Fuzzy-Taylor- Elephant Herd Optimization Inspired Deep Belief Network for DDoS Attack Detection and Comparison with State-of-the-Arts Algorithms

S. Velliangiri[a], Hari Mohan Pandey[b]

[a]Department of Computer Science, CMR Institute of Technology, Telangana, Hyderabad

velliangiris@gmail.com

[b]Department of Computer Science, Edge Hill University, UK

Pandeyh@edgehill.ac.uk

Abstract: Cloud computing environment support resource sharing as cloud service over the internet. It enables the users to outsource data into the cloud server that can be accessed remotely from various devices distributed geographically. Accessing resources from the cloud causes various security issues as the attackers try to illegally access the data. The distributed denial of service (DDoS) attack is one of the security concern in the cloud server. DDoS is a kind of cyber attack which disrupt normal traffic of targeted cloud server (or any other servers). In this paper, we propose an effective fuzzy and taylor-elephant herd optimization (FT-EHO) inspired by deep belief network (DBN) classifier for detecting the DDoS attack. FT-EHO uses taylor series and elephant heard optimization algorithm along with a fuzzy classifier for rules learning. The performance of the proposed FT-EHO is evaluated through rigorous computer simulations. Three standard benckmark databases, namely, KDD cup, database1 and database2 are used during simulations. Four quality measures such as accuracy, detection accurarcy, precision and recall are considered as a performance metrics. FT-EHO's performance is compared against the state-of-the-art methods considering the evaluation metrics. Results reveals that the proposed FT-EHO showed significantly higher value of evaluation metrics (accuracy (93.811%), detection rate (97.200%), precision (94.981%) and recall (93.833%)) as compared to other methods.

Keywords: Selector Engine, Elephant Herd Optimization (EHO), fuzzy system, Deep Belief Network (DBN).

1. Introduction

Cloud Computing (CC) is a new form to compute resources. It provide various service such as storage services, hardware equipment, operating systems, software applications and entire network

infrastructure and delivers the internet-based services to the users with low cost [6]. CC refers to both the services and the applications delivered to the hardware, internet, and software system in the data center that provides the cloud services [15]. The internet-based technology and the cloud metaphor referenced the accessibility and the availability of the resources to be computed [6]. CC provides the organizational users and the computing resources to be deployed as private, public, and community on the hybrid cloud [3]. CC provides various services like platform as a service (PaaS), expert as a service (EaaS), infrastructure as a service (IaaS) and software as a service (SaaS). The on-demand services and the computing resources allocated to the users are operated in the cloud [6].

CC is a paradigm where the resources of computing are shared as a cloud service on the internet [18]. CC is used in many organizations in the worldwide level as it supports computation and data storage with high performance. The cloud service relies on creating various dependencies using the number of holes with incompatibilities and vulnerabilities [17]. CC is a continuous development technology with abundant challenges in the service of security. The major concern in a cloud environment is the management of the cloud services [14]. The resources that are to be computed as the on-demand services are allocated to the user through the cloud storage [6].

The CC is used at most of the places to solve the cyber attacks as various vulnerabilities are existing in a cloud enviroment. The distributed denial of service (DDoS) attack is a kind of cyber attack which disrupt normal traffic of targeted cloud server. Hence, the DDoS attack is the critical [2]. Main characteristics of the DDoS attacks are defined: (a) It is very rigid to detect as it contains the equivalent flow in the regular user. DDoS attacks can be completed using a solitary node with limited flow of data and it results in low cost; (b) DDoS is a target insensitive attack becuase the attacked node identifies the attacker node. Thus, it affects the cloud computing services massively. The DDoS attacker compromises and collects various susceptible hosts referred to as zombies to attack against the targeted node; (c) DDoS attack increases sophistication, size and identifying the extortion is a major motive of this attack [3] [9]; (d) DDoS defense approach generally classifies the data packets as either malicious or legitimate packets [3]; (e) The DDoS attack exhausts the services and resources of the organization and individual by forwarding the useless traffic. Hence, legitimate users cannot access the services [4]; and (f) The DDoS detectors are situated in each host and their respective packet filters are disseminated through the virtual machines (VM) [4].

For the non-intrusive traffic, the network profile is generated by the detectors based on the network attributes of the selected statistics [4].

For most of the web applications (e.g. online auctions and online retail sales), security of the network is a major factor in the internet services [1]. Intrusion detection system (IDS) was noted as an effective method to detect DDoS attack and ensures the functional cloud services [6]. IDS detects the computer attacks by investigating the records, which are collected from the internet [1]. IDS was categorized into two types: (a) anomaly-based detection; and (b) signature or misuse based intrusion detection. Signature-based detection method uses the attacker signatures that are present in the knowledge database for identifying the attacks [3]. It is an effective technique to detect the known attacks [3]. On the other hand, anomaly-based method uses the behavioral pattern of the normal traffic with a period to compute whether the relevant patterns are deviated from the accepted behaviour [3]. The anomaly detection potentially detects a few or zero-day attacks [3]. Anomaly detection specifies the deviations that are obtained from the regular patterns whereas the signature detection utilized the patterns, which are related to the attacks to detect the intrusions [1].

Intrusion detection (ID) in CC is considered as an NP-hard problem. Metaheuristic algorithms give the best solution to the NP-hard problems [1]. The IDS is classified based on the source data as follows: (a) Host based IDS (HIDS); (b) Network-based IDS (NIDS); and (c) Distributed IDS (DIDS). The host-based detection system detects the intrusion using the sensors in the single host, whereas the network-based detection system focuses on the network arrangement. DIDS integrates the sensors and classifies the IDS into mobile agent-based IDS and grid-based IDS [10] [11].

The primary focuse of this research is to develop an algorithm for detecting the DDoS attack in the cloud. We propose an effective fuzzy and taylor-elephant herd optimization (FT-EHO) inspired by deep belief network (DBN) classifier for detecting the DDoS attack. The working of the FT-EHO involves three modules: (a) feature extraction; (b) feature selection; and (c) classification. The working starts when the user request is sent to the packet feature extraction module to extract the packet features. The packet informations such as *dest-bytes*, *duration*, *src-bytes*, and so on are extracted and the selective features are obtained by applying the holoentrophy into the feature selector engine. Finally, selected features are further processed by the classification module. The classification module detects the DDoS attacks using a fuzzy and taylor-elephant herd optimization (FT-EHO) inspired by deep belief network (DBN) classifier.

Commented [HP1]: **R3C2.** Motivation of the paper is not clearly explained so explain in a more proper way.

Response: The motivation of the paper is clearly mentioned in the Introduction section of the revised manuscript.

The proposed FT-EHO based DBN classifier is the integration of the DBN and fuzzy classifier. The rule learning approach based on the fuzzy classifier where the genetic algorithm (GA) in the adaptive genetic fuzzy system (AGFS) is the standard form. Moreover, the genetic algorithm has no guarantee for optimal solution and its one of state-of-the-art algorithms. Therefore, the genetic algorithm is replaced with the T-EHO algorithm, which is the integration of Taylor series and EHO algorithm. EHO is one of the ideal algorithms for finding a global optimization solution. EHO has been applied various optimization benchmark problems and real-life applications showing promising. Based on the extracted packet features, the proposed classifier based on a fuzzy and DBN classifier detects the DDoS attack by determining the node as an intruder or not.

The key contributions are elaborated as follows.

- We present a FT-EHO based DBN classifier for DDoS attack detection. Our approach utilized the merits of both fuzzy and DBN classified. In FT-EHO, we integraed Taylor serier and elephant heard optimization (EHO) algorithm along with fuzzy classifier for rules learning. In the proposed system, the tradional genetic algorithm (GA) is replaced by EHO algorithm.
- We systematically show how the packet features and packet information are extracted by the packet extraction module. The selective features are obtained by applying the holoentrophy into the feature selector engine.
- A comprehensive discussion is presented on the rule learning approach based on the fuzzy classifier, where GA is replaced by Taylor serier and EHO (T-EHO) algorithm.
- Extensive computer simulations are conducted to evaluated the performance of the proposed FT-EHO based DBN classifier. Three standard becnchmark databases, namely, KDD cup database [25], database 1 and database 2 are used during simulations. Four quality measures such as accuracy, detection accuracy, precision and recall are selected as a performance metrics.
- Finally, the comparative results and analysis is presented. FT-EHO based DBN classifer is compared against the state-of-the-art methods. The result indicates that FT-EHO based DBN classifier outperforms other methods.

The rest of the paper is organized as follows: Section 2 elaborates related work; Section 3 discusses the proposed algorithm; Simulation model is presented in the Section 4; Section 5 shows the conclusion of this research work.

Commented [HP2]: **R4C1**. The author proposes a novel effective fuzzy and Taylor-Elephant Herd optimization (T-EHO)-based Deep Belief Network (DBN) classifier to detect the DDoS attack. This paper seems interesting and can be very beneficial on to detect DDoS attack. There are some recommendations which the author can use to improve the quality of the paper as follows:

1.The author need to explain in introduction section why Taylor series and Elephant Heard Optimization algorithm with fuzzy classifier for as rule learning approach is used over genetic algorithm.

Response: The rule learning approach based on the fuzzy classifier, where the genetic algorithm in the AGFS is the standard form. Moreover, genetic algorithm has no guarantee for optimal solution and its one of state-of-the-art algorithm. Therefore, genetic algorithm is replaced with the T-EHO algorithm, which is the integration of Taylor series and EHO algorithm. Based on the extracted packet features, the proposed classifier based on fuzzy and DBN classifier detects the DDoS attack by determining the node as an intruder or not.

Commented [HP3]: **R1C1**: Introductory section is overcrowded with both relevant and irrelevant details. This should be summarized to highlight the research motivations and the proposed research carried out.

Response: Appreciating you for notifying this comment. The introduction is improved, and unnecessary sentences are removed. The revised introduction highlight the research motivation and proposed research carried out.

2. Related work

The review of literatures are deliberated in this section. Yasir Ali *et al.* [1] developed a detection and migration approach to enhance the performance and stability of the Network Control System (NCS). The unknown packets are dropped from the network by applying the filtering process to enhance data integrity. However, this approach did not apply to the physical applications in the Cloud Control System (CCS). Kesavamoorthy and Ruba Soundar [2] proposed an autonomous multi-agent approach to improve communication among the agents to accurately make the decision. The multiple agents communicate themselves and update the coordinating agent to detect the DDoS attacks. The stochastic based filtering was not applied to achieve optimization. Opeyemi *et al.* [3] developed a feature selection approach to enhance the optimal selection. The important features were identified based on the counter and the threshold and enhanced the detection accuracy. However, the performance attained using the labeled datasets is very less. Pandey *et al.* [4] introduced a distributed network filtering approach to distributing the filters among the virtual machines. The DDoS attacks were effectively detected and provide high scalability, whereas, the real-time detection was not performed. Loukas *et al.* [5] developed a cloud-based intrusion detection system using the deep learning approach. This system attained high accuracy and better detection latency. Collecting the data for the attack or normal behaviour was not considered. Hajimirzaei and Navimipour [6] introduced an intrusion detection system to create the training subset. The normal and the abnormal packets were identified using the multilayer perceptron in the network traffic. This approach achieved better performance, but the meta-heuristic methods were not applied effectively. Deng *et al.* [7] developed a cyber physical power system for accurate identification and detection of intrusions. This approach attained better performance in terms of scaleup and speedup. Even though the computing nodes were increased, the type of the noise data was not accurately determined. Chen *et al.* [8] proposed a fuzzy-based clustering algorithm to partition the dataset into clusters. Each node perfectly forwards the data packets to the neighboring node. The proposed approach identifies the malicious nodes under heavy traffic conditions, but the structure using a large number of nodes was not considered.

Yan et al. [26] showed the characteristics of DDoS attacks in cloud computing and presented a comprehensive survey on the DDoS attacks protection methods using software-defined networking (SDN). Major part had been analyzed and the studies about launching DDoS attacks on SDN and the techniques to detect DDoS attacks in SDN were discussed. From a rigorous

analysis, authors [26] concluded that the conflicting relationship between SDN and DDoS attacks had not been well addressed in the previous literatures. Further, authors [26] had provided information about how to make full use of SDN to beat DDoS attacks in cloud computing environments and how to prevent SDN itself from becoming a victim of DDoS attacks. Somani *et al*. [27] introduced developments related to DDoS attack mitigation solutions in the cloud. They presented an inclusive survey with a thorough insight into the characterization, prevention, detection, and mitigation techniques of these attacks. Also, they presented an inclusive solution taxonomy for classifying DDoS attack solutions. The authors provided a definite guideline on effective solution building and detailed solution requirements to assist the cybersecurity research community in designing defense techniques.

Gao et al. [9] modelled a semi-supervised learning approach in the cloud-based robotic system for detecting the intrusion. It constructs the ensemble labeled data using the ensemble learning, and utilized the unlabeled data for data analysis. However, modelling the generalization and improving the detection performance is a challenging task in the intrusion detection system. Dey et al. [10] developed a machine learning approach for intrusion detection in the cloud environment. Identifying the attributes in the detection system is a challenging task in the cloud-based environment. A Machine learning based intrusion detection system was developed to intercept the network traffic in the physical layer [11]. Detecting the intrusion using the ensemble learning classifier results in several challenges in the central storage server. A conceptual cloud mitigation framework is modelled for detecting the attacks in the cloud. Providing security to the cloud server is a challenging task in the cloud services. The availability of the training and the testing dataset results in attack patterns to generate the optimal features [12]. Patil et al [13] developed a multi-threaded based intrusion detection system to extract the accurate features in the cloud system. Delivering the proper services to the cloud user results many challenges in the cloud computing. In the real time, if the selective features are not accurately extracted, the DDoS attack will not be detected.

In order to address the above challenged, we have proposed Fuzzy-Taylor-Elephant Herd Optimization based Deep Belief Neural Network (FT-EHO-DBN) classifier for detecting the DDoS attack. When compared with the state-of-the-art algorithms, DBN has the more advantage in the pre-training with the fine-tuning learning technique and a multi-layer structure. These advantages formulate the DBN to extract the deep attributes of the training data. Hence, the DBN

solves the problems of low training efficiency, local optimum, and complex network attack detection. Here, the T-EHO algorithm is developed by integrating the Taylor series in the existing EHO for selecting the optimal weights and biases for the DBN classifier. EHO is one of the ideal algorithms for finding a global optimization solution. EHO has been applied to various optimization benchmark problems and real-life applications showing promising. Hence, the proposed FT-EHO-DBN classifier addresses the challenges of the research problem

3. Proposed Model

In this paper, Taylor-Elephant Herd Optimization based Deep Belief Neural Network (TEHO-DBN) classifier is developed for detecting the DDoS Attack. When compared to the conventional neural networks, DBN has the merits of pre-training with the fine-tuning learning technique and a multi-layer structure. These merits formulate the DBN to extract the deep attributes of the training data. Hence, the DBN solves the problems of low training efficiency, local optimum, and complex network attack detection. Here, the TEHO algorithm is developed for selecting the optimal weights and biases for the DBN classifier. EHO is one of the ideal algorithms for finding a global optimization solution. EHO has been applied to various optimization benchmark problems and real-life applications showing promising results in finding optimal solutions. Since EHO does not resort to any type of relaxations, EHO outperforms the state-of-the-art optimization algorithms. Hence, the proposed TEHO-DBN classifier addresses the challenges of the research problem.

The DDoS attack is detected using the proposed FT-EHO-DBN classifier. The block diagram of the proposed DDoS attack detection is depicted in Figure 1. We have used DBN classifier which is consists of 21 hidden layers. The number of hidden layers (N_h) can be determined using equation (1).

$$N_h = Ns / (Ni + No) * \alpha\alpha \quad (1)$$

Where, Ni = number of input neurons; No = the number of output neurons; Ns = number of samples in the training data set and αα = an arbitrary scaling factor, usually 2-10.

Commented [HP4]: **R1C2.** The related work needs to be elaborated more by adding more literature on deep networks for cloud computing security.

Response: The suggestion considered and therefore literatures have been added for cloud computing security.

Commented [HP5]: **R2C1.** In this paper authors have proposed an effective fuzzy and Taylor-Elephant Herd optimization (T-EHO)-based Deep Belief Network (DBN) classifier to detect the DDoS attack. The idea proposed in the paper is novel and well organized. Although, authors have covered most of the things but still the paper needs some improvement. My comments are on this paper is given as follows:

1.In section 2, a paragraph must be added at the end to show how this paper addresses the shortcoming of the existing algorithms.

Response: Thank you for pointing this to us. A paragraph is added to address this point.

Commented [HP6]: **R3C1**. The reason of replacing genetic algorithm need to be discussed.

Response: In this paper, Taylor-Elephant Herd Optimization based Deep Belief Neural Network (TEHO-DBN) classifier is developed for detecting the DDoS Attack. When compared to the conventional neural networks, DBN has the merits of pre-training with the fine-tuning learning technique and a multi-layer structure. These merits formulate the DBN to extract the deep attributes of the training data. Hence, the DBN solves the problems of low training efficiency, local optimum, and complex network attack detection.
Here, the TEHO algorithm is developed for selecting the optimal weights and biases for the DBN classifier. EHO is one of the ideal algorithms for finding a global optimization solution. EHO has been applied to various optimization benchmark problems and real-life applications showing promising results in finding optimal solutions. Since EHO does not resort to any type of relaxations, EHO outperforms the state-of-the-art optimization algorithms.
Hence, the proposed TEHO-DBN classifier addresses the challenges of the research problem.

Commented [HP7]: **R1C4**. Usually for Deep learning, the no of layers are at least 8 -10, how many layers in your proposed method?

Response: The number of hidden layers in the deep neural network is 21. The number layer can be calculated following equations.
Ni = number of input neurons.
No = the number of output neurons.
Ns = number of samples in the training data set.
αα = an arbitrary scaling factor, usually 2-10.
Nh = Ns / (Ni + No) * αα

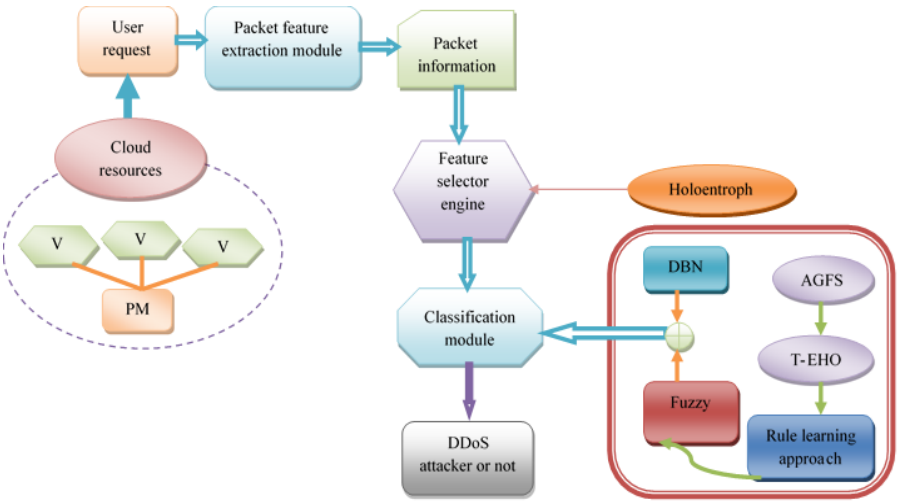


Figure 1. Architecture of the proposed DDOS attack detection.

The proposed DDoS attack detection classifier includes three modules are as follows: i) packet feature extraction module; ii) feature selector engine; and iii) classification module. Initially, the user request is send to the packet feature extraction module to extract the packet features. From each node, the packet information, like duration, dst-bytes, wrong fragment, and so on are extracted in the extraction module. The extracted features are passed into the feature selector engine, which selects the appropriate features accurately using the holoentrophy criteria. Finally, the selected features are further processed by the classification module. The classification module in the proposed approach detects the DDoS attacks using the FT-EHO-DBN classifiers, which is the hybridization of the DBN and fuzzy logic. The proposed FT-EHO-DBN classifier is employed for classification, where the training algorithm is based on T-EHO. In the fuzzy classifier, named AGFS [19], the genetic algorithm is replaced by the T-EHO algorithm, which is the rule learning approach. Based on the extracted packet features, the proposed classifier detects the DDoS attacks by determining the node as an intruder or not.

Initially, the log file is created in the attack detection module, and the created log file is represented as, H . The user utilizes the allocator or the resource scheduler to access the resources in the cloud model. In the resource allocation component, the devices are allocated to the user based on their requirement, and also it contains the device information. The resource scheduler records the log information of every user for creating the log file H .

Commented [HP8]: R1C3. The methodology discussed in section 3 looks interesting; however, this should be discussed in a manner that will facilitate understanding by a non-expert.

Response: We appreciate your concern on methodology. To improve big paragraph is divided in smaller one and an attempt is made to present the idea in lucid manner.

Commented [HP9]: R3C5. Figure 1 and 2 need more explanation so that the contribution needs to be explained in detailed manner.

Response: As per the reviewer suggestion, explanation is added in both Figure 1 and Figure 2.

Commented [HP10]: R2C2. Figure 1 needs explanation especially for different blocks used in the figure.

Response: We added explanation to cover different blocks used in Figure 1.

3.1 Packet feature extraction module

After creating the log file, packet features are extracted from the log file so that classifier can be trained. Extracted features are stored in the feature database, which is denoted as I . Extracted features are explained as follows:

duration: The duration is the first extracted packet feature, which is indicated in seconds, and it shows the time or length of the packet connection.

src-bytes: The nodes present in the cloud model is indicated as the source node or the destination node, as it performs the transmitting and receiving functions. The src-bytes indicate the total number of bytes transmitted from the sender to the receiver.

dst-bytes: This feature is similar to the src-bytes features, dst-bytes represents the total number of bytes transmitted from the receiver node to the sender node.

logged_in: logged_in feature is one of the discrete feature, when the user is log on to the system, then the logged_in feature is resulted as 1, otherwise the feature value will be 0.

count: This feature determines the total number of connections present in the same host.

srv-count: In certain cases, the nodes in the cloud model may provide the exact services, and is represented using the srv-count feature.

The connection of the node is situated under the rejection and the synchronization error, which is expressed using the below two features.

serror-rate: This feature reflects the percentage of total connections exists in the cloud using the SYN error.

rerror-rate: This feature shows the percentage of total connections exists in the cloud using REJ error.

diff-srv-rate: This feature represents the percentage of connections made to the various hosts, where the service connections are offered to different hosts.

same-srv-rate: Some services may have the same host as the destination. This feature symbolizes the total number of connections made in the same destination.

srv-diff-host-rate: This feature shows the service connection from single node to various destination hosts.

3.2 Feature selector engine using holoentrophy

The extracted packet features are subjected to the feature selector engine to select the considerable features using the holoentrophy criteria [20]. The holoentropy used in the feature selector engine is denoted as $X(j_k, j_l)$ and can be determined using equation (1).

$$X(j_k, j_l) = J.L(j_k, j_l) \tag{1}$$

where,

$$J = 2\left[1 - \frac{1}{1+\exp(-L(j_k, j_l))}\right] \tag{2}$$

$$L(j_k, j_l) = \sum_{k=1}^{m(j_t)} Y(j_k = k, j_l = l).\log(j_k = k, j_l = l) \tag{3}$$

Where, $m(j_t)$ represents the unique values of the selected features, j_k and j_l denotes the unique attributes of the selected fetaures.

3.3 FT-EHO-DBN classifier for attack detection

The proposed FT-EHO-DBN classifier to detect the DDoS attack in the cloud environment is elaborated in this section [22] [23]. The proposed FT-EHO-DBN classifier is employed for classification where the training algorithm is based on T-EHO. In the fuzzy classifier, named AGFS, the genetic algorithm is replaced by the T-EHO algorithm, which is the rule learning approach. Based on the extracted packet features, the proposed classifier detects the DDoS attacks by determining the node as an intruder or not.

3.3.1 Fuzzy classifier based on rule learning approach

The rule learning approach, termed as Adaptive T-EHO-based fuzzy system is applied into the fuzzy classifier by replacing the genetic in the AGFS system with the T-EHO algorithm. The Adaptive T-EHO-based fuzzy system is offered by integrating the fuzzy set, and T-EHO with the Genetic Algorithm (GA). The Adaptive T-EHO-based fuzzy system performs the classification using the fuzzy classifier, and generates the optimized rules by GA. Let I denotes the database, which is partitioned into two sets, as training I_{tr} and testing data set I_{te}. The training data is used to design the fuzzy system, and to generate the fuzzy rules. The testing data set is considered in

evaluating the performance of the classification. The overall procedure of Adaptive T-EHO based fuzzy system to generate the fuzzy rules is elaborated as follows:

a) Discretization

Discretization is a process of data pre-processing, which changes the data and range of values into a specific interval. In the discretization function, the training data set $I_{tr} = o_{np}; 0 \le n \le r \ and \ \ 0 \le p \le r$ with r number of attributes are considered. The minimum and the maximum values of the attributes are computed and are sorted in ascending order. However, the minimum and the maximum values are calculated for all the classes in the database, based on the vector.

b) An Algorithmic description of T-EHO algorithm for optimal weight selection

The T-EHO algorithm is introduced to select the optimal weight of the DBN classifier. The T-EHO algorithm is the integration of Taylor series [25] and the EHO [22] algorithm. The EHO algorithm uses the elephant herding characteristics to compute the optimal solution, and the solution space is updated using two operators, as i) clan updating, and ii) separating operator. The training process of the T-EHO algorithm is described as below:

Clan initialization: The solution space in the T-EHO algorithm is determined using the population of the clan. The solution space is considered with M population and is denoted as, P_{np}. In each clan, there exists various number of elephants.

Fitness evaluation: The fitness is evaluated by considering the position, clan and error-based fitness in the clan. The solution with the minimum error value is taken as the best fitness solution. Equation (4) is used to determine the fitness value.

$$N_{avg} = \frac{1}{r}\sum_{n=1}^{r}\left(h_i - o^n\right)^2 \tag{4}$$

Where, h_i is the output obtained from the classifier and o^n denotes the estimated output.

Solution update: Based on the matriarch elephant movement, the solution space is updated in the population. The updated solution space is represented using equation (5).

$$P_{new,qn,p} = P_{low,p} + \sigma \times \left(P_{best,qn} - P_{qn,p}\right) \times s \tag{5}$$

Where the elephant best position is represented as $P_{best,qn}$ and $P_{qn,p}$ is the old position of p^{th} elephant in clan qn and the term $P_{new,qn,p}$ represents the newly updated position and $P_{low,p}$ denotes the old position. The solution space in the clan is updated using the scaling factor s, with the value ranges from 0 to 1. The search space is refined by adopting the taylor series. The equation (5) is rearranged as shown in equation (6).

$$P_{new,qn,p} = P_{qn,p}(1-\sigma s) + \sigma \times P_{best,qn} \times s \tag{6}$$

The solution space based on the Taylor series is expressed using equation (7).

$$\begin{aligned} P(r+1) = {} & 0.5P(r) + 1.3591P(r-1) - 1.359P(r-2) + 0.6795P(r-3) - 0.2259P(r-4) + \\ & 0.0555P(r-5) - 0.0104P(r-6) + 1.38e^{-3}P(r-7) - 9.92e^{-5}P(r-8) \end{aligned} \tag{7}$$

Computing $P(r)$ from the above equation (7) is represented as equation (8).

$$P(r) = \frac{1}{0.5}\left[\begin{aligned} & P(r+1) + 1.3591P(r-1) - 1.359P(r-2) + 0.6795P(r-3) - 0.2259P(r-4) + \\ & 0.0555P(r-5) - 0.0104P(r-6) + 1.38e^{-3}P(r-7) - 9.92e^{-5}P(r-8) \end{aligned}\right] \tag{8}$$

Equation (8) is substituted in equation (6) to compute $P_{new,qn,p}$ as shown in equation (9).

$$\begin{aligned} P_{qn,p}(r+1) = {} & \frac{(1-\sigma s)}{0.5}\left[\begin{aligned} & P(r+1) + 1.3591P(r-1) - 1.359P(r-2) \\ & + 0.6795P(r-3) - 0.2259P(r-4) + \\ & 0.0555P(r-5) - 0.0104P(r-6) + 1.38e^{-3}P(r-7) \\ & - 9.92e^{-3}P(r-8) \end{aligned}\right] \\ & + \sigma \times P_{best,qn} \times s \end{aligned} \tag{9}$$

The value of $P_{qn,p}(n+1)$ is computed by solving the above equation as presented in equation (10).

$$\begin{aligned} P_{qn,p}(n+1) = {} & \frac{(1-\sigma s)}{0.5+\sigma s}\left[\begin{aligned} & 1.3591P(r-1) - 1.359P(r-2) \\ & + 0.6795P(r-3) - 0.2259P(r-4) + \\ & 0.0555P(r-5) - 0.0104P(r-6) + 1.38e^{-3}P(r-7) \\ & - 9.9.2e^{-5}P(r-8) \end{aligned}\right] \\ & + \sigma \times P_{best,qn} \times s\left(\frac{0.5}{0.5+\sigma s}\right) \end{aligned} \tag{10}$$

Equation (10) denotes the updated position of the matchairat elephant using Taylor series. The Taylor coefficient with degree six is used to enhance the optimization accuracy of T-EHO algorithm.

Computing the best solution: After the position gets updated, the optimal solution can be determined using equation (4).

Termination: The iteration to update the position is continued until; the best optimal solution is obtained.

c) Designing the fuzzy membership function

The fuzzy system is considered using the fuzzy rule and the fuzzy membership function.

***i)* Membership function of fuzzy:** The fuzzy membership function is computed using the triangular function, which contains three vertices, as $s, t,$ and u in the $y(z)$ fuzzy set. The membership value is calculated using equation (11).

$$y(z)=\begin{Bmatrix} 0 & if\ v \le s \\ \dfrac{v-s}{t-s} & if\ s \le v \le t \\ \dfrac{u-v}{u-t} & if\ t \le v \le u \\ 0 & if\ v \ge u \end{Bmatrix} \tag{11}$$

ii) Defining the membership function: The discretization function of each attributes is utilized to compute the number of membership function in the fuzzy classifier. The values of the vertices $s, t,$ and u is defined for each membership function. For each interval, the maximum value of u, and minimum value of s is selected to compute the value for t.

d) Classification using fuzzy system: The fuzzy membership function and the fuzzy rules are applied in the classification process of the fuzzy system.

i) Rule base in the fuzzy system: The fuzzy rule set is generated as $J = \{J_p; 1 \le p \le q - K\}$, which is ordered using the fuzzy rule based on the genetic algorithm. The rule base contains the fuzzy rule as, $L_1, L_2, L_3,$ and L_4, respectively as low, medium, very low, and very low decision.

ii) Membership function of fuzzy: The membership function used for every attributes and their respective values for the triangular membership is also generated.

iii) Classification using fuzzy data set: In the testing data set, the test data is converted into the fuzzified value, and the input of the fuzzified is matched with the fuzzy rules to generate the linguistic value, which is then further transformed into the fuzzy score. The optimal value is generated using the fuzzy score, and the output attained using the fuzzy classifier is denoted as, h_j.

3.3.2 Deep Belief Network for attack detection

DBN classifier is developed by incorporating two RBM layers and one MLP layer, which is depicted in Figure 2. In DBN, the connections exist between the visible and the hidden neurons, but there is no connection lies between the hidden neurons, and the visible neurons. The feature vector A is applied as input into the visible layer of the first RBM in DBN. The output obtained from the RBM 1 hidden layer is passed as input to the second RBM, and the output obtained from RBM layer 2 is fed as input to the MLP layer.

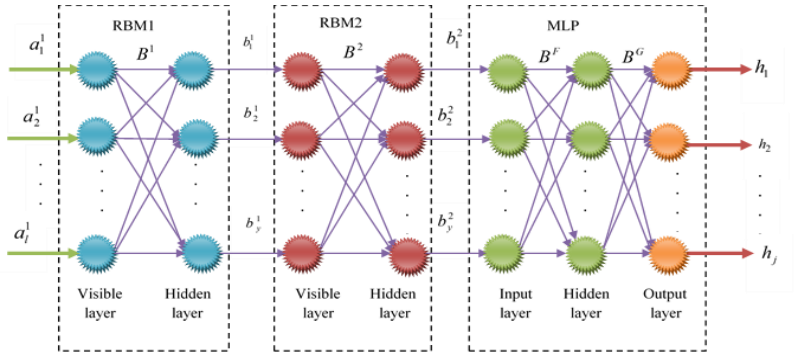


Figure 2. Architecture of DBN classifier.

The feature vector is the input to the visible layer of RBM1, and the output from the hidden layer of the RBM 1 is expressed as equation (12) and (13).

$$a^1 = \{a_1^1, a_2^1, \ldots, a_x^1, \ldots, a_{10}^1\}; 1 \le x \le l \quad (12)$$

$$b^1 = \{b_1^1, b_2^1, \ldots, b_c^1, \ldots, b_y^1\}; 1 \le c \le y \quad (13)$$

Where, b_c^1 denotes the c^{th} hidden neuron, a_x^1 is the x^{th} visible neuron of RBM 1, and y represents the number of hidden neurons. Each and every neuron in the hidden and the visible

Commented [HP11]: R2C3. Figure 2 need to be explained in detail.

Response: We added explanation for Figures 2.

Commented [HP12]: R3C5. Figure 1 and 2 need more explanation so that the contribution needs to be explained in detailed manner.

Response: As per the reviewer suggestion, explanation is added in both Figure 1 and Figure 2.

layer contains a bias. Let e and d represents the bias of the hidden and the visible layer. The two biases, which corresponds to the neurons of both the layers in RBM 1 is expressed as equation (14) and (15).

$$d^1 = \left\{d_1^1, d_2^1, \ldots, d_x^1, \ldots, d_l^1\right\} \tag{14}$$

$$e^1 = \left\{e_1^1, e_2^1, \ldots, e_c^1, \ldots, e_y^1\right\} \tag{15}$$

Where, e_c^1 denotes the bias of c^{th} hidden neuron, d_x^1 represents the bias of x^{th} visible neuron, and the weight of RBM 1 is represented as equation (16).

$$B^1 = \left\{B_{xc}^1\right\}; 1 \le x \le l; 1 \le c \le y \tag{16}$$

Where, B_{xc}^1 denotes the weight between the x^{th} visible neuron and the c^{th} hidden neuron, and the weight vector size is $l \times y$. Thus, the output of the first RBM hidden layer is calculated based on the weight and the bias associated with the visible neuron as represented as equation (17).

$$b_c^1 = \lambda\left[e_c^1 + \sum_x a_x^1 B_{xc}^1\right] \tag{17}$$

Where, λ denotes the activation function. Therefore, the output attained from the first RBM is expressed as equation (18).

$$b^1 = \left\{b_c^1\right\}; 1 \le c \le y \tag{18}$$

The learning process for RBM 2 is started using the output of hidden layer from RBM 1. The output attained from the first RBM is passed as input to the RBM 2 visible layer. Hence, the number of visible neurons in RBM 2 is similar to the number of neurons in the hidden layer in RBM 1. Thus, it can be expressed as equation (19).

$$a^2 = \left\{a_1^2, a_2^2, \ldots, a_y^2\right\} = \left\{b_c^1\right\}; 1 \le c \le y \tag{19}$$

Where, $\left\{b_c^1\right\}$ denotes the output vector of RBM 1. The hidden layer of RBM 2 is represented using equation (20).

$$b^2 = \left\{b_1^2, b_2^2, \ldots, b_c^2, \ldots, h_y^2\right\}; 1 \le c \le y \tag{20}$$

The bias of the visible layer, and the bias of the hidden layer are represented in the equation (14) and (15), and are represented as, d^2 and e^2, respectively. The weight vector of the second RBM is expressed as equation (21).

$$B^2 = \left\{B_{cc}^2\right\}; 1 \le c \le y \tag{21}$$

Where, B_{cc}^{2} is the weight between the c^{th} visible neuron and the c^{th} hidden neuron in RBM 2, hence the weight vector size is given as, $y \times y$. The output attained from the second RBM of c^{th} hidden neuron is expressed as equation (22).

$$b_c^2 = \lambda\left[e_c^2 + \sum_x a_x^2 B_{cc}^2\right] \forall a_x^2 = b_c^1 \tag{22}$$

Where, e_c^2 denotes the bias of c^{th} hidden neuron. Therefore, the output of the hidden layer is denoted as, equation (23).

$$b^2 = \left\{b_c^2\right\}; 1 \le c \le y \tag{23}$$

The output attained from the second RBM is passed as input to the MLP, which contains the number of neurons in the input layer as, y. The input of the MLP is expressed as equation (24).

$$f = \left\{f_1, f_2, \ldots, f_c, \ldots, f_y\right\} = \left\{b_c^2\right\}; 1 \le c \le y \tag{24}$$

Where, y denotes the number of neurons present in the input layer passed by the output of RBM2 hidden layer $\left\{b_c^2\right\}$. The MLP hidden layer is expressed as equation (25).

$$g = \{g_1, g_2, \ldots, g_C, \ldots, s_D\}; 1 \le C \le D \tag{25}$$

Where, D represents the number of hidden neurons. Let us assume E_C as the bias of C^{th} hidden neuron, where $C = 1,2,\ldots,D$. The output layer of MLP is expressed as equation (26).

$$h = \{h_1, h_2, \ldots, h_i, \ldots, h_m\}; 1 \le i \le m \tag{26}$$

Where, m denotes the neurons of the output layer. The MLP contains two weight vectors, as one is defined between the input and the hidden layer, and the second is defined between the hidden and the output layer. Let B^F denotes the weight between the input and the hidden layer, and is expressed as equation (27).

$$B^F = \left\{B_{cC}^F\right\}; 1 \le c \le y; 1 \le C \le D \tag{27}$$

Where, B_{cC}^F represents the weight between c^{th} input neuron and C^{th} hidden neuron, thus the size of B^F is denoted as, $y \times D$. According, to the weight and bias of the neuron the output of the hidden layer is computed as equation (28).

$$g_C = \left[\sum_{c=1}^{y} B_{cC}^F * f_c\right] E_C \forall f_c = b_c^2 \tag{28}$$

Where, E_C denotes the bias of the hidden neuron and $f_c = b_c^2$, as the output of RBM 2 is the input of the MLP. The weight between the hidden and the output layer is represented as B^G and is expressed as equation (29).

$$B^G = \left\{B_{Ci}^G\right\}; 1 \le C \le D; 1 \le i \le m \tag{29}$$

Based on the output of the weight B^G and the hidden layer, the output vector is calculated as equation (30).

$$h_i = \sum_{C=1}^{D} B_{Ci}^G * g_C \tag{30}$$

Where, B_{Ci}^G denotes the weight between the C^{th} hidden neuron and i^{th} output neuron and the hidden layer output is denoted as, g_C .

The RBM uses the gradient descent method to achieve the unsupervised learning, whereas the MLP uses the T-EHO algorithm. The MLP layer is trained using the T-EHO algorithm, and the training procedure is elaborated as follows:

i) The weight is applied to the input layer and then to the hidden layer using the random value.
ii) The features are extracted and the output of RBM 2 is fed into the MLP layer.
iii) The optimal weight is required in the MLP layer to compute the error value.
iv) The weight of the hidden and the input layer is updated using the T-EHO algorithm, which is expressed in equation (27).

c) Detection output based on the proposed FT-EHO-DBN classifier

The detection output attained obtained using the proposed FT-EHO-DBN classifier is represented as equation (31).

$$Z = \alpha.h_i + \beta.h_j \tag{31}$$

Where, h_i denotes the output of the fuzzy classifier and h_j represents the output of the T-EHO-based DBN classifier. The extracted features and the packet information are processed by the feature selector engine by using the holoentropy. Moreover, the classification process is performed based on the fuzzy classifier and hence, the attack detection is achieved.

4. Simulation model

Extensive computer simulations have been performed to evaluate the performance of the proposed system. The proposed FT-EHO-DBN classifier is implemented in the tool MATLAB with the PC of windows 10 OS, intel I3 processor, and 4 GB RAM.

4.1 Database description

The proposed FT-EHO-DBN classifier considers three databases to detect the DDoS attack and are explained as follows:

KDD cup database: The KDD cup database [24] is the standard database used to perform anomaly detection. It offers various features and the data that exist in the network connections helps to identify the nodes under attack.

Database 1: The database 1 have the total data generated as 2500, and the total number of users as 100 for DDoS detection.

Database 2: The database 2 contains the server log information with the total information contents as 150,000.

4.2 Performance metrics

The proposed Fuzzy and TEHO-based DBN classifier uses the metrics, like accuracy, recall, precision, and detection accuracy to analyze the performance of DDoS detection. The metrics are elaborated as follows:

Detection accuracy: Detection accuracy refers to the ratio of detected DDoS attacks with the ground truth information.

Accuracy: The accuracy determines the accurateness of the performance, and is measured as,

$$Accuracy = \frac{S + T}{S + T + Y + Z} \tag{32}$$

Where, T and S, , denotes the true negative and true positive, Z and Y represents the false negative and false positive respectively.

Precision: It is the ratio of the detected nodes with the actual number of nodes.

Recall: The recall refers to the ratio of the measure of normal nodes in the cloud with the actual number of normal nodes.

4.3 State-of-the-art method for comparison

The analysis is performed by comparing the proposed FT-EHO-DBN classifier with the existing methods, like SVM [19], NN [20], Ensemble [21], EHO [22], and TEHO-based DBN. The SVM [19] classifier is mainly used to detect the attacks in the cloud environment by fixing the decision boundary. NN [20] detects the attack by using the weight of the optimization algorithm. The ensemble [21] classifier is used to select the features to perform the attack detection. EHO [22] algorithm is mainly used to train the DBN classifier [23] to perform the classification.

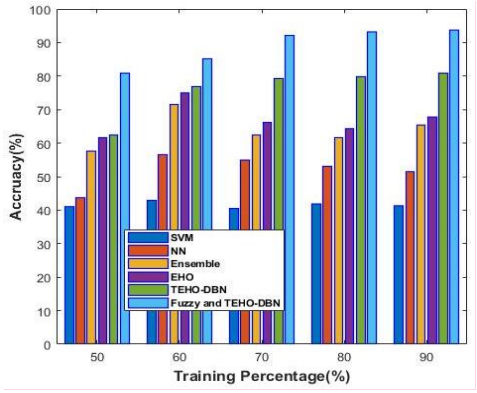
4.4 Results and Analysis

This section describes the comparative analysis of the proposed FT-EHO-DBN classifier and the results attained using different datasets are elaborated by varying the number of users and the percentage of data.

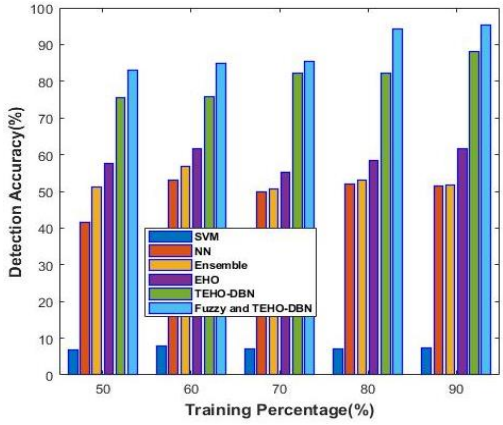
4.4.1 Comparative analysis using the KDD cup dataset

The comparative analysis using the KDD cup dataset for the proposed FT-EHO-DBN classifier is discussed briefly. In the comparative analysis, the percentage of data is varied with the large information in the KDD cup dataset. Figure 3 shows the comparative analysis of the proposed FT-EHO-DBN classifier by varying the percentage of data.

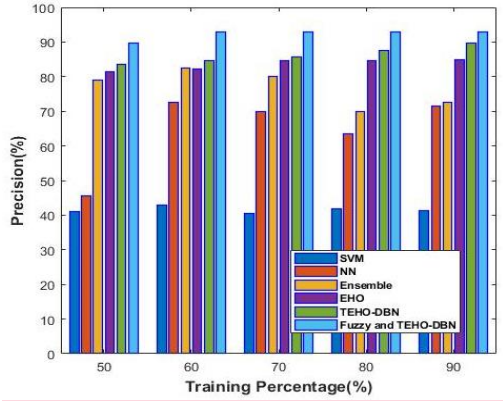
Figure 3 a) shows the analysis of accuracy by varying the percentage of data. When the raining data is 60%, the accuracy attained by the existing methods, like SVM, NN, Ensemble, EHO, and TEHO-DBN is 42.9762%, 56.5476%, 71.4286%, 75.0%, and 76.9231%, whereas the proposed FT-EHO-DBN classifier obtain the accuracy value as 85.2786%. For 90% training data, the accuracy computed by the existing methods, namely SVM, NN, Ensemble, EHO, and TEHO-DBN is 41.389%, 51.587%, 65.385%, 67.857%, and 81.0%, while the proposed FT-EHO-DBN classifier obtained increased accuracy value as 93.645%, respectively.


Accruacy(%)
Training Percentage(%)
SVM
NN
Ensemble
EHO
TEHO-DBN
Fuzzy and TEHO-DBN

(a)


Detection Accuracy(%)
Training Percentage(%)
SVM
NN
Ensemble
EHO
TEHO-DBN
Fuzzy and TEHO-DBN

(b)


Precision(%)
Training Percentage(%)
SVM
NN
Ensemble
EHO
TEHO-DBN
Fuzzy and TEHO-DBN

(c)

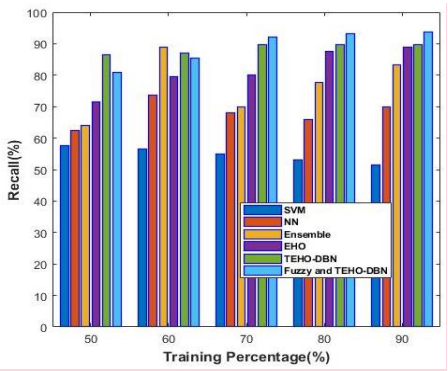


Figure 3. Comparative analysis using the KDD cup database by varying the percentage of data for (a) accuracy; (b) detection accuracy; (c) precision and (d) recall.

Figure 3 b) shows the analysis of detection accuracy by varying the percentage of data. When the percentage of training data is 60, the detection accuracy attained by the existing methods, like SVM, NN, Ensemble, EHO, and TEHO-DBN is 7.7844%, 53.2%, 56.6866%, 61.5385%, and 75.7633, whereas the proposed FT-EHO-DBN classifier obtained the detection accuracy value as 84.7872% respectively. For 90% training data, the detection accuracy computed by the existing methods, namely SVM, NN, Ensemble, EHO, and TEHO-DBN is 7.3852%, 51.6%, 51.6966%, 61.5385%, and 88.2353%, while the proposed FT-EHO-DBN classifier obtained increased detection accuracy value as 95.2697%, respectively.

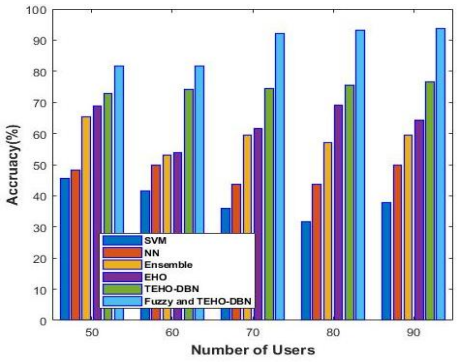
Figure 3 c) shows the analysis of precision by varying the percentage of data. When the training data is 60%, the precision attained by the existing methods, namely SVM, NN, Ensemble, EHO, and TEHO-DBN is 42.9880%, 72.7273%, 82.6087%, 82.1220%, and 84.6491%, whereas the proposed FT-EHO-DBN classifier obtained the precision value as 92.9672% respectively. For 90% training data, the precision value computed by the existing methods, namely SVM, NN, Ensemble, EHO, and TEHO-DBN is 41.3944%, 71.4286%, 72.7273%, 84.8718%, and 89.6154%, while the proposed FT-EHO-DBN classifier obtained better precision value as 92.9774%, respectively.

Figure 3 d) shows the analysis of recall by varying the percentage of data. When the training data is 60%, the recall attained by the existing methods, such as SVM, NN, Ensemble, EHO, and TEHO-DBN is 56.574%, 73.585%, 88.889%, 79.625%, and 87.048%, whereas the proposed FT-EHO-DBN classifier obtain the recall value as 85.297% respectively. For 90% training data, the

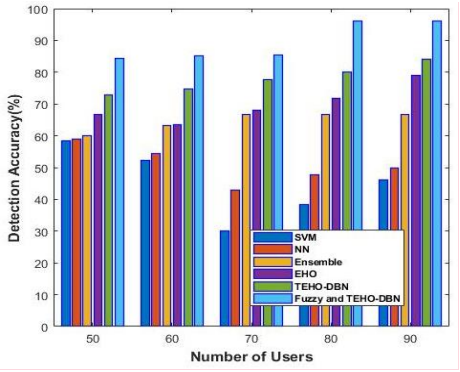
recall value computed by the existing methods, like SVM, NN, Ensemble, EHO, and TEHO-DBN is 51.594%, 69.811%, 83.333%, 88.889%, and 89.814, while the proposed FT-EHO-DBN classifier obtain better recall value as 93.670%, respectively.

4.4.2 Comparative analysis using the dataset 1

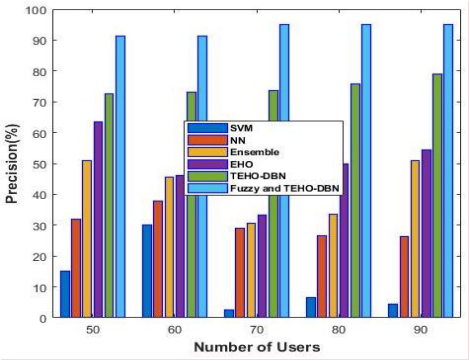
The comparative analysis of the proposed FT-EHO-DBN classifier using dataset 1 is discussed in this section. Figure 4 shows the analysis of the proposed FT-EHO-DBN classifier by varying the number of users.



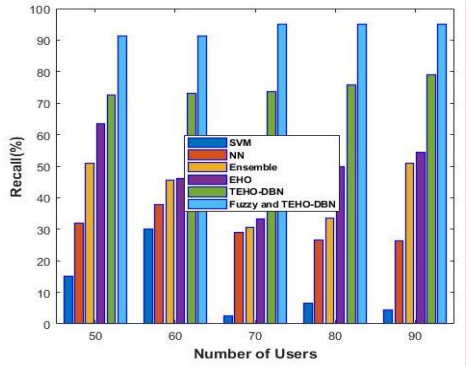
(a)



(b)



(c)



(d)

Figure 4. Comparative analysis using the database 1 by varying the number of users for (a) accuracy, (b) detection accuracy, (c) precision, and (d) recall.

Figure 4 a) shows the analysis of accuracy by varying the percentage of data. When the training data is 60%, the accuracy obtained by the existing methods, namely SVM, NN, Ensemble, EHO, and TEHO-DBN is 41.563%, 50.0%, 53.125%, 53.846%, and 74.186%, whereas the proposed FT-EHO-DBN classifier have the accuracy rate as 81.757% respectively. For 90% training data, the accuracy rate computed by the existing methods, namely SVM, NN, Ensemble, EHO, and TEHO-DBN is 37.872%, 50.0%, 59.574%, 64.286%, and 76.538%, while the proposed FT-EHO-DBN classifier obtain better accuracy value as 93.811%, respectively.

Figure 4 b) shows the analysis of detection rate by varying the percentage of data. When the percentage of training data is 60, the detection accuracy attained by the existing methods, like

SVM, NN, Ensemble, EHO, and TEHO-DBN is 52.381%, 54.545%, 63.158%, 63.636%, and 74.667%, whereas the proposed FT-EHO-DBN classifier obtain the detection accuracy value as 85.193% respectively. For 90% training data, the detection accuracy computed by the existing methods, namely SVM, NN, Ensemble, EHO, and TEHO-DBN is 46.154%, 50.0%, 66.667%, 78.947%, and 84.091%, while the proposed FT-EHO-DBN classifier obtain increased detection accuracy value as 96.030%, respectively.

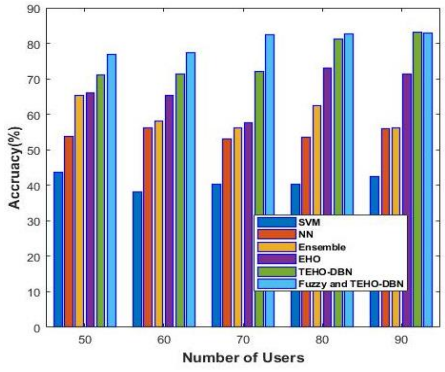
Figure 4 c) shows the analysis of precision by varying the percentage of data. When the training data is 60%, the precision value attained by the existing methods, namely SVM, NN, Ensemble, EHO, and TEHO-DBN is 30%, 37.826%, 45.555%, 46.153%, and 73%, whereas the proposed FT-EHO-DBN classifier have the precision value as 91.4% respectively. For 90% training data, the precision computed by the existing methods, namely SVM, NN, Ensemble, EHO, and TEHO-DBN is 4.444%, 26.364%, 50.870%, 54.545%, and 78.947%, while the proposed FT-EHO-DBN classifier obtain better precision value as 94.981%, respectively.

Figure 4 d) shows the analysis of recall by varying the percentage of data. When the training data is 60%, the recall value attained by the existing methods, namely SVM, NN, Ensemble, EHO, and TEHO-DBN is 26.667%, 60.0%, 60.870%, 66.667%, and 78.571%, whereas the proposed FT-EHO-DBN classifier obtain the recall value as 81.778% respectively. For 90% training data, the recall value computed by the existing methods, like SVM, NN, Ensemble, EHO, and TEHO-DBN is 15.926%, 25.0%, 66.667%, 75.0%, and 86.957, while the proposed FT-EHO-DBN classifier have better recall value as 93.833%, respectively.

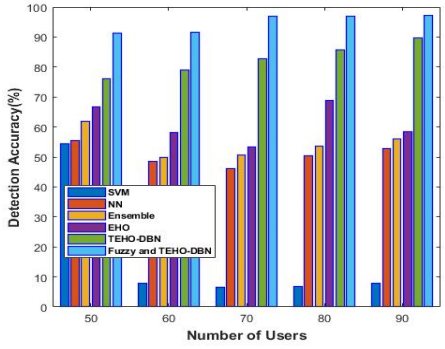
4.2.3 Comparative analysis using the dataset 2

The comparative analysis made using the dataset2 for the proposed FT-EHO-DBN classifier is discussed in this section. Figure 5 shows the analysis of the proposed FT-EHO-DBN classifier by varying the number of users.

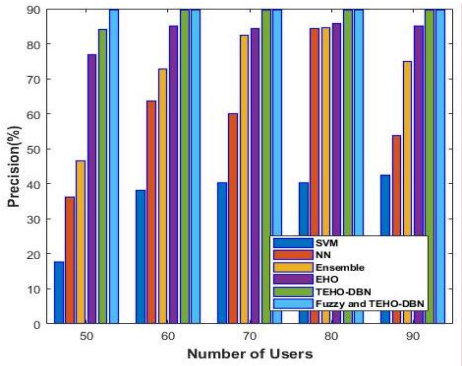
Figure 5 a) shows the analysis of accuracy by varying the percentage of data. When training data is 60%, the accuracy obtained by the existing methods, namely SVM, NN, Ensemble, EHO, and TEHO-DBN is 38.214%, 56.250%, 58.135%, 65.385%, and 71.429%, whereas the proposed FT-EHO-DBN classifier obtain the accuracy rate as 77.440% respectively. For 90% training data, the accuracy rate computed by the existing methods, namely SVM, NN, Ensemble, EHO, and TEHO-DBN is 42.579%, 55.952%, 56.250%, 71.429%, and 83.077%, while the proposed FT-EHO-DBN classifier obtain better accuracy value as 82.921%, respectively.


Accruacy(%)
Number of Users
SVM
NN
Ensemble
EHO
TEHO-DBN
Fuzzy and TEHO-DBN

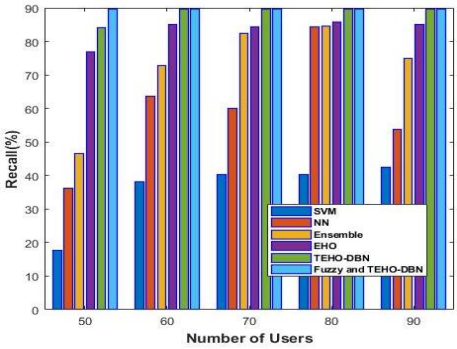
(a)


Detection Accuracy(%)
Number of Users
SVM
NN
Ensemble
EHO
TEHO-DBN
Fuzzy and TEHO-DBN

(b)


Precision(%)
Number of Users
SVM
NN
Ensemble
EHO
TEHO-DBN
Fuzzy and TEHO-DBN

(c)



(d)

Figure 5. Comparative analysis using database 2 by varying the number of users for (a) accuracy, (b) detection accuracy, (c) precision, and (d) recall.

Figure 5 b) shows the analysis of detection rate by varying the percentage of data. When the percentage of training data is 60, the detection accuracy attained by the existing methods, like SVM, NN, Ensemble, EHO, and TEHO-DBN is 7.784%, 48.4%, 50.0%, 58.283%, and 78.889%, whereas the proposed FT-EHO-DBN classifier obtain the detection accuracy value as 91.710% respectively. For 90% training data, the detection accuracy computed by the existing methods, namely SVM, NN, Ensemble, EHO, and TEHO-DBN is 47.784%, 52.8%, 56.088%, 58.333%, and 89.657%, while the proposed FT-EHO-DBN classifier obtain increased detection accuracy value as 97.2%, respectively.

Figure 5 c) shows the analysis of precision by varying the percentage of data. When the training data is 60%, the precision value attained by the existing methods, namely SVM, NN, Ensemble, EHO, and TEHO-DBN is 38.207%, 63.636%, 72.727%, 85.122%, and 89.619%, whereas the proposed FT-EHO-DBN classifier obtain the precision value as 89.688% respectively. For 90% training data, the precision computed by the existing methods, namely SVM, NN, Ensemble, EHO, and TEHO-DBN is 42.590%, 53.846%, 75.0%, 85.122%, and 89.645%, while the proposed FT-EHO-DBN classifier obtained better precision value as 89.701%, respectively.

Figure 5 d) shows the analysis of recall by varying the percentage of data. When the percentage of training data is 60, the recall value attained by the existing methods, namely SVM, NN, Ensemble, EHO, and TEHO-DBN is 58.167%, 70.0%, 73.585%, 84.211%, and 82.588%, whereas the proposed FT-EHO-DBN classifier obtain the recall value as 77.460% respectively. For 90% training data, the recall value computed by the existing methods, like SVM, NN, Ensemble, EHO, and TEHO-DBN is 55.976%, 73.585%, 87.5%, 89.623%, and 89.474%, while the proposed FT-EHO-DBN classifier obtained better recall value as 82.931%, respectively.

4.3 Discussion

The comparative discussion of the proposed Fuzzy-Taylor-EHO inspired DBN classifier is elaborated in this section. Table I shows the comparative results based on various performance metrics. The values computed by the existing methods, namely SVM, NN, Ensemble, EHO, and T-EHO-based DBN are also presented Table 1. The proposed FT-EHO-DBN classifier attained better performance than the existing methods for the metrics, like accuracy, detection accuracy, precision, and recall with the values of 93.811%, 97.200%, 94.981%, and 93.833, respectively.

Commented [HP13]: R5C1. The paper presents a proposal of Fuzzy-Taylor- Elephant Herd Optimization Inspired Deep Belief Network for DDoS Attack Detection and Comparison with State-of-the-Arts Algorithms. Authors discuss their algorithm and perform some experiments.
The paper presents several problems. First of all, the motivation for the paper is well explained. Although it might be clear the relevance of DDoS attack detection in the context of cloud computing, it is clearly mentioned why the proposal presented in the paper addresses the challenges of the research problem. Also, the selection of the proposed technique looks somehow ad-hoc, since the authors are not explaining adequately why their approach makes sense in the context of the problem.

Response: The proposed method has the advantages of DBN, EHO, and Taylor series, which are provided in sub-section 4.4 of the revised manuscript. Due to the advantages of the utilized methods, the propose technique detects the DDoS attacks more precisely.

Table 1. Comparative discussion.

Comparative techniques	Evaluation metrics			
	Accuracy(%)	Detection rate (%)	Precision(%)	Recall (%)
SVM	45.556	58.333	42.988	58.167
NN	56.548	59.091	84.444	73.585
Ensemble	71.429	66.667	84.615	88.889
EHO algorithm	75.000	78.947	85.714	89.623
TEHO-DBN classifier	83.077	89.657	89.645	89.814
Proposed FT-EHO-DBN	**93.811**	**97.200**	**94.981**	**93.833**

This result relevels that SVM (accuracy = 45.556%) and NN (accuracy = 56.548%) showed the worst performance as far as accuracy is concerned. In addition, we noted that the precision value for SVM is 42.988% lowest than any other methods implemented in this paper. Results presented in Table I concludes that SVM showed the worst performance whereas Ensemble, EHO, and TEHO-DBN demonstrated moderate performance, whilst the proposed FT-EHO-DBN classifier has outperformed on the performance matrics. The overall performance has achieved through FT-EHO-DBN classifier is mainly because integrating Taylor series and EHO algorithm helped in reaching to global optimum without getting stuck at local optimum. Initially, the user request is send to the packet feature extraction module, where the packet features are extracted. The extracted features are subjected to the feature selector engine, where the selective features is extracted using the holoentrophy criteria. The classification module uses the Deep Belief Network and the fuzzy classifier to detect the Distributed Denial of Service attack.

5. Conclusion

A FT-EHO-DBN classifier has been introduced in this paper to detect the DDoS attack. FT-EHo-DBN classifier has been developed by integrating the fuzzy and DBN classifier along with the T-EHO optimization algorithm. We showed the working of the proposed system in a well-organized manner so that researcher can utilize it in future. Initially, the user request is sent to the packet feature extraction module, where the packet features are extracted. The extracted features are subjected to the feature selector engine, where the selective features are extracted using the holoentrophy criteria. The classification module uses the DBN and fuzzy classifier to detect the DDoS attack. In FT-EHO-DBN system, the T-EHO algorithm has been implemented as rule learning approach of the fuzzy classifier. T-EHO algorithm has been implement as replacement of an adaptive genetic fuzzy system. Rigorous computer simulations have been performed to evaluate the performance of the proposed FT-EHO-DBN classifier. Three standard databases, namely, KDD cup database, Database 1 and Database 2 have been used for the simulations. Four performance metrics, namely, accuracy, detection accuracy,

Commented [HP14]: R2C4. Experimental design is detailed, and results have been discussed in significant detail but I would suggest authors to include a paragraph why/how this improvements have been achieved.

Response: Many thanks on this comment. In the revised manuscript we have added details.

Commented [HP15]: **R3C4.** Results presented in Table I shows clearly that the proposed method has brought significant improvement but how/why this improvement is achieved need to be discussed.

Response: The overall performance has achieved through Fuzzy and Deep Belief Network classifier based on the Taylor-Elephant Herd optimization algorithm, which is the integration of Taylor series with the Elephant Herd optimization algorithm. Initially, the user request is sent to the packet feature extraction module, where the packet features are extracted. The extracted features are subjected to the feature selector engine, where the selective features are extracted using the holoentrophy criteria. The classification module uses the Deep Belief Network and the fuzzy classifier to detect the Distributed Denial of Service attack.

Commented [HP16]: R4C3. A comparison results need to explain in the sense why the proposed method has brought improvement.

Response: Thank you for pointing this. It is included in the revised manuscript.

precision and recall have been identified to measure the result's quality. The performance of the FT-EHO-DBN DBN classifier was tested against the state-of-the-art methods and comparative results have been presented in Table I. The results reported in Table I reveals that the FT-EHO-DBN classifier outperformed over other state-of-the-art algorithms. Computational cost is found challenging during implementation mainly because of multiple hidden layers. Hence, The immediate future scope is to explore the possibility of developing new optimization algorithm (or replace EHO with other swarm algorithms) and combine with the DBN so that performance in terms of accuracy and detection rate with less computational cost can be achieved.

Commented [HP17]: **R5C4.** In the experiments section, the author has compared their results and performance state of the rat algorithm and its better accuracy and anyhow the author should mention their limitation in the proposed system.

Response: As per the suggestion, the limitation of the proposed system is added.

Commented [HP18]: **R4C4**. The author need to update the conclusion and highlight the future scope of this research.

Response: The immediate future scope is to explore the possibility of developing new optimization algorithm (or replace EHO with other swarm algorithms) and combine with the DBN so that performance in terms of accuracy and detection rate with less computational cost can be achieved.

Commented [HP19]: **R5C2**. A different aspect of the design such as the feature selection using Fuzzy-Taylor looks good. The operation process based on the creation of a log is also mentioned well. Considering the involvement of Deep Belief Networks, it would be important to evaluate whether the approach can operate in reasonable times, considering that the authors claim that it can detect the attack in earlier stages.

Response: In this research, we presented results on four quality measures, namely, accuracy, detection rate, precision and recall. Computational cost will be the agenda for further research and we highlighted this in the conclusion section.

References

[1]. Yasir Ali, Yuanqing Xia, Liang Ma, and Ahmad Hammad, " Secure design for cloud control system against distributed denial of service attack", Control Theory and Technology, vol.16, no.1, pp.14–24, February 2018.

[2]. R. Kesavamoorthy and K. Ruba Soundar, "Swarm intelligence based autonomous DDoS attack detection and defense using multi agent system", Cluster Computing, pp.1-8, 13 March 2018.

[3]. Opeyemi Osanaiye, Haibin Cai, Kim-Kwang Raymond Choo, Ali Dehghantanha, Zheng Xu, and Mqhele Dlodlo, " Ensemble-based multi-filter feature selection method for DDoS detection in cloud computing", EURASIP Journal on Wireless Communications and Networking, vol.2016, no.1, pp.130, 10 May 2016.

[4]. Vikash C Pandey, Sateesh K Peddoju, and Prachi S Deshpande, " A statistical and distributed packet filter against DDoS attacks in Cloud environment", Sādhanā, 43:32, vol.43, no.3, pp.32, March 2018.

[5]. Loukas G, Vuong T, Heartfield R, Sakellari G, Yoon Y and Gan D, "Cloud-based cyber-physical intrusion detection for vehicles using deep learning", IEEE Access, vol. 6, pp.3491-3508, 2018.

[6]. Hajimirzaei B and Navimipour NJ, "Intrusion detection for cloud computing using neural networks and artificial bee colony optimization algorithm", ICT Express, 2018.

[7]. Deng S, Zhou A.H, Yue D, Hu B and Zhu L.P, "Distributed intrusion detection based on hybrid gene expression programming and cloud computing in a cyber physical power system", IET Control Theory & Applications, vol. 11, no. 11, pp.1822-1829, 2017.

[8]. Chen M, Wang N, Zhou H and Chen Y, “FCM technique for efficient intrusion detection system for wireless networks in cloud environment”, Computers & Electrical Engineering, vol. 71, pp.978-987, 2018.

[9]. Gao Y, Liu Y, Jin Y, Chen J and Wu H, “A Novel Semi-Supervised Learning Approach for Network Intrusion Detection on Cloud-Based Robotic System”, IEEE Access, vol. 6, pp. 50927-50938, 2018.

[10]. Saurabh Dey, Qiang Ye, and Srinivas Sampalli, "A Machine Learning Based Intrusion Detection Scheme for Data Fusion in Mobile Clouds Involving Heterogeneous Client Networks", Information Fusion, vol. 49, ps. 205-215, September 2019.

[11]. Idhammad M, Afdel K and Belouch M, “Distributed Intrusion Detection System for Cloud Environments based on Data Mining techniques”, Procedia Computer Science, vol. 127, pp.35-41, 2018.

[12]. Osanaiye O, Choo K.K.R and Dlodlo M, “Distributed denial of service (DDoS) resilience in cloud: review and conceptual cloud DDoS mitigation framework”, Journal of Network and Computer Applications, vol. 67, pp.147-165, 2016.

[13]. Patil R, Dudeja H, Gawade S and Modi C, “Protocol specific Multi-threaded Network Intrusion Detection System (PM-NIDS) for D0S/DDoS Attack Detection in cloud”, IEEE International Conference on Computing, Communication and Networking Technologies (ICCCNT), pp. 1-7, July 2018.

[14]. Lonea A.M, Popescu D.E and Tianfield H, “Detecting DDoS attacks in cloud computing environment”, International Journal of Computers Communications & Control, vol. 8, no.1, pp.70-78, 2013.

[15]. JoSEP A.D, KAtz R, KonWinSKi A, Gunho L.E.E, PAttERSon D and RABKin A, “A view of cloud computing”, Communications of the ACM, vol. 53, no. 4, 2010.

[16]. Kholidy H.A. and Baiardi F, “CIDS: A framework for intrusion detection in cloud systems”, In IEEE Ninth International Conference on Information Technology-New Generations, pp. 379-385, April 2012.

[17]. Zhao, B, Fan P. and Ni M, “Mchain: A Blockchain-Based VM Measurements Secure Storage Approach in IaaS Cloud With Enhanced Integrity and Controllability”, IEEE Access, vol. 6, pp.43758-43769, 2018.

[18]. Hiremath S. and Kunte S, “A novel data auditing approach to achieve data privacy and data integrity in cloud computing”, In IEEE International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT), pp. 306-310, December 2017.

[19]. Dennis B. and Muthukrishnan S, “AGFS: Adaptive Genetic Fuzzy System for medical data classification”, Applied Soft Computing, vol. 25, pp.242-252, 2014.

[20]. Mane V.M. and Jadhav D.V., “Holoentropy enabled-decision tree for automatic classification of diabetic retinopathy using retinal fundus images”, Biomedical Engineering/Biomedizinische Technik, vol. 62, no. 3, pp.321-332, 2017.

[21]. Opeyemi Osanaiye, Haibin Cai, Kim-Kwang Raymond Choo, Ali Dehghantanha, Zheng Xu, and Mqhele Dlodlo, " Ensemble-based multi-filter feature selection method for DDoS detection in cloud computing", EURASIP Journal on Wireless Communications and Networking, vol.2016, no.1, pp.130, 10 May 2016.

[22]. Gai-Ge Wang, Suash Deb, and Leandro dos S. Coelho, " Elephant Herding Optimization", In Proceedings of the 3rd International Symposium on Computational and Business Intelligence (ISCBI), pp.1-5, 2015.

[23]. Hinton, G.E., “Deep belief networks,” Scholarpedia, vol. 4, no. 5, 2009.

[24]. KDD cup database http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html,b Accessed on November 2018.

[25]. S. Alamelu Mangai, B. Ravi Sankar and K. Alagarsamy, “Taylor Series Prediction of Time Series Data with Error Propagated by Artificial Neural Network”, International Journal of Computer Applications (0975 – 8887), Vol. 89, no.1, March 2014.

[26]. Yan, Q., Yu, F. R., Gong, Q., & Li, J. (2015). Software-defined networking (SDN) and distributed denial of service (DDoS) attacks in cloud computing environments: A survey, some research issues, and challenges. IEEE communications surveys & tutorials, 18(1), 602-622.

[27]. Somani, G., Gaur, M. S., Sanghi, D., Conti, M., & Buyya, R. (2017). DDoS attacks in cloud computing: Issues, taxonomy, and future directions. Computer Communications, 107, 30-48

Commented [HP20]: **R3C3**. More recent references must be added to present state-of-the art.

Response: As per the suggestion, some recent work was added into state-of-the art.

Commented [HP21]: **R4C2.** I think the references used in the paper are old. The author must update the new references and make the paper up to date.

Response: As per the reviewer suggestion, the revised manuscript has updated

Commented [HP22]: **R4C5**. The authors should add more recent work in the literature section as most of the references are outdated in the paper so it’s nice to update the references.

Response: The below mentioned references work were added into related work

- Yan, Q., Yu, F. R., Gong, Q., & Li, J. (2015). Software-defined networking (SDN) and distributed denial of service (DDoS) attacks in cloud computing environments: A survey, some research issues, and challenges. IEEE communications surveys & tutorials, 18(1), 602-622.
- Somani, G., Gaur, M. S., Sanghi, D., Conti, M., & Buyya, R. (2017). DDoS attacks in cloud computing: Issues, taxonomy, and future directions. Computer Communications, 107, 30-48.

Commented [HP23]: **R5C3**. The literature review is incomplete. Authors are not considering important literature on the area of DDoS attack detection in the context of cloud computing. This reviewer suggests considering the following references:
Yan, Q., Yu, F. R., Gong, Q., & Li, J. (2015). Software-defined networking (SDN) and distributed denial of service (DDoS) attacks in cloud computing environments: A survey, some research issues, and challenges. IEEE communications surveys & tutorials, 18(1), 602-622.
Somani, G., Gaur, M. S., Sanghi, D., Conti, M., & Buyya, R. (2017). DDoS attacks in cloud computing: Issues, taxonomy, and future directions. Computer Communications, 107, 30-48.

Response: As per the recommendation, above mentioned references are updated in the revised manuscript.