Flow based intrusion Detection system with Whale Optimization and Evolutionary Algorithms (EA) for Diversified Traffic Streams

k rajiv (krajivvnrvjiet@gmail.com)
VNR VJIET: VNR Vignana Jyothi Institute of Engineering and Technology
G Ramesh Chandra
VNR VJIET: VNR Vignana Jyothi Institute of Engineering and Technology
Vempaty Prashanthi
Gokaraju Rangaraju Institute of Engineering and Technology
V Akila
Gokaraju Rangaraju Institute of Engineering and Technology
D. Dakshayani Himabindu
Gokaraju Rangaraju Institute of Engineering and Technology

Research Article

Keywords: Intrusion detection systems, Machine learning, Deep learning techniques, Evolutionary algorithms, Artificial Neural Network (ANN), Swarm based algorithms

Posted Date: June 4th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-485941/v1

Flow based intrusion Detection system with Whale Optimization and Evolutionary Algorithms (EA) for Diversified traffic streams

*Dr.K Rajiv[1] Dr. G. Ramesh Chandra [2] * Vempaty Prashanthi [3] Dr.V.Akila [4] D.Dakshayani Himabindu[5]

[1,2]Department of Computer Science and Engineering, VNR Vignana Jyothi Institute of Engineering and Technology (VNR VJIET), Hyderabad.India,

rajivvnrvjiet@gmail.com , 8328653310

[3,4,5] Department of Information Technology, Gokaraju Rangaraju Institute of Engineering and Technology, Hyderabad,

Abstract: *The technological growth and advances in the internet led to the generation of huge volume of data that networks must be capable of transmitting. Providing security to this data is a challenging task. The development in the internet attracts several vulnerable attacks. The researchers in the literature proposed several machine learning, Deep learning and ANN based approaches for efficient attack detection. However, these approaches are prone to high false alarm rates and exhibits poor performance for diversified incoming traffic, because these methodologies relay on the packet level or transaction level features. The performance is inversely proposal to the diversity ratio of packet level features. To handle this, we introduced a combination of high-performed evolutionary algorithms and neural networks for attack classification at flow level with low false alarm rates and high detection accuracy. A unique set of flow features are defined to handle the traffic at flow level and optimal feature selection using whale Optimization Algorithm (WOA). The gravitational search (GS), and particle swarm optimization (PSO) combinations are used in attack detection phase to train the ANN and results proposed model as GSPSO-ANN with WOA. The performance of the proposed model is evaluated with NSL-KDD and CSE-CIC-IDS2018 datasets. The results are compared with other ANN based conventional methods. The results inferred that the proposed GSPSO-ANN with WOA attained maximum detection accuracy with low false alarm rates and processing time and also maintained consistency in the performance for diversified traffic.*

Key words: Intrusion detection systems, Machine learning, Deep learning techniques, Evolutionary algorithms, Artificial Neural Network (ANN) and Swarm based algorithms.

1. INTRODUCTION

The technological growth of the internet gained applications in various areas of human life like banking, public networking, online transactions, electronic trade etc. However, with the immense knowledge of attacker the vulnerabilities of the network have frequently been intruded with denial of service (DoS) attacks or Distributed Denial of Service (DDoS) attacks [1]. The DDoS attack is same as the traditional DoS attack, but huge amount of traffic is generated through various ends from the distributed environment using botnet. The DDoS attack denies the access of the victim system for the legitimate user requests. The attacker floods huge number of packets to launch the DDoS attack towards the victim or target network and exhausts the victim system resources like bandwidth, disk space, computing power, etc.

Flooding is one of the most powerful threats to internet. The attacker constitutes the botnet for generating the huge amount of traffic and floods this huge amount of traffic towards the victim. However, the flooding attacks are classified into network level/ transport level and application level based on the protocol used to generate and flood the traffic. Huge volume of traffic is generated using transport layer protocols such as TCP, UDP and ICMP to lunch the network/transport level DDoS attacks by exploiting the vulnerabilities. UDP flood, TCP flood and ICMP flood [2] are the examples of network/transport level DDoS attacks. Most of the web servers and applications widely used protocol in application layer is hypertext transfer protocol (HTTP). The increased HTTP traffic [3] and technological development of the internet invites the attackers to launch the HTTP- based attacks.

In recent studies, rule based datamining, artificial neural network (ANN) [4] , evolutionary algorithms and swarm intelligence have gained significant importance to address the Application layer DDoS attacks. However, the ANN based methods have two short comings. Firstly it exhibits low detection rates. Secondly, the performance of the detection system is unstable for large traffic patterns. One of the reasons for these shortcomings is that ANN based methods are feature dependent and failed to tackle the traffic from the diversified environments. The another reason is ANN provides the better results when it trained with less number of patterns, but the increase of traffic from diversified environments with diversified characteristics of features results uncertainty in the detection performance.

The ANN techniques [5] are prone to be over fitting wen the training dataset is from the diversified networks and contains the diversified characteristics of features to describe normal and attack behavior. It is obvious that the attacker launches DDoS attack with the help of bots or botnet, which exhibits diversified characteristics and distributed in diversified environments. The detection system in testing phase exhibits unexpected results such as unstable or low detection accuracy and high false alarms.One of the alternatives for the ANN methods is Meta-heuristic algorithms and the combination of ANN and these Meta-heuristic optimization algorithms exhibits better results to handle the Application Layer DDoS attacks [6]. The main reasons to consider these meta-heuristic algorithms are (i) these relay on features not the characteristics or values of them ;(ii) gradient information is not required; (iii) local minima is bypassed easily ; (iv) well suitable to address diversified characteristics of the traffic from the distributed ends.

The Nature-inspired or Bio-inspired meta-heuristic algorithms provide the solutions to optimization problems by imitating the biological or physical phenomena of the nature. These algorithms can be classified as Evolutionary based algorithms (EA), Physical-based algorithms (PA) and Swarm based algorithms (SA). The Evolution based algorithms are inspired by the natural evolution laws of creatures in the nature. The randomly generated population is used to start the search process and these will be evolved in the subsequent iterations. The strength of these evaluation based algorithms is that these choose the best individuals and combined these to define the individuals for the next iteration. This is achieved with feature optimization over the interactions. The popular evolution- based algorithms are Genetic Algorithms (GA), Evolution Strategy (ES), Genetic Programming (GP), Biogeography-Based Optimizer (BBO) and Probability-Based Incremental Learning (PBIL). The process of Evaluation-based algorithms is shown in figure 1.

The physical-based algorithms mimic the physical rules of the universe. The popular algorithms under this category are Big-Bang Big-Crunch (BBBC), Gravitational Local Search (GLSA), Charged System Search (CSS), and Gravitational Search Algorithm (GSA), etc.

The swarm-based algorithms mimic the social behavior of the group of animals in the nature. The Particle Swarm Optimization (PSO) is the most popular algorithm in swarm- based methods which mimics the behavior of bird flocking and uses ‘n ‘ number of particles around the search space for finding the optimal solution. The other popular examples other than PSO are Ant Colony Optimization (ACO) , Ant Bee colony (ABO), firefly Approach (FA), Bat algorithm (BA).,etc. These Meta-heuristic approaches are attractive because that PSO has proved to be very competitive with evolutionary based algorithms and swarm based algorithms poses some additional advantages over evolutionary algorithms (EA). For an instance, that the Evaluation algorithms (EA) discards the old information when a new population is designed, whereas swarm based methods save the information for the subsequent generations. The swarm based algorithms are easy to implement because it uses minimum operators compared to evolutionary approaches.

In this work, the aforementioned problems of ANN are solved with the combination of ANN method and the Evolutionary Algorithms (EA). A hybrid approach is proposed known as GS-ANN with the combination of ANN and Gravitational Search (GS) [7]. It is evident from the literature that the Swarm based algorithm namely Particle Swarm Optimization (PSO) [8] is faster than GS algorithm. The combination of GS and PSO is used to train the ANN which is known as GSPSO. Finally the proposed model is named as GSPSO-ANN. The GSPSO algorithm is proved to be competent in performance by training neural Networks for various datasets [9] [10] of different applications. The problem of ANN with the diversified characteristics of incoming traffic is solved with proposed flow based features (sec 3.2), where the flow based features are independent to the detection model. The whale Optimization Algorithm (WOA) reduces the dimensionality of the dataset by selecting the relevant features from the flow based features set. The performance of the proposed method GSPSO-ANN is validated with the NSL-KDD datasets [11], with the standard performance parameters such as detection rate, mean squared error (MSE), decision time and training time. The results are compared with traditional approaches.

The second section explores the detail analysis of flow based attack detection and swarm & evolutionary algorithms. Third section explains flow level features, whale optimization and GSPSO-ANN algorithm. Finally section 4 explores the results analysis followed by the conclusion.

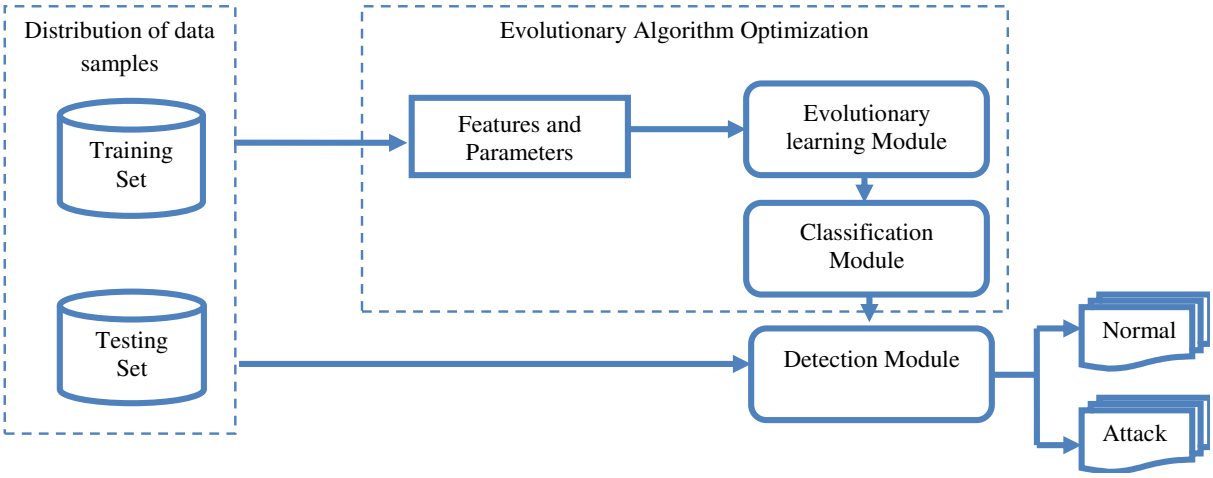

Figure 1: The process of Evaluation-based algorithms

2. RELATED WORK

This section explores the detail study of various flow based application layer DDoS attack detection methods and the evolution of swarm & evolutionary algorithms for attack detection. The detail analysis of the same is presented at the end of this section.

2.1 Review of Flow based intrusion detection

The authors [12] introduced a semi-supervised learning based DDoS attack detection and used entropy calculations to defined the flow size in a given time interval. Whenever the distribution of incoming traffic changes rapidly, then the traffic is divided into three clusters using co-clustering mechanisms. The information gain ratio is calculated to discriminate the abnormal cluster and its associated traffic. The extra-trees algorithm is used to classify the incoming traffic. The drawback of this paper is authors proposed many statistical approaches and these are not capable to handle the dynamic behavior of the traffic.

The authors [13] used packet header features along with the received packets count at a specific time interval as a key parameters to detect the DDoS attacks with Renyi's generalized entropy technique. Entropy index is calculated to discriminate normal traffic over DDoS attacks and the flash events. The limitation of this entropy based DDoS attack detection scheme is, it detects only high rate DDoS attacks.

The Constraint based group testing (GT) mechanism [14] is defined to discriminate the application layer DDoS attacks. The authors proposed partial, sequential and non-adaptive detection schemes to address the application layer DDoS attacks. The advantage of the proposed model is, these detection schemes effectively handled the low and high rate DDoS attacks with the parameters response time and arrival rate. But failed to detect the low rate DDoS attacks in early stages of transmission. The

authors [15] used generalized entropy to address the detection of various low and high rate DDoS attacks. The authors executed D-FACE detection method at ISP level and processed at victim level. Computed generalized entropy index, information distance and traffic rate on captured packet data for discriminating the DDoS attacks. The flow features played a major role while exploiting the detection of low rate DDoS attacks.

The authors [16] introduced a new data structure to detect the application layer DDoS attacks in association with hash tables called as sketch. The distribution of the traffic is stable with randomized hash function for normal traffic. The deviation between the sketches is calculated with Hellinger distance and bloom filters are used to discriminate the normal traffic from the attack traffic. The proposed approach exhibits good detection accuracy for the detection of application layer DDoS attacks and produced high false alarm rates. The punith and mala [17] proposed a flow based intrusion detection system with flow features instead of request level parameters. The HTTP requests are differentiated from the normal DDoS attacks with flow features such as count of GET requests, GET request type and Service time. The low application layer attacks are detected successfully with Support Vector Machine (SVM) model and this model identifies request flood attacks only.

An improved Bio-inspired CUCKOO search algorithm[18] is applied on flow features extracted from the network for validating the traffic at flow level as normal or DDoS attacks. The traffic flow size is calculated with absolute time interval and sessions are described with unique set of features. The feature source diversity ratio is applied to find the diversified sources involved in the time interval. The Authors proved good detection accuracy, but this method is applicable only for HTTP flooding attack.

In [19] the detection of application layer DDoS attacks is implemented with SVM, Machine learning (ML), DBSCAN and K-Means algorithms. The browsing characteristics were used to differentiate the attacks from the normal traffic and used Principle component analysis (PCA) to validate the HTTP requests from the traffic. This combination of algorithms defined in this paper addresses only the flooding attacks. The application layer based GET-flood attacks are discriminated from HTTP traffic with parameters like response index, repetition index, request index and popularity index in web browser. In [20] the attacks are categorized into constant rate attack, repeated attack, flash attack and dominant page attack. The authors kim et al [21] proposed a defense method to mitigate flow based application layer DDoS attacks in wireless ad-hoc networks. The number of requests dumped over the session in a particular time frame is used to validate the flooding attacks. The statistical parameters such as variance and standard deviation are used to measure the deviation of packet count in a time frame. The sources of flooding attack are detected with two detection methods namely relay based and originator based transmissions techniques defined in blacklist. The packet level simulator is used to evaluate the proposed method and failed to address various type of application layer DDoS attacks.

The detection of HTTP based application layer DDoS attacks are implemented with flow feature in distributed environments for diversified traffics. The authors[22] mainly focused on the slow rate DDoS attacks with HTTP requests targeting server HTTP. The bots from the botnet establishes a connections or sessions with incomplete GET or POST methods of HTTP requests to launch the slow rate attacks. The authors presented various HTTP based application layer detections mechanism with their pros and corns and also listed the open issue to the budding researchers. The slow communication rate[23] HTTP based application layer DDoS attacks were launched with application specific protocols by

exploiting the timeout period at server and these slow rate attacks are not application specific. The bots targets the victim server by establishing multiple sessions from the botnet towards the server. It floods the huge volume of keep-alive, broken request and incomplete requests towards the server with low transmission bytes repeatedly in low transmission time.

- The survey reveals that the application layer DDoS attacks block the availability of victim server by flooding the huge amount of requests to the server using HTTP GET/POST methods. The list of challenges to be addressed is given below to increase the detection accuracy with low false alarms in diversified traffic flows.
- Most of the methods presented in the literature are used the packet level or request level features for calculating the detection metrics. However, this increases the algorithm complexity and failed to handle the diversified incoming corpus. The behavior of the traffic is well classified using the flow properties of flow-based intrusion detection system.
- In existing techniques, the diversified behavior of the network traffic is not addressed. When the quantum of incoming traffic is raised then the diversity among the traffic features are also increased. For example two packets with different values for the features represent the same flooding attack.
- The literature reveals that the majority of the researchers proposed detection models in network and transport layer DDoS attacks and very few focused on applications layer DDoS attacks.
- Many researchers proposed machine learning algorithms and metaheurastic algorithms for detecting application layer DDoS attack. However the machine learning algorithms relay on the features used to train the system. The researchers failed to address the diversity of the features while training the system, but this diversification behavior of the features values effect the performance of the detection process. The selection of flow based features rather than request level features will solve the diversity behavior efficiently.

2.2 Review of Swarm and Evolutionary algorithms (EA)

The frequently used Evolutionary algorithms (EA) are Genetic Algorithm (GA), and Genetic Programming (GP) and these Evolutionary algorithms are motivated by the biological process of living creatures. These evolutionary algorithms are also known as population based algorithms. The GA was designed with computational methods to mimic the natural process of generating off-springs for obtaining the optimal solutions to the complex real world problems [25]. The GA is heuristic algorithm which mimics natural selection process to acquire expert solutions for specific problem context by using nature inspired operators such as mutation and cross over. The EA's provides the solutions for single objective problems, multi objective problems, combinational and non-deterministic problems [24]. Evaluating the fitness values for high dimensional problems and Multi model complex problems is time consuming process and also computationally very expensive. Hence the genetic algorithms are failed to solve such problems efficiently. The intrusion detection system (IDS) uses EA's for feature optimization in defining classifiers to classify the attacks.

In [26] GA and Differential Evolution (DE) Evolutionary algorithms are proposed for attack detection. The Evolutionary Algorithms (EA) uses the features and calculates the fitness value to know the significance of it. The classification module of EA is trained with the selected features from the dataset and performs classification with various derived attack patterns. The GNP was designed with fuzzy association rules to handle continues and discrete attributes of the attack dataset [27]. The intrusive

patterns in the traffic or dataset are detected and extracted by designing best rules with directed graph structure. Usually, the combinations of GA's and rule-based systems are called as learning classifier systems [28].

The GA extracted the information from the network flows to generate the attack signatures with network system flow analysis. The fuzzy logic classifier along with the trained classifier pattern is used to detect the specific traffic instance as normal or attack [29]. The Differential Evolution (DE) [30] is introduced to detect the anomalies or attacks from the incoming traffic with selected features in IDS. However, the candidate solutions are obtained by applying self-mutation with large population and new candidate solutions are defined with a weighted difference mechanism among individuals. The authors [31] proposed the comparative analysis of GA, DE and PSO. When the traffic classification is implemented with SVM , these three algorithms are used only for feature selection. In KDD99 dataset among 41 features 16, 15 and 31 features respectively are extracted by the GA, PSO and DE. The DE outperforms PSO and GA in classifying the incoming traffic as attack or normal.

The self-organized clustering [32] mechanism is designed with a hybrid approach using support vector machine (SVM) and Ant Colony Optimization (ACO). This hybrid approach gets advantage of clustering efficiency from self-organized ACO and efficiency of classification from SVM. However, the combination of SVM and ACO is efficient to select the optimal features from the original feature set.

The Particle Swarm Optimization (PSO) is one of the global optimization algorithms with birds flickering behavior or fishes schooling behavior for converging a common goal with group of agents by perceiving the feedback from others in a swarm [33]. However, the swarm is a collection of distributed agents of similar behavior, which interact each other to obtain optimal solution. The PSO is normally used to provide the solutions for non-linear problems, discontinuous problems, and non-differentiable problems. The Ant Bee colony (ABC) algorithm for IDS is used in combination with the learning algorithms for better classification of incoming traffic as abnormal or attack traffic. In [34] the SVM parameters are identified with standard ABC algorithm and optimum feature set is obtained using binary ABC algorithm. The fitness value is defined as an accuracy rate.

The firefly Algorithm (FA) has been used in [35] for selection of optimized features from the dataset to improve the detection accuracy of classifying DDoS attacks by improving the efficiency of the classifier. The accurate detection of network anomalies or attacks is attained by minimizing cluster initialization problem with the combination of k-harmonic means algorithm and FA. The initialization of k-cluster problem is optimized with the combination of FA and k-harmonic means algorithm. The echolocation behavior of bats is the main inspiration for designing the Bat algorithm (BA) and it provides efficient solution to solve single objective and multi objective optimization problems. The BA is used in IDS for feature optimization process, defining the classifier parameters, detecting attacks and classified these into proposed attack classes. In [36] the BA is used to optimize the parameters and SVM kernel parameters. The information gain algorithm along with BA is used to define the optimized feature set and detecting the DDoS attacks accurately.

In [37] network anomaly-based IDS is introduced, where improved version of BA is combined with SVM. The input parameters of SVM are optimized with normal BA and binary variant of BA is defined as wrapper-based feature selection to select the features from the dataset. The BA algorithm extends the accuracy of the attack detection in both the methods and it is implemented in exploitation

phase and exploration phase. The Proposed method outperformed SVM, general BA and PSO methods, when the experimentation carried with NSL-KDD dataset. In [38] pollination based optimization method is used for feature optimization and the optimized features of an IDS have been utilized as local and global properties of FPA. The dataset is defined as plants, the population is assumed to be the number of samples from the dataset, features are assumed as pollinators in the feature selection process. The two classifiers NB and J48 extract the values for the sleeted features from the incoming traffic. The list of advantages of EA's is given below and the evaluation of various methods is explored in table1;

- The EA's acquire intrinsic parallelism characteristic and easily handles large volume of attack data.
- It provides population of solutions instead of single solution, which is applied for behavior based IDS where user profiles are considered for detecting the attacks or intrusions in the network.
- EA's can be easily retrained and provides the better adaptability. Hence, these algorithms can be easily applied to variable environment conditions for detecting the DDoS attacks.
- The controlling parameters of EA's such as cross over and mutation probabilities changed overtime and these are suitable to extract the rules from the detection system.

Though The EA's are efficient for handling DDoS attack detection, but these also have some limitations which are given below.

- Formulation of fitness function, space search representation and defining feature space for EA's is very difficult task.
- Selection of suitable values for control parameters is very difficult and need appropriate local function to select efficient solutions from search space.
- EA's consumed more number of iterations to calculate the fitness values for converting the optimal solution.

Table 1: Evaluation of various ANN and Deep learning methods

Authors	Method	Advantages	Disadvantages
S. Elsayed, R. Sarker,[39]	The proposed DE model used binomial crossover, simple selection, single vector mutation operator for attack detection and classification.	The model effectively classifies the attack traffic with appropriate selection of features.	It failed to validate the traffic for multi class data and exhibited poor performance.
E.-S. M. El-Alfy [40]	Run-time analysis and MapReduce implementation of Sequential GA and Parallel GA .	The experimental results reveal that the parallel GA is more consistent than sequential GA.	The number of parameters and the volume of incoming traffic play the vital role in calculating the performance.
M. G. Raman, N. Somu [41]	The hybrid approach GA-SVM and SVM-GA is used define the Detection of incoming traffic flow.	The detection accuracy is improved with low false alarm rates by using the feature optimization with GA. The SMM-SA combination provides better scalability and adaptability.	Diversity of the feature values is not addressed and the diversified behavior of the incoming flow failed to maintain the consistency of detection method.
A. H. Hamamoto, L. F. Carvalho [42]	The hybrid approach Fuzzy logic-GA with Binary tournament selection.	The Proposed combination Fuzzy Logic-GA is independent to the detection process and provides the better detection accuracy.	False alarm rate is very high and scalability is very less.
H. M. Rais, [43]	The combination of ACO-SVM is used for feature selection and attack detection in the network.	The classification accuracy of the method is improved with feature selection using ACO.	Time consuming process and unable to handle diversified traffic.
Y. Wan, M. Wang[44]	The combination of Binary ACO and GA for feature optimization and attack detection.	The evolutionary fitness curve is used to define the results and address the traffic diversity.	False alarm rate is high and bench mark datasets are not used to evaluate the proposed method.
M. H. Ali, B. A. D. Al Mohammed [45]	The FLN and PSO combination is used for feature optimization and attack detection.	Attained improved detection accuracy with FLN-PSO combination. The accuracy is proportional to the number of neurons used in the system.	When the volume of the traffic is increased, the performance of the system is decreased and exhibited high false alarm rates.
H. Li, W. Guo, G. Wu[46]	The PSO-RF combination for feature optimization.	Achieved high attack detection rate and classification rate using RF and PSO.	Improved TPR and FPR values.
. Hajisalem, S. Babaie,[47]	The combination of ABC-AFS is proposed for HTTP attack detection.	The computation cost and time is very less and it also addressed the	The performance is proportional to the incoming traffic size.

		diversified web traffic. The false alarm rate is 0.01.	
J. Yang, Z. Ye, L. Yan, W. Gu, R. Wang [48]	The ABC and MNB combination is used for feature optimization and application layer attack detection.	The processing time is very less because of ABC algorithm and provides the better solutions to the diversity of the traffic features.	The detection accuracy is less i.e 90% and high false alarm i.e 10%
S. A. R. Shah,[49]	The combination of SVM-FA for feature optimization and attack detection is proposed.	The processing time very less and provides less false alarms.	Detection accuracy is poor for the high speed networks and not addressed the diversity.
B. Selvakumar, K. Muneeswaran[50]	The combination of C4.5, NB and FA is used for feature optimization and detection of HTTP based flooding attacks.	The computational cost and detection accuracy is improved with feature selection algorithm FA.	The detection accuracy of HTTP flood attacks is very poor and exhibits high false alarms.
A.-C. Enache,[51]	The combination of C4.5, SVM and binary BA is used for feature optimization and application layer DDoS attack detection	The number of iterations is very less in feature optimization which minimizes the time and improves the performance of the process.	False alarms are high and accuracy is poor for large datasets.
W. Park, S. Ahn [52]	The combination of SVM and FPA is used to detect the application layer DDoS attacks.	The detection accuracy is improved with FPA for linear and non-linear classes.	The performance is poor for diversified data and exhibits high false alarms.
Zhang, Y., Li, P., & Wang, X,[53]	The number of hidden layers and DBN input nodes are defined with GA	The detection accuracy is 97% and false alarms are 7%	The performance is not reliable. The performance is inversely proposition to the input volume.

3. PROPOSED WORK

This section explores the flow based Detection of HTTP based Application layer DDoS attacks with Whale Optimization Algorithm (WOA) and Swarm & Evolutionary algorithms GSPSO-ANN. The first subsection presents the frame of proposed application layer DDoS detection, second subsection presents the unique set of proposed flow level features to address the traffic at flow level, Third section defines detailed analysis of Whale Optimization Algorithm (WOA), Fourth section presents the detail explanation of GSPSO-ANN algorithm for traffic classification, finally Evaluation of proposed with experimental results.

3.1 Proposed Detection model Framework

Figure 1 shows the framework of the application layer DDoS attack detection. However, the work of each module in the given framework has been described in the "Experimental results" section (i.e., Sect. 4) for further understanding.

*ANN design phase***:** The network traffic is considered as the input for ANN training and testing process at packet level or request level with N number of attributes. The packet level or request level features are dependent to the performance of the attack detection process. When the volume of the traffic is increased, it is difficult to maintain the consistency in the performance of the detection process with request level features due to the diversity of the transactions. To overcome these limitations, flow level features are described in the section 3.2 and extracted the values for these flow level features from the input traffic.

In this work, ANN is designed with Multilayered ANN also known as Multilayered Perceptron (MLP) and this multilayered ANN contains minimum three layer of nodes such as input layer, hidden layer and output layer. The non-linear activation function is executed by the hidden layers or middle layers and each node is considered as one neuron. The ANN inter-layer weights acted as the input for the hidden layer weights (V) and passed to hidden layers to output layers (W). This MLP is designed as M biases for the input layer to single output layer. The output layer is only one and during the testing or detection phase for the processed input instances a threshold is calculated to publish result of the detection phase. This MLP uses supervised learning approach called back propagation. These MLPs are more frequently used for pattern recognition and classification. The MLPs are also provides the solutions for nonlinearly separable problems.

Training of ANN with dataset: The figure 2 demonstrates the architecture of DDoS attack detection in HTTP flow streams. In the data input step the network traffic dataset is given as an input to training and testing process of ANN to update the knowledge such as V, W, weights and biases with GSPSO algorithm. The values for the flow features are extracted from these input data, because the proposed approach is flow based approach and it avoids the request level or packet level feature dependency. The whale optimization Algorithm (WOA) is adopted for feature selection due to the massive and large scale input dataset with huge volume. This WOA eliminates the unimportant features from flow feature set and reduces the dataset dimensionality to minimize the training and testing time.

The GSPSO algorithm is utilized in two phases of ANN learning. Firstly, the initial parameter weights such as V, W and biases are set for each layer and later these weights are updated for each iteration. The number of first layer nodes is defined based on the number of features selected after feature section phase. In this paper multilayered perceptron is used for classification with the K input layers, N hidden layers and with one output or prediction node. This architecture is denoted as K:N:1 and K X N

numbers of weights are existed with M bias in the hidden layer. The Preprocessing step is defined as a result of the use of GSPSO-ANN as classifier and two stages of processing is implemented. The sigmoid function is shown below.

$$f_{sig}(x) = \frac{1}{1 + e^{-x}}$$

The mean squared error (MSErr) is used as the minimization function and the MSE function is given below. Where NoPtt denotes the number of input instances from raining dataset.

$$MSErr = \frac{1}{NOP} \sum_{i=1}^{NOP} (OP_i - tar_i)$$

Testing ANN: The testing of ANN classifier is performed after completing the training of ANN with training dataset. The predicted output in the testing phase is validated with closest match of any target class and selective action is taken based on this matched output class for the current instance from the testing dataset.

3.2 Defining flow level features

The proposed method of application layer DDoS attack detection is implemented at flow level instead of transaction level. The incoming traffic is processed in flow intervals rather than transactions and the list of such are are given below.

Feature – id	Feature-name	Description
F1	Average byte stream rate per flow	The average amount of bytes transferred in a flow
F2	Average durations flows	The average time duration of each flow
F3	Percentage of Asymmetric flows	The asymmetric flow percentage in each flow of time frame.
F4	Asymmetric flows variation rate in a given time frame	The deviation in asymmetric flow in a time frame.
F5	Percentage of tiny packet flows	The contribution of small length flows in time frame.
F6	Percentage of Client connections	Establishment of connection with the incoming flow or request.
F7	Percentage of messages with urgent or keep-alive data	The contribution of data/keep-alive messages in a specific flow.
F8	Average Packet size	The urgent/keep-alive packet average size in a flow.
F9	Average interval time between messages	Elapsed between two continues keep alive or urgent messages in a given time interval.
F10	Percentage of GET/POST requests	Contribution of GET/POST requests alone in each flow.
F11	Average of GET/POST interval	Elapsed between two continues GET or POST messages in a given time interval.

F12	Ratio of incoming requests	The percentage of incoming requests from individual clients.
F13	Ratio of GET and POST sequence requests	The ratio of GET and POST message combinations in a given flow of request.
F14	Client total service time	The amount of time allocated to a specific client by the server.
F15	Bandwidth consumptions	This feature defines the bandwidth consumed in each session.
F16	Source Diversity Ratio (SR)	The number of sources involved to generate the incoming traffic in a specific time frame.
F17	Average server waiting time	The amount of time sever is waiting to finish the acceptance of client request.

3.3 Extracting the Flow features

The proposed method is a flow based classification method, where the features defined (sec 3.1) for flow or time interval plays the vital role in classifying the incoming traffic as normal or attack traffic. The design and evaluation of these features truly impacts the performance of the detection algorithm. This section explores the evaluation of flow features used to define the behavior of the incoming traffic at flow level as Normal flow or attack flow.

- *Average byte stream rate per unit time:* The attacker floods huge volume of traffic with high bandwidth links in short time to block the network devices. However, it results the average number of bytes transmitted is higher than the normal time. This behavior is addressed with this feature and measures the average byte stream rate from the incoming traffic flows per unit time. The calculation is given as follows.

$$abt = \frac{bt_{Ti_n} - bt_{Ti_{n-1}}}{Ti_n - Ti_{n-1}}$$

- *Average durations of each traffic flow:* In continuation with the above feature it observes the length or duration of each flow and finds the average length or duration of each flow in a given time frame. It plays a vital role to analyze the frequency of flows involved in each time frame and it is calculated as given below.

$$adu = \frac{\sum_{i=0}^{fl\_n_{Ti}} du_i}{fl\_n_{Ti}}$$

- *Percentage of Asymmetric flows in a time frame***:** The symmetric or symmetric nature of the traffic is also one of the parameter which affects the performance of the attack detection. In Normal scenario the traffic in the network from source to destination is in symmetric and it is bi-directional. Hence the percentage of symmetric flows is very high. This feature finds the number of such asymmetric flows in each time frame and also calculates the deviation of this behavior from the normal scenario.

$$perpf = \frac{\sum_{i=0}^{fl\_n_{Ti}} N_{p\_f}}{\sum_{i=0}^{fl\_n_{Ti}} N_{p\_f} + \sum_{i=0}^{fl\_n_{Ti}} N_{s\_f}}$$

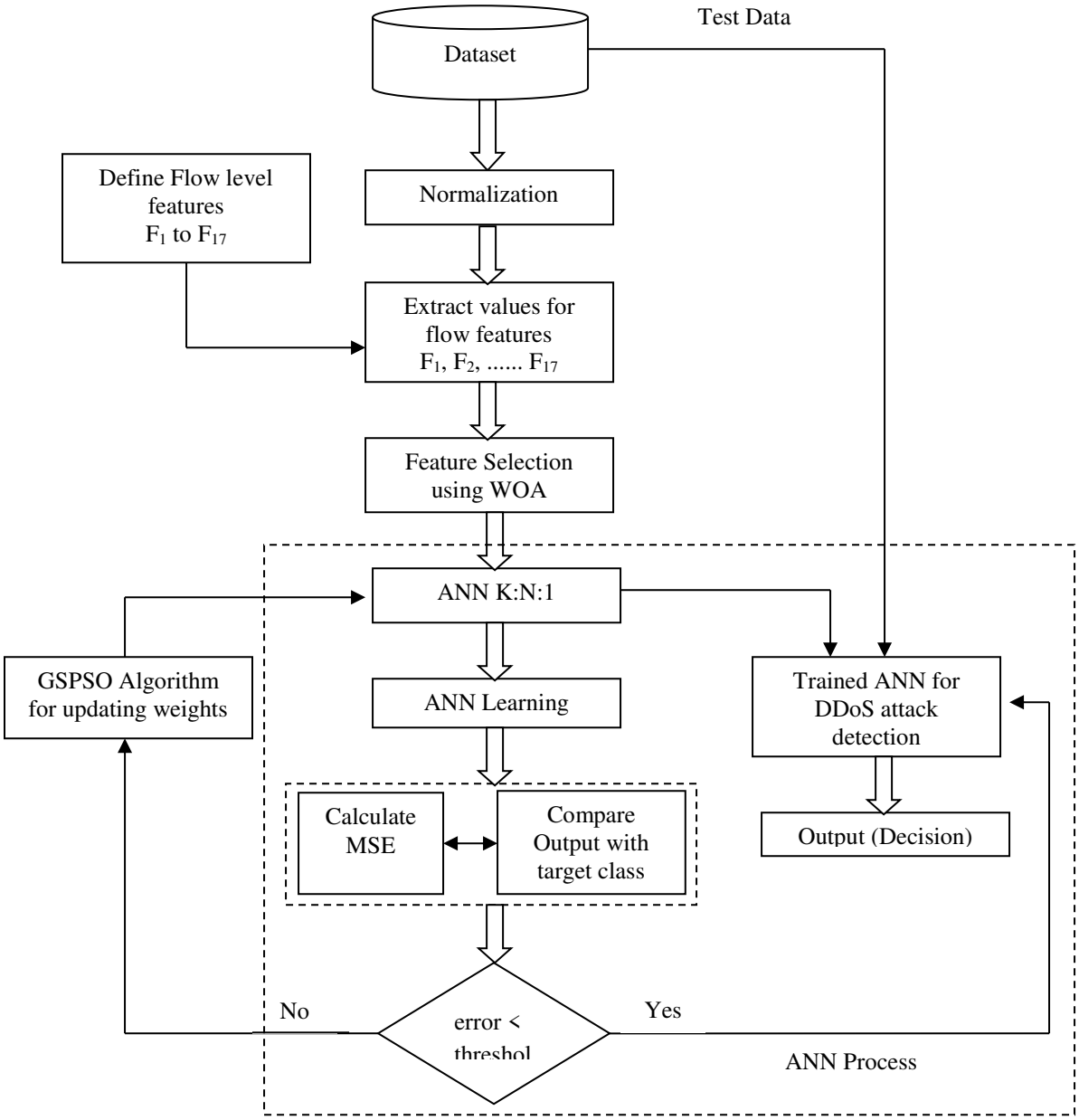

Figure 2: Framework of the proposed DDoS Detection System.

- *Asymmetric flows variation rate in a given time frame:* The abnormal behavior of the network flow is accomplished with asymmetric flows. These one side flows causes the DDoS attack which maximizes the asymmetric flows by minimizing the symmetric flows in the network. This feature address this issue, for genuine traffic the symmetric flows increases proportionally to the incoming traffic over the time frame, but in case of attack traffic the asymmetric flows increases in a short span. When the asymmetric flow of traffic increases drastically then the victim severs failed to give responses to the client requests.

$$varsf = \frac{fl\_n_{Ti} - \sum_{i=0}^{fl\_n_{Ti}} N_{s\_f}}{Ti_n - Ti_{n-1}}$$

- *Percentage of tiny packet flows:* To down the network quickly, the attacker floods more number of packets in a short time interval and this is achieved with huge number of tiny packets in each traffic flow. This feature identifies such behavior by calculating the percentage of tiny packet flows in a given time frame. It is measured as follows.

$$perfl = \frac{\sum Fl_i(N_{pkt} < V)}{fl\_n_{Ti}}$$

- *Percentage of Client connections per flow:* The communication is established with HTTP request and reply. This feature defines the number of connections established with the victim server in a flow and also calculates the average of such connections in all the flows of a given time frame. It is essential to predict the slow rate attacks from the flows and also analyzes the number of client sources involved in the flow to evaluate the source diversity ratio.

For each traffic flow f_k , k=1 to $|f_k|$

$$\sum_{i=1}^{|f|} \frac{\sum_{j=1}^{|c_i|} |r_j|}{|c_i|}$$

- *Percentage of messages with urgent or keep-alive data :* The percentage of keep alive or urgent HTTP data requests towards the victim server from the clients in each flow of specific time frame is measured.

For each traffic flow f_i ,

$$\sum_{i=1}^{|f|} \frac{\sum_{j=1}^{|r_i|} |ur_j|}{|r_i|}$$

- *Average Packet size per flow:* The average of all urgent data or keep-alive message sizes are calculated from the incoming flows to identified the priority based request load in each flow.

For each traffic flow,

$$\sum_{i=1}^{|p|} \frac{\sum_{j=1}^{|p_i|} |p_j|}{|p_i|}$$

- *Average interval time between messages:* To discriminate slow rate HTTP attacks the average interval time between keep alive or urgent messages in a specified flow of incoming traffic is calculated in a given time frame. This is measured if the messages are successive in transaction.

$$\sum_{i=0}^{|s_b|} \frac{(sb_{i+1} - sb_i)}{|s_b|}$$

- *Percentage of GET/POST requests in a flow:* The number of GET/POST requests from each flow is calculated to analyze the percentage of such GET/POST request successes and to find the load on the webserver. This is validated with this feature.

$$\sum_{i=1}^{|r|} \frac{|r_P| + |r_G|}{|r|}$$

- *Average of GET/POST interval in a flow:* This feature finds the amount of time taken by the request for success processing at server in flow and also finds the average time consumed by the request for successive completion in a flow from the incoming traffic.

- *Ratio of incoming requests per flow:* The successive GET/POST requests from the server always creates an healthy environment, whereas the unsuccessful or partial requests blocks the server side buffer and deny the services to the legitimate requests from the genuine users. This feature validates the percentage of such partial or unsuccessful GET/POST requests directed to the victim server from a flow.

For each traffic flow f_k , k=1 to $|f|$

$$\sum_{i=1}^{|f|} \frac{|r_i|}{|f|}$$

- *Ratio of GET and POST sequence requests in a flow:* This feature counts the number of HTTP requests with GET and POST methods in an incoming traffic flow. This count helps to analyze the percentage of load on the server and to study the pattern of access with GET and POST methods in a flow of specified Time frame.

$$\sum_{i=1}^{|f|} \frac{\sum_{j=0}^{|r_i|} \frac{\left|r_{PG_j}\right|}{|r_i|}}{|f|}$$

- *Client total service time per flow:* This feature defines the amount of time blocked the server with HTTP requests to acquire services. It helps to analyze the percentage of load in which each client is contributing for each flow of incoming traffic and also number of diversions among them with diversified characteristics. When the number of clients in each flow of particular time frame is increased, then the diversified characteristics of the flow are increased.

$$\sum_{i=1}^{|f|} \frac{\sum_{j=1}^{|c|} \frac{|c_j|}{|c|}}{|f|}$$

- *Bandwidth consumptions per flow***:** This feature defines the bandwidth consumed in each flow of incoming traffic and calculates the average bandwidth consumed in a specific time frame. This feature defines the load on the channel and severs with incoming requests.

$$\sum_{i=1}^{|f|} \frac{B_i}{|f|}$$

- *Source Diversity Ratio (SR) per flow:* This feature defines the average number of sources involved in flow from incoming traffic of specific time frame.

$$\sum_{i=1}^{|f|} \frac{W_i}{|f|}$$

- *Average server waiting time per flow*: When the HTTP requests are sent in multiple tiny packets , then the waiting time of the server to receive the complete message after receiving the partial message from the flow of incoming traffic is defined as server waiting time. This feature analyzes the server waiting time and average interval time consumed by the client to complete the transaction as multiple tiny packets.

3.4 Whale Optimization Algorithm (WOA) for feature selection

This Section introduces a new meta-heuristic optimization algorithm known as Whale Optimization Algorithm (WOA) [54] which imitates humpback whales hunting behavior. The key difference between the existing bio inspired meta-heuristic approaches and this WOA is, it exhibits best optimization results with its hunting behavior by using best search agent to hunt the prey and with bubble-net attacking behavior. The bubble-net attacking behavior is simulated with spiral approach. The optimization results are exploring that the WOA is competitive that existing methodologies.

Motivation of WOA:

- The whales are highly intelligent animals with emotions than other animals after humans. The spindle cells in the humans and whales makes responsible for emotions, judgment and social behaviors. The whales have twice of these cells quantity than an adult human, which creates its smartness.
- It has been proved that whales can learn, think, judge, communicate with others using own language and emotional like humans, but exhibits less smartness than humans.
- The social behavior of the whales is quite interesting, because they live in groups or alone and some of their species like humpback whales live as a family for their whole life span. The preferred prey for these whales is krill and small fish.
- The hunting behavior of humpback whales is very much interested, because it uses bubble-hunt feeding strategy. These humpback whales hunts the krill or small fishes close to the surface by generating unique bubbles along a circle or “9” fashioned path which is shown in figure 3.
- In this paper we considered the unique bubble-net feeding behavior of humpback whale and modeled mathematically a spiral bubble-net feeding technique to implement optimization.

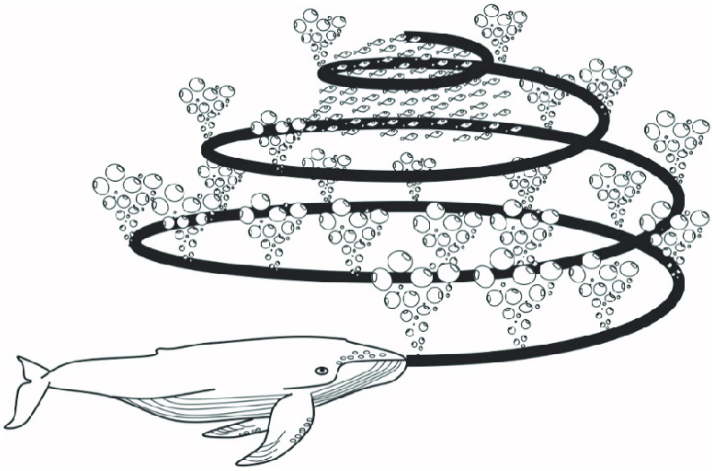

Figure 3. Humpback whales Bubble-net feeding behavior

Mathematical model and optimization algorithm

The hunting process of the humpback whales for krill or small fishes on the surface of the water contains three phases namely encircling prey phase, spiral bubble-net hunt phase and prey hunting phase. This section explores all these phases with mathematical modeling and proposed the whale Optimization Algorithm (WOA).

Encircling prey phase: The humpback whales encircle the prey by identifying the prey location, because in search space the position of the optimal design is not known in advance. It is assumed that the target prey or candidate close to the optimum is the present best solution in WOA algorithm. The remaining search agents update their positions towards the selected best search agent. The following equations demonstrate this behavior.

$$\vec{A}_w\,(t+1) = \vec{A}_m - \vec{U} * \vec{V} \qquad (1)$$

$$V = \left|\vec{P} * \vec{A}_m - \vec{A}_w\right| \qquad (2)$$

$$U = 2\vec{\tau} * \vec{k}_1 - \vec{\tau} \qquad (3)$$

$$\vec{P} = 2.\vec{k_2} \qquad (4)$$

Where the current iteration is represented as, $\vec{A}_m$ denotes the current best solution, $\vec{A}_w$ denotes position vector. The search agents are the whales which updates their positions with reference to the best known solution towards the prey. The updating of the vectors $\vec{U}$ and $\vec{V}$ in the search space controls the whales towards the prey. The parameters $\vec{k}_1$ and $\vec{k}_2$ denotes random values from 0 to 1 and $\vec{\tau}$ value linearly decreased over time from 2 to 0 using the following equation.

$$\tau = 2 - \left[\frac{2t}{MaxIte}\right] \qquad (5)$$

Bubble-net attacking phase: This phase exhibits the bubble-net hunting behavior of the humpback whale and represents the exploitation phase. In exploitation phase the whales randomly search for the prey. To model the bubble-net attacking behavior of the humpback whales mathematically spiral updating method is deployed.

The reduction of encircling behavior of bubble-net hunting is accomplished by minimizing the value of τ with the above equation. The neighbor search agent position is created with the spiral method which is shown in the following equation.

$$\vec{A}_w(i+1) = \vec{V} * e^{bc} * \cos(2\pi c) + \vec{A}_m \quad (6)$$

The humpback whales swim along a spiral path and palely around the prey within a decreased length of circle path. In the optimization process the probability of 50% is considered to select the two mechanisms and this process considers r is a random number from 0 to 1. It is represented in the following equation.

$$\vec{A}_w(i+1) = \begin{cases} \vec{A}_m - \vec{U}\vec{V} & if\ r < 0.5 \\ \vec{V} * e^{bc} * \cos(2\pi c) + \vec{A}_m & if\ r \geq 0.5 \end{cases} \quad (7)$$

Search for prey (exploration) phase: In this phase search randomly for prey based on each other positions by the hunting humpback whales. For this the value of $\vec{U}$ is assigned with a random number between +1 and -1, to force the search agents to transfer to longer positions from the selected search agent. Hence, $\vec{A}_{rand}$ is used to update a search agent position rather than using best agent identified. This process is mathematically modeled as follows.

$$\overrightarrow{A}_w(i+1) = \vec{A}_{rand} - \vec{U}\vec{V} \quad (8)$$

$$V = \left|\vec{P} * \vec{A}_{rand} - \vec{A}_w\right| \quad (9)$$

In Whale Optimization approach (WOA) , whales are defined as the randomly selected features and learning algorithm are used to evaluate the fitness of individual feature subset. The search agent is defined as subset of features with best solution. The best features subset is used to update the other whales' position with bubble-net hunting method. In the next iteration the updated features are used as the whale's position and this process continues repeatedly until the final subset contains best informative subset. The selected features are the input for the detection algorithm to detect the incoming traffic as attack prone or normal traffic. The algorithm of WOA is given in below.

Algorithm : Whale Optimization Algorithm

Input: $DS, \overline{a}$ and $F(x, y)$
Output : $\emptyset$
Initialize $B_{x_{fit}} \leftarrow \infty, B_{y_{fit}} \leftarrow \infty, B_{z_{fit}} \leftarrow \infty, B_x \leftarrow \vec{A}_{rand}, B_y \leftarrow \vec{A}_{rand}, B_z \leftarrow \vec{A}_{rand}$
$n_j \leftarrow \{x_1, x_2, x_3, \ldots\ldots x_m\} \in DS \mid j \in \{1,2,3, \ldots\ldots, m\}$
for $i = 1$ to $MaxIte$ do
  for $j = 1$ to m do

Compute $F(n_j, \bar{a})$
$fn \leftarrow F(n_j, \bar{a})$
$fn \leftarrow F(n_j, \bar{a})$
if $fn < B_{x_{fit}}$ then
$B_{x_{fit}} \leftarrow fn$
$B_x \leftarrow a_{new}$
else if $fn < B_{y_{fit}}$ then
$B_{y_{fit}} \leftarrow fn$
$B_y \leftarrow a_{new}$
else if $fn < B_{z_{fit}}$ then
$B_{z_{fit}} \leftarrow fn$
$B_z \leftarrow a_{new}$
end if
end for
update τ using $2 - \left[\frac{2t}{MaxIte}\right]$
for $j = 1$ to m do
Compute $U = 2\vec{\tau} * \vec{k}_1 - \vec{\tau}$ and $V = \left|\vec{P} * \vec{A}_m - \vec{A}_w\right|$
if $r < 0.5$ then
if $|U| \geq 1$ then
update $\vec{A}_w$ using $\vec{A}_w(i+1) = \vec{A}_{rand} - \vec{U}\vec{V}$
else
$\vec{A}_{rand} \leftarrow B_x$
update $\vec{A}_w$ using $\vec{A}_w(i+1) = \begin{cases} \vec{A}_m - \vec{U}\vec{V} & if\ r < 0.5 \\ \vec{V} * e^{bc} * \cos(2\pi c) + \vec{A}_m & if\ r \geq 0.5 \end{cases}$
end if
else
update $\vec{A}_w$ using $\vec{A}_w(i+1) = \vec{V} * e^{bc} * \cos(2\pi c) + \vec{A}_m$
end if
$n_j \leftarrow \vec{A}_w$
end for
end for
$\emptyset \leftarrow \{B_x, B_y, B_z\}$

3.5 Training of ANN with hybrid algorithms

In this section the combination of GS and PSO is used as a detection classifier. The artificial neural Network (ANN) classifiers are generally trained with back propagation method and many of them block to find the local optimum. This problem is successfully handled with two GS and PSO. The Artificial Neural Network (ANN) is trained with this GSPSO hybrid method. The GSPSO-ANN algorithm is defined below.

The GSPSO system mathematically similar to GS algorithm with isolated system of agents and ensures the Newtonian laws of motion and gravity. More specifically the agents follow the law of gravity and motion [55]. The calculation of various mathematical formulas required to define GSPSO-ANN are given as follows.

Let us assume a system with G agents and the position of the kth agent defined as follows

$$A_k = (a_k^1, a_k^2, \dots, a_k^d .., a_k^g) \text{ for } 1 \le k \le G$$

Where a_k^d denotes k^{th} agent position in the d^{th} dimension.

The gravitational force at time ti on mass x to mass y is represented as follows

$$FN_{xy}^d = GC(ti) \frac{GM_{pX}(ti) * GM_{ay}(ti)}{ED_{xy}(ti) + \epsilon} \left(a_y^d(ti) - a_x^d(ti)\right) \qquad (3)$$

In the above equation (3) $GC(ti)$ denotes the gravitational constant at a time ti, the passive gravitational mass is denoted as GM_{pX} for x , the active gravitational mass for agent y is denoted as GM_{ay}. The GC value and masses for agents in time function is defined using equation (4). The , i.e gravitational constant is denoted as $GC(ti)$ and the initial value for the same is denoted as GC_0 with maximum iteration value MI.

$$GC(ti) = GC_0 * \frac{-\eta * ti}{MI} \qquad (4)$$

where $bt(ti)$ and $wt(ti)$ represents the minimum and maximum fitness values.

The fitness value fv_x is calculated for all possible positions of agents in A, where $x = 1, 2, \dots, G$ and G denotes agents count in the search space. The equations (6)-(9) are defined to calculate the masses of agents with fitness values. Here $wt(ti)$ and $bt(ti)$ symbolizes the maximum and minimum fitness values.

$$GM_{ax} = GM_{px} = GM_{xx} = GM_x ; x = 1, 2, \dots, G \qquad (5)$$

Here GM_{xx} symbolizes agent x initial mass and GM_x denotes the agent x gravitational mass.

$$gm_x(ti) = \frac{fv_x(ti) - wt(ti)}{bt(ti) - wt(ti)} \qquad (6)$$

$$GM_x(ti) = \frac{gm_x(ti)}{\sum_y^G gm_y(ti)} \qquad (7)$$

In equation (6) , the agent x fitness value at time ti is defined as $fv_x(ti)$ and mathematically $bt(ti)$ and $wt(ti)$ for global minimization problem is defined as

$$bt(ti) = \min\{fv_y(ti)\}; \forall y \in \{1,2, \dots \dots, G\} \qquad (8)$$

$$wt(ti) = \max\{fv_y(ti)\}; \forall y \in \{1,2, \dots \dots, G\} \qquad (9)$$

The Euclidean distance ED_{xy} between x and y entities is defined with the following equations.

$$ED_{xy}(ti) = \|A_x(ti), A_x(ti)\|_2 \qquad (10)$$

$$ED_{xy}(ti) = \left(\sum_{p=1}^{g} \left(A_x^p(ti) - A_y^p(ti) \right)^2 \right)^{\frac{1}{2}} \qquad (11)$$

The stochastic characteristic for the GSPSO algorithm is defined as total which act on g agent in d dimension is defined as the randomly weighted sum of the forces extracted from d^{th} components of other agents.

$$FN_x^d(ti) = \sum_{y=1, y \neq x}^{G} \alpha_y FN_{xy}^d(ti) \qquad (12)$$

The randomized characteristic of the search is adopted using α_y (random number) with values ranging from 0 to 1. The mass of the agent and field Fn_g are used to evaluate the acceleration of each agent at time ti.

$$acc_g^d(ti) = \frac{Fn_g(ti)}{GM_g(ti)} \qquad (13)$$

The equations (14)-(16) are used to define the velocity and position of the agent g. Here weighting function is denoted as f , the velocity of agent g at time ti is represented as $V_g\ (ti)$, the accelerations coefficients are $a\acute{c}_1$and $a\acute{c}_2$, the best fitness is shown with gBt, the position of agent g at time ti is represented as $A_g(ti)$, the acceleration of agent g is symbolized as acc_g, two random numbers in the range [0,1] is defines as θ_1 and θ_2.

$$\chi_1 = \left(a\acute{c}_1 * \theta_1 * acc_g * ac_g(ti) \right) \qquad (14)$$

$$\chi_2 = \left(a\acute{c}_2 * \theta_2 * \left(gBt - A_g(ti) \right) \right) \qquad (15)$$

$$V_g\ (ti) = \left(wf * V_g\ (ti) \right) + \chi_1 + \chi_2 \qquad (16)$$

The process of GSPSO is described above and GSPSO algorithm is defined to train the ANN with input parameters as its weights (X , Y and biases).

Algorithm 2 GSPSO–ANN for DDoS attack Detection

procedure GSPSO-ANN-Attack detection
The dataset(attack and normal records) is given as input (training and testing)
The input dataset is Normalized
ANN parameters are Initialized with input data matrix: P, Q, weights (X , Y and biases).
Initialize the following
$MaxIte$, force FO: training parameters,
ma, GM_0:masses of agents,
G_0, inertial weights, $a\acute{c}_1, a\acute{c}_2, d$ etc: initial position of agents,
Initialize with large values : $gBt, gBtSc$
while $iteration\ (ti) \leq Max\ Ite$ do
Gravitational constant $GC(ti)$ is updated using $GC(ti) = GC_0 * \frac{-\eta * ti}{MI}$
for each agents do
Position of agents are used to set weights of ANN (X , Y and biases) using

$$V_g\,(ti) = \left(wf * V_g\,(ti)\right) + \chi_1 + \chi_2$$

for each epoch do
Compute error of current agents (Err).
S = S+ Err
end for
$MSErr = S/NoPtt$
Current Fitness of agent g: $CFA(g) = MSErr$
if $gBtSc > CFA(g)$ then
$gBtSc = CFA(g)$
$gBt = G(g,:)$
end if
end for
for each agents do
Calculate force using $FN_x^d(ti) = \sum_{y=1,y\neq x}^{G} \alpha_y FN_{xy}^d\,(ti)$

Calculate acceleration using $acc_g^d(ti) = \frac{Fn_g(ti)}{GM_g(ti)}$
Update Velocity $\chi_1 = \left(a\acute{c}_1 * \theta_1 * acc_g * ac_g(ti)\right)$
Update new position of agents $G\ using\ V_g\,(ti) = \left(wf * V_g\,(ti)\right) + \chi_1 + \chi_2$
end for
end while
end procedure

In GSPSO algorithm, initially the number of ANN weights is calculated and generated the initial population by computing the agent size as much as the weights calculated in the initial step. Once the first population is generated successfully, then the initial parameters are defined randomly for each agent such as initial positions of agents, velocity, masses of agents, Force F0, initial weights, etc. The fitness of each agent is calculated in the next step. The algorithm continues until it reached the maximum iterations or attained the error threshold. The parameters such as agent position, velocity and each agent masses are updated to generate the new set of agents or solutions and again calculated the fitness of each agent for nest iteration. In the next step the best solution Global and each agent best solution are updated. The mathematical formula for updating agent position, velocity, masses calculations and other formula required for GSPSO-ANN are explained in the above. The GSPSO method is adopted because of its local search skills of GS and abilities of PSO in social movement behavior.

4. EXPERIMENTAL RESULTS

4.1 Attack datasets

The proposed GSPSO-ANN with WOA is evaluated with NSL-KDD [56] and CSE-CIC-IDS2018 [57] datasets.

NSL-KDD dataset: The commonly used dataset for validating any intrusion detection system is NSL-KDD and each record in this dataset is represented with 41 features. The dataset addresses 24 type of attacks and classified into four classes namely Denial of service (DoS) attack, Probe attack, User Root (U2R) attack and Remote Local (R2L) attack. The DoS attack blocks the resources which are not

available to the legitimate users; Probe attack collects the confidential information to get the higher level privileges, U2R attack try to grab the root information with fake credentials to exploit the resources and R2L attack trying to grab the local system information to access the local system and resources. The classification of features in NSL-KDD is defined in table 2.

Table 2: NSL-KDD feature Description

Features	**Description**
1-9	Primary features related to Network Connection
10-22	Content-related features
23-31	Time Related features
32-41	Host Related features

CSE-CIC-IDS2018 dataset : In 2018 the Canadian Institute of Cyber Security (CIC) created an Intrusion Detection Dataset on AWS (Amazon Web Services) known as CSE-CIC-IDS2018 [39]. The real time attacks were launched and collected as CSE-CIC-IDS2018. It is a publicly available dataset and updated from CSE-CICIDS2017 dataset and consists of various known attack types by following necessary standards. It contains six types of attack related datasets namely Botnet, Brute force, Web Attacks, DoS/DDoS and infiltration. The list of 83 features of each sample is given in the table3 as follows. The number of requests considered for both training and testing is given in table 4.

Table 3: CSE-CIC-IDS2018 feature Description

Features	Description
1-4	Primary features related to Network Connection
5-16	Network packet related features
17-22	Network Flow related packets
23-45	Network Flow related statistical features
46-63	Content-related traffic features
64-67	Network subflows related features
68-79	General purpose traffic features
80-83	Primary features related to Network Connection

Table 4. Distribution of the benchmark datasets

Dataset name	Traffic class	Size of Training dataset	Size of Testing dataset	Total Size of dataset
NSL-KDD	Normal	61956	8934	70890
	DoS	42253	6861	49114
	R2L	915	2656	3571
	Probe	10724	2227	12951
	U2R	480	620	1100
CSE-CIC-IDS2018	Benign	29440	7360	36800
	Bot	14720	3680	18400
	DDoS attack -UDP	1273	318	1592
	DDoS attack - HOIC	14720	3680	18400

	DDoS attacks – HTTP	14720	3680	18400
	DoS attacks goldenEye	14720	3680	18400
	DoS attacks –Hulk	14720	3680	18400
	Bruteforce-SSH	14720	3680	18400
	Bruteforce-FTP	347	281	628
	Bruteforce-Web	1045	652	1697
	Bruteforce-XSS	1067	482	1549
	Infiltration	14720	3680	18400
	SQL Injection	679	251	930

The parameter setting for ANN and training algorithms plays a vital role in performance evaluation of proposed method. The list of parameters used for evaluating the proposed method is given in Table5. These parameters are considered based on the various machine learning applications defined earlier with GS and PSO.

Table 5: Parameters assumed in GSPSO evolutionary algorithm

Parameter	Value
Number of generations	10
Chromosome Population Size	50
Crossover rate	0.5
Cross over type	Single
Mutation rate	0.3
Mutation type	Uniform
Population size of GSPSO	30
(inertia weight) w.	0.9
G0	0

4.2 Evaluation metrics:

The evaluation of the proposed method is done with True Positive (TP), True Negative (TN), False Positive (FP), False Negative (FN) Precision, Recall, F-Measure and Detection Accuracy values. The True positive is define as the percentage of requests correctly identified s attacks, True Negatives refers that the percentage of requests correctly identified as Normal, False Positives refers the number of requests wrongly identified as attacks and finally false Negatives defines the number requests wrongly identified as Normal requests. The performance metrics are calculated as follows.

Precision: It is a ratio between true positive rate and sum of true positive and false positive. The calculation is given as follows.

$$\text{Precision} = \frac{TP}{\text{TP+FP}}$$

Recall: It quantifies the number of attacks truly detected out of all attack requests from the dataset. The calculation for the same is defined as follows.

$$\text{Recall} = \frac{TP}{\text{TP+FN}}$$

F-measure or balanced F-score: It is quantified based on the harmonic mean of the precision and recall values and it is calculated as follows.

$$\text{F-measure} = \frac{2XPrecision}{\text{Precision+Recall}}$$

Detection Accuracy: It quantifies the incoming traffic by detecting the attack traffic and normal traffic correctly from the given instances of the request set. The calculation of detection accuracy is given s follows.

$$\text{Detection Accuracy} = \frac{TP+TN}{\text{TP+TN+FP+FN}}$$

4.3 Performance of the Whale Optimization Algorithm (WAO) in feature selection

In the proposed approach the feature optimization is implemented with Whale Optimization Algorithm (WAO) for selecting the necessary features by excluding the unnecessary or unimportant features to improve the performance of the detection process. The proposed method is evaluated with NSL-KDD and CSE-CIC-IDS2018 datasets, but these datasets contains records at request level. The NSL-KDD dataset contains the records with 41 features, whereas the CSE-CIC-IDS2018 data contains dataset with 83 features. The features of NSL-KDD and CSE-CIC-IDS2018 datasets doesn't address the diversity of the features and diversified characteristics of the traffic, which will plays the vital role while evaluating the traffic as normal or attacks for the incoming traffic from the distributed environments. Hence, flow based features are proposed to overcome this limitation and address the diversified characteristics of the traffic. The advantage of the flow based features is, these are independent to the detection process and improves the performance. The values for the flow features are extracted from the NSL-KDD and CSE-CIC-IDES2018 datasets and evaluate the performance of detection system with the optimized parameters of the flow features set, which is given in table 6. This section explores the performance of the Whale Optimization Algorithm (WAO) for selection of important features at request level from NS-KDD and CSE-CIC-IDS2018 datasets. The processing time required evaluating the traffic with WAO for the two datasets and flow features are given in figure 4.

Table 6: The performance of the WAO with NSL-KDD and CSE-CIC-IDS2018 datasets

Dataset name	Traffic class	Optimized features	Detection accuracy	Optimized features from flow features set (Total 17 features)	Detection accuracy
NSL-KDD (41 Features)	Normal	8	94.2	6	98.6
	DoS	11	89.4	5	97.6
	R2L	14	91.7	4	99.1
	Probe	7	89.2	6	97.3
	U2R	9	90.8	8	97.3
CSE-CIC-IDS2018 (83 Features)	Benign	27	89.6	7	93.5
	Bot	31	86.3	6	98.5
	DDoS attack -UDP	16	91.4	9	97.3
	DDoS attack -HOIC	32	92.6	5	96.4

	DDoS attacks – HTTP	21	84.9	8	94.7
	DoS attacks goldenEye	18	88.1	6	95.7
	DoS attacks –Hulk	25	89.7	11	97.3
	Bruteforce-SSH	18	92.5	8	96.8
	Bruteforce-FTP	14	95.3	7	98.2
	Bruteforce-Web	19	92.3	5	97.5
	Bruteforce-XSS	13	89.3	6	98.2
	Infiltration	16	87.9	5	97.5
	SQL Injection	19	92.3	4	98.7

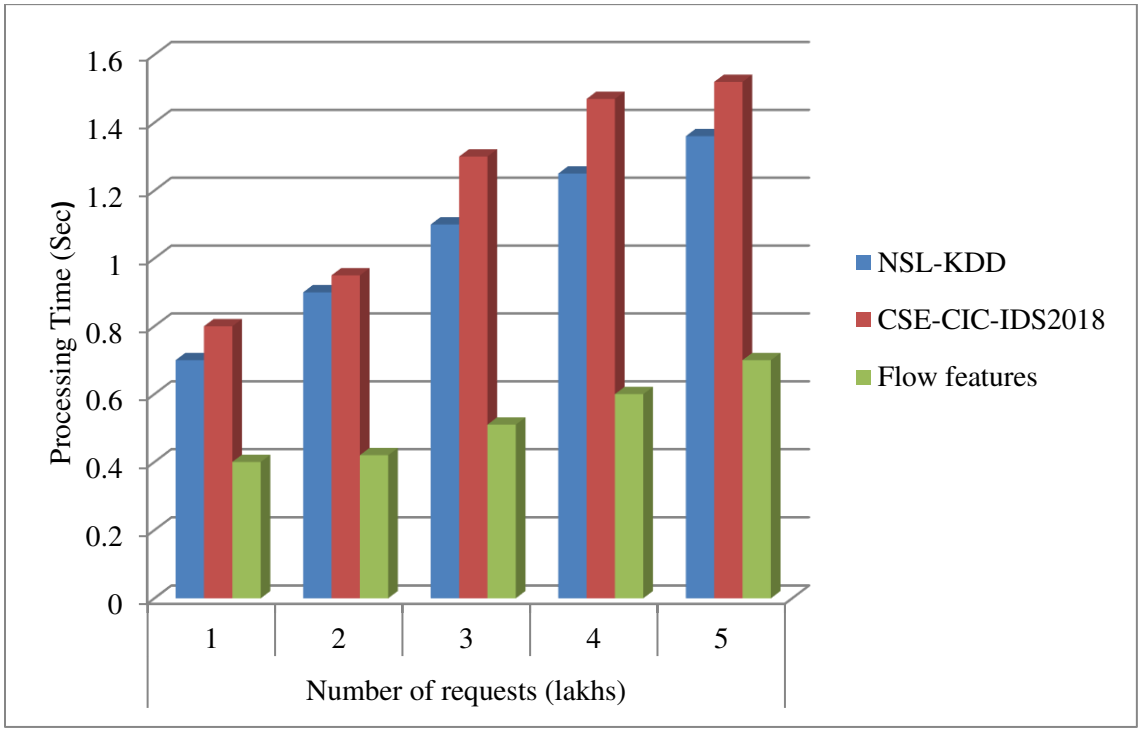

Figure 4: Comparison of Processing Times of various Data sets and flow features

4.4 Performance of the proposed method with different algorithms

The simulation process is repeated for 10 times for all the techniques to calculate the average result for best suitable comparisons. The statistical metrics like mean, standard deviation (std), maximum and minimum are calculated from the ten simulations. The MSE and training time is considered as the performance parameters and the detail evaluation of the same for NSL-KDD and CSE-CIC-IDS2018 training dataset is defined in table 7 and table 8. The MSE and Training time of proposed approach with whale Optimization Algorithm (WOA) for original dataset of NSL-KDD with 41 features and CSE-CIC-IDS2018 dataset of 83 features against the 17 flow features of the same is evaluated and displayed in the table 9 to table 10. The comparison reveals that the GSPSO-ANN outperforms the existing methodologies and also the proposed system further improves the MSE and Training times with WOA. The optimized flow features with WOA exhibited better results than regular packet level features with WOA. For example the MSE for GSPSO-ANN is 0.029, which is very less compared to other methods.

However, this is still minimized to 0.012 with proposed approach and this is further improved with flow level features with WOA for NSL-KDD as 0.09. Hence, the argument is that the proposed method GSPSO-ANN with WAO provides better results at flow level rather than packet or request level features.

Table 7 Mean square Error (MSE) calculated for various methods for NSL-KDD with and packet and flow features with WOA

Detection Method	**Mean square Error (MSE)**							
	NSL-KDD with WOA				*Flow based features of NSL-KDD with WOA*			
	Minimum	*Maximum*	*Mean*	*Standard Deviation*	*Minimum*	*Maximum*	*Mean*	*Standard Deviation*
DT	0.045	0.07	**0.3625**	± 0.009	0.036	0.056	**0.3148**	± 0.0065
GD-ANN	1.156	1.32	**1.1563**	± 0.1125	1.023	1.16	**1.1298**	± 0.0963
GA-ANN	0.101	1.985	**0.5462**	± 0.4953	0.100	1.745	**0.5025**	± 0.4158
PSO-ANN	0.051	1.72	**0.4926**	± 0.5725	0.048	1.586	**0.4654**	± 0.5239
GS-ANN	0.036	1.87	**0.3956**	± 0.5642	0.032	1.74	**0.3645**	± 0.4956
GSPSO-ANN	0.029	1.83	**0.4023**	± 0.5863	0.021	1.65	**0.3856**	± 0.5032
GSPSO-ANN with WOA (Proposed Method)	0.019	1.28	**0.3268**	± 0.4562	0.009	1.423	**0.2589**	± 0.3012

Table 8: Mean square Error (MSE) calculated for various methods for CSE-CIC-IDS2018 Dataset with and packet and flow features with WOA

Detection Method	**Mean square Error (MSE)**							
	CSE-CIC-IDS2018 dataset *with WOA*				*Flow based features of* CSE-CIC-IDS2018 dataset *with WOA*			
	Minimum	*Maximum*	*Mean*	*Standard Deviation*	*Minimum*	*Maximum*	*Mean*	*Standard Deviation*
DT	0.046	0.08	**0.3543**	± 0.01	0.041	0.075	**0.3265**	± 0.009
GD-ANN	1.162	1.36	**1.1546**	± 0.1256	1.158	1.325	**1.1452**	± 0.1156
GA-ANN	0.123	2.03	**0.5029**	± 0.5126	0.105	2.000	**0.4853**	± 0.4953
PSO-ANN	0.0545	1.86	**0.4756**	± 0.5985	0.0512	1.71	**0.4568**	± 0.5259
GS-ANN	0.0349	1.90	**0.4012**	± 0.5763	0.0323	1.83	**0.3945**	± 0.5159
GSPSO-ANN	0.0246	1.89	**0.4265**	± 0.6123	0.0241	1.75	**0.4685**	± 0.5823
GSPSO-ANN with WOA (Proposed Method)	0.156	1.34	**0.3856**	± 0.4963	0.0326	1.523	**0.3026**	± 0.3125

The training time of various methods are evaluated and compared with proposed GSPSO-ANN with WOA method using NSL-KDD dataset and CSE-CIC-IDS2018 dataset with original features at packet or request level with WOA and Flow level features defined with WOA. The proposed method consumed less training time compared to other models. For example the training time of the proposed model with optimized NSL-KDD request level features is 81.45 seconds and optimized CSE-CIC-IDS2018 dataset is 82.63 and with optimized flow features for the same are 68.3 and 70.123 respectively. It is proved that the training time of the proposed method is less for flow features rather than request level features of the given datasets.

Table 9 :Training Time calculated for various methods for NSL-KDD Dataset with and packet and flow features with WOA.

Detection Method	***Training Time (in sec)***							
	NSL-KDD with WOA				*Flow based features of NSL-KDD with WOA*			
	Minimum	*Maximum*	*Mean*	*Standard Deviation*	*Minimum*	*Maximum*	*Mean*	*Standard Deviation*
GD-ANN	196.23	289.54	**245.89**	± 29.56	132.3	201.6	**186.5**	± 18.2
GA-ANN	128.95	169.15	**146.38**	± 12.885	101.1	112.5	**96.5**	± 9.5363
PSO-ANN	74.25	95.4	**86.6**	± 5.4563	52.3	74.6	**56.9**	± 3.2659
GS-ANN	158.5	182.6	**172.5**	± 4.1235	104.9	125.9	**101.2**	± 3.1235
GSPSO-ANN	85.60	115.6	**99.7**	± 9.5863	58.2	84.9	**75.6**	± 6.5638
GSPSO-ANN with WOA (Proposed Method)	70.32	96.53	**81.45**	± 8.536	45.36	72.63	**68.23**	± 5.8962

Table 10 :Training Time calculated for various methods for CSE-CIC-IDS2018 Dataset with and packet and flow features with WOA

Detection Method	***Training Time (in sec)***							
	CSE-CIC-IDS2018 dataset *with WOA*				*Flow based features of* CSE-CIC-IDS2018 dataset *with WOA*			
	Minimum	*Maximum*	*Mean*	*Standard Deviation*	*Minimum*	*Maximum*	*Mean*	*Standard Deviation*
GD-ANN	191.93	310.56	**286.25**	± 29.56	146.23	226.53	**192.3**	± 20.326
GA-ANN	136.54	185.9	**162.59**	± 12.885	109.56	145.3	**106.35**	± 10.265
PSO-ANN	79.65	109.3	**87.6**	± 5.4563	56.9	86.	**63.25**	± 4.2369
GS-ANN	162.87	193.65	**184.62**	± 4.1235	112.36	136.52	**112.56**	± 3.0265
GSPSO-ANN	89.56	129.3	**102.45**	± 9.5863	62.5	95.36	**82.63**	± 7.0235
GSPSO-ANN with WOA (Proposed Method)	69.63	102.63	**82.63**	± 6.963	48.63	75.63	**70.123**	± 6.025

The Detection time of various methods are evaluated and compared with proposed GSPSO-ANN with WOA method using NSL-KDD dataset and CSE-CIC-IDS2018 dataset with original features at packet or request level with WOA and Flow level features defined with WOA is given in the Table 11 and Table 12. The proposed method consumed less Detection time compared to other models. For example the Detection time of the proposed model with optimized NSL-KDD request level features is

0.19 seconds and optimized CSE-CIC-IDS2018 dataset is 0.23 and with optimized flow features for the same are 0.06 and 0.09 seconds respectively. It is proved that the detection time of the proposed method is less for flow features rather than request level features of the given datasets.

Table 11: Testing Time calculated for various methods for NSL-KDD Dataset with and packet and flow features with WOA.

Detection Method	***Detection Time (in sec)***							
	NSL-KDD with WOA				*Flow based features of NSL-KDD with WOA*			
	Minimum	*Maximum*	*Mean*	*Standard Deviation*	*Minimum*	*Maximum*	*Mean*	*Standard Deviation*
DT	1.35	1.47	**1.42**	± 0.0396	0.91	1.02	**0.94**	± 0.0265
GD-ANN	0.71	1.53	**1.09**	± 0.2426	0.52	1.1	**0.89**	± 0.2125
GA-ANN	0.65	1.29	**0.84**	± 0.1865	0.46	0.91	**0.69**	± 0.1536
PSO-ANN	0.82	1.46	**0.96**	± 0.0875	0.63	0.86	**0.72**	± 0.0658
GS-ANN	0.89	1.26	**1.02**	± 0.1126	0.71	0.89	**0.79**	± 0.0936
GSPSO-ANN	0.79	1.39	**0.91**	± 0.1526	0.69	0.92	**0.82**	± 0.1356
GSPSO-ANN with WOA (Proposed Method)	0.12	0.32	**0.19**	± 0.0923	0.05	0.09	**0.06**	± 0.005

Table 12: Detection Time calculated for various methods for CSE-CIC-IDS2018 Dataset with and packet and flow features with WOA.

Detection Method	***Detection Time (in sec)***							
	CSE-CIC-IDS2018 dataset *with WOA*				*Flow based features of* CSE-CIC-IDS2018 dataset *with WOA*			
	Minimum	*Maximum*	*Mean*	*Standard Deviation*	*Minimum*	*Maximum*	*Mean*	*Standard Deviation*
DT	1.42	1.52	**1.44**	± 0.0412	0.95	1.12	**0.99**	± 0.0265
GD-ANN	0.76	1.59	**1.12**	± 0.2625	0.57	1.16	**0.92**	± 0.2125
GA-ANN	0.69	1.35	**0.89**	± 0.2066	0.50	0.98	**0.73**	± 0.1536
PSO-ANN	0.84	1.50	**0.99**	± 0.0925	0.68	0.90	**0.79**	± 0.0658
GS-ANN	0.92	1.29	**1.23**	± 0.1022	0.76	0.99	**0.83**	± 0.0936
GSPSO-ANN	0.81	1.41	**0.99**	± 0.1632	0.72	1.02	**0.89**	± 0.1356
GSPSO-ANN with WOA (Proposed Method)	0.14	0.41	**0.23**	± 0.1226	0.08	0.11	**0.09**	± 0.006

The performance of the proposed work is evaluated with the metrics such as Detection accuracy, precision, recall and F-Measure which is shown in the table 13. From the experimentation it is revealed that the proposed method with flow features attains the maximum Precision, Recall, F-measure, and Accuracy when compared with packet level features of NSL-KDD and CSE-CIC-IDS2018 datasets. The metrics are evaluated for the packet level features of the datasets with WOA and flow level features of the datasets with WOA separately. The detection accuracy of the same is given in figure 5 and figure 6.

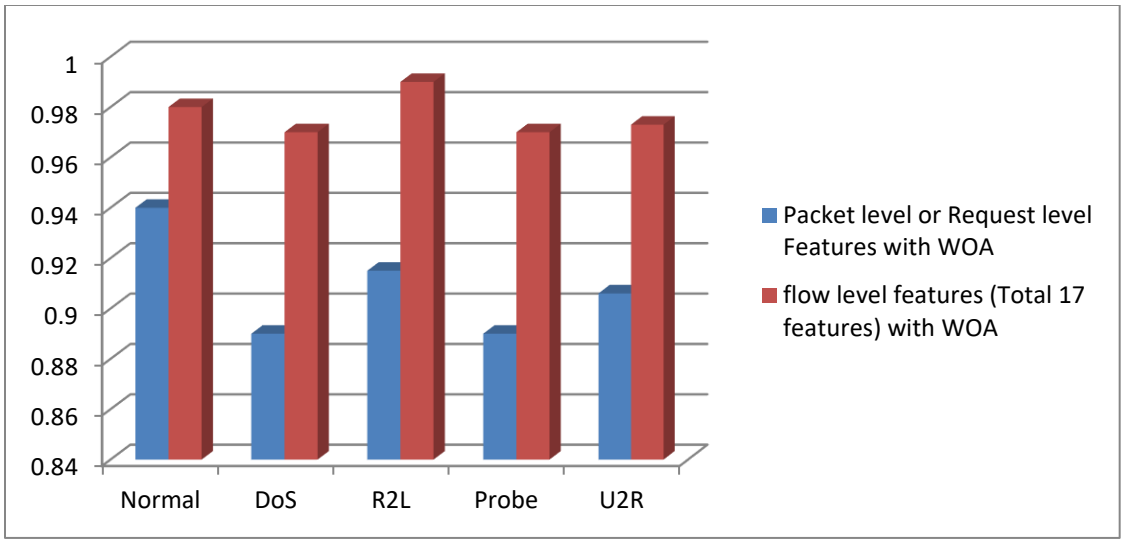

Figure 5: Detection accuracy of the proposed model with NSL-KDD dataset using packet level and Flow level features with WOA.

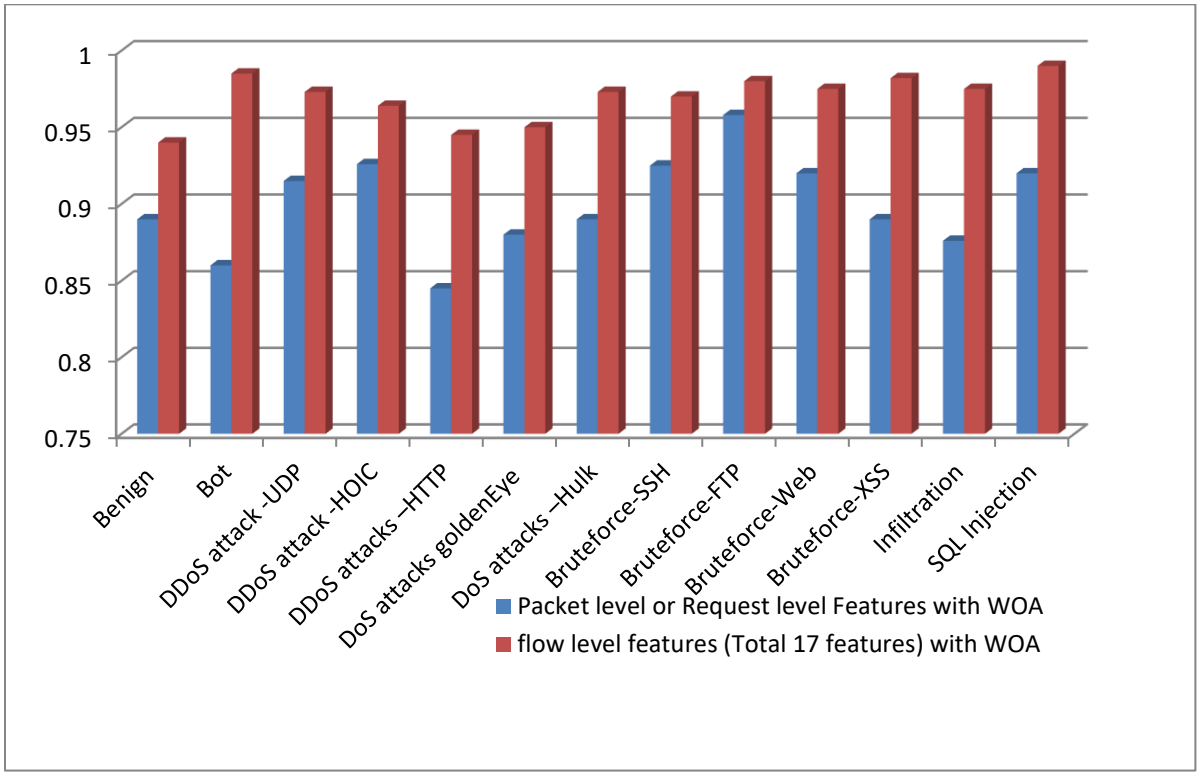

Figure 6: Detection accuracy of the proposed model with CSE-CIC-IDS2018 dataset using packet level and Flow level features with WOA.

4.5 Comparative analysis of the proposed method with existing works:

The detection rate of all the contemporary models is given in the table 14 (a) & table 14(b). The results explored that the detection rate of GSPSO-ANN with WAO is better than GSPSO-ANN and other

methods. From the table it is inferred that the detection accuracy of GSPSO-ANN with WAO at flow level with unique set of flow features exhibit better accuracy than with request or transaction level features. The Whale Optimization Algorithm (WAO) selects the important features both at packet level and flow level. Though the GSPSO-ANN exhibits impressive results, but the WAO improves the performance of the GSPSO-ANN approach. For example, the GSPSO-ANN produces a detection accuracy of 94.32% with detection time as 0.99 seconds, whereas GSPSO-ANN with WAO produced 96.5 % detection accuracy with 0.23 seconds of detection time. However, the proposed approach GSPSO-ANN with WAO at flow level with optimized flow features outperformed with 99.2 % detection accuracy in 0.09 seconds. It is also reliable because it is getting reduced from trapping of local minima. However, the GS-ANN results poor detection accuracy due to exploitation ability of ANN with slow searching ability.

Table 14 (a) Detection Accuracy calculated of various methods for NSL-KDD Dataset with and packet and flow features with WOA

Detection Method	***Detection Accuracy***							
	NSL-KDD with WOA				*Flow based features of NSL-KDD with WOA*			
	Minimum	*Maximum*	*Mean*	*Standard Deviation*	*Minimum*	*Maximum*	*Mean*	*Standard Deviation*
DT	91.2	95.6	**93.4**	± 1.236	93.64	98.65	**94.8**	± 1.1253
GD-ANN	91.9	93.54	**92.6**	± 1.2723	92.5	97.6	**95.63**	± 3.9856
GA-ANN	89.6	93.5	**91.3**	± 1.7652	93.5	98.65	**96.84**	± 2.1256
PSO-ANN	92.56	96.5	**94.36**	± 1.75362	94.62	97.85	**96.63**	± 2.2456
GS-ANN	92.56	96.56	**93.48**	± 2.2369	95.68	98.56	**97.53**	± 2.3891
GSPSO-ANN	92.63	98.13	**94.32**	± 2.659	94.65	99.45	**97.65**	± 1.1356
GSPSO-ANN with WOA (Proposed Method)	94.53	98.6	**96.5**	± 3.456	98.5	99.5	**99.2**	± 0.952

Table 14 (b) : Comparison of Detection Accuracy of various methods for CDE-CIC-IDS2018 Dataset with and packet and flow features with WOA

Detection Method	***Detection Accuracy***							
	CSE-CIC-IDS2018 dataset *with WOA*				*Flow based features of* CSE-CIC-IDS2018 dataset *with WOA*			
	Minimum	*Maximum*	*Mean*	*Standard Deviation*	*Minimum*	*Maximum*	*Mean*	*Standard Deviation*
DT	92.5	94.6	**93.74**	± 1.387	93.26	97.85	**96.86**	± 2.286
GD-ANN	92.65	94.56	**93.56**	± 3.258	91.87	96.25	**95.63**	± 3.4586
GA-ANN	92.9.74	93.65	**93.53**	± 1.986	93.5	95.46	**95.23**	± 3.259
PSO-ANN	93.56	97.53	**95.36**	± 2.365	96.78	97.85	**97.22**	± .4563
GS-ANN	90.256	94.563	**93.58**	± 3.258	95.68	97.25	**96.86**	± 2.9635
GSPSO-ANN	91.85	95.68	**94.2**	± 1.8569	95.84	97.86	**96.85**	± 1.2563
GSPSO-ANN with WOA (Proposed Method)	95.3	98.67	**97.56**	± 3.1234	98.6	99.4	**99.1**	± 1.786

Table 13: simulation of performance metrics for the Proposed method using NSL-KDD and CSE-CIC-IDS2018 datasets with WAO.

Dataset name	Traffic class	Packet level or Request level Features with WOA				flow level features (Total 17 features) with WOA			
		Detection Accuracy	Precession	Recall	F-Measure	Detection Accuracy	Precession	Recall	F-Measure
NSL-KDD (41 Features)	Normal	0.94	0.89	0.88	0.82	0.98	0.92	0.93	0.89
	DoS	0.89	0.843	0.89	0.86	0.97	0.89	0.92	0.93
	R2L	0.915	0.869	0.90	0.89	0.99	0.965	0.98	0.97
	Probe	0.89	0.85	0.86	0.79	0.97	0.93	0.92	0.93
	U2R	0.906	0.88	0.88	0.89	0.973	0.91	0.95	0.94
CSE-CIC-IDS2018 (83 Features)	Benign	0.89	0.827	0.79	0.816	0.94	0.92	0.94	0.96
	Bot	0.86	0.84	0.84	0.828	0.985	0.92	0.98	0.945
	DDoS attack -UDP	0.915	0.86	0.88	0.89	0.973	0.953	0.965	0.983
	DDoS attack -HOIC	0.926	0.95	0.90	0.89	0.964	0.923	0.92	0.89
	DDoS attacks –HTTP	0.845	0.83	0.82	0.815	0.945	0.92	0.893	0.91
	DoS attacks goldenEye	0.88	0.85	0.87	0.82	0.95	0.93	0.912	0.936
	DoS attacks –Hulk	0.89	0.83	0.88	0.819	0.973	0.953	0.925	0.916
	Bruteforce-SSH	0.925	0.93	0.89	0.88	0.97	0.92	0.899	0.935
	Bruteforce-FTP	0.958	0.914	0.92	0.88	0.98	0.935	0.946	0.893
	Bruteforce-Web	0.92	0.88	0.86	0.827	0.975	0.95	0945	0.963
	Bruteforce-XSS	0.89	0.837	0.83	0.857	0.982	0.936	0.942	0.956
	Infiltration	0.876	0.85	0.81	0.84	0.975	0.962	0.923	0.94
	SQL Injection	0.92	0.88	0.90	0.86	0.99	0.956	0.943	0.958

The comparison of precision, recall and F-measure of the proposed method with the contemporary methods with NSL-KDD and CSE-CIC-IDS2018 datasets are given in table 15 , figure 7 and figure 8. The features selection process is employed in the proposed approach with WAO and which reduces the training and testing time as shown in the above tables. The proposed method exhibited better values for precision, recall and F-Measure compared to contemporary methods.

Table 15; Comparison of Precision, Recall, F-Measure of the proposed approach with other methods

Detection Method	***Detection Accuracy***					
	NSL-KDD dataset with WOA			CSE-CIC-IDS2018 dataset *with WOA*		
	Precision	*Recall*	*F-measure*	*Precision*	*Recall*	*F-measure*
DT	0.823	0.756	0.816	0.856	0.806	0.796
GD-ANN	0.819	0.796	0.835	0.819	0.796	0.856
GA-ANN	0.845	0.842	0.795	0.847	0.827	0.825
PSO-ANN	0.825	0.835	0.846	0.817	0.829	0.835
GS-ANN	0.862	0.856	0.862	0.858	0.848	0.875
GSPSO-ANN	0.912	0.905	0.915	0.926	0.913	0.926
GSPSO-ANN with WOA (Proposed Method)	0.936	0.956	0.958	0.946	0.946	0.968

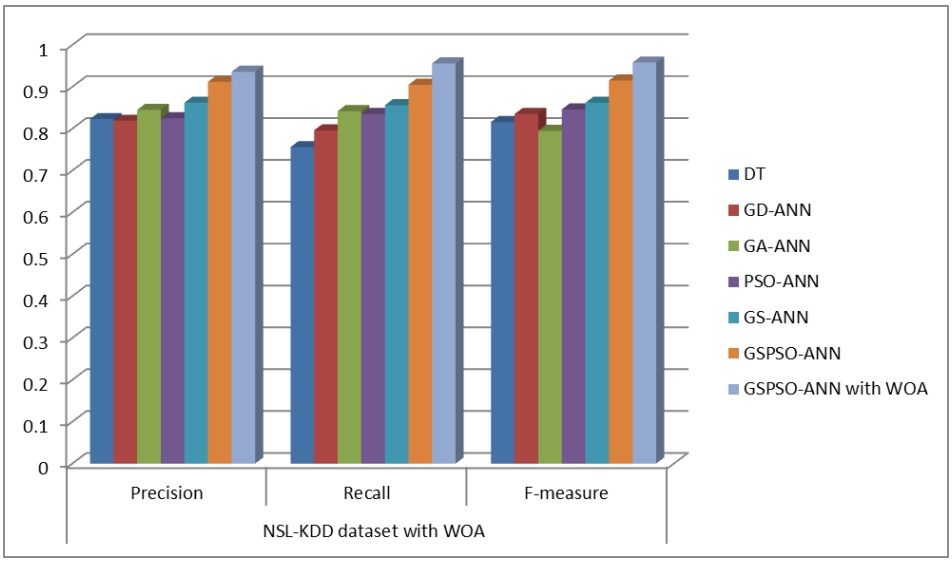

Figure 7: Comparison of Precision, Recall and F-measure with Contemporary methods (NSL-KDD dataset)

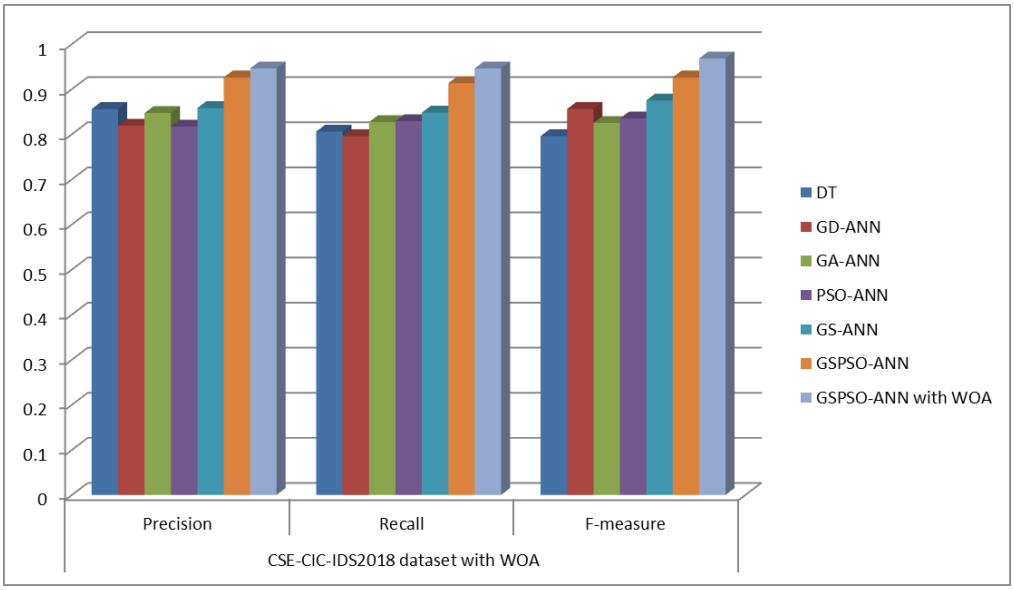

Figure 8: Comparison of Precision, Recall and F-measure with Contemporary methods (CSE-CIC-IDS2018 dataset)

The proposed method exhibits low false alarm rate, when compared with the contemporary methods. The false alarm rate of the proposed method and comparison with the existing methodologies are shown in figure 9.

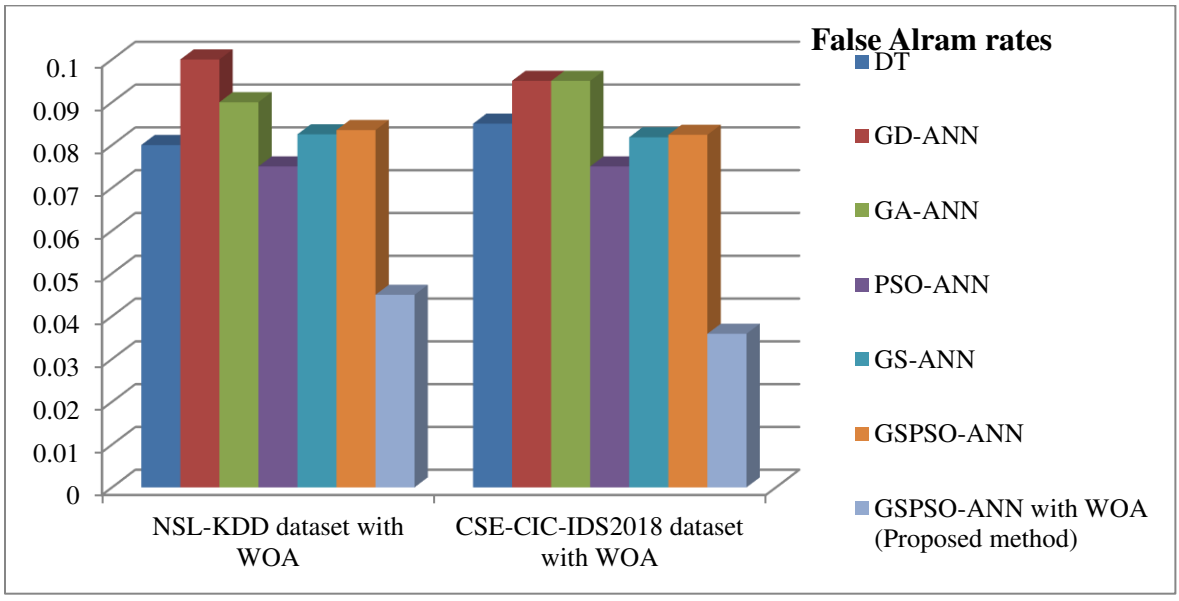

Figure 9: Comparison of false alarm rates with Contemporary methods.

4.6 Performance of the proposed method with diversified flows or diversified characteristics

Though the performance of the GSPSO-ANN is satisfactory, but it failed to maintain the same for the diversified flows from the distributed environments or distributed characteristics of the features used to represent the transaction or the request. One of the key arguments is that the existing methodologies

such as DL, GS-ANN,GD-ANN,GA-ANN and GD-PSO results better performance for homogeneous and neglected the diversity of the data. However, most of the machine learning, deep learning and ANN algorithms relay on the features selected to train the system. The diversity of the data plays a prominent role while training the system with selected features and for example attack request can be designed with multiple values for the selected features. The existing methods neglected the diversity and produced the results as efficient. In this paper the diversity of the traffic or data is addressed with flow level features rather than request level, because request level or transaction level features are always process dependent.

The flow features are independent to the methodology and efficiently handles the diversity of the requests. The feature source diversified ratio evaluates the diversity of the incoming traffic. This section explores the performance of the proposed GSPSO-ANN with WOA for diversified traffic and the comparison with the existing methodologies. The comparison of detection accuracy for diversified traffic is given in table 16 and table 17. The performance of the proposed system for various diversified values is given in figure 10 and figure 11.

Table 16: Comparison of detection accuracy at different diversified values for NSL-KDD dataset

Detection Method	***Detection Accuracy (Diversified flows)***				
	NSL-KDD dataset with WOA				
	Diversion d=0.2	*Diversion d=0.4*	*Diversion d=0.6*	*Diversion d=0.8*	*Diversion d=1.0*
DT	0.93	0.92	0.90	0.87	0.85
GD-ANN	0.91	0.905	0.895	0.88	0.84
GA-ANN	0.92	0.91	0.90	0.87	0.83
PSO-ANN	0.90	0.90	0.885	0.86	0.81
GS-ANN	0.93	0.92	0.905	0.885	0.855
GSPSO-ANN	0.92	0.915	0.896	0.875	0.834
GSPSO-ANN with WOA (Proposed Method)	0.96	0.959	0.945	0.935	0.92

Table 17: Comparison of detection accuracy at different diversified values for CSE-IS-IDS2018 dataset

Detection Method	***Detection Accuracy (Diversified flows)***				
	CSE-CIC-IDS2018 dataset with WOA				
	Diversion d=0.2	*Diversion d=0.4*	*Diversion d=0.6*	*Diversion d=0.8*	*Diversion d=1.0*
DT	0.95	0.935	0.912	0.895	0.854
GD-ANN	0.94	0.925	0.905	0.875	0.853
GA-ANN	0.935	0.918	0.906	0.886	0.845
PSO-ANN	0.925	0.915	0.90	0.895	0.835
GS-ANN	0.945	0.935	0.915	0.905	0.874
GSPSO-ANN	0.935	0.924	0.915	0.895	0.856
GSPSO-ANN with WOA (Proposed Method)	0.99	0.975	0.965	0.96	0.945

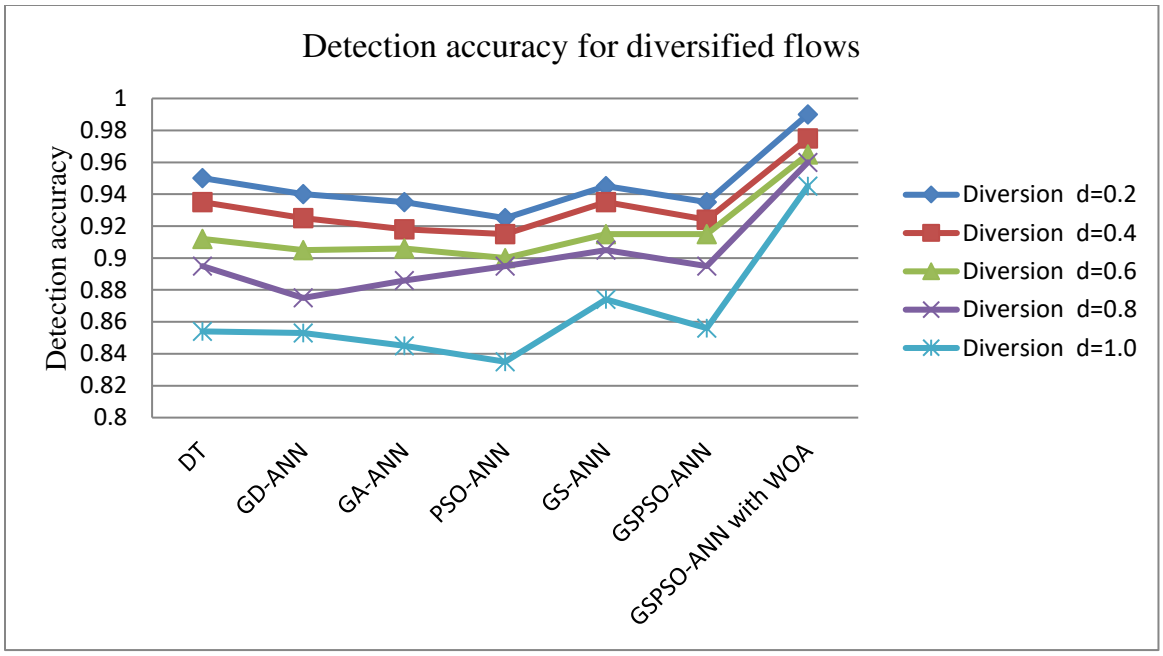

Figure 10: Comparison of detection accuracy with other methods for diversified NSL-KDD dataset

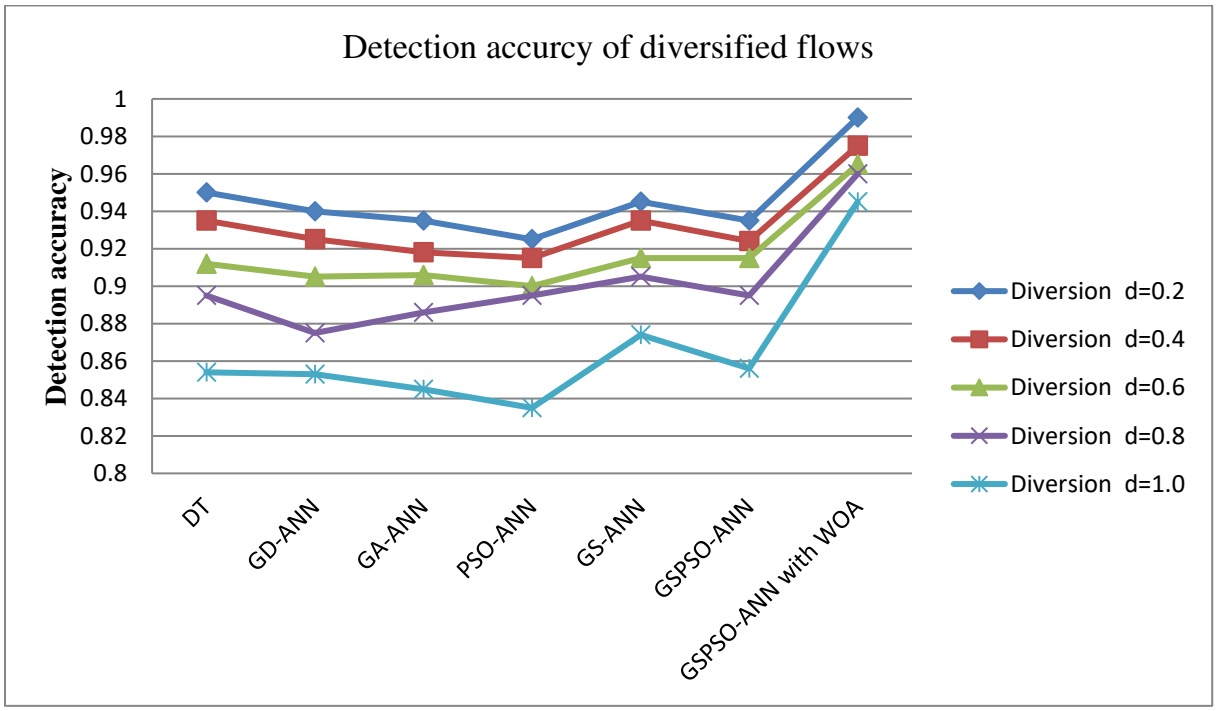

Figure 11: Comparison of detection accuracy with other methods for diversified CSE-CIC-IDS2018 dataset

Conclusion

In this paper, we introduced a new hybrid method with high-performed evolutionary algorithms and neural networks for attack classification at flow level rather than at request level with low false alarm rates and high detection accuracy. A unique set of flow features are defined to handle the traffic at flow level and optimal feature selection using whale Optimization Algorithm (WOA). The gravitational search (GS), and particle swarm optimization (PSO) combinations are used in attack detection phase to train the ANN and results proposed model as GSPSO-ANN with WOA. The performance of the proposed model is evaluated with NSL-KDD and CSE-CIC-IDS2018 datasets. The results are compared with other ANN based conventional methods like Decision Tree (DT), GS-ANN, gradient descent (GD-ANN), genetic algorithm with ANN (GA-ANN), ANN with particle Swarm Optimization (PSO-ANN), and GSPSO-ANN. The results inferred that the proposed GSPSO-ANN with WOA attained maximum detection accuracy with low false alarm rates and processing time and also maintained consistency in the performance for diversified traffic.

Declarations:

Conflict of interest: NO CONFLICT OF INTEREST FROM THE AUTHORS

Data availability: The dataset is available in http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html . and http://www.unb.ca/cic/datasets/ids-2018.html

References

[1] Prasad, Dr.A.Rama Mohan Reddy, Dr K.Venugopal Rao, K. (2014). DoS and DDoS Attacks: Defense, Detection and Traceback Mechanisms - A Survey. *Global Journal Of Computer Science And Technology,* https://computerresearch.org/index.php/computer/article/view/1081.

[2] Aggarwal A, Gupta A (2015) Survey on data mining and IP traceback technique in DDoS attack. Int J Eng Comput Sci 4:12595–12598.

[3] Ahmed L, Iqbal MM, Aldabbas H et al (2020) Images data practices for semantic segmentation of breast cancer using deep neural network. J Ambient Intell Human Comput. https://doi.org/10.1007/ s12652-020-01680-1.

[4] Ahmed L, Iqbal MM, Aldabbas H et al (2020) Images data practices for semantic segmentation of breast cancer using deep neural network. J Ambient Intell Human Comput. https://doi.org/10.1007/ s12652-020-01680-1.

[5] Singh, K., Singh, P. and Kumar, K. (2017) 'Application layer HTTP-get flood DDoS attacks: research landscape and challenges', Computers & Security, Vol. 65, pp.344–372, doi: https://doi.org/10.1016/j.cose.2016.10.005.

[6] Diro, A. A., & Chilamkurti, N. (2018). Distributed attack detection scheme using deep learning approach for Internet of Things. Future Generation Computer Systems, 82, 761–768.

[7] Rashedi E, Nezamabadi-pour H, Saryazdi S (2009) GSA: a gravitational search algorithm. Inf Sci 179:2232–2248.

[8] Eberhart R, Kennedym J (1995) A new optimization using particle swarm theory. In: Sixth international symposium on micro machine and human science, MHS'95, IEEE, pp 39–43

[9] Mirjalili S, Hashim SZM, Sardroudi HM (2012) Training feedforward neural networks using hybrid particle swarm optimization and gravitational search algorithm. Appl Math Comput 218:11125– 11137.

[10] Dash T, Nayak SK, Behera HS (2015a) Hybrid gravitational search and particle swarm based fuzzy MLP for medical data classification. In: Computational intelligence in data mining, vol 1. Springer, India, pp 35–43.

[11] Tavallaee M, Bagheri E, LuW, Ghorbani A (2009) A detailed analysis of the KDD CUP'99 dataset. In: Proceedings of the IEEE symposium on computational intelligence for security and defense applications, pp 53–58.
[12] dhammad, M., Afdel, K. and Belouch, M. (2018) 'Semi-supervised machine learning approach for DDoS detection', Applied Intelligence, Vol. 48, No. 10, pp.3193–3208.
[13] Behal, S., Kumar, K. and Sachdeva, M. (2018b) 'A generalized detection system to detect distributed denial of service attacks and flash events for information theory metrics', Turkish Journal of Electrical Engineering & Computer Sciences, Vol. 26, No. 4, pp.1759–1770.
[14] Xuan, Y., Shin, I., Thai, M.T. and Znati, T. (2010) 'Detecting application denial-of-service attacks: a group-testing-based approach', IEEE Transactions on Parallel and Distributed Systems, Vol. 21, No. 8, pp.1203–1216.
[15] Behal, S., Kumar, K. and Sachdeva, M. (2018a) 'D-face: an anomaly based distributed approach for early detection of DDoS attacks and flash events', Journal of Network and Computer Applications, Vol. 111, pp.49–63, doi: https://doi.org/10.1016/j.jnca.2018.03.024.
[16] wang, C., Miu, T.T.N., Luo, X. and Wang, J. (2018) 'Skyshield: a sketch-based defense system against application layer DDoS attacks', IEEE Transactions on Information Forensics and Security, Vol. 13, No. 3, pp.559–573.
[17] Punitha, V. and Mala, C. (2018) 'SVM based traffic classification for mitigating http attack', in 2018 The 3rd International Symposium Mobile Internet Security (MobiSec 18), KIISC Research Group on 5G Security, University of San Carlos, No. 10, pp.1–9.
[18] Munivara Prasad, K., Rama Mohan Reddy, A. and Venugopal Rao, K. (2017) 'BIFAD: bio-inspired anomaly based http-flood attack detection', Wireless Personal Communications, Vol. 97, No. 1, pp.281–308.
[19] Luo, X., Di, X., Liu, X., Qi, H., Li, J., Cong, L. and Yang, H. (2018) 'Anomaly detection for application layer user browsing behaviour based on attributes and features', in Journal of Physics: Conference Series, IOP Publishing, Vol. 1069, p. 012072.
[20] Singh, K., Singh, P. and Kumar, K. (2018) 'User behavior analytics-based classification of application layer HTTP-get flood attacks', Journal of Network and Computer Applications, Vol. 112, pp.97–114, doi: https://doi.org/10.1016/j.jnca.2018.03.030.
[21] Kim, J. and Bohacek, S. (2018) 'Efficient modeling of network flooding performance with proactive retransmissions in mobile ad hoc networks', Wireless Networks, Vol. 25, No. 5, pp.2423–2436.
[22] Mantas, G., Stakhanova, N., Gonzalez, H., Jazi, H.H. and Ghorbani, A.A. (2015) 'Application-layer denial of service attacks: taxonomy and survey', International Journal of Information and Computer Security, Vol. 7, Nos. 2–4, pp.216–239.
[23] Cambiaso, E., Papaleo, G. and Aiello, M. (2017) 'Slow comm: design, development and performance evaluation of a new slow dos attack', Journal of Information Security and Applications, Vol. 35, pp.23–31, doi: https://doi.org/10.1016/j.jisa.2017.05.005.
[24] H. Aytug, M. Khouja, F. Vergara, Use of genetic algorithms to solve production and operations management problems: a review, International Journal of Production Research 41 (17) (2003) 3955–4009
[25] J. H. Holland, Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence, MIT press, 1992.
[26] S. Binitha, S. S. Sathya, et al., A survey of bio inspired optimization algorithms, International Journal of Soft Computing and Engineering 2 (2) (2012) 137–151.
[27] S. Mabu, C. Chen, N. Lu, K. Shimada, K. Hirasawa, An intrusion detection model based on fuzzy class-association-rule mining using genetic network programming, IEEE Transactions on Systems, Man, and Cyber1550 netics, Part C (Applications and Reviews) 41 (1) (2011) 130–139.
[28] K. Shafi, H. A. Abbass, An adaptive genetic-based signature learning system for intrusion detection, Expert Systems with Applications 36 (10) (2009) 12036–12043.
[29] A. H. Hamamoto, L. F. Carvalho, L. D. H. Sampaio, T. Abrao, M. L. Proen¸ca Jr, Network anomaly detection system using genetic algorithm and fuzzy logic, Expert Systems with Applications 92 (2018) 390–402.

[30] A. Zamuda, J. Brest, Self-adaptive control parameters randomization frequency and propagations in differential evolution, Swarm and Evolutionary Computation 25 (2015) 72–99.
[31] S. Zaman, M. El-Abed, F. Karray, Features selection approaches for intrusion detection systems based on evolution algorithms, in: Proceedings of the 7th International Conference on Ubiquitous Information Management and Communication, ACM, 2013, p. 10.
[32] . Feng, Q. Zhang, G. Hu, J. X. Huang, Mining network data for intrusion detection through combining SVMs with ant colony networks, Future Generation Computer Systems 37 (2014) 127–140.
[33] . Shi, R. C. Eberhart, Parameter selection in particle swarm optimization, in: International conference on evolutionary programming, Springer, 1998, pp. 591–600.
[34] . Wang, T. Li, R. Ren, A real time IDSs based on artificial bee colonysupport vector machine algorithm, in: Advanced computational intelligence (IWACI), 2010 third international workshop on, IEEE, 2010, pp. 91–96.
[35] M. H. Adaniya, M. F. Lima, J. J. Rodrigues, T. Abrao, M. L. Proen¸ca, Anomaly detection using dsns and firefly harmonic clustering algorithm, in: Communications (ICC), 2012 IEEE International Conference on, IEEE, 2012, pp. 1183–1187.
[36] A.-C. Enache, V. Sgarciu, Anomaly intrusions detection based on support vector machines with an improved bat algorithm, in: Control Systems and Computer Science (CSCS), 2015 20th International Conference on, IEEE, 1850 2015, pp. 317–321.
[37] A.-C. Enache, V. Sgarciu, An improved bat algorithm driven by support vector machines for intrusion detection, in: International Joint Conference, Springer, 2015, pp. 41–51.
[38] S. Dubb, Y. Sood, Feature selection approach for intrusion detection system based on pollination algorithm, International Journal of Advanced Engineering Research and Technology 2016 4 (6).
[39] S. Elsayed, R. Sarker, J. Slay, Evaluating the performance of a differential evolution algorithm in anomaly detection, in: 2015 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2015, pp. 2490–2497.
[40]] E.-S. M. El-Alfy, M. A. Alshammari, Towards scalable rough set based attribute subset selection for intrusion detection using parallel genetic algorithm in MapReduce, Simulation Modelling Practice and Theory 64 (2016) 18–29.
[41] M. G. Raman, N. Somu, K. Kirthivasan, R. Liscano, V. S. Sriram, An efficient intrusion detection system based on hypergraph-Genetic algorithm for parameter optimization and feature selection in support vector machine, Knowledge-Based Systems 134 (2017) 1–12.
[42] A. H. Hamamoto, L. F. Carvalho, L. D. H. Sampaio, T. Abr˜ao, M. L. Proen¸ca Jr, Network anomaly detection system using genetic algorithm and fuzzy logic, Expert Systems with Applications 92 (2018) 390–402.
[43] H. M. Rais, T. Mehmood, Dynamic Ant Colony System with Three Level Update Feature Selection for Intrusion Detection, IJ Network Security 1645 20 (1) (2018) 184–192.
[44] Y. Wan, M. Wang, Z. Ye, X. Lai, A feature selection method based on modified binary coded ant colony optimization algorithm, Applied Soft Computing 49 (2016) 248–258.
[45] M. H. Ali, B. A. D. Al Mohammed, A. Ismail, M. F. Zolkipli, A new intrusion detection system based on fast learning network and particle swarm optimization, IEEE Access 6 (2018) 20255–20261.
[46]] H. Li, W. Guo, G. Wu, Y. Li, A RF-PSO Based Hybrid Feature Selection Model in Intrusion Detection System, in: 2018 IEEE Third International Conference on Data Science in Cyberspace (DSC), IEEE, 2018, pp. 795– 802.
[47] Hajisalem, S. Babaie, A hybrid intrusion detection system based on ABC-AFS algorithm for misuse and anomaly detection, Computer Networks 136 (2018) 37–50.
[48] J. Yang, Z. Ye, L. Yan, W. Gu, R. Wang, Modified Naive Bayes Algorithm for Network Intrusion Detection based on Artificial Bee Colony Algorithm, 1770 in: 2018 IEEE 4th International Symposium on Wireless Systems within the International Conferences on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS-SWS), IEEE, 2018, pp. 35–40.
[49] S. A. R. Shah, B. Issac, Performance comparison of intrusion detection systems and application of machine learning to Snort system, Future Generation Computer Systems 80 (2018) 157–17.

[50] B. Selvakumar, K. Muneeswaran, Firefly algorithm based feature selection for network intrusion detection, Computers & Security 81 (2019) 148–155.
[51] A.-C. Enache, V. Sgˆarciu, A feature selection approach implemented with the binary bat algorithm applied for intrusion detection, in: Telecommunications and Signal Processing (TSP), 2015 38th International Conference on, IEEE, 2015, pp. 11–15.
[52] W. Park, S. Ahn, Performance comparison and detection analysis in snort and suricata environment, Wireless Personal Communications 94 (2) (2017) 241–252.
[53] S. X. Wu, W. Banzhaf, The use of computational intelligence in intrusion detection systems: A review, Applied soft computing 10 (1) (2010) 1–35.
[54] S. Mirjalili and A. Lewis, ‘‘The whale optimization algorithm,’’ Adv. Eng. Softw., vol. 95, pp. 51–67, May 2016.
[55] Rashedi E, Nezamabadi-pour H, Saryazdi S (2009) GSA: a gravitational search algorithm. Inf Sci 179:2232–2248
[56] KDD data set, 1999; <http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html>.
[57] http://www.unb.ca/cic/datasets/ids-2018.html

Figures

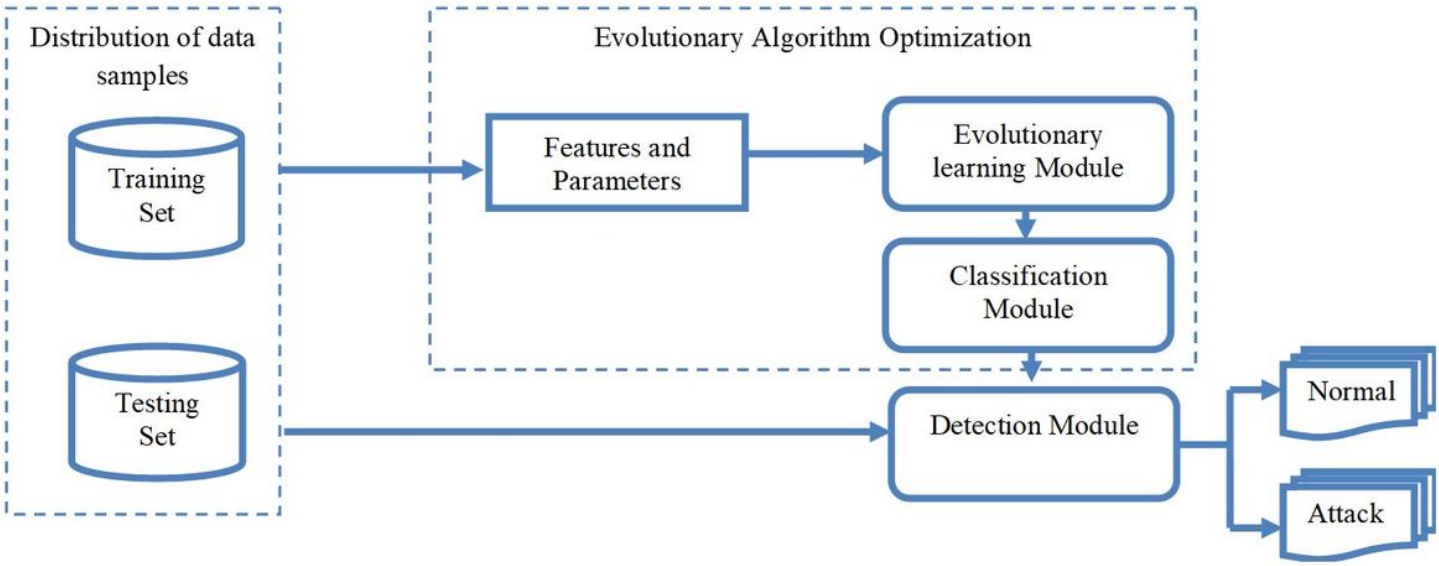

Figure 1

The process of Evaluation-based algorithms

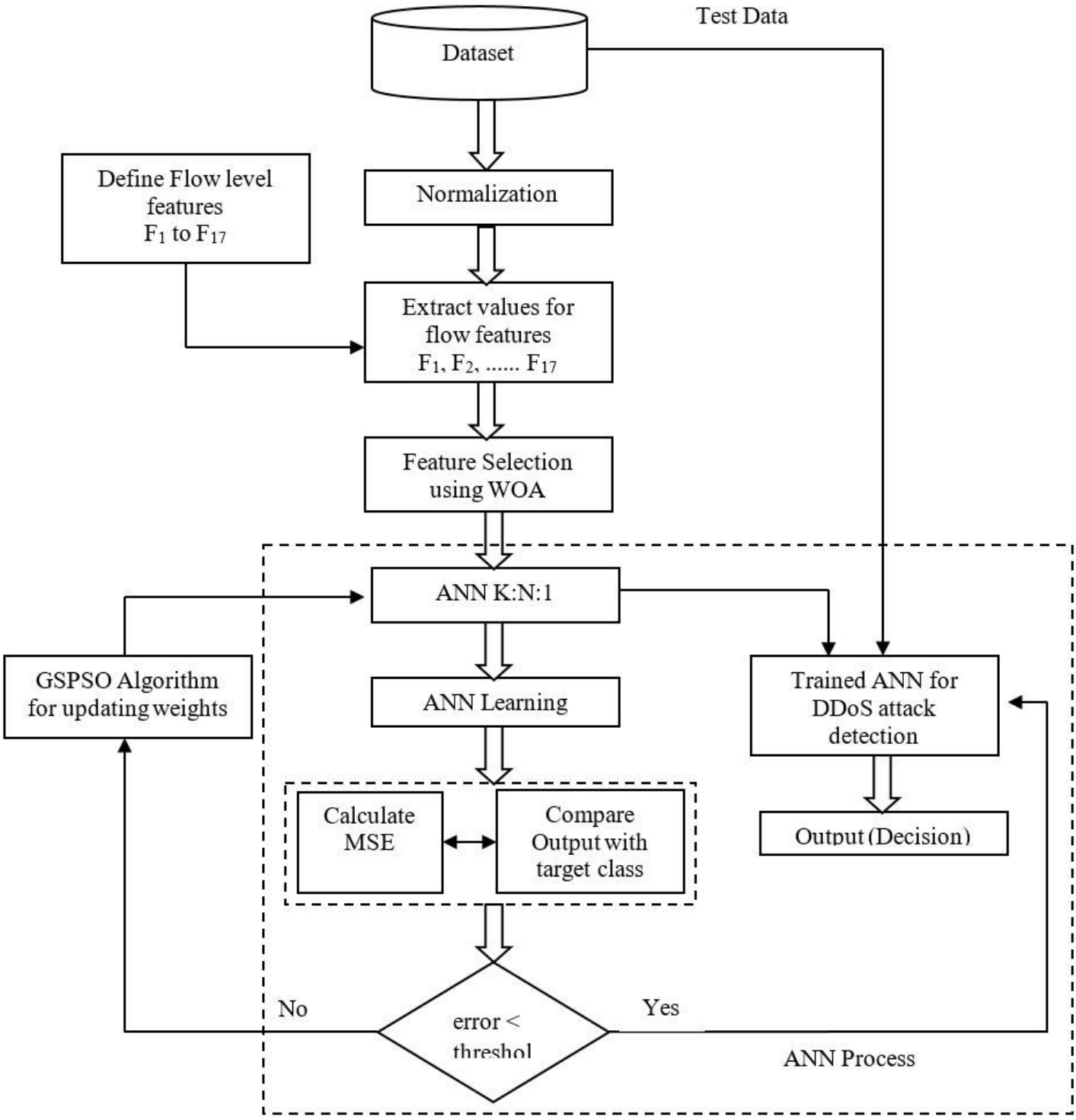

Figure 2

Framework of the proposed DDoS Detection System.

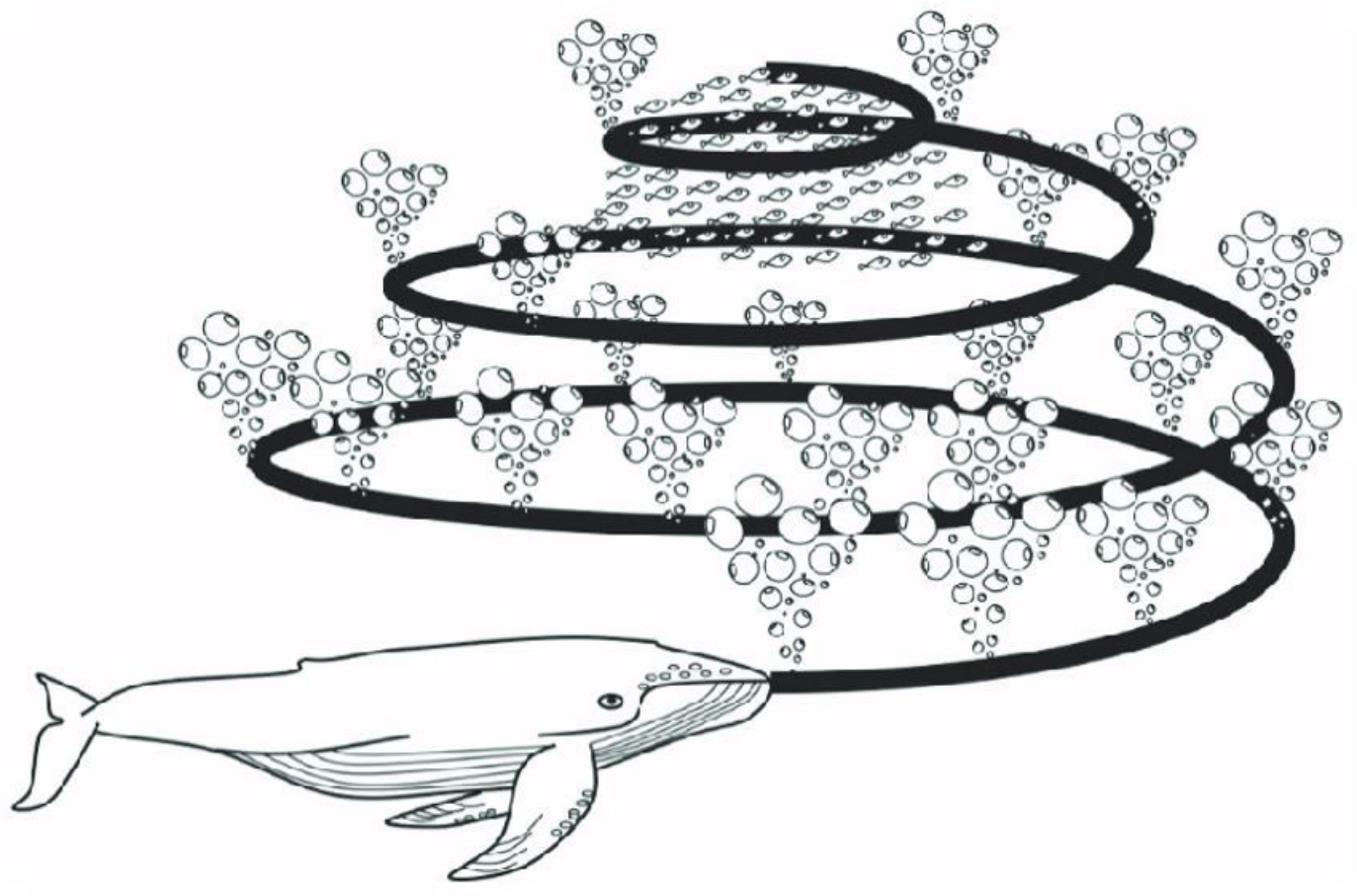

Figure 3

Humpback whales Bubble-net feeding behavior

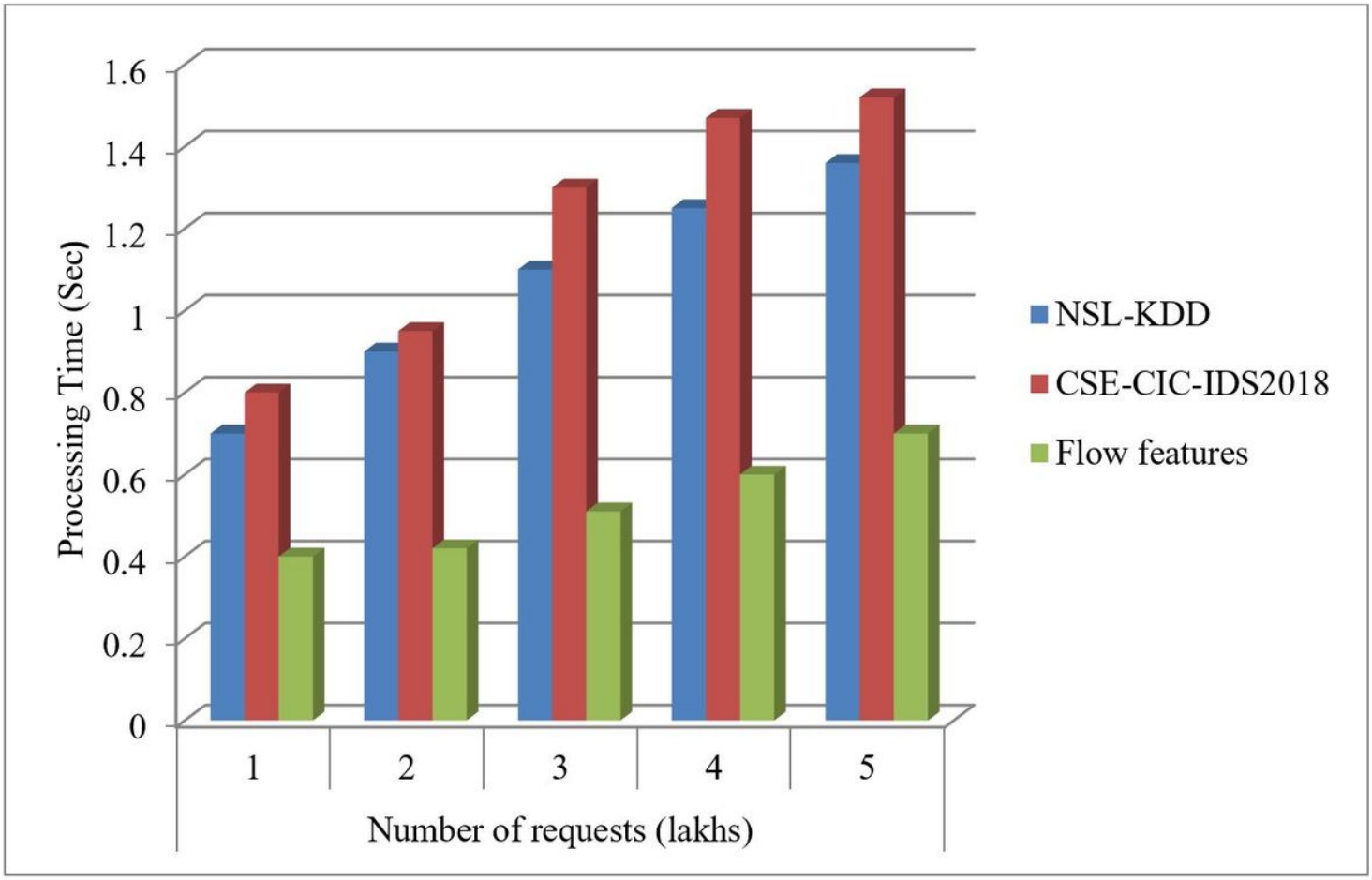

Figure 4

Comparison of Processing Times of various Data sets and flow features

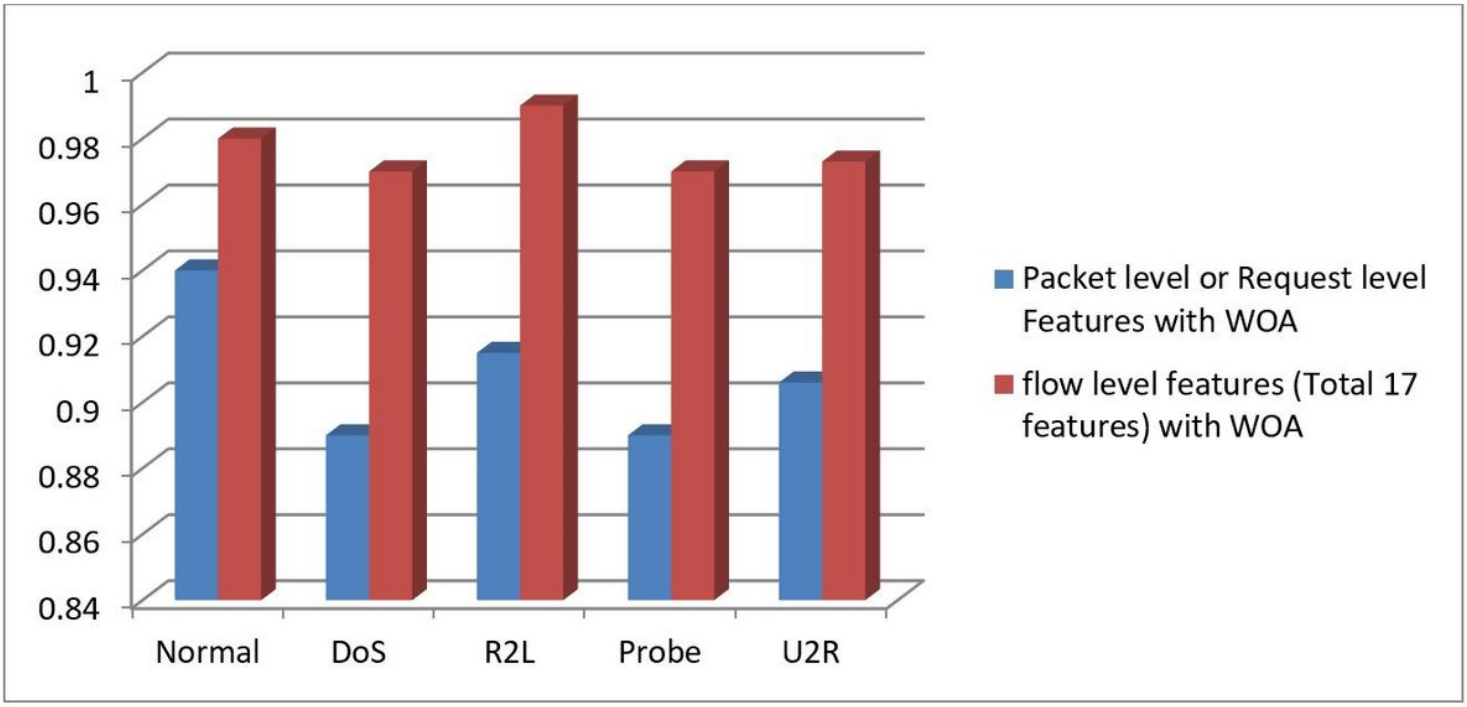

Figure 5

Detection accuracy of the proposed model with NSL-KDD dataset using packet level and Flow level features with WOA.

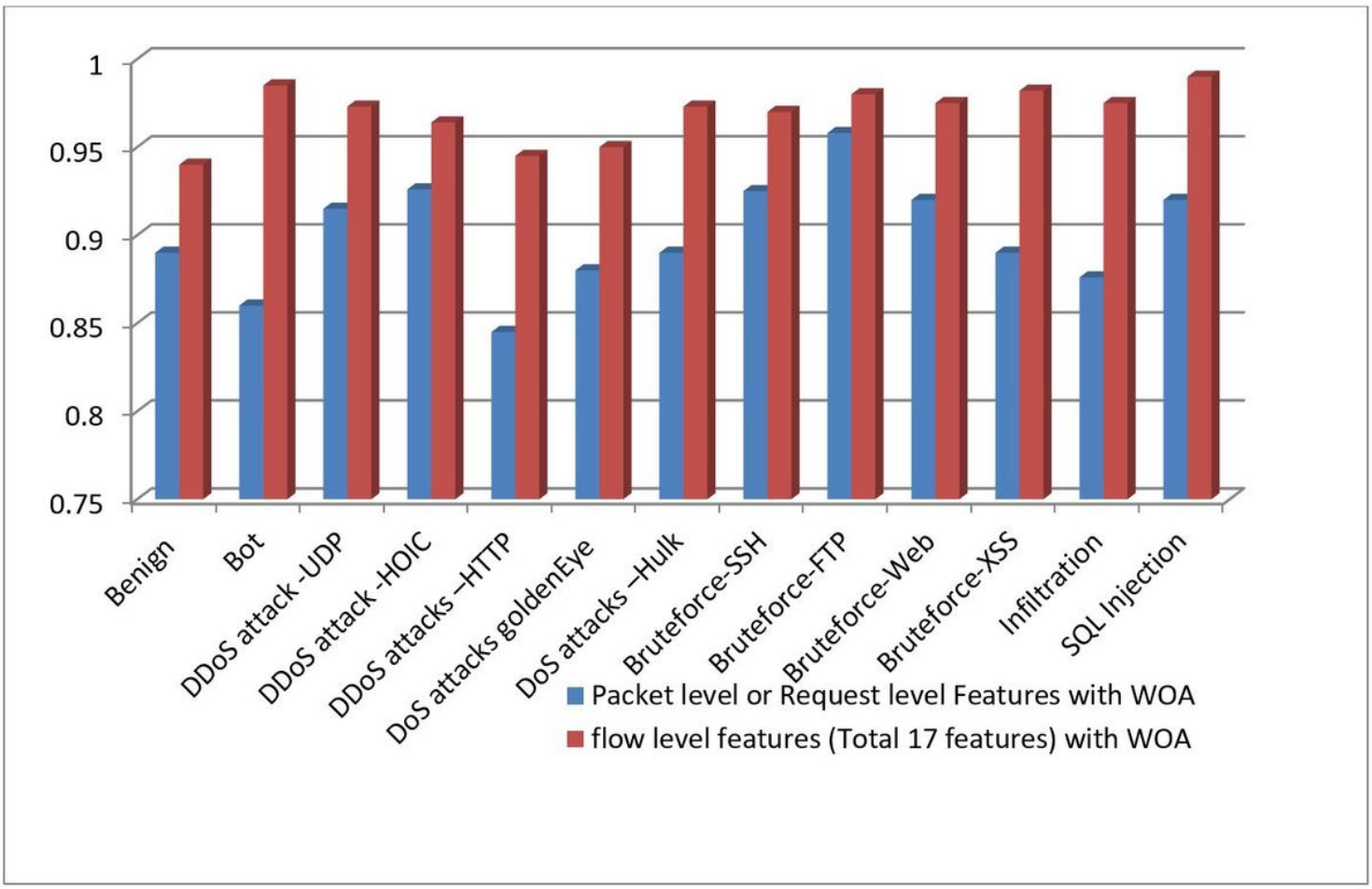

Figure 6

Detection accuracy of the proposed model with CSE-CIC-IDS2018 dataset using packet level and Flow level features with WOA.

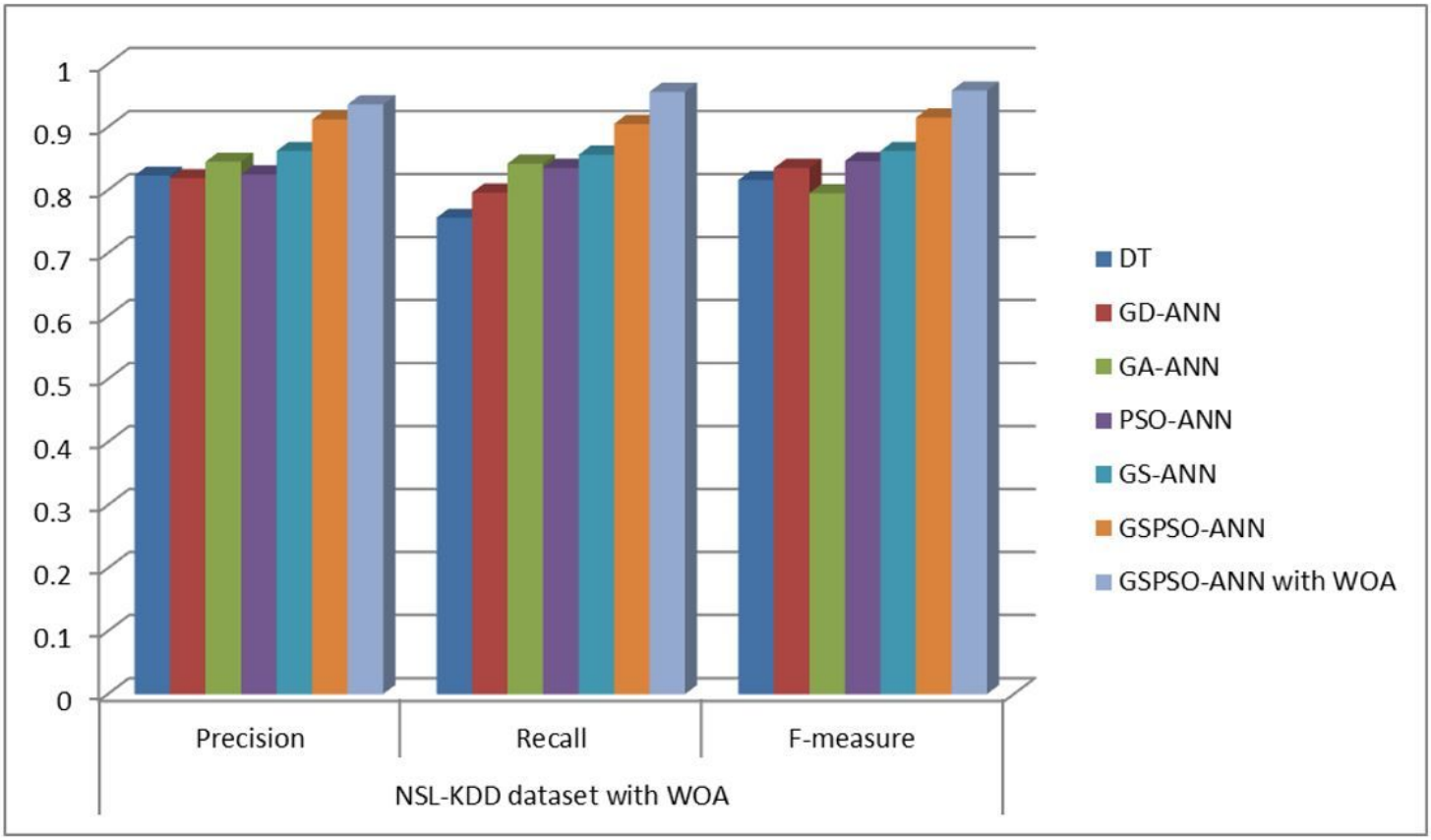

Figure 7

Comparison of Precision, Recall and F-measure with Contemporary methods (NSL-KDD dataset)

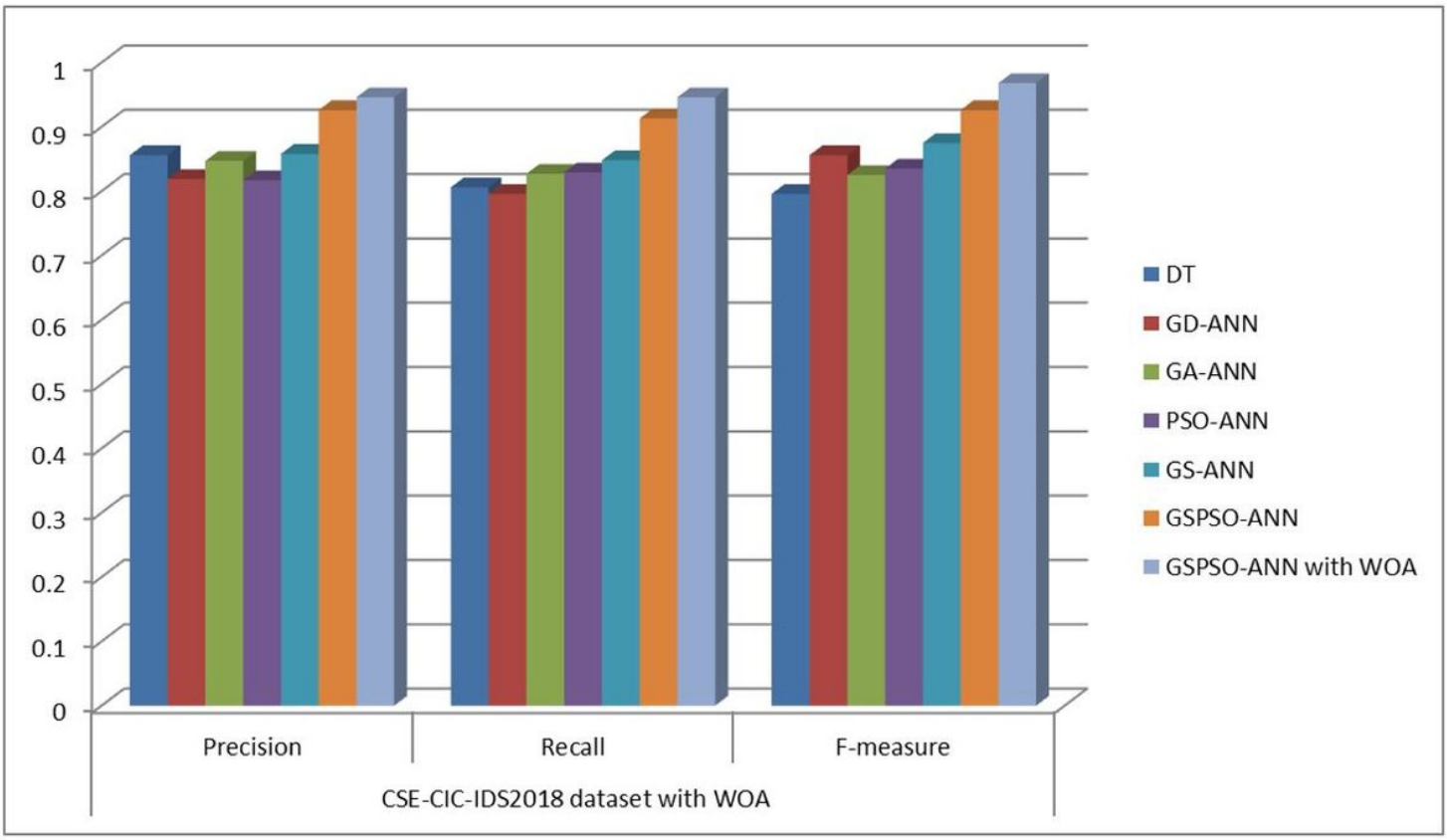

Figure 8

Comparison of Precision, Recall and F-measure with Contemporary methods (CSE-CIC-IDS2018 dataset)

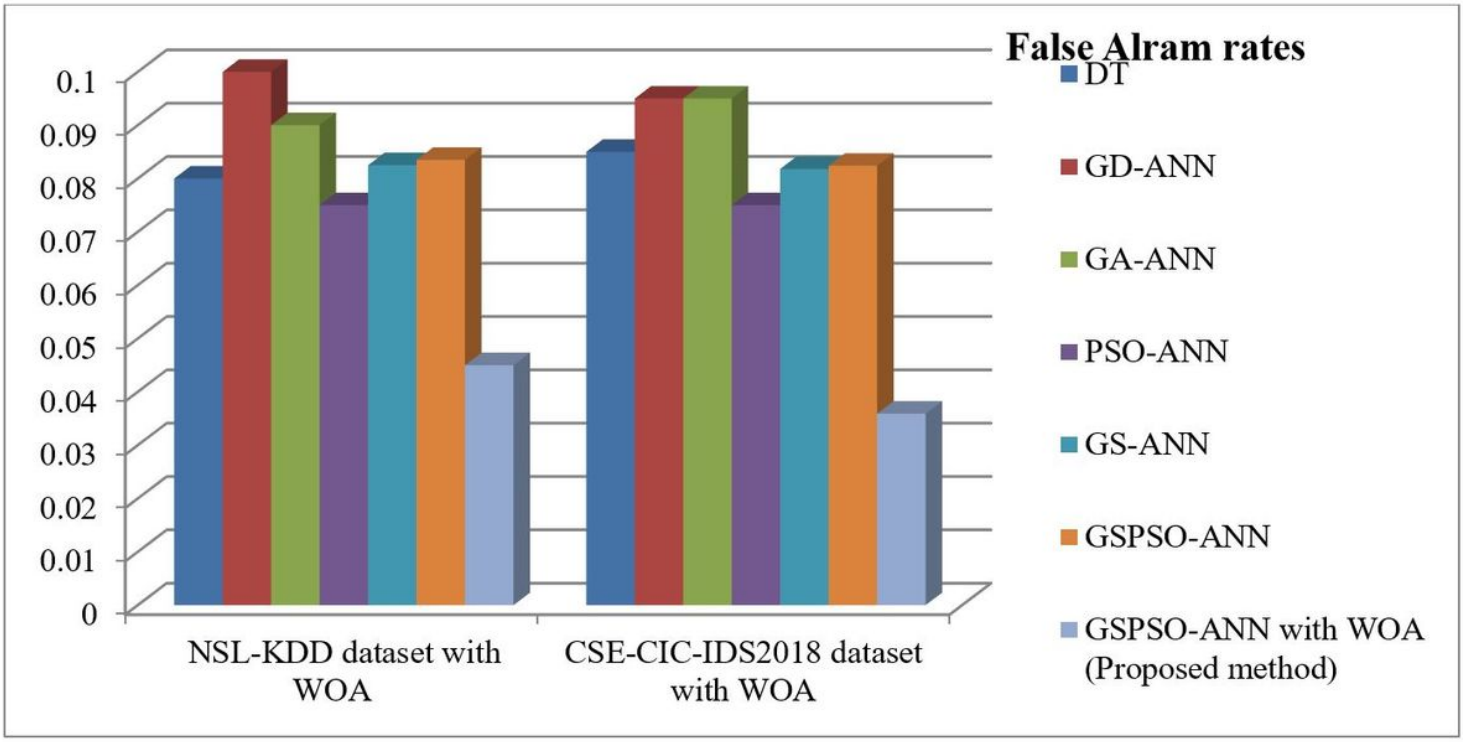

Figure 9

Comparison of false alarm rates with Contemporary methods.

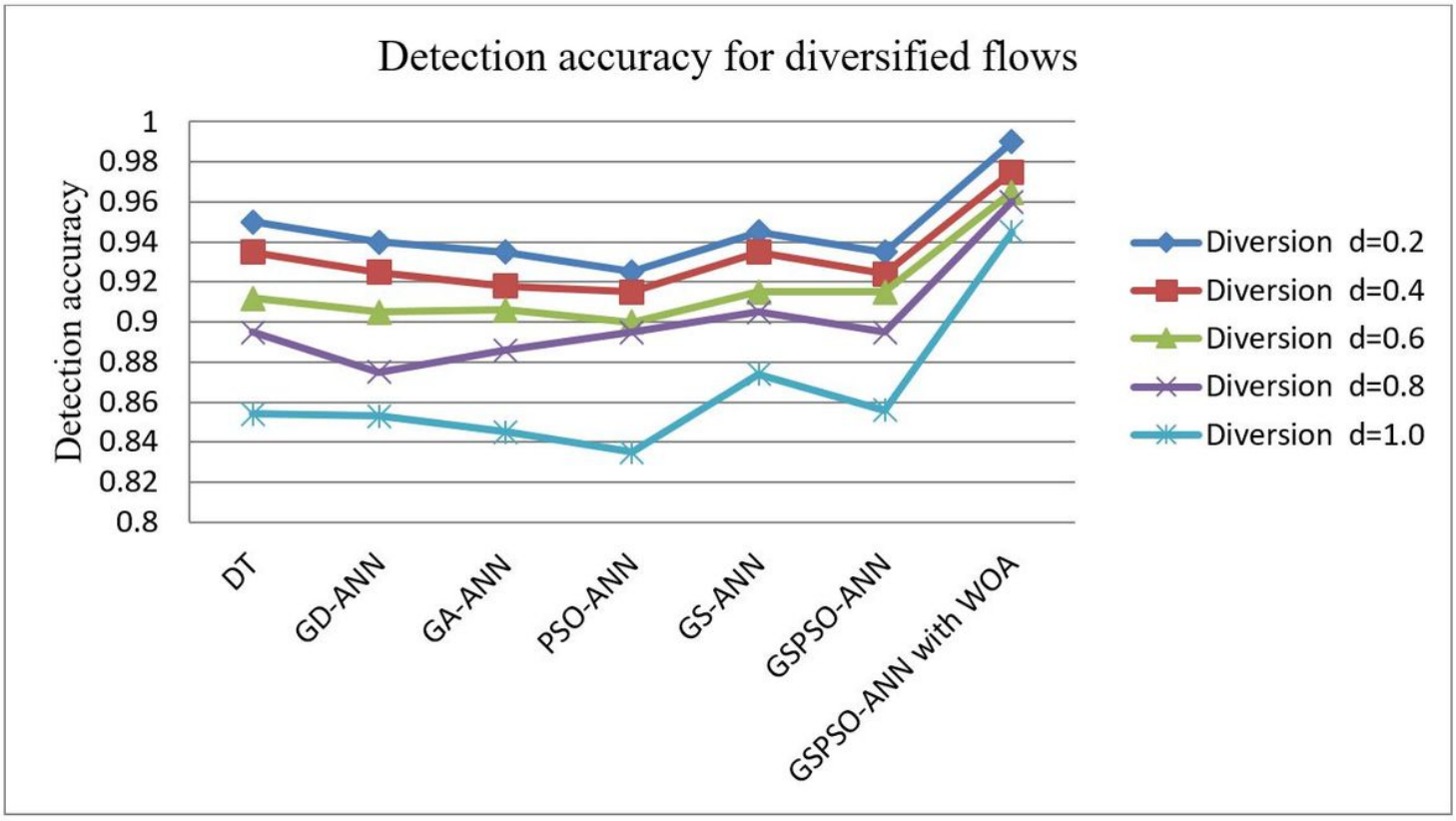

Figure 10

Comparison of detection accuracy with other methods for diversified NSL-KDD dataset

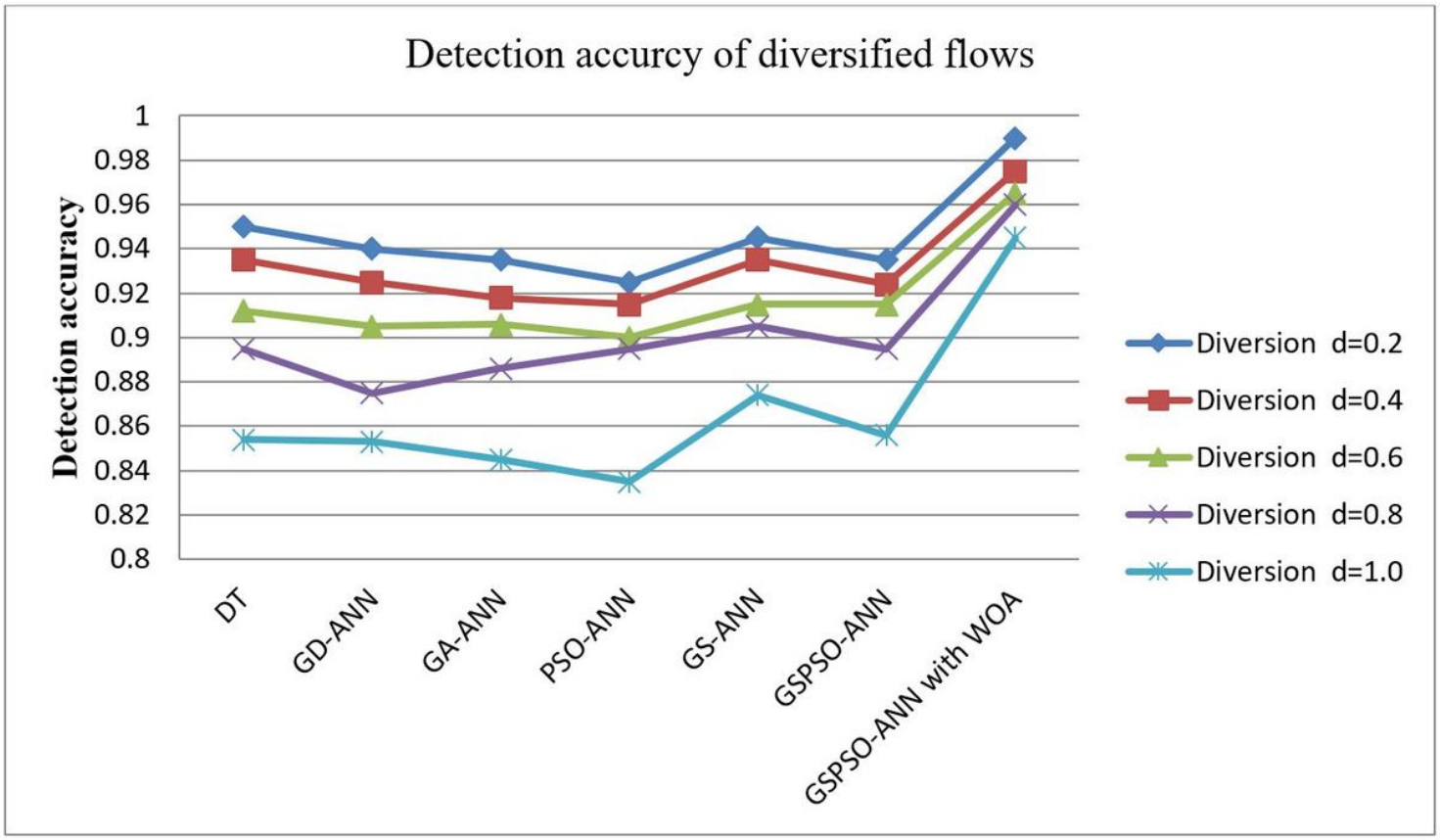

Figure 11

Comparison of detection accuracy with other methods for diversified CSE-CIC-IDS2018 dataset