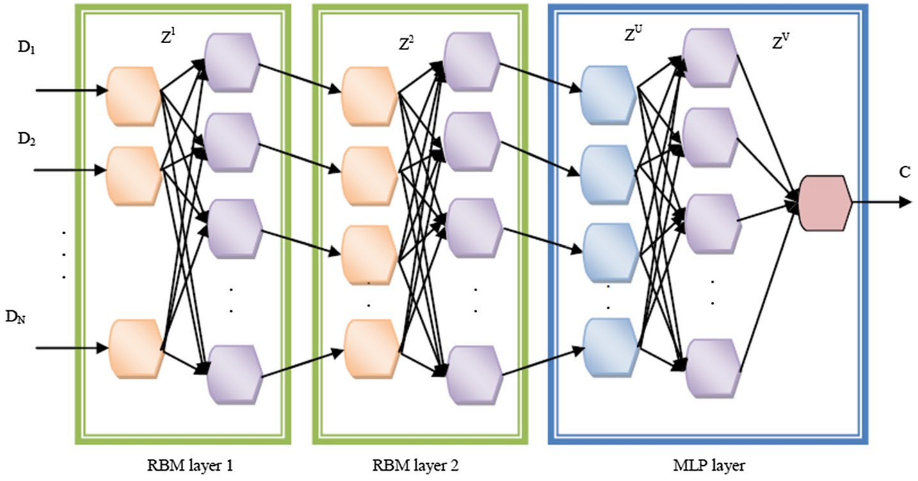

Fig. 3 Architecture of the DBN with the Crow-AFL algorithm

weight value computed in the above expression, the output of the RBM layer one is computed, and it is expressed as follows,

$$A_j^1 = \beta \left[q_j^1 + \sum_j D_i^1 Z_{ij}^1 \right] \tag{21}$$

where the term β represents the activation function for the RBM layer 1, and the final output of the RBM layer one is represented as follows,

$$A^1 = \left\{ A_j^1 \right\}; 1 \le j \le b \tag{22}$$

The output of the RBM layer one is provided as the input to the RBM layer 2, and hence, the visible layer of the RBM layer 2 has b layers. The expression for the visible layer and the hidden layer of the RBM layer two is expressed by the following equations,

$$D^2 = \left\{ D_1^2, D_2^2, \dots, D_b^2 \right\} = \left\{ A_j^1 \right\}; 1 \le j \le b \tag{23}$$

$$A^2 = \left\{ A_1^2, A_2^2, \dots, A_j^2, \dots, A_b^2 \right\}; 1 \le j \le b \tag{24}$$

Similarly, as the RBM layer 1, the RBM layer 2 has the weight vectors, and they are represented as,

$$Z^2 = \left\{ Z_{jj}^2 \right\}; 1 \le j \le b \tag{25}$$

Based on the weights and the bias present in the hidden layer, the output of the RBM layer two is calculated, and the following equation expresses it,

$$A_j^2 = \beta \left[q_j^2 + \sum_i D_i^2 Z_{jj}^2 \right] \forall D_i^2 = A_j^2 \tag{26}$$

where q_j^2 indicates the bias present in the jth neuron of the hidden layer in the RBM layer 2. Then, the final expression for the output of the RBM layer two is given as follows,

$$A^2 = \left\{ A_j^2 \right\}; 1 \le j \le b \tag{27}$$

Then, finally the output of the RBM layer two forms the input layer of the MLP layer. The MLP layer comprises of the input layer, hidden layer, and the output layer. The input layer of the MLP layer is described in the expression (28),

$$u = \left\{ u_1, u_2, \dots, u_j, \dots, u_b \right\} = \left\{ A_j^2 \right\}; 1 \le j \le b \tag{28}$$

where u_j represents the jth input neuron of the MLP layer, and similarly, the hidden layers present in the MLP layer is represented as,

$$v = \left\{ v_1, v_2, \dots, v_x, \dots, v_y \right\}; 1 \le x \le y \tag{29}$$

where y refers to the size of the hidden neurons present in the MLP layer. In this work, the presence of intrusion in the database is indicated through the output layer of the MLP, and thus, the class value in the output layer has only one neuron represented by C. For the computation of the output neuron of the hidden layers, the MLP layers constitute of the weights both in the input and the hidden layer. The following expressions (30) and (31) represent the weights present in the input and the hidden layer.

$$Z^U = \left\{ Z^U_{jx} \right\}; 1 \le j \le b; 1 \le x \le y \tag{30}$$

$$Z^V = \left\{ Z^V_x \right\}; 1 \le x \le y \tag{31}$$

where Z^U_{jx} refers to the weight present between the jth input neuron and the xth hidden layer of the MLP layer. Besides the presence of the weight, the hidden layer of the MLP has the bias, and it is represented as follows,

$$z_x = \left[\sum_{j=1}^{b} W^U_{jx} * u_j \right] B_X \forall u_j = A^2_j \tag{32}$$

where W^U_{jx} is the weight of the jth input neuron and the xth hidden layer of the MLP, and B_x refers to the bias present in the hidden neuron of the MLP layer. The final expression for the output of the hidden layer is expressed as follows,

$$C = \sum_{x=1}^{y} Z^V_x * z_x \tag{33}$$

where the term Z^V_x represents the weight present in each hidden neuron of the MLP layer.

2.4.1 Solution Encoding for the Weight Selection of DBN

The MLP layer present in the DBN contains the weights at both the input and the hidden neurons. The required weights for the input and the visible neurons of the MLP layer are trained using the proposed Crow-AFL algorithm. Here, the proposed Crow-AFL algorithm acts as an optimization algorithm to find the suitable weights for the MLP layer training.

2.4.2 Training Phase of the DBN with the Proposed Crow-AFL Algorithm

The weights in the RBM layers and the MLP layer get trained with the use of the proposed Crow-AFL algorithm. Initially, the DBN utilized the gradient descent- backpropagation algorithm for finding the optimal weights, and this work uses the Crow-AFL for further refining the search process. The training of the weights in the two RBM layers is done through the gradient descent- backpropagation algorithm, whereas the training of the weights in the MLP layer is performed through the proposed Crow-AFL algorithm.

I. Training of RBM layer 1 with the gradient descent- backpropagation

The training procedure for obtaining the optimal weights in the RBM layer 1 is presented in the following steps,

(i) The compact data D is given as the input to the RBM layer 1, and the equivalent vector for the training of each input present in the training sample is generated based on the expression (18).

(ii) Then, the equivalent probability value of each hidden neuron in the hidden layers of the RBM 1 is computed as follows,

$$P\left(A_j^1 = 1 \middle| \bar{D}^1\right) = \beta\left[q_j^1 + \sum_i D_i^1 Z_{ij}^1\right] \tag{34}$$

(iii) The probability value of the hidden neurons relate to the positive gradient value of the RBM 1, and hence, it is expressed as follows,

$$G^+ = D^1.P_A^T \tag{35}$$

The positive gradient of the hidden layer in the RBM 1 has the size of $N * b$.

(iv) The reconstruction of the visible neurons in the visible layer of the RBM 1 constitutes the probability of every visible neuron, and it is done through the sampling. The following expression describes the probability of each hidden neurons present in the RBM 1.

$$P\left(D_i'^1 = 1 \middle| \bar{A}^1\right) = \sigma\left[p_i^1 + \sum_j A_j^1 W_{ij}^1\right] \tag{36}$$

(v) In the next step, the probability of reconstruction of hidden neurons is found through resampling, and it is represented as,

$$P\left(A_j'^1 = 1 \middle| \bar{A}^1\right) = \sigma\left[q_j^1 + \sum_i D_i'^1 Z_{ij}^1\right] \tag{37}$$

(vi) The probability values computed in the above steps, are employed to find the negative gradient value of the RBM layer 1, and it is expressed as follows,

$$G^- = D'^1 \cdot A'^{1T} \tag{38}$$

(vii) Based on the learning rate, positive and the negative gradients, the weights in the RBM layer one gets updated. The expression for the weight update in the RBM layer one is given as follows,

$$\Delta Z_{ij} = \chi\left(G^+ - G^-\right) \tag{39}$$

where the term χ refers to the learning rate for updating the weights in the RBM layer.

(viii) Now, the weight update equation for the next iteration in the RBM layer one is expressed as,

$$Z_{ij}^1(t+1) = Z_{ij}^1(t) + \Delta Z_{ij} \tag{40}$$

The value of the ΔZ_{ij} is obtained using Eq. (39).

(ix) Finally, the expression for the energy based on the computed weights for the visible and the hidden neurons is computed and is expressed as,

$$M\left(\bar{D}^1,\bar{A}^1\right) = -\sum_{i,j} Z^1_{ij} D^1_i A^1_j - \sum_i p^1_i D^1_i - \sum_j q^1_j D^1_j \tag{41}$$

(x) The training procedure is repeated until the weights produce minimal energy, and finally, at the end of the iteration, the weight with the minimal energy is taken as the optimal weight for the RBM layer 1.

II. Training the weights of RBM layer 2 with the gradient descent- backpropagation

The training procedure for obtaining the optimal weights for the RBM layer 2 is same as the training procedure for the RBM layer one, but the output of the RBM layer one is fed as the training input for the RBM layer 2. The optimal weights for the RBM layer two are expressed as Z^2.

III. Training the weights of the MLP layer with the proposed Crow-AFL algorithm

The optimal weights of the MLP layer are obtained with the proposed Crow-AFL algorithm, which is discussed in the Sect. 3.1. Based on the optimization procedure, the weights of the MLP layer is trained. The training procedure for obtaining the optimal weights in the MLP layer of the DBN is presented below,

(i) Initially, the weights present in the input Z^U and the hidden layer Z^V of the MLP is randomly initialized based on the Eqs. (30) and (31).

(ii) The output of the RBM layer two is provided as the input to the MLP layer, and hence, the training input to the MLP is represented as $\left\{A^2_j\right\}$.

(iii) Now, compute the values of z_x based on the Eq. (32) and the output C, accordingly.

(iv) From the output value C, compute the average error performance, which is the deviation of the actual performance from the desired response. The expression for the average error is given as follows,

$$R_{avg} = \frac{1}{N}\sum_{i=1}^{N}\left(C_i - O_i\right)^2 \tag{42}$$

where C_i refers to the output of the DBN and O_i is the desired response.

(v) The weights present in the MLP layer is computed based on the average error response, and hence, in this step, the weights present in the input and the hidden layer are computed, and they are represented as follows,

$$\Delta Z^U_{jx} = -\chi\frac{\partial R_{avg}}{\partial Z^U_{jx}} \tag{43}$$

$$\Delta Z^V_x = -\chi\frac{\partial R_{avg}}{\partial Z^V_x} \tag{44}$$

(vi) Now, based on the gradient descent backpropagation algorithm, the weight of the MLP layer is updated, and the updated expressions for the weights in the input and the hidden layer of the MLP are expressed as follows,

$$Z^U_{jx(Back)}(t+1) = Z^U_{jx}(t) + \Delta Z^U_{jx} \tag{45}$$

$$Z^V_{x(Back)}(t+1) = Z^V_{x}(t) + \Delta Z^V_{x} \tag{46}$$

(vii) In this step, the weights in the MLP layer are updated based on the proposed Crow-AFL algorithm. The Eq. (47) represents the updated weight in the input layer of the MLP using the Crow-AFL algorithm.

$$Z^U_{t+1} = \frac{1 - h.J^U_t}{\left[1 - h.J^U_t - \eta\right]}\left[\frac{1}{2}\eta\, Z^U_{t-1} - \frac{\eta * h.J^U_t * (o_t)}{1 - h.J^U_t}\right] \tag{47}$$

Similarly, the weights present in the hidden neurons of the MLP is updated based on the Crow-AFL algorithm, and it is given as follows,

$$Z^V_{t+1} = \frac{1 - h.J^V_t}{\left[1 - h.J^V_t - \eta\right]}\left[\frac{1}{2}\eta\, Z^V_{t-1} - \frac{\eta * h.J^V_t * (o_t)}{1 - h.J^V_t}\right] \tag{48}$$

(viii) Now, utilize the Eq. (42) for computing the error based on the output computed by the updated gradient descent backpropagation algorithm, and it is represented as,$R_{avg(back)}$

(ix) Compute the error function $R_{avg(Crow\_AFL)}$ based on the Eq. (42), for the updated weights using the proposed Crow-AFL algorithm.

(x) The error value obtained from both the algorithms is compared, and the weight update provided by the algorithm with the minimal error is selected as the optimal weight. The expression for the optimal weight of the input and the hidden neurons in the MLP layer is given by the following equation,

$$Z^U_{jx}(t+1) = \begin{cases} Z^U_{jx(Crow\_AFL)}(t+1); & if\, R_{avg(Crow\_AFL)} < R_{avg(back)} \\ Z^U_{jx(back)}(t+1); & otherwise \end{cases} \tag{49}$$

$$Z^V_{x}(t+1) = \begin{cases} Z^V_{x(Crow\_AFL)}(t+1); & if\, R_{avg(Crow\_AFL)} < R_{avg(back)} \\ Z^V_{x(back)}(t+1); & otherwise \end{cases} \tag{50}$$

(xi) The steps are repeated until the end of the iteration, and the optimal weights for the MLP layer is returned.

2.4.3 Testing Phase: Detection of Intrusion in the Test Data

The DBN algorithm provides the required intrusion information for the test data L, based on the optimal weights presented through the training process. The proposed Crow-AFL algorithm found the suitable weights and based on the weights, the DBN detects the presence of the intrusion information in the test data. The final output of the proposed algorithm along with the DBN will be the intrusion class given as follows,

$$C = DBN\{L\} \tag{51}$$

where the term L represents the test data, and C refers to the intrusion class. The intrusion class provides the value as 1 for the presence of intrusion and the value 0 for other conditions.

2.5 Pseudo-Code for the Proposed IDS with the Crow-AFL Algorithm

The entire process involved in this work for detecting the presence of the intrusion in the database is briefed in the following pseudo-code. Table 1 briefs the pseudo code of the

Table 1 Description of the proposed IDS using Crow-AFL algorithm

Pseudo code: Intrusion detection using the proposed Crow-AFL algorithm

```
Input: Database $I$
Output: Intrusion class $C$
Begin
    Call the Crow-AFL algorithm
        Begin
            Define the fitness
            Find the optimal centroids
            Cluster the database into $N$ clusters based on the optimal centroids obtained
        End
    Call the HSDT algorithm
        Begin
            For each cluster (1 to N)
                Identify the intrusion in the cluster
                Return the intrusion information in the class
            End for
        End
    Generate the compact data for all $N$
    Call the DBN
        Begin
            //Training phase
            Begin
                Input the training data
                Initialize the weights and the bias of the RBM layer 1
                Initialize the weights and the bias of the RBM layer 2
                Initialize the weights and the bias of the MLP layer
                Find the weights of the RBM layer 1 and RBM layer 2 with the gradient
                backpropagation algorithm
                Find the weights of the MLP layer with the proposed Crow-AFL
                algorithm
                Find the error based on the weights
                Return the optimal weights
            End
            //Testing phase
            Begin
                Input  the test data $L$
                Find the intrusion class $C$ based on the optimal weights
            End
        End
    Return the intrusion class $C$
End
```

proposed system with the Crow-AFL algorithm. Initially, the database I of size $Y \times Q$ is provided as the input to the proposed Crow-AFL algorithm for the clustering, and the clustering process yields N data groups of size $Y \times 1$. The HSDT algorithm is enabled in each cluster to identify the intrusion information, and the compact data is generated from the output of the each HSDT classifier. The DBN is trained with the use of the compact data and provides the intrusion class C.

3 Results and Discussion

This section presents the experimental results achieved by the IDS model with the proposed Crow-AFL algorithm. The simulation results achieved by the proposed Crow-AFL algorithm is compared with various existing algorithms, and the metrics, such as True Positive Rate (TPR), True Negative Rate (TNR), and Accuracy along with ROC curve, analyze the performance of each model.

3.1 Experimental Setup

The experimentation of the proposed intrusion detection algorithm is implemented in the MATLAB 2018.a. The experimentation of the proposed Crow-AFL algorithm required the PC with the configurations of Windows 10 OS, Intel I3 processor, and the 4 GB RAM. The experimentation is done under two criteria (without PCA and with PCA) using the DARPA's KDD cup dataset 1999.

3.1.1 Dataset Description

The simulation setup used the DARPA's KDD cup dataset 1999 [31] for the experimentation. The DARPA's KDD cup dataset 1999 has been considered as the standard dataset for the intrusion detection, and it contains the data collected from the military environment, which is connected through the LAN. The dataset contains various cyber attacks, such as DoS, R2L, U2R, and the probing.

3.1.2 Evaluation Metrics

The intrusion detection done by the proposed Crow-AFL and the comparative models is analyzed under various metrics, TPR, TNR and accuracy, and their mathematical expression is given as follows,

TPR TPR defines the trueness of the algorithm to identify the actual intrusion information in the database, and it is expressed as,

$$TPR = \frac{TP}{TP + FN} \tag{52}$$

TNR TNR defines the correct prediction about the absence of the intrusion in the database, and it is expressed by the following equation,

$$TNR = \frac{TN}{TN + FP} \tag{53}$$

Accuracy The accuracy metric defines the efficiency of the algorithm to predict the intrusion information, and the mathematical expression for the accuracy is defined as follows,

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{54}$$

where *TP* indicates the true positive, *TN* refers to the true negative, *FN* represents the false negative, and *FP* denotes the false positive achieved by the algorithms.

3.1.3 Comparative Models

The performance of the proposed Crow-AFL algorithm is compared with various techniques, such as Intrusion Detection using ADDOFL and Hyperbolic Secant-based Decision Tree Classifier (I-AHSDT), ADDOFL + DT [29], C-ADDOFL + DBN [29], PSO + SVM [21], PSO + ANFIS [21], PSO + Naive Bayes [21], HML-IDS [32], and ICNN [33]. The I-AHSDT algorithm is developed through the modification of the HSDT with the ADDOFL algorithm. The ADDOFL + DT model is the combination of the ADDOFL with the decision tree, while the C-ADDOFL + DBN algorithm is the modified ADDOFL

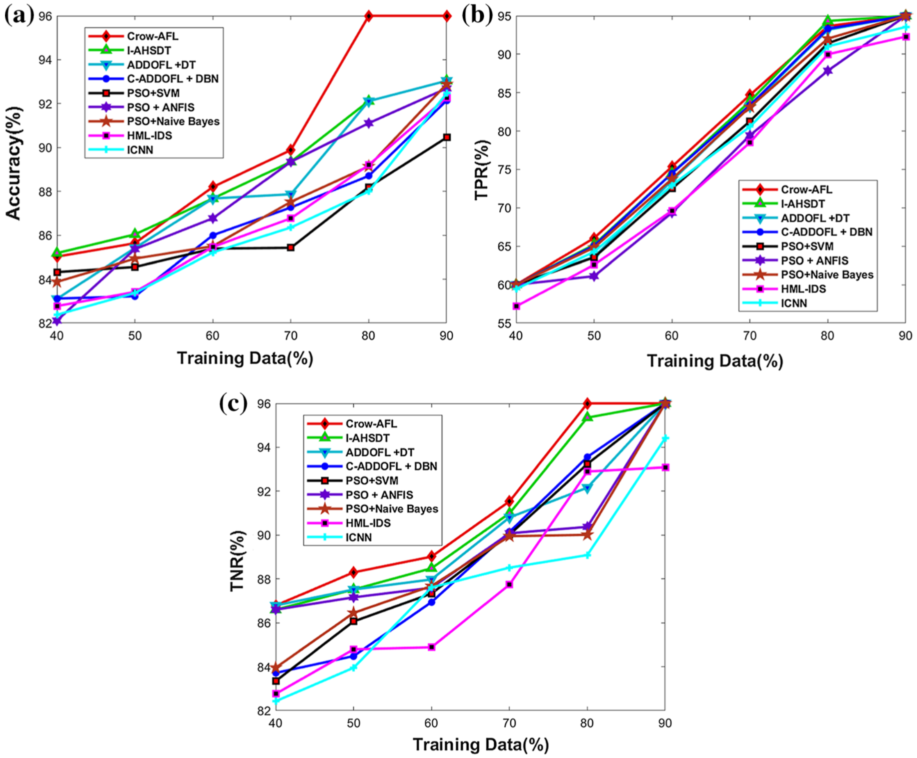

Fig. 4 Comparative analysis of the dataset without PCA for varying training percentage based on **a** accuracy, **b** TPR, and **c** TNR

algorithm along with the DBN. The existing models PSO + SVM, PSO + ANFIS, and PSO + Naive Bayes is the modified PSO algorithm with the SVM, ANFIS, and the naive Bayes, respectively.

3.2 Comparative Analysis of the Dataset Without the PCA

Here, the performance of the models is analyzed when implemented in the dataset without the application of the PCA.

3.2.1 Analysis of Crow-AFL for Varying Training Percentage

Figure 4 shows the performance of each model for the varying training percentage of the dataset without the application of the PCA. Figure 4a presents the comparative analysis of the models based on the accuracy metric. At 90% training of the database, existing models I-AHSDT, ADDOFL + DT, C-ADDOFL + DBN, PSO + SVM, PSO + ANFIS, PSO + Naive Bayes, HML-IDS, and ICNN have the accuracy values of 93.0433%, 93.0433%, 92.1396%, 90.4526%, 92.6968%, 92.8923%, 92.2755%, and 92.4706%, respectively, while the proposed Crow-AFL algorithm has achieved better accuracy value of 96%. Figure 4b shows the comparative analysis of the model based on the TPR versus training percentage using the database without PCA. At 90% of the training data, the existing I-AHSDT, ADDOFL + DT, C-ADDOFL + DBN, PSO + SVM, PSO + ANFIS, PSO + Naive Bayes,

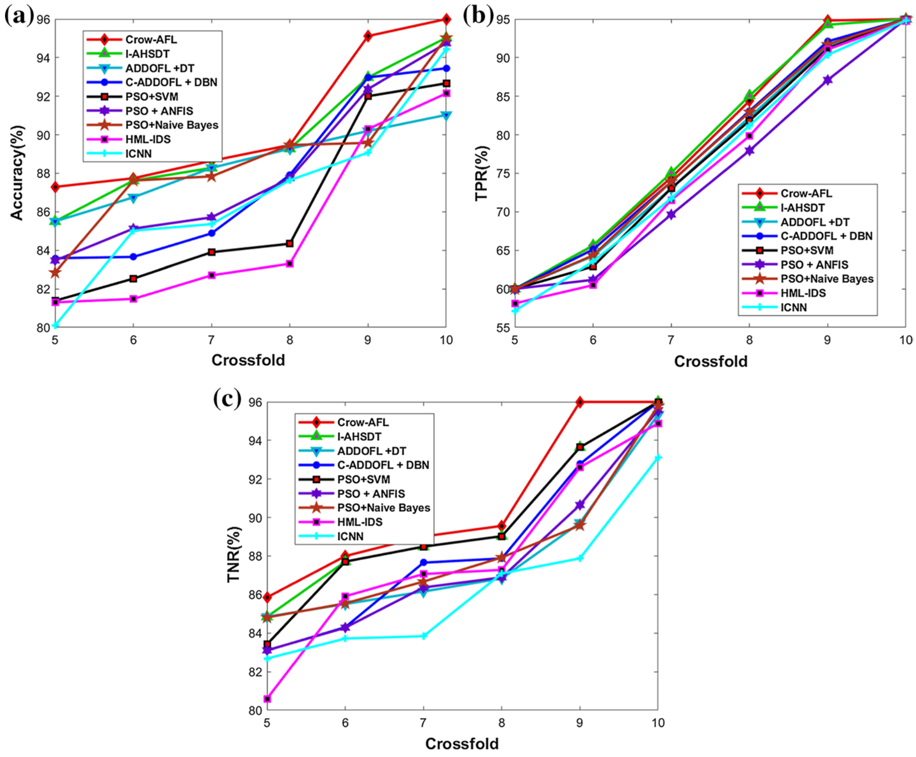

Fig. 5 Comparative analysis of the dataset without PCA for varying crossfold validation based on **a** accuracy, **b** TPR, and **c** TNR

HML-IDS, and ICNN models reached TPR value of 95%, 95%, 95%, 95%, 95%, 95%, 92.2902%, and 93.5558%, respectively, and the proposed model has high TPR value of 95%. Analysis based on the TNR is presented in the Fig. 4c, and from the graph, it is evident that the proposed Crow-AFL algorithm achieved maximum TNR value of 96%.

3.2.2 Analysis of Crow-AFL for Varying Crossfold

Figure 5 shows the performance of each model for varying crossfold validation of the dataset without the application of the PCA. Figure 5a presents the comparative analysis of the models based on the accuracy metric for varying crossfold values of the database without PCA. At the crossfold value = 9 of the database, existing models I-AHSDT, ADDOFL + DT, C-ADDOFL + DBN, PSO + SVM, PSO + ANFIS, PSO + Naive Bayes, HML-IDS, and ICNN have the accuracy values of 92.9728%, 90.1979%, 92.9728%, 92.0019%, 92.3749%, 89.5794%, 90.2991%, and 89.0735%, respectively, while the proposed Crow-AFL algorithm has achieved the better accuracy value of 95.1185%. Figure 5b shows the comparative analysis of the model based on the TPR Vs crossfold validation under the database without PCA. When the crossfold value = 9, the existing I-AHSDT, ADDOFL + DT, C-ADDOFL + DBN, PSO + SVM, PSO + ANFIS, PSO + Naive Bayes, HML-IDS, and ICNN models reached low TPR value of 95%, 95%, 95%, 95%, 95%, 95%, 94.7609%, and 94.8282%, respectively, and the proposed model has TPR value of 95%. Analysis based on the TNR is presented in the Fig. 5c, and from the graph it is evident that the proposed Crow-AFL algorithm achieved TNR value of 96% when the croofold value = 10, which is higher than that of the existing I-AHSDT, ADDOFL + DT, C-ADDOFL + DBN, PSO + SVM, PSO + ANFIS, PSO + Naive Bayes, HML-IDS, and ICNN have achieved the values of 96%, 95.3402%, 96%, 96%, 95.5728%, 95.7995%, 94.8839%, and 93.1182%.

3.2.3 ROC Analysis Without PCA

Here, the performance of the models based on the ROC analysis for the database without PCA is analyzed and is shown in Fig. 6. For the FPR value of 20, the existing I-AHSDT, ADDOFL + DT, C-ADDOFL + DBN, PSO + SVM, PSO + ANFIS, PSO + Naive Bayes

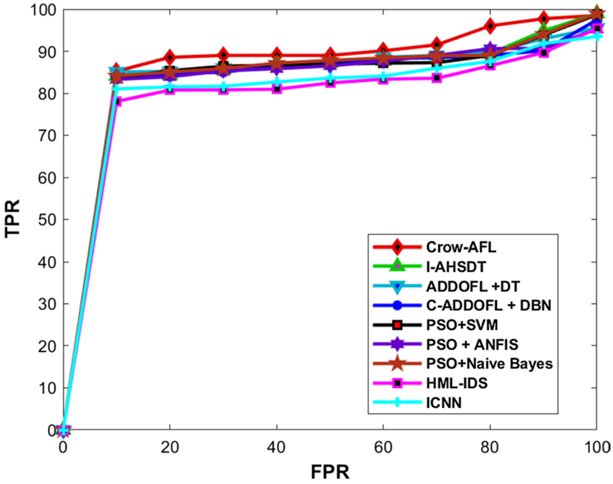

Fig. 6 ROC analysis of the models under the dataset without PCA

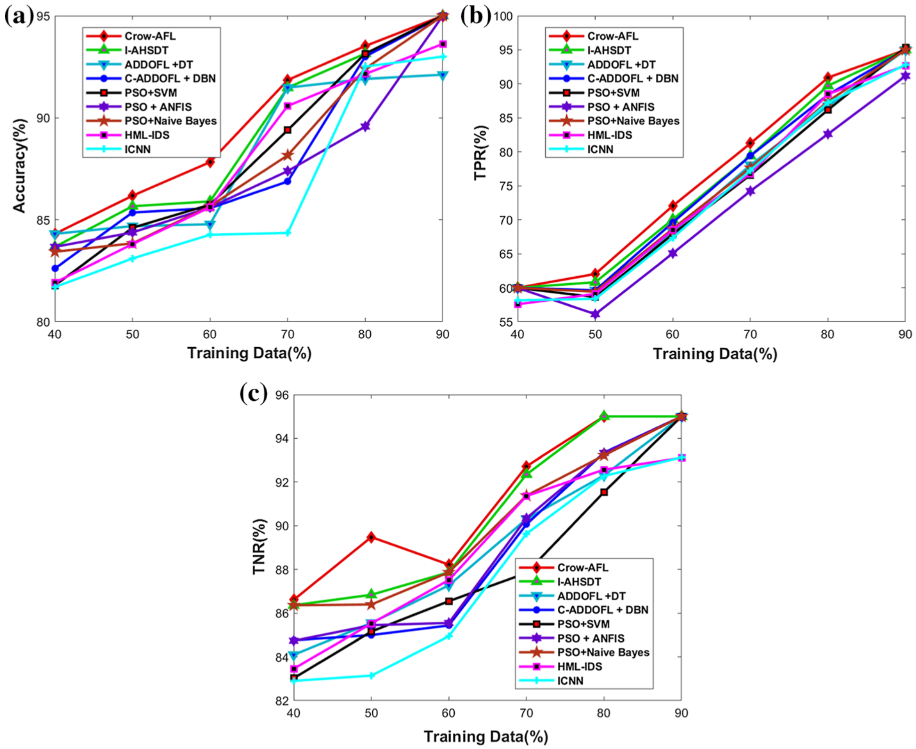

Fig. 7 Comparative analysis of the dataset with PCA for varying training % based on **a** accuracy, **b** TPR, and **c** TNR

algorithm, HML-IDS, and ICNN have achieved the TPR value of 84.3591%, 85.21059%, 84.6455%, 85.3484%, 83.9254%, 85.0253%, 80.7521%, and 81.5258%, respectively. The proposed Crow-AFL algorithm has achieved better TPR value than each comparative model with the values of 88.5461% at the FPR value of 20.

3.3 Comparative Analysis of the Dataset with the PCA

In this analysis, the performance of the models is analyzed by applying the PCA to the database.

3.3.1 Analysis of Crow-AFL for Varying Training Percentage

Figure 7 shows the performance of each model for the varying training percentage of the dataset with the application of the PCA. Figure 7a presents the comparative analysis of the models based on the accuracy metric for the varying percentage of the database with PCA. At the 70% training of the database, existing models I-AHSDT, ADDOFL + DT, C-ADDOFL + DBN, PSO + SVM, PSO + ANFIS, PSO + Naive Bayes, HML-IDS, and ICNN have the accuracy values of 91.4827%, 91.4827%, 86.87919%, 89.4026%, 87.3920%, 88.1649%, 90.5859%, and 84.3545%, respectively, while the proposed Crow-AFL algorithm has the achieved better accuracy value of 91.8486%. Figure 7b shows the comparative analysis of the model based on the TPR Vs training percentage under the database with PCA. For 70% of the training data, the existing methods, such as I-AHSDT, ADDOFL + DT, C-ADDOFL + DBN, PSO + SVM, PSO + ANFIS, PSO + Naive Bayes, HML-IDS, and ICNN reached low TPR value of 79.4156%, 78.0569%, 79.4156%, 76.6005%, 74.2337%, 77.6507%, 76.7774%, and 77.2869%, respectively, and the proposed model has high TPR value of 81.2458% at the training percentage of 70. Analysis based on the TNR is presented in the Fig. 7c, and from the graph, it is evident that the proposed Crow-AFL algorithm achieved TNR value of 92.7242%, which is higher than the existing I-AHSDT, ADDOFL + DT, C-ADDOFL + DBN, PSO + SVM, PSO + ANFIS, PSO + Naive Bayes models, HML-IDS, and ICNN have achieved the values of 92.3548%, 90.3105%, 90.0658%, 87.8426%, 90.3474%, 91.3797%, 91.3623%, and 89.6324%, for the database training of 70%.

3.3.2 Analysis of Crow-AFL for Varying Crossfold

Figure 8 shows the performance of each model for the varying crossfold validation of the dataset with the application of the PCA. Figure 8a presents the comparative analysis of the models based on the accuracy metric for the varying crossfold validation of the database without PCA. At the crossfold validation = 8 of the database, existing models I-AHSDT, ADDOFL + DT, C-ADDOFL + DBN, PSO + SVM, PSO + ANFIS,

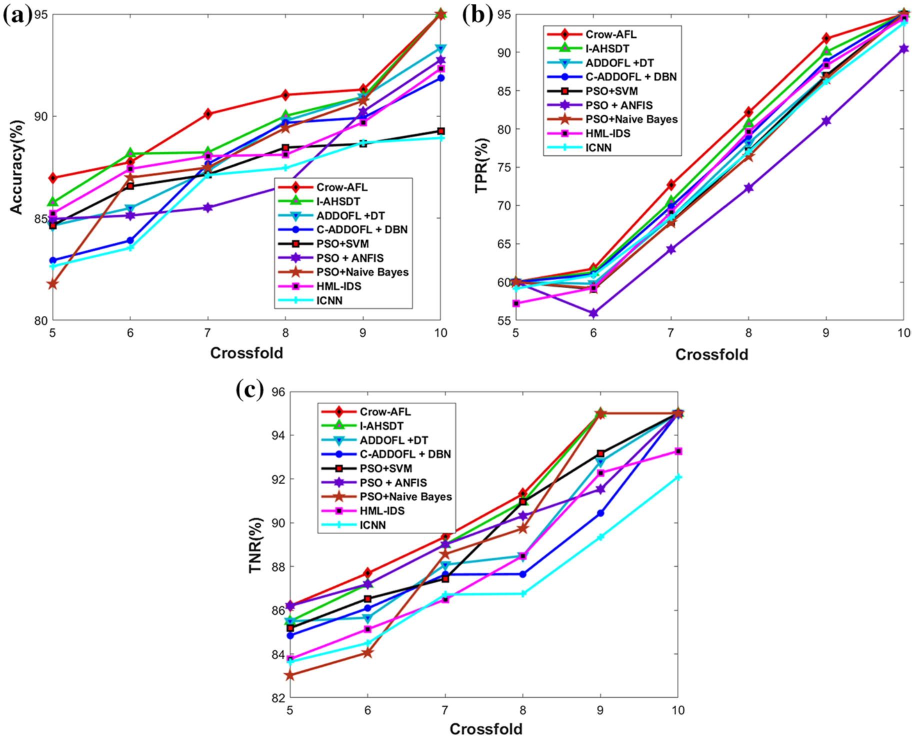

Fig. 8 Comparative analysis of the dataset with PCA for varying crossfold validation based on **a** accuracy, **b** TPR, and **c** TNR

PSO + Naive Bayes, HML-IDS, and ICNN have the accuracy values of 90.0146%, 89.7694%, 89.6769%, 88.4637%, 86.5844%, 89.4257%, 88.1108%, and 87.4592%, respectively, while the proposed Crow-AFL algorithm has achieved better accuracy value of 91.0467%. Figure 8b shows the comparative analysis of the model based on the TPR Vs crossfold validation under the database without PCA. The existing I-AHSDT, ADDOFL + DT, C-ADDOFL + DBN, PSO + SVM, PSO + ANFIS, PSO + Naive Bayes models, HML-IDS, and ICNN reached low TPR value of 80.6803%, 78.15%, 79.0412%, 77.0946%, 72.2983%, 76.4092%, 79.6437%, and 77.0167%, respectively, and the proposed model has TPR value of 82.1664% at the crossfold validation = 8. Analysis based on the TNR is presented in the Fig. 8c, and from the graph, it is evident that the proposed Crow-AFL algorithm achieved TNR value of 95%, which is higher than that of the existing I-AHSDT, ADDOFL + DT, C-ADDOFL + DBN, PSO + SVM, PSO + ANFIS, PSO + Naive Bayes models, HML-IDS, and ICNN have achieved the values of 95%, 92.7975%, 90.4319%, 93.1691%, 91.5348%, 95%, 92.2827%, and 89.3475%, respectively, for the crossfold validation = 9.

3.3.3 RoC Curve Analysis with PCA

Here, the performance of the models based on the RoC curve analysis for the database with PCA is analyzed and is shown in Fig. 9. For the FPR value of 20, the existing I-AHSDT, ADDOFL + DT, C-ADDOFL + DBN, PSO + SVM, PSO + ANFIS, PSO + Naive Bayes, HML-IDS, and ICNN have achieved the TPR value of 84.1808%, 83.7566%, 84.1409%, 83.558%, 84.3592%, 84.0014%, 81.6932%, and 81.3579%, respectively. The proposed Crow-AFL algorithm has achieved better TPR value than each comparative model with the value of 84.3592%, at the FPR value of 20.

3.4 Comparative Discussion

Table 2 presents the comparative analysis of each model with the proposed Crow-AFL algorithm based on various evaluation metrics. This table discusses the best performance of the comparative models under different scenarios. From the comparative discussion, the

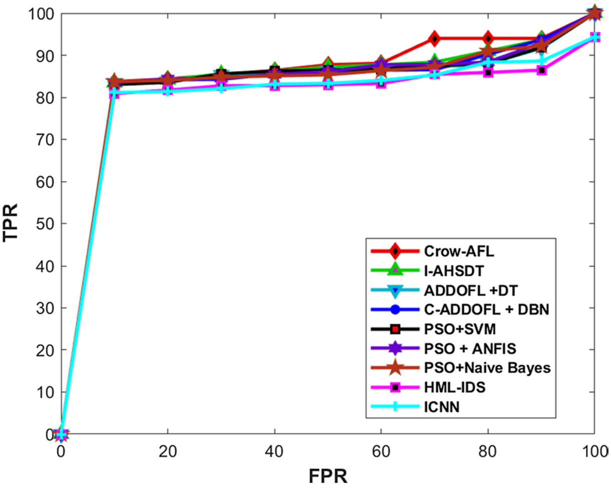

Fig. 9 RoC analysis of the models under the dataset with PCA

Table 2 Comparative discussion

Comparative models	Evaluation metrics		
	Accuracy (%)	TPR (%)	TNR (%)
Proposed Crow-AFL algorithm	96	95	96
I-AHSDT	95.0307	95	96
ADDOFL + DT	93.3372	95	96
C-ADDOFL + DBN	95	95	96
PSO + SVM	95	95	96
PSO + ANFIS	95	95	96
PSO + Naive bayes	95.0307	95	96
HML-IDS	93.6174	94.7609	94.8839
ICNN	94.4408	94.8282	94.4133

proposed model has achieved overall better performance with the value of 96%, 95%, and 96%, for the accuracy, TPR and TNR, respectively, which is comparatively better than the other competing techniques.

4 Conclusion

The system has proposed the Crow-AFL algorithm by combining the CSA algorithm with the existing ADDOFL algorithm. The database containing the data from several users is subjected to the clustering using the proposed Crow-AFL algorithm. Then, the presence of the intrusion in each cluster is identified by the use of the HSDT classifier and the output of each HSDT classifier is collected together to form the compact data. Besides, DBN is trained using the compact data and the proposed Crow-AFL algorithm is integrated with the DBN for obtaining the optimal weights for the training process. The simulation of the proposed Crow-AFL algorithm is done with the DARPA's KDD cup dataset 1999, which is one of the standard databases for the intrusion detection. The experimental results of the Crow-AFL are obtained through various training percentage and the crossfold validation, and the results are compared with the recent works. The metrics, accuracy, TPR, and TNR, measure the performance of the proposed Crow-AFL algorithm, and it has achieved better performance with the value of 96%, 95%, and 96%, respectively.

References

1. Singh, S., & Silakari, S. (2009). A survey of cyber-attack detection systems. *International Journal of Computer Science and Network Security, 9*(5), 1–10.
2. Acharya, N., & Singh, S. (2017). An IWD-based feature selection method for intrusion detection system. *Soft Computing, 22,* 1–10.
3. Ben-Asher, N., & Gonzalez, C. (2015). Effects of cyber security knowledge on attack detection. *Computers in Human Behavior, 48,* 51–61.
4. Boulaiche, A., & Adi, K. (2018). An auto-learning approach for network intrusion detection. *Telecommunication Systems, 68*(2), 277–294.
5. Zhang, H., Wang, Y., Chen, H., Zhao, Y., & Zhang, J. (2017). Exploring machine-learning-based control plane intrusion detection techniques in software defined optical networks. *Optical Fiber Technology, 39,* 37–42.

6. Orojloo, H., & Azgomi, M. A. (2017). A game-theoretic approach to model and quantify the security of cyber-physical systems. *Computers in Industry, 88,* 44–57.
7. Mitchell, R., & Chen, R. (2013). On survivability of mobile cyber physical systems with intrusion detection. *Wireless Personal Communications, 68*(4), 1377–1391.
8. Folino, G., & Pisani, F. S. (2016). Evolving meta-ensemble of classifiers for handling incomplete and unbalanced datasets in the cyber security domain. *Applied Soft Computing, 47,* 179–190.
9. Raman, M. G., Somu, N., Kirthivasan, K., Liscano, R., & Sriram, V. S. (2017). An efficient intrusion detection system based on hypergraph-Genetic algorithm for parameter optimization and feature selection in support vector machine. *Knowledge-Based Systems, 134,* 1–12.
10. Kuang, F., Xu, W., & Zhang, S. (2014). A novel hybrid KPCA and SVM with GA model for intrusion detection. *Applied Soft Computing, 18,* 178–184.
11. Wang, G., Hao, J., Ma, J., & Huang, L. (2010). A new approach to intrusion detection using Artificial Neural Networks and fuzzy clustering. *Expert Systems with Applications, 37*(9), 6225–6232.
12. Sadhasivan, D. K., & Balasubramanian, K. (2017). A novel LWCSO-PKM-based feature optimization and classification of attack types in SCADA network. *Arabian Journal for Science and Engineering, 42*(8), 3435–3449.
13. Pajouh, H. H., Dastghaibyfard, G., & Hashemi, S. (2017). Two-tier network anomaly detection model: A machine learning approach. *Journal of Intelligent Information Systems, 48*(1), 61–74.
14. Liao, H. J., Lin, C. H. R., Lin, Y. C., & Tung, K. Y. (2013). Intrusion detection system: A comprehensive review. *Journal of Network and Computer Applications, 36*(1), 16–24.
15. Veeraiah, N., & Krishna, B. T. (2018). Intrusion detection based on piecewise fuzzy C-means clustering and fuzzy Naïve Bayes rule. *Multimedia Research, 1*(1), 27–32.
16. Powalkar, S., & Mukhedkar, M. M. (2015). Fast face recognition based on wavelet transform on PCA. *International Journal of Scientific Research in Science, Engineering and Technology (IJSRSET), 1*(4), 21–24.
17. Tsai, C. F., Hsu, Y. F., Lin, C. Y., & Lin, W. Y. (2009). Intrusion detection by machine learning: A review. *Expert Systems with Applications, 36*(1), 11994–12000.
18. Huang, J., Zhu, Q., Yang, L., Cheng, D., & Wu, Q. (2017). A novel outlier cluster detection algorithm without top-n parameter. *Knowledge-Based Systems, 121,* 32–40.
19. Daga, B. S., Ghatol, A. A., & Thakare V. M. (2017). Silhouette based human fall detection using multimodal classifiers for content based video retrieval systems. In *Proceedings of the international conference on intelligent computing, instrumentation and control technologies (ICICICT)* (pp. 1409–1416).
20. Wu, M., Song, Z., & Moon, Y. B. (2019). Detecting cyber-physical attacks in CyberManufacturing systems with machine learning methods. *Journal of Intelligent Manufacturing, 30*(3),1111–1123.
21. Bamakan, S. M. H., Wang, H., Yingjie, T., & Shi, Y. (2016). An effective intrusion detection framework based on MCLP/SVM optimized by time-varying chaos particle swarm optimization. *Neurocomputing, 199,* 90–102.
22. Devi, R., Jha, R. K., Gupta, A., Jain, S., & Kumar, P. (2017). Implementation of intrusion detection system using adaptive neuro-fuzzy inference system for 5G wireless communication network. *AEU-International Journal of Electronics and Communications, 74,* 94–106.
23. Raman, M. G., Somu, N., Kirthivasan, K., & Sriram, V. S. (2017). A hypergraph and arithmetic residue-based probabilistic neural network for classification in intrusion detection systems. *Neural Networks, 92,* 89–97.
24. Wang, H., Gu, J., & Wang, S. (2017). An effective intrusion detection framework based on SVM with feature augmentation. *Knowledge-Based Systems, 136,* 130–139.
25. Azad, C., & Jha, V. K. (2017). Fuzzy min–max neural network and particle swarm optimization based intrusion detection system. *Microsystem Technologies, 23*(4), 907–918.
26. Moon, D., Im, H., Kim, I., & Park, J. H. (2015). DTB-IDS: an intrusion detection system based on decision tree using behavior analysis for preventing APT attacks. *The Journal of Supercomputing, 73*(7), 2881–2895.
27. Jing, L., & Bin, W. (2016). Network intrusion detection method based on relevance deep learning. In *International conference on intelligent transportation, big data & smart city (ICITBS)*, Changsha, China (pp. 237–240).
28. Askarzadeh, A. (2016). A novel metaheuristic method for solving constrained engineering optimization problems: crow search algorithm. *Computers & Structures, 169,* 1–12.
29. Chander, S., Vijaya, P., & Dhyani, P. (2018). Multi kernel and dynamic fractional lion optimization algorithm for data clustering. *Alexandria Engineering Journal, 57*(1), 267-276.
30. Chandanapalli, S. B., Sreenivasa Reddy, E., & Rajya Lakshmi, D. (2017). FTDT: Rough set integrated functional tangent decision tree for finding the status of aqua pond in aquaculture. *Journal of Intelligent & Fuzzy Systems, 32,* 1821–1832.

31. The UCI KDD Archive KDD Cup 1999 Data. Retrieved from October 2017 http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.
32. Khan, I. A., Pi, D., Khan, Z. U., Hussain, Y., & Nawaz, A. (2019). HML-IDS: a hybrid-multilevel anomaly prediction approach for intrusion detection in SCADA systems. *IEEE Access, 7,* 89507–89521.
33. Yang, Hongyu, & Wang, Fengyan. (2019). Wireless network intrusion detection based on improved convolutional neural network. *IEEE Access, 7,* 64366–64374.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

R. Ganeshan is currently working as Assistant Professor in Koneru Lakshmaiah Education Foundation, Vaddeswaram, AP, India

Paul Rodrigues is currently working as Professor in King Khalid University, Abha, Saudi Arabia