International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-9 Issue-1, October, 2019

An Effective System of Intrusion Detection on Deep Neural Network by Hybrid Optimization in Cyber Security

Thupakula Bhaskar, Tryambak Hiwarkar, K. Ramanjaneyulu

Abstract: ***In present trends organizations are very much interested to protect data and prevent malware attack by using well flourished and excellent tools. Many algorithms are used for the intrusion detection system (IDS) and it has pros and cons. Here we proposed a novel method of intrusion detection using hybrid optimization techniques such as Gravity search algorithm with gray wolf optimization (GSGW). In this method the gray wolf technique has a leader for the continuous monitoring of the attacker and has a low false alarm rate and a high detection rate. The performance evaluation is done by the feature selection in NSL-KDD dataset. In the proposed method the experimental result reveals less false alarm rate, better accuracy and high Detection when compared to previous analysis.***

Keywords: FAR (False Alarm Rate), DR (Detection Rate), IDS, Gravity search Gray wolf optimization (GSGW)

I. INTRODUCTION

Nowadays, Internet has become a crucial part in various organization to survive technological terms. [1]. The anomaly detection is applied to detect dissimilar requests in other areas such as credit card fraud detection, safety critical systems in fault detection, and aims to classify abnormal actions in medical field. Though, the anomaly detection process may offer a high false alarm rate, and need wide instruction sets to attain a dependable performance results [2]. The high perception of Supervisory Control and Data Acquisition (SCADA), ICT systems known as Information and Communications Technology are consistent to one another, following a difficult susceptibility through esteem cyber intrusions [3]. DoS attack prevents the communication through the classification ways it have been well scheduled, so this is the furthest available designs in intrusion area [4].

By learning DoS attack its properties are proved without any difficulties. [5].Energy constraint persists a normal connection for dissimilar thought of muggers, that they may modify their rules in attack [6]..

Revised Manuscript Received on October 30, 2019.

* Correspondence Author

Thupakula Bhaskar *, Research-Scholar, Department of Computer Science and Engineering, Sri Satya Sai University of Technology & Medical Sciences, Bhopal, M. P. India.

Dr.Tryambak Hiwarkar, Professor, Department of Computer Science and Engineering, Sri Satya Sai University of Technology & Medical Sciences, Bhopal, M. P. India.

Dr.K. Ramanjaneyulu, Professor, Department of ECE, Prasad V. Potluri Siddhartha Institute of Technology, Vijayawada, A. P. India.

II. MOTIVATION AND PROBLEM DEFINITION

Internet is trendy among variety of users and numerous cyber-attacks are generated against internet. Because of the growing usage of Internet primarily based services cyber security in IDS is a challenge. The cyber criminals are easy to access the information, so we need, machine learning based data processing to face the challenges of cyber security. The purpose of intrusion detection analysis is to beat the drawbacks of existing approaches in Internet security. High detection time, low accuracy and low flexibility are the common drawbacks of intrusion detection approaches. Machine learning based intrusion detection system looks problem in whole dataset because of its size and imbalanced character which results in biased performance and over-fitting. So it is needed to diagnose intrusion from intruder by correct feature learning and suitable algorithms.

III. PROPOSED METHODOLOGY

In this proposed methodology, we recommend an efficient intrusion detection framework with adaptive Jaya optimization (AJO) [9] to concurrently do parameter initialization and feature selection for modified Deep Neural Network (MDNN) [8]. MDNN classifier is presented to classify the various kinds of attacks in cyber security. The values related to weights are updated using hybrid Gravity Search Algorithm with Gray Wolf Optimization (GSGW) to minimize the classification error. The main think is to integrate the potential of exploitation in Gray wolf with the capability of exploration in GSA to synthesize each algorithms' strength. The experimental analysis reveals that the hybrid algorithm has high capability to flee from native optimums with quicker convergence than the quality GSA and GWO

3.1 Flowchart of the proposed System

In the proposed System the features are selected from the NSL dataset using Adaptive Jaya optimization. The GSGW is used to calculate the fitness value and is applied to the Modified deep neural network to detect the intrusion.

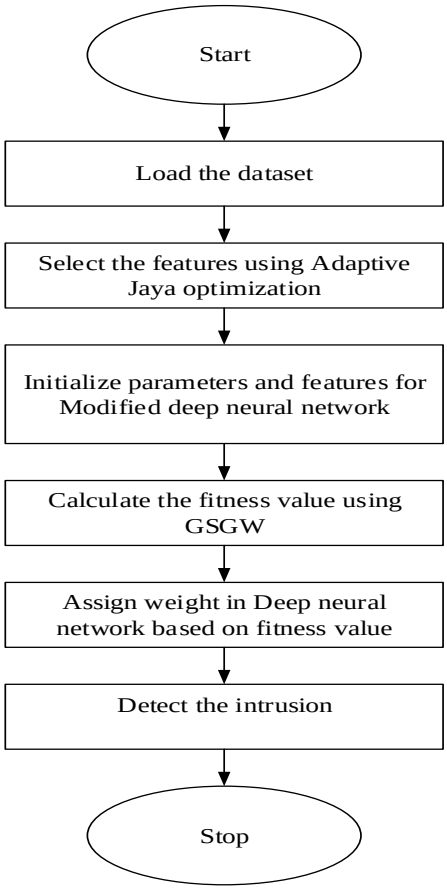

Figure1: Flow Chart of System

In the proposed system we use the NSL-KDD dataset. The fig1 shows the flow of the proposed system. Initially the data is loaded and the best features among the data is selected by using Adaptive Jaya optimization technique. The features are given as input for the Modified deep neural network. The deep neural network consists of several layers such as input layer, Hidden layer and Output layer. In modified deep neural network it comprises input layer, four hidden layers and output layer. The increasing number of hidden layer produces better performance in the output. In each layer of the modified deep neural network a weight value is assigned based on the fitness value calculated for the particular layer. The fitness value is calculated by using hybrid Gravity search Algorithm with Gray wolf optimization (GSGW) and the modified deep neural network process the input values for all the four hidden layers and produce better output. By using the proposed system the performance in detection rate is increased and produces less false alarm rate.

3.2 Working principle of Gravity search algorithm with grey wolf optimization (GSGW) on MDDN

3.2.1 Modified deep neural network (MDDN):

Neural network is termed as deep learning of a process and is composed of one hidden layer and modified deep neural network has several hidden layers.

In each layer the weighted value is randomly selected based on the fittest value calculated by hybrid gravity search algorithm with gray wolf optimization (GSGW). In this, the input is the selected best features in NSL-KDD dataset by using Jaya optimization.

In the modified deep neural network let M denotes the quantity of layers. The layer one is termed as input layer and therefore the layer m is termed as output layer. The mediate hidden layers are painted as layer2 and layer M-1. The value of every node is calculated by multiplying the value with the load W1, W2…Wm. The load is updated within the modified deep neural network by exploitation (GSGW).The values for every node is denoted as Ci,j. The process is recurrent for every layer and therefore the values are calculated. The load for every layer won't be of zero entities. The network is of totally connected is shown in figure 2.

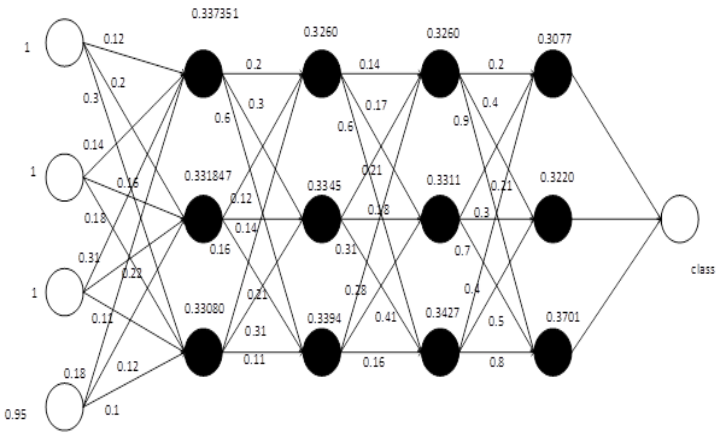

Figure2: Modified Deep Neural Network Classifier

The formula used to calculate the attacks in the modified deep neural network is termed as

$$H = f(G_1^1 Z_1 \mid G_1^2 Z_2 \mid G_1^3 Z_3 \mid .....G_1^m Z_m)$$

$$H = f(\sum_j G_i^j Z_j)$$

G represents the load allotted to the link between the layers and Z represents the neurons present within the network.

The softmax function is applied to each hidden layers in the modified deep neural network. The values are calculated by using the formula

$$P(y = j \mid \theta^i) = \frac{e^{\theta(i)}}{\sum_{j=0}^{k} e^{\theta_k(i)}}$$

Where

$$\theta = W_0 X_0 + W_1 X_1 + ...+ W_k X_k$$

$$\theta = \sum_{i=0}^{k} W_i X_i = W^T X$$

The value is calculated for each and every neurons present in the hidden layer and is represented as Y_i, $e^{\theta_k(i)}$

The figure2 represents the modified deep neural network with four hidden layers. The inclusion of many hidden layers produce better performance.

The fitness value calculated by GSGW is 0.35 is used for the above calculation and is multiplied with the assigned weight of range [0, 1].

Neural networks are designed to recognize patterns as the work done by the human brain. As the human brain understand the data the neural network interpret the data by machine learning. It can understand text, sound, image recognition. The neural network helps to cluster and classify the data.

3.2.2 Hybrid Gravity search algorithm with grey wolf optimization

Gravitational search algorithm (GSA) [7] is a technique used to combine the particles based on the gravity and mass. This algorithm depends on newton's law of motion that indicates the magnitude and direction. The searching of the neighbour objects are done and they attract each other based on the heavier mass and the gravitational force. The object with heavier mass gives the best solution and the lower mass gives the worst solution. In this proposed system this algorithm is used as a search agent to find the intruder in the system. It describes the direction of the nearby system to give the clue about the searching of the intruder.

The formula can be represented as

$$F = k\frac{p1p2}{u^2}$$

k->Gravitational constant

P1-> Mass of first object

P2-> Mass of second object

U->distance between two objects

The gravitational constant is represented using the formula

$$K(t)=K_0 e^{-\propto t/T}$$

The values of K_0 and $\propto$ is initialised in the beginning

In GSA algorithm the agents are selected randomly based on some criteria of the mass and position of the objects which represents a solution to the problem. Here we used several iteration by changing the position of the objects with respect to velocity, fitness value and acceleration in each iteration.

Consider the system with M agents and the ith agent position is defined as

$$Z_i = (z_i^1 \ldots z_i^h \ldots . z_i^m) \qquad for \;\; i = 1,2,3 \ldots M$$

z_i^h represents the ith agent position in the hth dimension and m is the dimension of the pursuit space.

The force foster on the two masses i and j at a given time is denoted as

$$F_{ij}^h = k(t)\frac{p_{ai}(t)\times p_{sj}(t)}{d_{ij}+\varepsilon}(z_j^h(t)-z_i^h(t))$$

$k(t)$ ->Gravitational constant

$p_{ai}(t)$ ->Gravitational mass of agent i

$p_{sj}(t)$ -> Gravitational mass of agent j

ε ->represents a small constant

The Euclidean detachment between the two agents i and j is denoted as

$$d_{ij}(t) = \left| z_i(t).z_j(t) \right|$$

The total force stand-in on mass$_i$ in the hth element in time t is denoted as

$$F_i^h(t) = \sum_{j\in kbest, j\neq i}^{M} random_j F_{ij}^h(t)$$

Where is an arbitrary number of range [0,1], k best is the arrangement of first k agents with the best fitness value.

The acceleration identified with mass i in time t in the hth element is denoted as

$$a_i^h = \frac{F_i^h(t)}{p_{ii}(t)}$$

Where inertial mass $p_{ii}(t)$ of ith agent

The updated velocity of an object is computed by taking product of random value and current velocity in addition to acceleration. The same is calculated by using the below given formula

$$q_i^d(t+1) = random_i q_i^d(t) + a_i^h(t)$$

$$z_i^d(t+1) = z_i^d(t) + q_i^d(t+1)$$

$random_i$ is an random number of range [0,1]

The mass are updated by using

$$p_{ai} = p_{sj} = p_{ii} = p_i \qquad i = 1,2,..M$$

$$p_i(t) = \frac{fitest_i(t) - worst(t)}{best(t) - worst(t)}$$

The $fitest_i(t)$ signifies the fittest value of agent i at time t

$$P_i(t) = \frac{p_i(t)}{\sum_{j=1}^{m} p_j(t)}$$

The worst and best value is calculated by using the formula

$$best(t) = \min_{j\in(1\ldots m)} fitest_j(t)$$

$$worst(t) = \max_{j\in(1\ldots m)} fitest_j(t)$$

By using the algorithm the particles are combined grounded on the mass of the object. In this proposed method this algorithm is used to find the direction of the object and the position are updated by using the gray wolf optimization .

The fitness value is calculated based on the feature values of NSL- KDD dataset.

3.2.3 Grey wolf optimization

The gray wolf algorithm depicts the leadership quality in wolves and also it lives in a pack. It leaves for hunting as a pack. In the pack there is a leader wolf known as alpha and his commands should be followed by the pack. The subordinate wolf termed as beta helps the leader in decision making. The lower level wolf in the pack is termed as omega. There is an intermediate level termed as delta who dominates omega and subordinate to alpha and beta. This algorithm proposes a leader in searching the prey. On seeing the prey to hunt, the pack involves in hunting as per the sign produced by the alpha. The instructions are obeyed by the subordinates to cache the prey. Initially the wolf encircle the prey to make a pivot point to attack the prey. Next by the movement of the prey the positions of the wolf are updated to cache it.

The grey wolf algorithm [10] includes several steps

- Initialize the search agents S_a , design variable size S_b and vectors b,Z,f and the iteration criteria it_{max}

$$\vec{Z} = 2\vec{b}.r_1 - \vec{b}$$

$$\vec{f} = 2.r_2$$

For each iteration the values of b decreased from 2 to 0.

- The wolves are represented as

Wolves=

$$\begin{bmatrix} S_1^1 & S_2^1 & S_3^1 & \dots & .. & .. & .. & S_{sb-1}^1 & S_{sb}^1 \\ S_1^2 & S_2^2 & S_3^2 & .. & .. & .. & .. & S_{sb-1}^2 & S_{sb}^2 \\ . & \dots & .. & .. & \dots & \dots & \dots & \dots & \dots . \\ . & \dots . & \dots . & \dots . & \dots & \dots . & \dots . & \dots . & \dots . \\ S_1^{sa} & S_1^{sa} & S_1^{sa} & \dots & .. & \dots & \dots & S_{sb-1}^{sa} & S_{sb}^{sa} \end{bmatrix}$$

Where Sij is the initial value of the ith pack of the jth wolves

- The fitness value is calculated using

$$\vec{h} = \left| \vec{f}.\vec{S}_p(t) - \vec{S}(t) \right|$$

$$\vec{S}(t+1) = \vec{S}_p(t) - \vec{Z} \,.\, \vec{h}$$

- Calculate the best hunt values

$$\vec{h}_\alpha = \left| \vec{f}_1.\vec{S}_\alpha - \vec{S} \right|$$

$$\vec{h}_\beta = \left| \vec{f}_2.\vec{S}_\beta - \vec{S} \right|$$

$$\vec{h}_\delta = \left| \vec{f}_3.\vec{S}_\delta - \vec{S} \right|$$

$$\vec{S}_1 = \vec{S}_\alpha - \vec{Z}_1(\vec{h}_\alpha)$$

$$\vec{S}_2 = \vec{S}_\beta - \vec{Z}_2(\vec{h}_\beta)$$

$$\vec{S}_3 = \vec{S}_\delta - \vec{Z}_3(\vec{h}_\delta)$$

The new location of the wolf is calculated by

$$\vec{S}(t+1) = F\frac{\vec{S}_1 + \vec{S}_2 + \vec{S}_3}{3}$$

The value of F is taken from the gravity search algorithm and clubbed with the grey wolf algorithm for better result. The new fitness value is calculated and based on that the positions are updated until the condition is satisfied.

3.2.3.1 Calculation GSGW Algorithm:

Fitness

[38978.26003331 73779.0372579 78484.95302316 838326.5630996 24337.66872029 147618.91102804 23099.83020553 48373.99988401 13028.99315072 127280.69518529 87722.61209052 28510.70901802 39233.56175001 29292.9189462 357570.22205822 111550.93472763 19303.52557341]

Worst: 838326.5630995962

Best: 13028.9931507225

First: 0.9685576841281687

Second 0.06732586277956637

Eps = 2.220446049250313e-16

Random =171.74815363608246

Distance =16.564389786001854 -14.097013168758508 = 2.467376617243346

Dij= 37.62388831350286 *21.496048611951167 = 808.7649321576789

In the proposed method we use Adaptive jaya optimization technique to select the best features from the NSL-KDD dataset. The selected best features are, service, Src_bytes, Dst_bytes, hot, , num_root, protocol_type, num_file_creations, count, flag, srv_count, num_compromised,dst_host_count, dst_host_srv_count, rerror_rate, duration dst_host_same_src_port_rate, logged_in. The selected best features are utilised as an input to the modified neural network to detect various kinds of attacks in cyber security.

IV. RESULTS AND DISCUSSIONS

In our proposed AJOMDNN-GSGW methodology it produces better DR, low FAR and high accuracy.

Table 2: The attacks with & without feature selection

class	1) By AJOMDNN-GSGW without FS						2) By AJOMDNN-GSGW with FS					
	Predicted attack						Predicted attack					
	R2L	Probe	Normal	U2R	DoS	DR	R2L	Probe	Normal	U2R	DoS	DR
R2L	2732	8	4	7	3	98.55	2743	4	4	2	1	99.52
Probe	0	0	1	199	2	98.68	0	0	1	201	0	99.38
Normal	22	13	14	18	7389	99.74	5	4	4	6	7437	99.87
U2R	6	2400	6	5	4	83.26	2	2407	5	4	3	91.78
DoS	12	11	9663	10	14	99.68	6	7	9684	6	7	99.85

The table shows the best features selected using Adaptive Jaya optimization for the modified deep neural network. The numbers inside the table reveals the field numbers of the features in NSL-KDD dataset.

Probe: The intrusion will scan the network to accumulate data about the system.

Denial of service (DoS): The intrusion makes the machine unavailable to the user by engaging resources.

User to root (U2R): The intrusion access the root as normal and then attack the system privilege.

Remote to user (R2L): The intrusion try to access the remote by sending packets through network and then exploit the machine .The intruder do not have an account in the local system

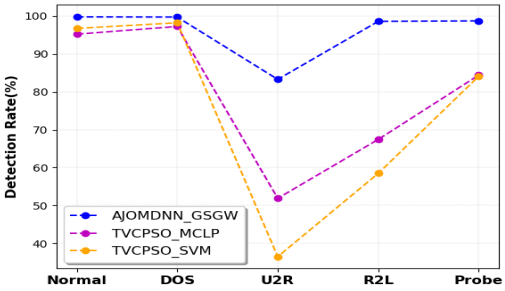

Figure 3: DR without feature selection

Table 1

The Table1 describes the best features selected in our proposed system from NSL-KDD dataset

Basic features	{1,2,3,4,5,6}
Content features	{10,12,13,16,17}
Features of Time based	{23,25,28}
Features of Host based	{32,33,36}

In Fig 3 shows the performance evaluation of detection rate without feature selection. AJOMDNN_GSGW is the proposed method. It shows overall 95.98% detection rate when compared to the earlier methods. The attacks mentioned here are Normal, DOS, U2R, R2L, Probe and the detection rate occurred are 99.74, 99.68, 83.26, 98.55, 98.68 respectively without feature selection

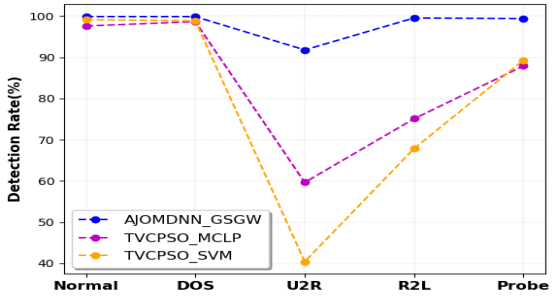

Figure 4: DR with feature selection

In Fig 4 shows the performance evaluation of detection rate with feature selection .The features are selected from NSL-KDD dataset using AJO. It shows overall 98.07% detection rate when compared to the earlier methods. The attacks mentioned here are Normal, DOS, U2R, R2L, Probe and the detection rate

occurred are 99.87, 99.85, 91.78, 99.52, 99.38 respectively.

Comparing both the graphs the graph representing the various attack with feature selection shows better detection rate than without feature selection. The values of the proposed method is contrasted with the existing TVCPSO-MCLP and TVCPSO-SVM [1].

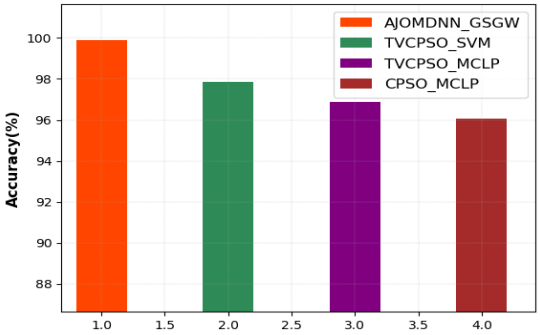

Figure 5: Accuracy without feature selection

In Fig 5 shows the performance evaluation of accuracy of our proposed system. It reveals the accuracy without feature selection. AJOMDNN_GSGW show a higher accuracy percentage of 99.71 when compared to earlier methods TVCPSO-MCLP, TVCPSO-SVM, CPSO-MCLP representing 94.69, 95.75, 92.47 respectively.

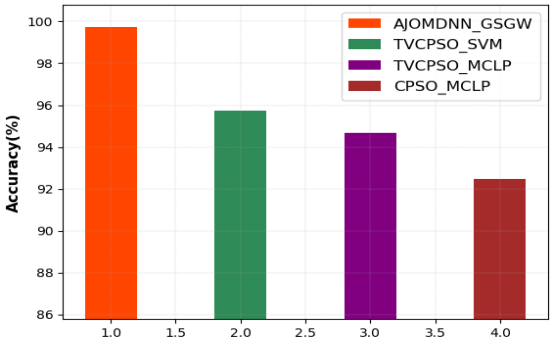

Figure 6: Accuracy with feature selection

In Fig 6 shows the performance evaluation of accuracy of our proposed system. It reveals the accuracy with feature selection. AJOMDNN_GSGW show a higher accuracy percentage of 99.87 when compared to earlier methods TVCPSO-MCLP, TVCPSO-SVM, CPSO-MCLP representing 96.88, 97.84, 96.06 respectively. The features are selected from NSL-KDD dataset using AJO. The graph represented with feature selection shows better accuracy than without feature selection. The values of the proposed system is contrasted with the existing TVCPSO-MCLP (Time varying chaos particle swarm optimization- multiple criteria linear programming),TVCPSO-SVM,CPSO-MCLP(chaos particle swarm optimization).AJOMDNN-GSGW is the proposed methodology termed as Adaptive Jaya optimization modified deep neural network-Gravity search grey wolf algorithm.

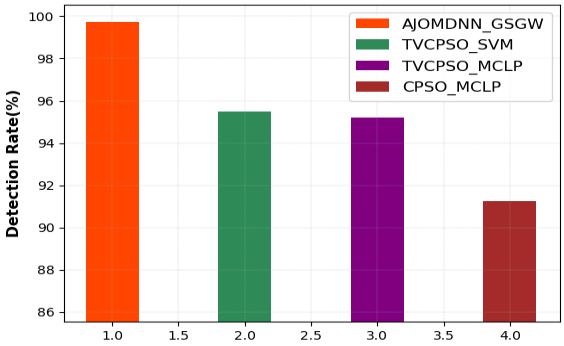

Figure 7: DR without feature selection

In Fig 7 reveals the performance evaluation of the detection rate of our proposed system without feature selection.AJOMDNN_GSGW show a higher detection rate of 95.98 % when compared to earlier methods TVCPSO-MCLP, TVCPSO-SVM, CPSO-MCLP representing 94.69, 95.75, 92.47 respectively.

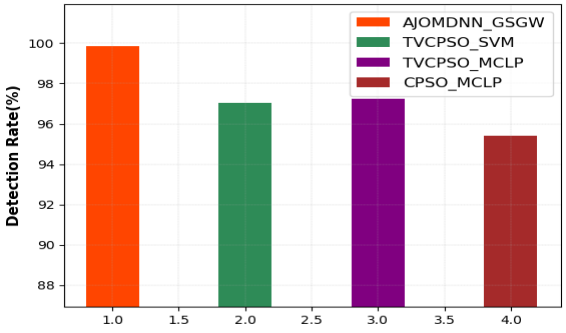

Figure 8: DR without feature selection

In Fig 8 reveals the performance evaluation of the detection rate of our proposed system with feature selection.AJOMDNN_GSGW show a higher detection rate of 98.07 % when compared to earlier methods TVCPSO-MCLP, TVCPSO-SVM,CPSO-MCLP representing 94.69, 95.75, 92.47 respectively.

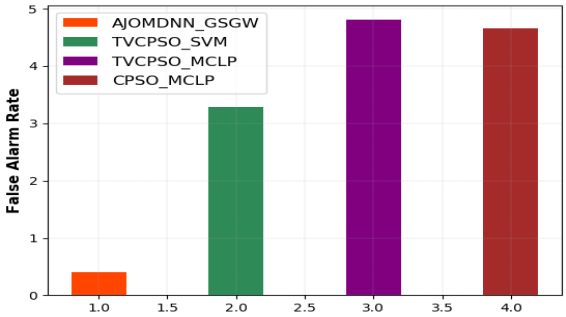

Figure 9: FAR without feature selection

In Fig 9 displays the performance evaluation of FAR of our proposed system withoutFS(FeatureSelection). AJOMDNN_GSGW shows a low FAR of 0.0028 % when compared to previous methods TVCPSO-MCLP, TVCPSO-SVM, CPSO-MCLP representing 4.81, 3.29, 4.66 respectively.

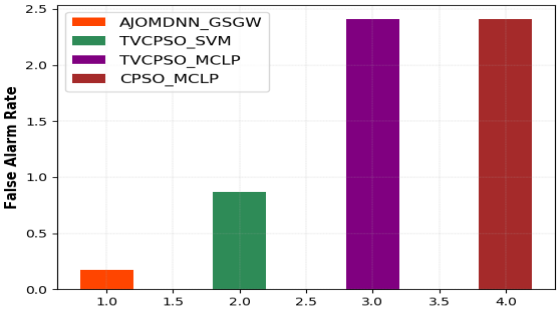

Figure 10: FAR with feature selection

In Fig 10 indicates the performance evaluation of the FAR of our proposed system with FS. AJOMDNN_GSGW shows a low FAR of 0.0012% when compared to previous methods TVCPSO-MCLP, TVCPSO-SVM, CPSO-MCLP representing 2.41, 0.87, 2.41 respectively.

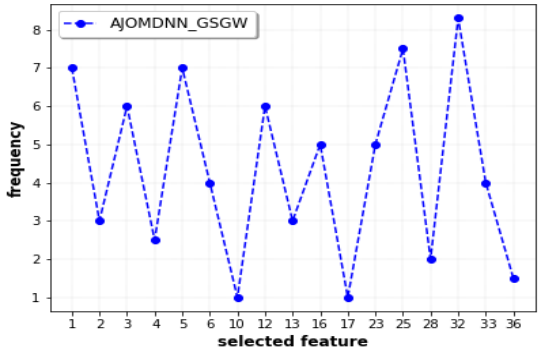

Figure 11: Feature selection using AJO

Table 4:Comparison between existing[1] & proposed method

Metrics	TVCP-MCLP	TVCP-SVM	CPSO- MCLP	AJOMDNN-GSGW
Parameter without selected feature				
Accuracy	94.69	95.75	92.47	**99.71**
Detection rate (DR)	95.19	95.49	91.26	**95.98**
False alarm rate (FAR)	4.81	3.29	4.66	**0.0028**
Parameter with selected feature				
Accuracy	96.88	97.84	96.06	**99.87**
Detection rate (DR)	97.23	97.03	95.42	**98.07**
False alarm rate (FAR)	2.41	0.87	2.41	**0.0012**

V. CONCLUSION & FUTURE WORKS:

In this paper the proposed methodology includes an intelligent intrusion detection context with Adaptive Jaya Optimization (AJO) to concurrently initialize parameters and FS for Modified Deep Neural Network (MDNN). MDNN classifier is presented to classify the various kinds of attacks in cyber security. The values related to weights are updated using hybrid Gravity Search Algorithm with Gray Wolf Optimization (GSGW) to minimize the classification error. Here we used 17 best features from KDD cup data set for high DR and low FAR. In future work a new set of features can be selected by using another technique for better performance.

In Fig 11 describes the frequency of the selected features used in the proposed system as it is described in table7. The x-axis represents the selected features from NSL- KDD dataset using AJO.

Table 3: Parameters used in the proposed method

Variables	Values for AJOMDNN-GSGW
Maximum iteration	400
Particles used	17
Range values of F	[0,1]

During implementation of this paper, we used the above mentioned parameters in proposed method. The Following Table 4 gives the comparison between proposed method (AJOMDNN-GSGW) and exisising techniques.

Acknowledgments:

"A Machine Learning Based Hybrid Intrusion Detection System in Cyber Security" has been a subject with tremendous scope to research upon, which leads to explore new heights in the field of Computer Science & Engineering, and its miscellaneous applications. I 'm thankful to my Research guide: Dr.Tryambak Hiwarkar & Research Co-guide:Dr.K. Ramanjaneyulu whose guidance helped me to work successfully. Their guidance will always encourage me to do work perfectly and professionally.

REFERENCES

1. Bamakan, Seyed Mojtaba Hosseini, Huadong Wang, Tian Yingjie, and Yong Shi. "An effective intrusion detection framework based on MCLP/SVM optimized by time-varying chaos particle swarm optimization." Neurocomputing 199 (2016): 90-102.
2. Ji, S.Y., Jeong, B.K., Choi, S. and Jeong, D.H., 2016. A multi-level intrusion detection method for abnormal network behaviors. Journal of Network and Computer Applications, 62, pp.9-17.
3. Hong, J. and Liu, C.C., 2019. Intelligent electronic devices with collaborative intrusion detection systems. IEEE Transactions on Smart Grid, 10(1), pp.271-281.
4. Amin, S., Cárdenas, A.A. and Sastry, S.S., 2009, April. Safe and secure networked control systems under denial-of-service attacks. In International Workshop on Hybrid Systems: Computation and Control (pp. 31-45). Springer, Berlin, Heidelberg.
5. Zuba, M., Shi, Z., Peng, Z. and Cui, J.H., 2011, December. Launching denial-of-service jamming attacks in underwater sensor networks. In Proceedings of the Sixth ACM International Workshop on Underwater Networks (p. 12). ACM.
6. S. Amin, X. Litrico, S. S. Sastry, and A. M. Bayen, "Cyber security of water scada systems-part i: Analysis and experimentation of stealthy deception attacks," IEEE Transactions on Control Systems Technology, vol. 21, no. 5, pp. 1963–1970, 2013.
7. Rashedi E, Nezamabadi-pour H and Saryazdi S (2009) GSA: a gravitational search algorithm. Inform. Sciences. 179(13), 2232-2248.

8. Thupakula Bhaskar, Tryambak Hiwarkar, K. Ramanjaneyulu. A Modified Deep Neural Network Based Hybrid Intrusion Detection System in Cyber Security, IJITEE, ISSN: 2278-3075, Volume-8 Issue-8, June 2019.
9. Thupakula Bhaskar, Tryambak Hiwarkar, K. Ramanjaneyulu.A novel approach for feature selection technique in NSL-KDD data set of cyber security, IJAAI,Volume 6, Issue 6, June 2019.
10. E. Emary, Hossam M. Zawbaa, and Crina Grosan Experienced Gray Wolf Optimization through Reinforcement Learning and Neural Networks. IEEE transactions on neural networks and learning systems, vol. 29, no. 3, march 2018.

AUTHORS PROFILE

Thupakula Bhaskar is currently a PhD student in Sri Satya Sai University of Technology & Medical Sciences, Bhopal, M.P., and India. He received his M.Tech (CSE) from JNTU Hyderabad, India in 2011. His current research interests include Machine Learning, Deep Learning and Cyber security. He has presented and published more than 20+ papers (5 IEEE & 5 Scopus) at National and International level. .

Dr.Trayambak Hiwarkar was born in Maharashtra, India in 1965. He received the B.Tech (1994), M.tech (1996) & Ph.D. Degree (2003) in CSE from Bundelkhand University Jhansi. He was published many papers in National / International level Journals. He has life memberships in IEEE, ACM, Institution of Engineers (India), MCSI, IETE and many more etc.

Dr.K. Ramanjaneyulu received the Ph.D. degree from CoE, Andhra University, Visakhapatnam, Andhra Pradesh, India, in 2012. He is currently working as professor in PVPSIT,Vijayawada,AndhraPradesh,India.He was published 18 papers in National and International Journals and 22 papers presented in various conferences National and International level.