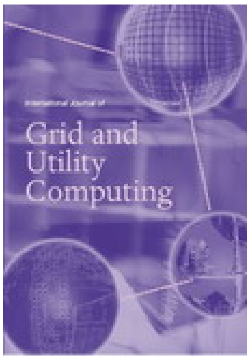
INDERSCIENCE PUBLISHERS
Linking academia, business and industry through research



International Journal of Grid and Utility Computing

ISSN online: 1741-8488 - ISSN print: 1741-847X
https://www.inderscience.com/ijguc

Optimal attack detection using an enhanced machine learning algorithm

Reddy Sai Sindhu Theja, Gopal K. Shyam, Shanthi Makka

DOI: 10.1504/IJGUC.2022.10064192

Article History:

Received:	05 August 2022
Last revised:	19 November 2022
Accepted:	25 November 2022
Published online:	12 January 2025

Optimal attack detection using an enhanced machine learning algorithm

Reddy Sai Sindhu Theja*

Department of CSE,
Vardhaman College of Engineering,
Hyderabad, Telangana, India
Email: thejasindhu@gmail.com
*Corresponding author

Gopal K. Shyam

School of CSE, Cloud Computing Lab,
Presidency University,
Bengaluru, Karnataka, India
Email: gopalshyambabu@gmail.com

Shanthi Makka

Department of CSE,
Vardhaman College of Engineering,
Hyderabad, Telangana, India
Email: dr.shanthimakka@gmail.com

Abstract: As computer network and internet technologies advance more quickly today, the importance of network security is widely acknowledged. This research intends to introduce a new security platform for SaaS framework, which comprises two major phases: (1) Optimal Feature Selection and (2) Classification. Initially, the optimal features are selected from the data set. A novel algorithm named Accelerator updated Rider Optimisation Algorithm (AR-ROA), a modified form of ROA and Deep Belief Network (DBN) based Attack Detection System is proposed. The optimal features that are selected form AR-ROA are subjected to DBN classification process, in which the presence of attacks is determined. The proposed model outperforms other traditional models in aspects of Accuracy (95.3%), Specificity (98%), Sensitivity (86%), Precision (92%), Negative predictive value (97%), F1-score (86%), False positive ratio (2%), False negative ratio (10%), False detection ratio (10%), and Matthew's correlation coefficient (0.82%).

Keywords: software-as-a-service; SaaS framework; security; ROA; optimisation; DBN; attack detection system; feature selection; classification; machine learning algorithms; security issues.

Reference to this paper should be made as follows: Theja, R.S.S., Shyam, G.K. and Makka, S. (2025) 'Optimal attack detection using an enhanced machine learning algorithm', *Int. J. Grid and Utility Computing*, Vol. 16, No. 1, pp.1–14.

Bibliographical notes: Reddy Sai Sindhu Theja is working as an Associate Professor at Vardhaman College of Engineering, Hyderabad, India. She received her PhD degree from REVA University, School of Computing & IT, Bangaluru, India, under the guidance of Dr. Gopal K. Shyam, MTech degree from KL University, Vijayawada and BTech degree from MRR Institute of Science and Technology, JNTU Hyderabad. Her research interests include cloud computing, machine learning, etc. She has handled several subjects for UG/PG students like Database Management Systems, Mathematical Foundations of Computer Science, Machine Learning, Object Oriented programming through Java, Software Project Management, Real-Time Operating Systems, Computer Networks, Agile project management and Cloud Computing. She has published many papers in highly reputed National/International Conferences like IEEE, Springer, etc., and 5 papers in Journals with high-impact factor like *IJKSUCIS*, *ASOC* and *IJGUC*.

Gopal K. Shyam is an Associate Professor in School of Computing and IT at Presidency University. He received his BE, MTech and PhD degrees in Computer Science and Engineering from VTU, Belagavi. He has handled several subjects for UG/PG Students like Design and

Analysis of Algorithms, Computer Networks, Web programming, Advanced Computer Architecture, Information Security, Computer Concepts and C Programming. His research interest includes cloud computing, grid computing, high performance computing, etc. He has published about many papers in highly reputed national/international conferences like IEEE, Elsevier, Springer, etc. and 8 papers in journals with high-impact factor like *JNCA*, *ASOC* and *IJCC*.

Shanthi Makka completed her PhD degree from Birla Institute of Technology, Ranchi (Mesra), MTech degree from Galgotia College of Engineering, Greater Noida, Uttar Pradesh and BTech degree from GITAM, Visakhapatnam, Andhra Pradesh. She has 19 years of experience in Academics. Her research interests include Blockchain technology, data science and theoretical computer science, authored a book on 'Identification of Parallel Modules' and also published book chapters in Springer and IGI Global. She has worked as Head of the Departments in various Engineering Colleges, and published a patent on Wireless School Security System based on IoT using GSM & Micro-controller.

1 Introduction

Software-as-a-Service (SaaS) is at the top of the cloud stack. It takes on the Platform-as-a-Service (PaaS) and Infrastructure-as-a-Service (IaaS) security issues (Jangjou and Sohrabi, 2022; Lee et al., 2022). Customers can access it over a network. According to the OSI model, abstractions corresponding to lower-level functions are used to control the network administrators. The complexity in terms of security, integrity and confidentiality has increased the risk of preserving the user data by introducing the bugs and errors at the time of implementation (Zhang et al., 2022a; Abdou et al., 2018). Network virtualisation increases the complexity of data preservation, which puts tremendous pressure on programmers to create a faultless software solution (He et al., 2018). This pressure further creeps onto cloud computing, and one of the major issues in cloud computing is DoS attacks (Kalkan et al., 2018; Li et al., 2018).

The term 'SaaS' refers to a model for delivering and licencing software. Software deployment refers to the delivery of ready-to-use software to a user. In other words, software deployment is the process that occurs between the acquisition and execution of software (Wang et al., 2022; Liu et al., 2018; Yoon et al., 2017). In this paradigm, the software is maintained and stored in the cloud by the software supplier. The cloud-based software is accessible to end users (i.e., software consumers) via a web interface. There are plenty of attacks committed here because the software service is cloud-based and user-visible (Han et al., 2018). The application management, operating systems and the underlying infrastructure are managed and controlled by the cloud provider, and the consumer controls the settings on the user-specific application (Saidi et al., 2022; Zhang et al., 2022b; Ghosh et al., 2016). Hence, there is poor security in SaaS in the cloud, the primary focus of modern software security research is SaaS data security. Specific tools and measures are used to enforce information governance through data security (Hyun et al., 2018; Wang et al., 2017; El-Dine Atta et al., 2022).

In the world of cloud computing, the Cloud Service Providers (CSP) and unauthorised users may try to access outsourced data as well as divulge and infer private information about clients, which is thought to be valuable for conducting current data leakage attacks (Belguith et al., 2022). On the other side, web application vulnerabilities might make any SaaS product insecure. The internet has been used by opponents to hack into users' computers and carry out damaging deeds like stealing private information (Ouda et al., 2022). Similar to other internet applications, SaaS apps have security concerns, but typical security measures fall short in defending them from attacks, forcing the development of alternative strategies. One of the major issues with cloud security management is detecting the malicious software (Liu et al., 2022).

One of the major threats to online security is the botnet (Aziz et al., 2022). Botnet spreads itself periodically by infecting an increasing number of computers, laptops, servers and mobile devices. To increase data security in cloud computing, botnet detection research is still in its infancy (Onyema et al., 2022). Cybercrimes including DDoS, click fraud, phishing fraud, key logging, bit coin fraud, spamming, traffic sniffing, spreading new malware and Google AdSense misuse can all be carried out by a botmaster using bots (Salek et al., 2022).

In the existing research of SaaS security, High Availability and Integrity Layer (HAIL) plays a crucial role in cloud data storage. It ensures data integrity and availability through error-correcting layers. The main drawback of the existing work is, it can provide security to the static data but not to the dynamic data. The next more focused approach is Trusted Third Party. It is efficient in providing authentication, integrity and data confidentiality and the communication medium in a cloud environment via public key infrastructure. It failed due to its non-availability of data backup and recovery (Zhang et al., 2018). As a result, a specific method is required to eliminate the shortcomings of conventional models.

The main objectives of the paper are (i) Introducing a new algorithm named Accelerator updated Rider Optimisation Algorithm (AR-ROA). (ii) The proposed AR-ROA is validated on N-BaIoT data set. (iii) Feature selection is performed through AR-ROA algorithm. (iv) Further classification is performed through Deep Belief Network (DBN). (v) AR-ROA hits with Accuracy (95.3% over other conventional models), Specificity (98%), Sensitivity (86%), Precision (92%), Negative predictive value (97%), F1-score (86%), False positive ratio (2%), False negative ratio (10%), False detection ratio (10%) and Matthew's correlation coefficient (0.82%).

This document is organised as follows: Section 2 offers illustrations of the SaaS framework literature review. Section 3 expresses a detailed explanation of the proposed SaaS framework. Section 4 provides a description of the proposed AR-ROA for the optimal feature selection. Section 5 illustrates the DBN-based attack detection technique. Section 6 presents the findings and discussions. Conclusions and recommendations for further study are presented in Section 7.

An outline of the paper's primary contributions is provided below:

- To present a secure SaaS framework for detecting the botnet attacks. This attack detection system comprises two major phases namely Optimal Feature Selection and Classification.
- Two best algorithms were chosen namely Accelerator updated Rider Optimisation Algorithm (AR-ROA) and Deep Belief Network (DBN) that improves the ability to get rid of premature convergence.
- To validate the performance of the proposed work with other conventional algorithms using a benchmark data set with different metrics like Accuracy, Sensitivity and Specificity, etc.

2 Literature review

2.1 Related works

Tripathy et al. (2020) introduced an algorithm to determine SQL attacks. The paper uses machine learning classifiers to investigate the good and bad features. This methodology is divided into six phases viz., problem description, data cleaning and collection, feature extraction, training the model, and estimation. The results prove that the proposed work can distinguish the normal and malicious payloads with higher detection rate.

An application paradigm to create high-level security in SaaS applications hosted in a private cloud environment was proposed by Ghuge et al. (2020). The Least Significant Bit (LSB) strategy is utilised to provide security utilising a revolutionary video steganography approach, Advanced Encryption Standard (AES) is deployed to encrypt sensitive data and Hidden Markov model is employed as an intrusion detection technique.

Zhang et al. (2018) introduced an innovative User Security Authentication Scheme on SaaS Platform of Enterprises to correlate the characters of the champion-centred multi-tenant business and to combine the enterprise alliance-oriented independent database. With the help of a data encryption configuration method and a data query decryption technique, this model had secured the multi-tenant business data.

An attention-based Recurrent Convolutional Neural Network (RCNN) was put up by Prabhakaran and Kulandasamy (2021) to determine whether text data is being intruded upon. The Modified Flower Pollination Algorithm with Elliptical Curve Cryptography (MFP-ECC), is introduced that creates an end-to-end encryption system to raise security levels. This method depicts highest breaking time and can survive with various attacks when related with other models.

Saleh et al. (2015) introduced Sec Place for SaaS in the cloud environment for enhancing the security level for the tenants that were sharing the same infrastructure. The proposed model had reduced the hazard of co-resident tenants, which were available in the similar database server. The tenant subscription data was utilised by the Sec Place to improve the security of private data further.

Zhou (2018) projected a novel real-time security-based approach in SaaS referred to as a reliable and dynamic approach to find SDN devices (SDN-RDCD devices) that are compromised. The affected SDN devices were initially assessed while the switches and controller were in a trust-less condition. The backup controllers handled and audited the data created in the switches and the primary controller.

Fawcett et al. (2018) developed a new distributed SDN security framework that integrates SDN's efficiency with distributed system scalability and flexibility monitoring. This model was based on the multi-level remediation mechanism, and there was lightweight visibility throughout the information flows due to the unique security pipeline. Over the current models, the proposed model was evaluated, and the resultant of the analysis was low latency in the detection and had high scalability in extensive network monitoring.

Elsayed and Zulkernine (2019) proposed Security Diagnosis as a Service (SDaaS) to assess the security state of SaaS applications and identify data flow issues. The findings demonstrate that the methodology provides the best protection against problems like integrity and confidentiality. The outcomes demonstrate the diagnosis' increased accuracy, scalability and efficient use of resources.

The characteristics and problems with SaaS are shown in Table 1. Each method has a specific advantage that helps to reduce various issues. Also, apart from this, various challenges need to be addressed.

Table 1 Features and challenges of SaaS

Author and citation	*Adopted methodology*	*Features*	*Challenges*
Tripathy et al. (2020)	Machine Learning Classifiers	Detecting the attacks in application level	• Limits experimenting on huge data sets. • No security guarantee for users
Ghuge et al. (2020)	Hidden Markov Model, Advanced Encryption Standard (AES), Least Significant Bit (LSB)	Protect the SaaS application running in a private cloud.	• Working with huge data where data size grows exponentially remained as a challenge.
Zhang et al. (2018)	User Security Authentication Scheme	Supported customised data storage security	• Repeated encryption is required for mixed encryption
Prabhakaran and Kulandasamy (2021)	Recurrent Convolutional Neural Network (RCNN), Modified Flower Pollination Algorithm (MFP), Elliptical Curve Cryptography (ECC).	Enhances high-data security	• Not applicable to real service
Saleh et al. (2015)	Model for allocating resources – Sec Place.	• Resource allocation is high • Increase the level of security. • Gradually decrease the risks regarding the security of co-located tenants.	• CSRF attacks are not minimised • High cost
Zhou (2018)	A method for detecting corrupted SDN devices in real time.	• Prevents the central controller from being vulnerable. • Low cost	• Non-malicious problems are uncovered
Fawcett et al. (2018)	Distributed SDN security Framework.	• Scalability of a distributed • System is high	• High cost • High complexity
Elsayed and Zulkernine (2019)	Security Diagnosis as a Service (SDaaS)	• The violations regarding integrity and confidentiality should be shielded.	• Weak cryptography, hashing and randomness

SaaS security comprises getting clients on board with cloud-based corporate data assurance applications. As the name implies, SaaS distributes application software and transmits a lot of complex data through the Internet. In this type of computing service, Cloud Service Provider (CSP) oversees all aspects, including hardware and application software (Parast et al., 2022). Although SaaS has generated a lot of excitement, there are still a few security issues that need to be fixed (Vinoth et al., 2022). Some of the security issues to be addressed by SaaS are botnet attacks, phishing, DoS attacks, data breaches, data security, network security, etc.

2.2 *Related work on rider optimisation algorithm (ROA)*

A team of riders racing toward a destination is the foundation for ROA. Binu and Kariyappa (2019) employed ROA to diagnose faults in analogue circuitry. Additionally, a classifier known as RideNN is built utilising Neural Networks (NN). For the purpose of tackling engineering optimisation issues, Wang et al. (2019) suggested a hybridised algorithm called GSSROA that combines the Gravitational Search Strategy (GSS) and the Rider Optimisation Algorithm (ROA). Results demonstrate that the proposed task is very robust and feasible.

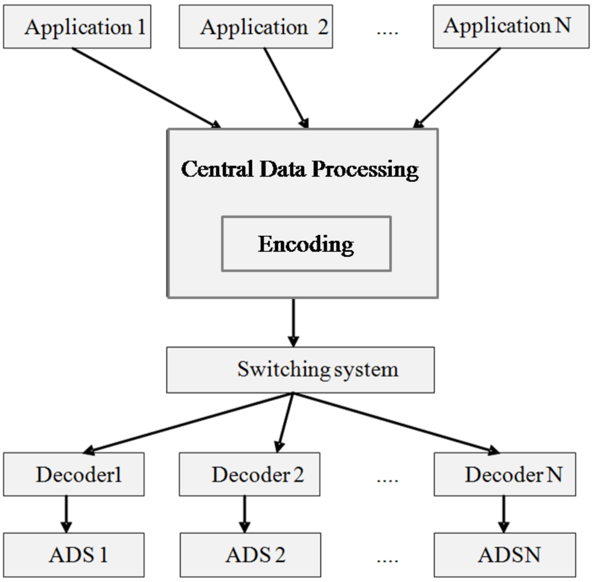
Alazab et al. (2021) proposed a multi-objective cluster head selection strategy for IoT networks in smart cities for employing an upgraded rider optimisation algorithm. To pick the optimal Cluster Head (CH), optimisation variables in IoT devices are considered.

The best features are selected using the Modified Gear and Steering-based Rider Optimisation Method (MGS-ROA), an enhanced meta-heuristic algorithm, which is utilised in Jadhav et al. (2021) to update the weight in the DBN. When employed in analogue circuits for fault diagnostics, ROA has been successful in producing findings that are at the global optimum. The innovative AR-ROA is introduced in this work for faster convergence.

3 Systematic description of proposed SaaS framework

SaaS applications can interrelate with other data and applications within a broad range of platforms and environments. This paper introduces a new SaaS framework. This framework comprises Central Data Processing (CDP), Encoder, Decoder, Switching system and Attack Detection System (ADS). Initially, the features are taken from nine applications to CDP. To provide security for such a large amount of data RSA algorithm (More et al., 2022) is used. Owing to its prime factorisation, it is very difficult to find the key. The switching system then sends the encoded data to the appropriate decoder. It ensures that the data is sent to correct destination without any loss of information. Then, the decoded data is transferred to ADS where the presence of the attacks is determined. In Figure 1, the suggested SaaS framework is depicted.

Figure 1 Modelling of SaaS framework



The data set used comprises of different botnet attacks and it is a collection of nine applications. The nine applications are downloaded from UCI Machine Learning Repository (Detection_of_IoT_botnet_attacks_N_BaIoT, 2021). The given input data consists of nine data sets each with 115 features (F_t) where some of the features are HH_jit_L1_ mean, H_L5_weight M_dir_L5_variance HH_L0.1_weight HH_jit_L4_weight Hp_L0.01_magnitude H_L5_mean MI_dir_L1_weight HH_L0.1_mean HH_jit_L4_mean Hp_L0.01_radius. Because of the vast quantity of features F_t in the input data, the optimal selection of features F_t^* is necessary for classification without losing any information. Hence, to select the optimal features F_t^* from F_t, this work aims to introduce a novel AR-ROA. In the ADS, the proposed AR-ROA algorithm selects the optimal features. The proposed algorithm is stated in detail in the next section. Once optimal features are selected, these features are given for classification, where the DBN predicts the presence of attacks.

4 Improved rider optimisation algorithm for optimal feature selection: AR-ROA algorithm

4.1 Conventional ROA algorithm

ROA (Binu and Kariyappa, 2019) is the Rider Optimisation Algorithm, which is developed by the bunch of riders who race towards the target. In this, each rider's group is dispersed with the equivalent count of riders, and the groups are bypass rider, attacker, follower, and overtaker, respectively. Every group practises diverse strategy for reaching their destination and is explained as follows:

1) The bypasser intends to ride towards destination by crossing the prime rider.

2) All around the axis, the follower follows the prime rider.

3) The overtakers chase along their path to arrive at their target, to the closer location of prime rider.

4) To quickly approach the target, the attacker seizes the rider's path.

The riders mostly chase the pre-arranged strategies, though, the core features like accelerator, brake, gear and steering, to reach the destination. These parameters are synchronised in correspondence to the alteration in the rider's location at each time along with the target. They are inversely correlated with the separation between the rider's position and the goal. The riders' overall riding time is used to evaluate this ongoing process. The rider who arrives at the finish line first is soon declared the winner. The following are the steps of the ROA algorithm

a) *Initialisation*: The initiation is made using the four groups of riders and is expressed as $\hat{L}$ with random initiated position. Equation (1) portrays the group initiation. Here, the number of riders is given as *Ct* and equivalent to $\hat{L}$, the co-ordinates dimension is estimated as *D*, the *i*-th location of the rider in time *t* instant is expressed as $S^t\left(\tilde{i},\tilde{j}\right)$. The estimation on riders count within every group is initiated using equation (2). In this stage, the letters *B*, *F*, *O* and *A* stand for the number of bypass riders, followers, over-taker and attackers, respectively. As a result, equation (3) is used to illustrate the relationship between these parameters.

$$S^t = \left\{S^t\left(\tilde{i},\tilde{j}\right)\right\};1 \le \tilde{i} \le Ct;1 \le \tilde{j} \le D \tag{1}$$

$$Ct = B + F + O + A \tag{2}$$

$$B = F = O = A = \frac{Ct}{4} \tag{3}$$

All the four riders pose the range between the intervals, [S_1, $S_{Ct/4}$], [$S_{Ct/4+1}$, $S_{Ct/2}$], [$S_{Ct/2+1}$, $S_{3Ct/4}$] and [S_{3Ct}, S_{Ct}] correspondingly. Consequently, the parameter initiation of steering, gear, accelerator and brake is performed as well. Equation (4) explains the steering angle at time *t*. Here, *i*-th rider's steering angle is denoted as $A^t_{\tilde{i}\tilde{j}}$ equation (5) describes the steering angle at first position with starting time.

$$A^t = A^t_{\tilde{i}\tilde{j}};1 \le \tilde{i} \le Ct;1 \le \tilde{j} \le D \tag{4}$$

$$A^t_{\tilde{i}\tilde{j}} = \begin{cases} \theta_i; & \text{if } \tilde{j} = 1 \\ A^t_{\tilde{i}\tilde{j}} + \phi; & \text{if } \tilde{j} \ne 1 \text{ and } A^t_{\tilde{i}\tilde{j}} + \phi \le 360 \\ A^t_{\tilde{i}\tilde{j}} + \phi - 360; & \text{otherwise} \end{cases} \tag{5}$$

b) *Update procedure of the bypass-rider*: Since this rider bypasses the primary route, without chasing the prime riders will set update position arbitrarily and is symbolised using equation (6). Here, θ and β specifies the random values that range from 0 to 1, η and ζ indicates the arbitrary number in the range [1, RN].

$$S^B_{t+1}\left(\tilde{i},\tilde{j}\right) = \theta\left[S_t\left(\eta,\tilde{j}\right) + \beta\left(\tilde{j}\right) + S_t\left(\zeta,\tilde{j}\right) * \left[1 - \beta\left(\tilde{j}\right)\right]\right] \tag{6}$$

c) *Update procedure of the follower*: By updating the position of the follower who is chasing the primary rider in this scenario, the riders successfully reach their target. The position update of the follower is created using the coordinate selection in *E* and is directed at the chosen values in equation (7).

$$S_{t+1}^{F}\left(\tilde{i},\tilde{k}\right)=S^{E}\left(E,\tilde{k}\right)+\left[cos\left(Tg_{\tilde{i}\tilde{k}}^{t}\right)*\left[S^{E}\left(E,\tilde{k}\right)*dt_{\tilde{i}}^{t}\right]\right] \quad (7)$$

d) *Update procedure of the overtaker*: This process hangs on the factors like success rate, direction indicator and denoted as $G\hat{k}_t(x)$ in time *t* and is given in equation (8). As a result, in order to achieve a higher success rate, the computation of each rider's direction indicator is developed based on the success rate, or taken into account with success rate ratio.

$$S_{t+1}^{o}\left(\tilde{i},\tilde{k}\right)=S_{t}\left(\tilde{i},\widetilde{k}\right)+\left[G_{t}^{\tilde{i}}*S^{E}\left(E,\tilde{k}\right)\right] \quad (8)$$

e) *Procedure to update the attacker*: It engages the position of the front rider and is deployed similarly to the follower. The attacker's updated position is defined according to equation (9).

$$S_{t+1}^{A}\left(\tilde{i},\tilde{j}\right)=S^{E}\left(E,\tilde{k}\right)+\left[cos\left(A_{\tilde{i}\tilde{j}}^{t}\right)*\left[S^{E}\left(E,\tilde{k}\right)*dt_{\tilde{i}}^{t}\right]\right] \quad (9)$$

f) *Determining the success rate*: The success rate of all the riders can be evaluated after the completion of the position update. The new rider's position is substituted by the foremost rider, which is attained to be maximum.

g) *Update parameters of the rider*: The parameters of a rider require an update, to identify the effective optimal solution. In this, the update is handled by introducing a new parameter named activity counter.

(i) *Activity counter*: If the rider's success rate goes beyond $t+1$, the activity counter attains 1 or else 0, stated as per the equation (10).

$$AY^{t+1}\left(\tilde{i}\right)=\begin{cases}1 & \text{if } \hat{w}_{t+1}\left(\tilde{i}\right)>\hat{w}_{t}\left(\tilde{i}\right)\\ 0 & \text{otherwise}\end{cases} \quad (10)$$

(ii) *Steering angle*: This modification is built using activity counter as per equation (11).

$$TS_{\tilde{i},\widetilde{j}}^{t+1}=\begin{cases}TS_{\tilde{i}+1,j}^{t} & \text{if } AY^{t+1}\left(\tilde{i}\right)=1\\ TS_{\tilde{i}-1,j}^{t} & \text{if } AY^{t+1}\left(\tilde{i}\right)=0\end{cases} \quad (11)$$

(iii) *Gear*: The gear update for $t+1$ depends on activity counter and gear's peak value is engaged by the rider using equation (12).

$$Ge_{\tilde{i}}^{t+1}=\begin{cases}Ge_{\tilde{i}}^{t}+1, & \text{if } AY^{t+1}\left(\tilde{i}\right)=1 \text{ and } Ge_{\tilde{i}}^{t+1}\neq\left|Ge\right|\\ Ge_{\tilde{i}}^{t}-1, & \text{if } AY^{t+1}\left(\tilde{i}\right)=1 \text{ and } Ge_{\tilde{i}}^{t+1}\neq 0\\ Ge_{\tilde{i}}^{t+1} & \text{otherwise}\end{cases} \quad (12)$$

(iv) *Accelerator*: The accelerator update is made with gear and is given in equation (13), in which explains the count of gear.

$$At_{\tilde{i}}^{t+1}=\frac{Ge_{\tilde{i}}^{t+1}}{\left|Ge\right|} \quad (13)$$

(v) *Break*: The break update is close as that of accelerator updates, even so with the subtraction from unity, and is stated as per the equation (14).

$$Bk_{\tilde{i}}^{t+1}=\left[1-\frac{Ge_{\tilde{i}}^{t+1}}{\left|Ge\right|}\right] \quad (14)$$

(vi) *Riding off time*: The iteration of these steps, as mentioned earlier, is continued up to the TI_{OFF} time reaches. At the end, the person who arrives first to the destination is announced as the winner.

4.2 Proposed AR-ROA algorithm

ROA is the new well-known optimisation concept introduced according to the group of riders who compete towards their destination location. This algorithm is considered the fascinating one because of its superior prediction rate over several applications. Still, the algorithm needs some enhancement to be implanted for attaining the effective outcomes like fast convergence. Hence, this research introduced a novel algorithm, AR-ROA, which is the enhanced model of ROA. In this proposed algorithm, the update is exploited within the Accelerator update, in which a new parameter is defined in equation (13). By using this parameter, the new accelerator update is formulated and is stated in equation (12). Equation (13), refers to the current fitness and denotes the total fitness. This accelerator update speeds up the execution and reduces the cost expenditure. Algorithm 1 depicts the proposed AR-ROA Algorithm.

Algorithm 1: Pseudo-code for proposed AR-ROA algorithm

```
Start
Input: Random position of riders
Output: Prime rider, $S^E$
Population initialisation
Initialisation of the rider's constraints
Find the success rate While $t < T_{OFF}$
For $\tilde{i}=1..Ct$
update bypass-rider location using equation (3)
update follower location using equation (4)
update over-taker location using equation (5)
update attacker location using equation (6)
Rank given to the riders based on the success rate
Rider with maximum success rate is selected as prime rider
TS, Ge, Bk constraints are updated
Update the accelerator as per the new improved equation in equation (15) which are equations (17), (18), (19) and (20)
Return $S^E$
$t = t+1$
End For loop
End While loop
End
```

Accelerator: We consider *auxrate*, as the new accelerator update which is formulated and is given in equation (12). In equation (13), f_c refers to the current fitness and f_t denotes the total fitness.

$$At_{\tilde{i}}^{t+1} = \frac{Ge_{\tilde{i}}^{t+1}}{|Ge|} + \left[\frac{1}{auxrate(\tilde{i})}\right] \tag{15}$$

$$auxrate = \frac{f_c}{\min(f_t)} \tag{16}$$

The accelerator update is calculated. Therefore, the updated locations of bypass-rider, follower, over-taker and attacker are given below:

$$\begin{aligned} &S_{t+1}^{B}(\tilde{i},\tilde{j}) \\ &= \theta\left[S_t(\eta,\tilde{j}) + \beta(\tilde{j}) + S_t(\zeta,\tilde{j}) * \left[1-\beta(\tilde{j})\right]\right] + At_{\tilde{B}}^{t+1} \end{aligned} \tag{17}$$

$$\begin{aligned} &S_{t+1}^{F}(\tilde{i},\tilde{k}) \\ &= S^{E}(E,\tilde{k}) + \left[cos\left(Tg_{\tilde{i},\widetilde{k}}^{t}\right) * \left[S^{E}(E,\tilde{k})\right] * \left[dt_{\tilde{i}}^{t}\right]\right] + At_{\tilde{F}}^{t+1} \end{aligned} \tag{18}$$

$$S_{t+1}^{0}(\tilde{i},\widetilde{k}) = S_t(\tilde{i},\widetilde{k}) + \left[G_t^{\tilde{i}} * S^{E}(E,\tilde{k})\right] + At_{\tilde{i}}^{t+1} \tag{19}$$

$$\begin{aligned} &S_{t+1}^{A}(\tilde{i},\tilde{j}) \\ &= S^{E}(E,\tilde{k}) + \left[cos\left(A_{\tilde{i},\widetilde{k}}^{t}\right) * \left[S^{E}(E,\tilde{k})\right] * \left[dt_{\tilde{i}}^{t}\right]\right] + At_{\tilde{F}}^{t+1} \end{aligned} \tag{20}$$

After updating the rider's positions, the rider who got the maximum rank will be selected. Thus, in each iteration, the riders who got the maximum rank will be considered as optimal features. Once the feature selection is completed, they are given to DBN for classification, in which it detects whether there is an attack.

5 Proposed DBN-based attack detection system

DBN (Tang et al., 2016; Mannepalli et al., 2017) is considered as a gifted approach that was proposed in 1986 by Smolensky (Qiao et al., 2022). The DBN is deployed within the Attack Detection System, which in turns verifies whether there exists any attack or not. In the testing phase, the attack detection system probably tells the presence of attack via the DBN model. This Attack Detection process is given in Figure 2 and comprises two stages namely, Optimal Feature Selection and Classification.

The data set is trained with DBN initially. In the testing phase, the attack detection system probably tells the presence of attack via the DBN model. Because of the vast quantity of features in the input data, the optimal selection of features is necessary for classification without losing any information. Hence, to select the optimum features, AR-ROA is used. As we find the optimal features, they are given to DBN which predicts the presence of attacks. Figure 3 depicts the DBN model architecture.

Figure 2 Proposed DBN-based attack detection system

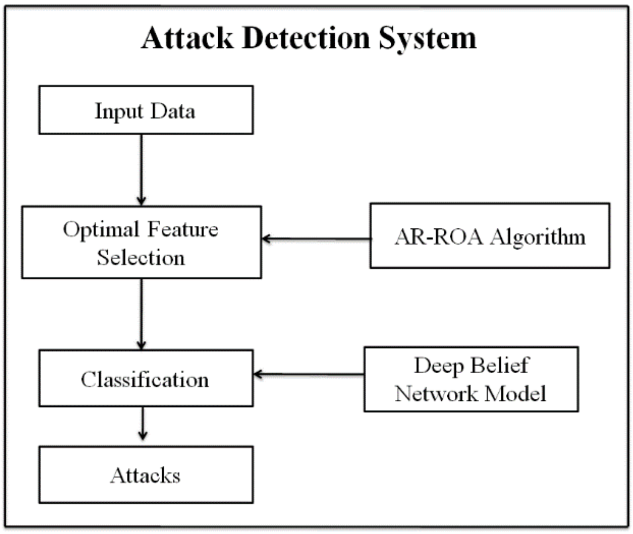
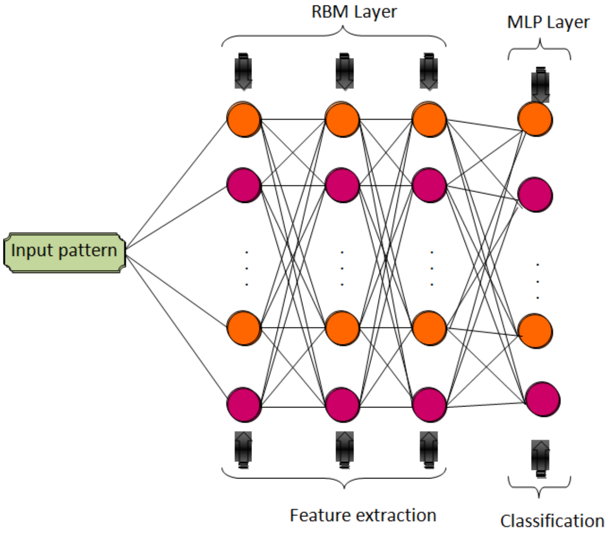


Figure 3 DBN model architecture (see online version for colours)



The DBN model is fundamentally made up of a number of layers. The input layer of each layer contains visible neurons, whereas the output layer is made up of hidden neurons. Every hidden neuron thus insists on a connection to the input neurons. The hidden and visible neurons, on the other hand, are not connected. The link between the hidden and visible neurons is specifically asymmetric. This stochastic neuron method produces precise results for the input. Because the Boltzmann network's output is probabilistic in nature, the output is calculated as per equation (21) and the probability in sigmoid function is represented using equation (22), where *Tem* signifies the pseudo temperature. Equation (23) depicts the stochastic model.

$$pl_r(\zeta) = \frac{1}{1+e^{\frac{-\zeta}{Tem}}} \tag{21}$$

$$v = \begin{cases} 1 \text{ when} 1 - pl_r(\zeta) \\ 0 \quad \text{when} \qquad pl_r(\zeta) \end{cases} \tag{22}$$

$$pl_r\left(\zeta\right) = \frac{1}{1+e^{\frac{-\zeta}{Tem}}} = \begin{cases} 0 & \text{for } \zeta < 0 \\ \frac{1}{2} & \text{for } \zeta = 0 \\ 1 & \text{for } \zeta > 0 \end{cases} \tag{23}$$

A multi-layer perceptron is used for classification in Figure 3 while a set of RBM layers are used for feature selection. This is an example of the DBN approach in action. The arithmetical representation in enlightening the Boltzmann machine's energy for the neuron state composition q is related using equation (24), where $we_{x,y}$ exemplifies the weights and, θ_x defines the biases.

$$\Delta E\left(q_x\right) = \sum_y q_x we_{x,y} + \theta_x \tag{24}$$

The energy descriptions of the visible (u) and hidden (v) neurons are portrayed using equations (25) and (26). In which, u_x explains the visible unit's binary state x, v_y denotes the hidden neuron's binary state y and g_x, h_x denote the biases that apply within the network.

$$\Delta E\left(q_x, \bar{v}\right) = \sum_y we_{xy} v_y + g_x \tag{25}$$

$$\Delta E\left(\bar{u}, v_x\right) = \sum_y we_{xy} u_x + h_y \tag{26}$$

In reality, the assigned probabilities can be maximised using the RBM training, and the weight assignment is found as per the equation (27). For each possible pair of vectors, equation (28) gives the probability allotted with the RBM approach, in which Z explains the partition function specified in equation (29).

$$we^{(0)} = \max_{we} \prod_{\bar{u}e\bar{U}} pl\left(\bar{u}\right) \tag{27}$$

$$pl\left(\bar{u}, \bar{v}\right) = \frac{1}{z} e^{-E\left(\bar{u}, \bar{v}\right)} \tag{28}$$

$$Z = \sum_{\bar{u}, \bar{v}} e^{-E\left(\bar{u}, \bar{v}\right)} \tag{29}$$

As the sampling under the distribution attainment is a critical task, usage of Contrastive Divergence (CD) learning is implemented in this paper, which is explained below:

1) Decide visible neurons and the training samples.
2) Determine the probability of hidden neurons pl_v, by identifying the product of wt and u using $pl_v = \sigma(u, wt)$, which is as per the equation (30).

$$pl\left(\bar{v}_y \mid \bar{u}\right) = \sigma\left(h_y + \sum_x v_x wt_{x,y}\right) \tag{30}$$

3) Check on the hidden states v from these probabilities pl_v.
4) Determine the vector's external product u and pl_v; call it a positive gradient $\phi^+ = u.pl_v^{Tem}$.
5) Check on the visible state restoration u from v using equation (31).

$$pl\left(u_y \rightarrow\mid \bar{v}\right) = \sigma\left(g_x + \sum_x u_y wt_{x,y}\right) \tag{31}$$

6) Determine the final product of u and v. The negative gradient = $\phi = u.pl_u^{Tem}$.
7) Calculate the new weight as per equation (32), in which h indicates the learning rate. The weight update with new values is performed using equation (33).

$$\Delta we = \eta\left(\phi^+ - \phi^-\right) \tag{32}$$

$$we'_{x,y} = \Delta we_{x,y} + we_{x,y} \tag{33}$$

The learning process using the Multi-Layer Perceptron algorithm is begun by assuming the training patterns $\left(I^0, J^0\right)$, in which 0 portrays the training pattern counts, $1 \leq O \leq 0$, I^0, and J^0 symbolises the input and desired output vector in a respective manner. Equation (34) computes the error of each neuron in y within the output layer. As a result, equation (35) gives the pattern's squared error 0 followed using mean squared error within equation (36).

$$Er_y^0 = I^0 - J^0 \tag{34}$$

$$Er_0^{mean} = \frac{1}{n_k}\sum_{y=1}^{n_k}\left(Er_y^0\right)^2 = \frac{1}{n_k}\sum_{y=1}^{n_k}\left(I^0 - J^0\right) \tag{35}$$

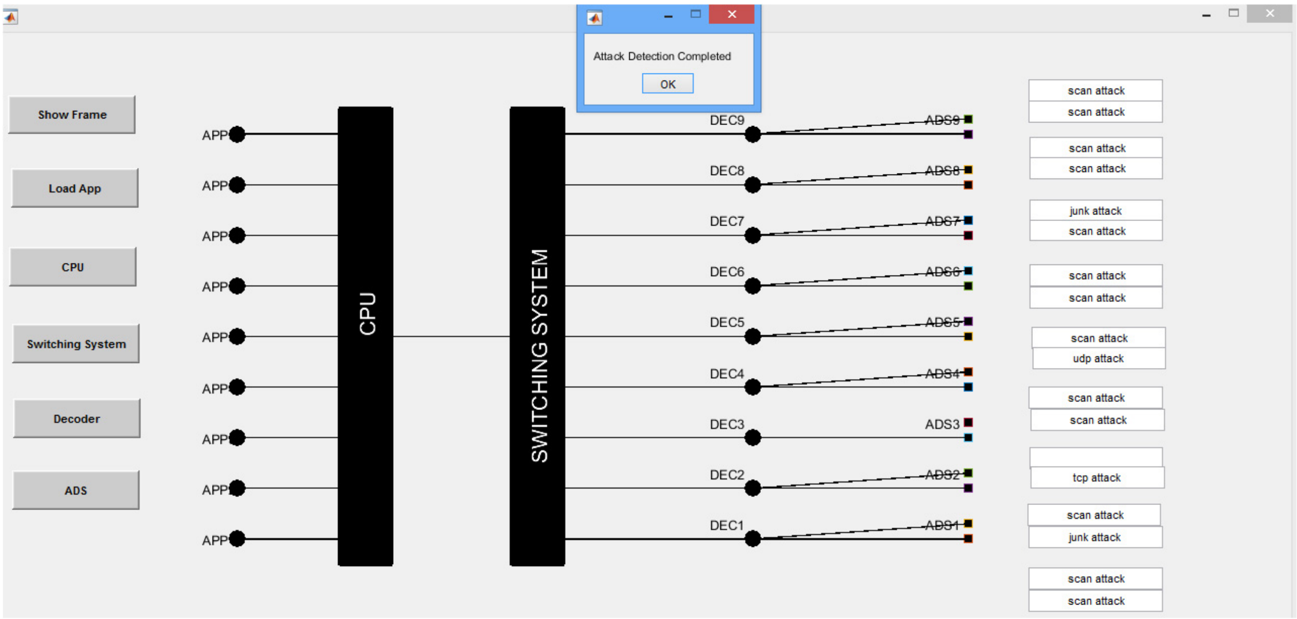
$$Er_{avg} = \frac{1}{0} Er_0^{mean} \tag{36}$$

The DBN training process integrates the RBM, which is the pre-training and MLP defines that these following steps offer normal training.

1) Using the selected weights, biases and various appropriate parameters, DBN gets started.
2) Initially, RBM is initialised with the input data using unsupervised learning.
3) The successive layer input is accomplished by sampling the possibilities explained in the earlier layer's hidden neurons. Subsequently, it uses unsupervised learning.
4) These processes will be repeated until the correct number of layers is obtained. So, until the RBM reaches the MLP layer, the pre-training stage is attained.
5) The refined learning is offered by the MLP layer in supervised format and is continued till the target error rate is obtained.

Finally, after classification, the attacks detected by ADS are shown in Figure 4.

Figure 4 Attack detection (see online version for colours)



Attack Detection Completed

6 Experimental results

a) *Description of the data set*: The data set we have considered is N_baIoT data set for detecting the botnets. It consists of 115 features that contains stream aggregation, time frame, statistics mined from packet stream like weight, magnitude, radius, etc. The characteristics of the data set are multivariate and sequential and the attribute characteristics are real. Missing values are nil.

- *The data gathering process*: We aimed to accurately represent IoT devices deployed in an enterprise context, infected with actual botnets and carrying out actual attacks in our trials. In this work we considered a data set that contains 9 camera collections that are freely accessible. We gathered the traffic information from IoT devices which are linked via Wi-Fi to various access points in order to simulate a typical organisational data flow. Since this data set was used to test the effectiveness of the proposed methodology, we used a MATLAB-based implementation. The 9 applications are namely (i) Danmini_ Doorbel, (ii) Ecobe _Thermostat, (iii) Enio doorbell, (iv) Phillips_B120N1_Baby _Monitor, (v) Provision_ PT_73E_Security, (vi) Provision_PT_ 83_Security, (vii) Samsung_ SH_ 1011_N, (viii) Simple Home _XCS7 _1002_WT and (ix) SimpleHome_XS7 _1003_WT.
- *Botnets deployed*: We concentrated on BASHLITE and Mirai, are the most prevalent IoT botnet families (Mallek et al., 2022).
- *BASHLITE*: BASHLITE is one of the most well-known IoT botnets, and IoT malware often makes use of its code and behaviours. When a new bot connected to it and came under its control, this server was able to instruct the infected device to start an attack.
- *Mirai*: Using its publicly available source code, it is the second botnet present in the network. When a device became infected, it immediately began checking the network for new victims.
- *Attacks executed*: The list of attacks that were carried out and tested is as follows:
  1. *BASHLITE attacks* include the following: (i) Scan: searching the network for weak points; (ii) Junk: sending spam; (iii) Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) flooding and (iv) COMBO: simultaneously sending spam and establishing a connection with given IP address.
  2. *Mirai attacks*: (i) Acknowledgement flooding, (ii) Syn flooding, (iii) UDP flooding (iv) UDP plain flooding. (v) Scan: Automatically scan for weak points in the system.

b) *Execution setup*: The proposed work is implemented in MATLAB2019. Applications that were used in this work have been downloaded from UCI Machine Learning repository (Detection_of_IoT_botnet_ attacks_N_BaIoT, 2021). The performance of the proposed model is analysed and executed over the other Machine Learning models like Genetic Algorithm (GA) (Losantos et al., 2022), Particle Swarm Optimisation (PSO) (Zhang et al., 2022c), Firefly Algorithm (FF) (Shaban et al., 2022), Whale Optimisation Algorithm (WOA) (Zhou et al., 2022) and Rider Optimisation Algorithm (ROA) (Binu and Kariyappa, 2019) in terms of performance metrics that are represented in Table 2, where True Positive (TP), False Negative (FN), False Positive (FP) and True Negative (TN).

Table 2 Performance measures

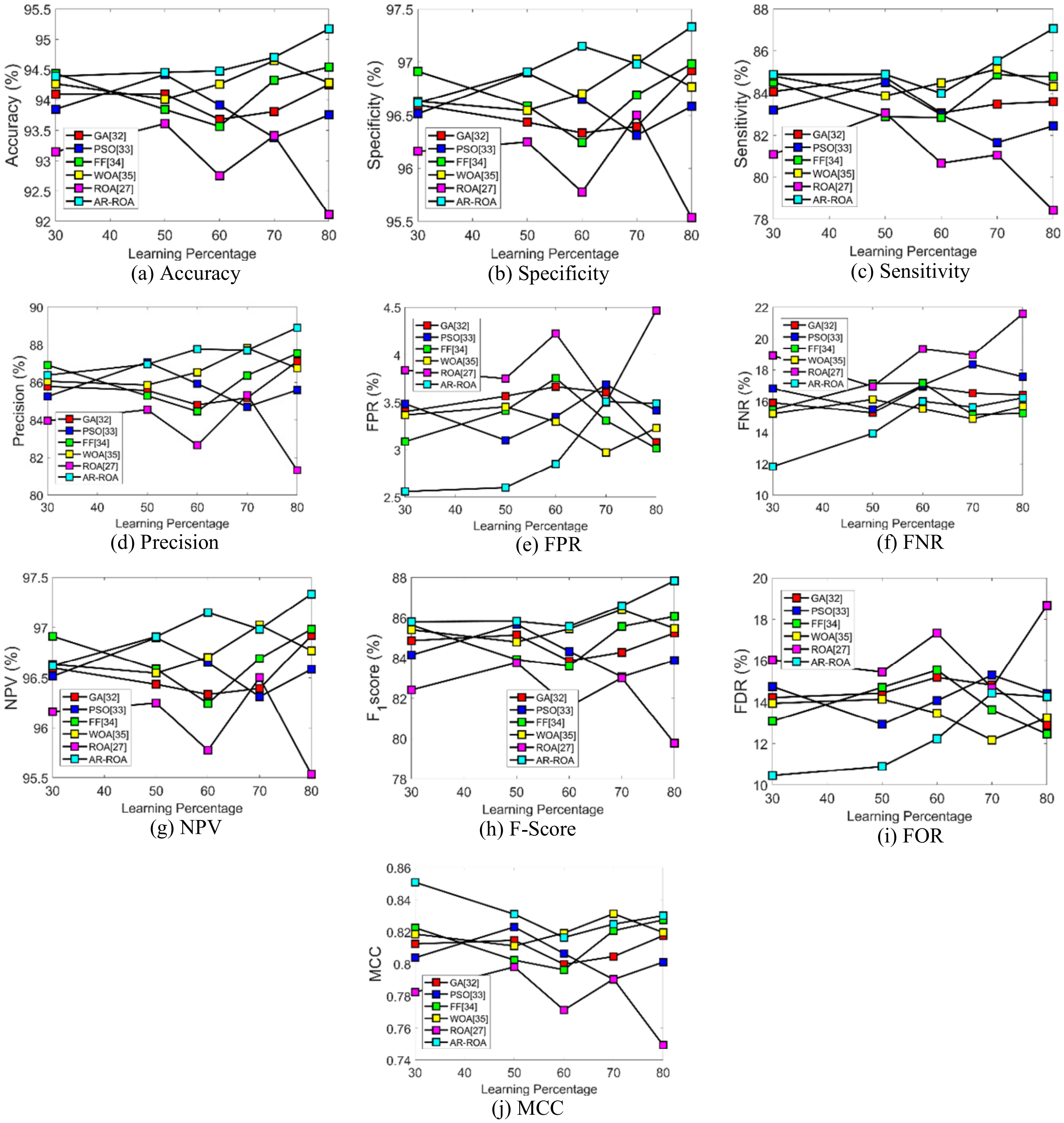
Performance metrics	*Explanation*	*Evaluation*
Accuracy	Proportion of two accurate forecasts	(TP+TN)/(TP+TN+FP+FN)
Specificity	The precise positive cases that have been determined to be accurate.	TN/(TN+FP)
Sensitivity	Normal data were found compared to the data set's overall data availability.	TP/(TP+FN)
Precision	The ratio of normal data to all abnormal data is normal.	TP/(TP+FP)
FPR (False-Positive Rate)	Incorrectly declining the null hypothesis for a specific test.	FP/(FP+TN)
FNR (False-Negative Rate)	Fraction of positives that get negative results for the test	FN/(FN + TP)
NPV (Negative Predictive Value)	The ratio of predicted negatives which are real negatives	TN/(TN+FP)
F1-score	Mean of precision and sensitivity.	(2 * TP) / (2 * TP + FP + FN)
FDR (False-Detection Rate)	Predictable ratio of type I faults	FP/TP+FP
MCC (Matthews Correlation Coefficient)	Correlation coefficient between the observed as well as identified binary classifications.	((TP * TN) - (FP * FN)) / sqrt ((TP + FP) * (TP + FN) * (TN + FP) * (TN + FN))

c) *Performance Analysis of AR-ROA over other models under attack detection*: The performance analysis is shown in Table 3. This analysis is made with varying learning percentages, and the results are given in Figure 5, in which the proposed model has verified its betterments in terms of detecting or predicting the attacks over other conventional models. In Genetic Algorithm, the optimum solution varies each time, and the convergence is always dependent upon the initial solution. Particle Swarm Optimisation (PSO), Whale Optimisation Algorithm (WOA) and Firefly Algorithm (FF), have slow convergence and when exploring the search space. To speed up the process in ROA, the accelerator update is calculated and implemented as AR-ROA. The proposed algorithm gives the best results even though the data set is high dimensional.

Table 3 Performance analysis of AR-ROA over other models under attack detection

Performance metrics	*Percentage of learning*	*Algorithms*					
		GA	*PSO*	*FF*	*WOA*	*ROA*	*AR-ROA*
Accuracy	60	94.2	93.6	94.4	94.3	92	95.3
Specificity	60	96.7	96.5	96.8	96.6	96.68	97.3
Sensitivity	80	83	82	84.5	84	78	87.6
Precision	80	86.3	85.3	86.5	86.2	81	92
FPR	30	3.4	3.5	3.1	3.3	3.8	2.5
FNR	30	16	16.9	15.9	15.8	19	12
NPV	80	96.8	96.55	96.9	96.6	95.5	97.4
F1-score	80	85	83.2	85.9	85.2	79.9	88
FDR	30	14.2	14.9	13.1	14	16	10.5
MCC	50	0.818	0.812	0.822	0.82	0.78	0.845

Figure 5 Performance analysis of AR-ROA model over other conventional models under attack detection (see online version for colours)



(a) Accuracy

(b) Specificity

(c) Sensitivity

(d) Precision

(e) FPR

(f) FNR

(g) NPV

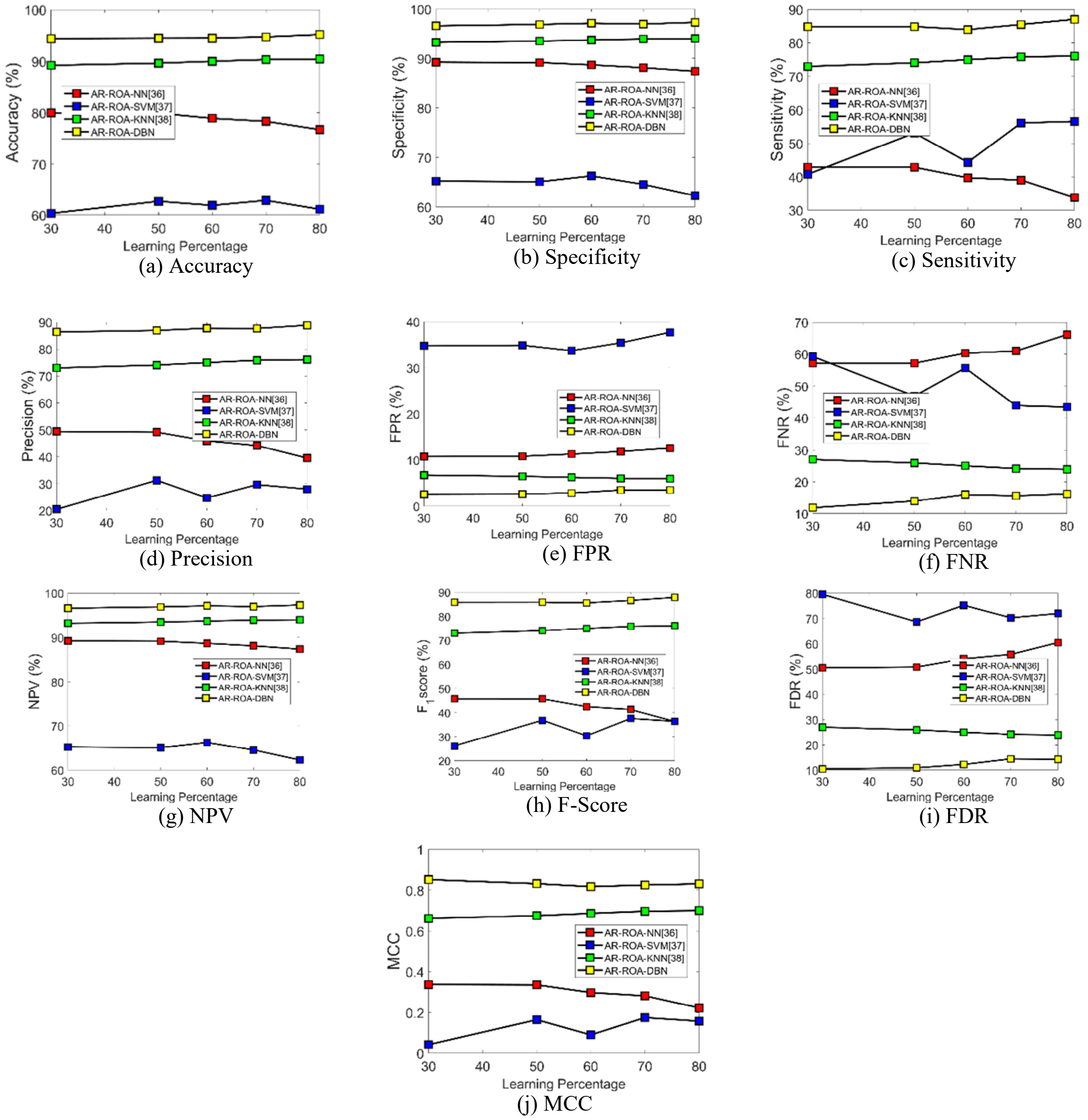
(h) F-Score

(i) FOR

(j) MCC

d) *Analysis of DBN classifier*: This segment illustrates the routine of the AR-ROA-based DBN classifier with various classifiers like Neural Network (NN) (Zeng and Long, 2022), Support Vector Machine (SVM) (Xia et al., 2022) and *K*-means Nearest Neighbour (KNN) (Mallek et al., 2022), and the results are given in Figure 6. The DBN is a complete bipartite graph with nodes at each layer. The classification is relatively simple and less expensive when compared to other classifiers because the nodes in the same layer are not coupled. CD learning is also employed for distribution accomplishment. This kind of network produces prompt and accurate results. It is evident from these data that the DBN classifier outperformed the other classifiers in terms of results. The analysis is performed by changing the learning-percentage to 30%, 40%, 50%, 60%, 70% and 80%, respectively. Hence, the performance altogether has revealed better results over other classifier models, represented in Table 4.

Figure 6 The performance analysis of AR-ROA-DBN classifier model over the other classifiers (see online version for colours)



(a) Accuracy

(b) Specificity

(c) Sensitivity

(d) Precision

(e) FPR

(f) FNR

(g) NPV

(h) F-Score

(i) FDR

(j) MCC

Table 4 Performance analysis of AR-ROA-DBN classifier model over the other classifiers

Performance measures	*Learning percentage*	*Algorithms*			
		AR-ROA NN	*AR-ROA SVM*	*AR-ROA KNN*	*AR-ROA DBN*
Accuracy	80	76	27	73	95.3
Specificity	80	88	62	93	98
Sensitivity	80	32	54	73	86
Precision	80	40	27	73	90
FPR	30	12	38	8	2
FNR	30	50	80	29	10
NPV	80	88	62	90.4	97
F1-score	80	48	28	72	86
FDR	30	50	80	20.8	10
MCC	30	0.38	0.12	0.63	0.82

7 Conclusions and future work

The primary goal of the study was to create a secure SaaS framework by providing a new optimisation technique for reliable and effective identification of the attacks. The new security platform for SaaS framework has two stages: (1) Optimal Feature Selection and (2) Classification. Initially, the best features from the data were chosen because each data set contained additional properties that might frequently increase processing complexity. As a result, a novel AR-ROA algorithm has been proposed for optimal feature selection. Following the selection of these ideal qualities, the classification procedure was carried out utilising the DBN model, which allowed for the detection of attacks. The performance of the suggested work was compared to that of other models, and its effectiveness was validated. Results indicate that, in contrast to other traditional models, the performance of the suggested model is more effective.

Future research will focus on the various values that emerge during the application's dynamic execution. The encryption's fixed values or the volatility of different algorithms can be used to determine whether or not an application runs a clear refinement. This motivates developers to enhance the framework's capability to include the identification of other weaknesses like poor randomisation and weak cryptography. Cross validation will be used in subsequent work to evaluate the models' accuracy. Additionally, we intend to put into practice the attack mitigation system using dynamic data sets.

References

Abdou, A., Van Oorschot, P.C. and Wan, T. (2018) 'Comparative analysis of control plane security of SDN and conventional networks', *Proceedings of the IEEE Communications Surveys and Tutorials*, IEEE, Vol. 20, No. 4, pp.3542–3559.

Alazab, M., Lakshmanna, K., Reddy, T., Pham, Q-V. and Maddikunta, P.K.R. (2021) 'Multi-objective cluster head selection using fitness averaged rider optimization algorithm for IoT networks in smart cities', *Proceedings of the Sustainable Energy Technologies and Assessments*, Vol. 43. Doi: 10.1016/j.seta.2020.100973.

Aziz, I.T., Abdulqadder, I.H. and Jawad, T.A. (2022) 'Distributed denial of service attacks on cloud computing environment', *Proceedings of the Cihan University-Erbil Scientific Journal*, Vol. 6, No. 1, pp.47–52.

Belguith, S., Kaaniche, N. and Hammoudeh, M. (2022) 'Analysis of attribute-based cryptographic techniques and their application to protect cloud services', *Proceedings of the Transactions on Emerging Telecommunications Technologies*, Vol. 33, No. 3, pp.1–19.

Binu, D. and Kariyappa, B.S. (2019) 'RideNN: a new rider optimization algorithm-based neural network for fault diagnosis in analog circuits', *Proceedings of the IEEE Transactions on Instrumentation and Measurement*, IEEE, Vol. 68, No. 1, pp.2–26.

Binu, D. and Kariyappa, B.S. (2019) 'RideNN: a new rider optimization algorithm-based neural network for fault diagnosis in analog circuits', *Proceedings of the* IEEE Transactions on Instrumentation and Measurement, Vol. 68, No. 1, pp.2–26.

Detection_of_IoT_botnet_attacks_N_BaIoT (2021) *Detection_of_IoT_botnet_attacks_N_BaIoT*, UC Irvine Machine Learning Repository. https://archive.ics.uci.edu/ml/datasets/detection_of_IoT_botnet_attacks_N_BaIoT# (accessed on July 2021).

El-Dine Atta, M.E., Ibrahim, D.K., Gilany, M. and Zobaa, A.F. (2022) 'Adaptive scheme for detecting induction motor incipient broken bar faults at various load and inertia conditions', *Proceedings of the Sensors*, Vol. 22, No. 1. Doi: 10.3390/s22010365.

Elsayed, M. and Zulkernine, M. (2019) 'Offering security diagnosis as a service for cloud SaaS applications', *Proceedings of the Journal of Information Security and Applications*, Vol. 44, No. 1, pp.32-48.

Fawcett, L., Scott-Hayward, S., Broadbent, M., Wright, A. and Race, N. (2018) 'Tennison: a distributed SDN framework for scalable network security', *Proceedings of the IEEE Journal on Selected Areas in Communications*, Vol. 36, No. 12, pp.2805–2818.

Ghosh, N., Chatterjee, D., Ghosh, S.K. and Das, S.K. (2016) 'Securing loosely-coupled collaboration in cloud environment through dynamic detection and removal of access conflicts', *Proceedings of the IEEE Transactions on Cloud Computing*, IEEE, Vol. 4, No. 3, pp.349–362.

Ghuge, S.S., Kumar, N. and Savitha, S. and Suraj, V. (2020) 'Multilayer technique to secure data transfer in private cloud for SaaS applications', *Proceedings of the Second International Conference on Innovative Mechanisms for Industry Applications (ICIMIA'20)*, pp.646–651.

Han, Z., Li, X., Huang, K. and Feng, Z. (2018) 'A software defined network-based security assessment framework for CloudIoT', *Proceedings of the IEEE Internet of Things Journal*, IEEE, Vol. 5, No. 3, pp.1424–1434.

He, J., Zhang, Y., Lu, J., Wu, M. and Huang, F. (2018) 'Block-stream as a service: a more secure, nimble, and dynamically balanced cloud service model for ambient computing', *Proceedings of the IEEE Network*, IEEE, Vol. 32, No. 1, pp.126–132.

Hyun, S. et al. (2018) 'Interface to network security functions for cloud-based security services', *Proceedings of the IEEE Communications Magazine*, IEEE, Vol. 56, No. 1, pp.171–178.

Jadhav, A.S., Patil, P.B. and Biradar, S. (2021) 'Optimal feature selection-based diabetic retinopathy detection using improved rider optimization algorithm enabled with deep learning', *Proceedings of the Evolutionary Intelligence*, Vol. 14, No. 4, pp.1431–1448.

Jangjou, M. and Sohrabi, M.K. (2022) 'A comprehensive survey on security challenges in different network layers in cloud computing', *Proceedings of the Archives of Computational Methods in Engineering*, pp.1–22.

Kalkan, K., Altay, L., Gür, G. and Alagöz, F. (2018) 'JESS: joint entropy-based DDoS defence scheme in SDN', *Proceedings of the IEEE Journal on Selected Areas in Communications*, IEEE, Vol. 36, No. 10, pp.2358–2372.

Lee, S.P., Kim, K. and Park, S. (2022) 'Investigating the market success of software-as-a-service providers: the multivariate latent growth curve model approach', *Proceedings of the Information Systems Frontiers*, pp.1–20.

Li, Q., Chen, Y., Lee, P.P.C., Xu, M. and Ren, K. (2018) 'Security policy violations in SDN data plane', *Proceedings of the IEEE/ACM Transactions on Networking*, IEEE, Vol. 26, No. 4, pp.1715–1727.

Liu, Y., Lu, Y., Qiao, W. and Chen, X. (2018) 'A dynamic composition mechanism of security service chaining oriented to SDN/NFV-enabled networks', *Proceedings of the IEEE Access*, IEEE, Vol. 6, pp.53918–53929.

Liu, Z., Xu, B., Cheng, B., Hu, X. and Darbandi, M. (2022) 'Intrusion detection systems in the cloud computing: a comprehensive and deep literature review', *Proceedings of the Concurrency and Computation: Practice and Experience*, Vol. 34, No. 4. Doi: 10.1002/cpe.6646.

Losantos, R. et al. (2022) 'Parameter characterization of HTPEMFC using numerical simulation and genetic algorithms', *International Journal of Hydrogen Energy*, Vol. 47, No. 4, pp. 4814–4826.

Mallek, A., Klosa, D. and Büskens, C. (2022) 'Enhanced k-nearest neighbour model for multi-steps traffic flow forecast in urban roads', *Proceedings of the IEEE International Smart Cities Conference (ISC2)*, pp.1–5.

Mannepalli, K., Sastry, P.N. and Suman, M. (2017) 'A novel adaptive fractional deep belief networks for speaker emotion recognition', *Proceedings of the Alexandria Engineering Journal*, Elsevier, Vol. 56, No. 4, pp.485–497.

More, S.S., Narain, B. and Jadhav, B.T. (2022) 'Personal identification using fuzzy approach and RSA algorithm', *Proceedings of the ICT Analysis and Applications*, Springer, pp.99–106.

Onyema, E.M., Dalal, S., Romero, C.A.T., Seth, B., Young, P. and Wajid, M.A. (2022) 'Design of intrusion detection system based on cyborg intelligence for security of cloud network traffic of smart cities', *Proceedings of the Journal of Cloud Computing*, Vol. 11, No. 1, pp.1–20.

Ouda, A.J., Yousif, A.N., Hasan, A.S., Ibrahim, H.M. and Shaya, M.A. (2022) 'The impact of cloud computing on network security and the risk for organization behaviours', *Proceedings of the Webology*, Vol. 19, No. 1, pp.195–206.

Parast, F.K., Sindhav, C., Nikam, S., Yekta, H.I., Kent, K.B. and Hakak, S. (2022) 'Cloud computing security: a survey of service-based models', *Proceedings of the Computers and Security*, Vol. 114, No. 1. Doi: 10.1016/j.cose.2021.102580.

Prabhakaran, V. and Kulandasamy, A. (2021) 'Integration of recurrent convolutional neural network and optimal encryption scheme for intrusion detection with secure data storage in the cloud', *Proceedings of the Computational Intelligence*, Vol. 37, No. 1, pp.344–370.

Qiao, C. et al. (2022) 'Deep belief networks with self-adaptive sparsity', *Proceedings of Applied Intelligence*, Vol. 52, No. 1, pp.237–253.

Saidi, A., Nouali, O. and Amira, A. (2022) 'SHARE-ABE: an efficient and secure data sharing framework based on ciphertext-policy attribute-based encryption and Fog computing', *Proceedings of the Cluster Computing*, Vol. 25, No. 1, pp.167–185.

Saleh, E., Sianipar, J., Takouna, I. and Meinel, C. (2015) 'SecPlace: a security-aware placement model for multi-tenant SaaS environments', *Proceedings of the IEEE 11th International Conference on Ubiquitous Intelligence and Computing*, pp.596–602.

Salek, M.S., Khan, S.M., Rahman, M., Deng, H.W., Islam, M., Khan, Z. and Shue, M. (2022) 'A review on cybersecurity of cloud computing for supporting connected vehicle applications', *Proceedings of the IEEE Internet of Things Journal*, Vol. 9, No. 11, pp.8250–8268.

Shaban, W.M., Elbaz, K., Amin, M. and Ashour, A.G. (2022) 'A new systematic firefly algorithm for forecasting the durability of reinforced recycled aggregate concrete', *Proceedings of the Frontiers of Structural and Civil Engineering*, pp.1–18.

Tang, B., Liu, X., Lei, J., Song, M. and Dong, F. (2016) 'Deep chart: combining deep convolutional networks and deep belief networks in chart classification', *Proceedings of the Signal Processing*, Elsevier, Vol. 124, pp.156–161.

Tripathy, D., Gohil, R. and Halabi, T. (2020) 'Detecting SQL injection attacks in cloud SaaS using machine learning', *Proceedings of the 6th International Conference on Big Data Security on Cloud (Bigdata Security) and International Conference on High Performance and Smart Computing, (HPSC) and IEEE International Conference on Intelligent Data and Security (IDS)*, pp.145–150.

Vinoth, S., Vemula, H.L., Haralayya, B., Mamgain, P., Hasan, M.F. and Naved, M. (2022) 'Application of cloud computing in banking and e-commerce and related security threats', *Proceedings of the Materials Today*, Vol. 51, pp.2172–2175.

Wang, G., Yuan, Y. and Guo, W. (2019) 'An improved rider optimization algorithm for solving engineering optimization problems', *Proceedings of the IEEE Access*, Vol. 7, pp.80570–80576.

Wang, M., Liu, J., Mao, J., Cheng, H., Chen, J. and Qi, C. (2017) 'Route guardian: constructing secure routing paths in software-defined networking', *Proceedings of the Tsinghua Science and Technology*, IEEE, Vol. 22, No. 4, pp.400–412.

Wang, R., Gu, C., He, S., Shi, Z. and Meng, W. (2022) 'An interoperable and flat industrial internet of things architecture for low latency data collection in manufacturing systems', *Proceedings of the Journal of Systems Architecture*, Vol. 129. Doi: 10.1016/j.sysarc.2022.102631.

Xia, J. et al. (2022) 'Performance optimization of support vector machine with oppositional grasshopper optimization for acute appendicitis diagnosis', *Proceedings of the Computers in Biology and Medicine*. Doi: 10.1016/j.compbiomed.2021.105206.

Yoon, C. et al. (2017) 'Flow wars: systemizing the attack surface and defences in software-defined networks', *Proceedings of the IEEE/ACM Transactions on Networking*, IEEE, Vol. 25, No. 6, pp. 3514–3530.

Zeng, X. and Long, L. (2022) *Beginning Deep Learning with TensorFlow: Work with Keras, MNIST Data Sets, and Advanced Neural Networks*, Springer, Apress, Berkeley, CA, pp.191–234.

Zhang, L. et al. (2022c) 'Sound classification using evolving ensemble models and particle swarm optimization', *Proceedings of the Applied Soft Computing*, Elsevier, Vol. 116. Doi: 10.1016/j.asoc.2021.108322.

Zhang, L., Li, X., Tang, Y., Xin, J. and Huang, S. (2022a) 'A survey on QoT prediction using machine learning in optical networks', *Proceedings of the Optical Fibre Technology*, Vol. 68. Doi: 10.1016/j.yofte.2021.102804.

Zhang, X., Duan, M., Xu, R., Rao, H. and Deng, J. (2022b) 'EdgeCloudSim based computing resource configuration strategy analysis of cloud-edge system in power distribution internet of things', *Proceedings of the 7th Asia Conference on Power and Electrical Engineering (ACPEE)*, pp.2013–2018.

Zhang, Y., Sheng, H., Wang X. and Hua, J. (2018) 'User security authentication scheme under SaaS platform of enterprises', *Proceedings of the International Conference on Virtual Reality and Intelligent Systems (ICVRIS)*, Vol. 1, No. 1, pp.147–151.

Zhou, H. (2018) 'SDN-RDCD: a real-time and reliable method for detecting compromised SDN devices', *Proceedings of the IEEE/ACM Transactions on Networking*, Vol. 26, No.5, pp.2048–206.

Zhou, J., Zhu, S., Qiu, Y., Armaghani, D.J., Zhou, A. and Yong, W. (2022) 'Predicting tunnel squeezing using support vector machine optimized by whale optimization algorithm', *Proceedings of the Acta Geotechnica*, Springer, Vol. 17, No. 4, pp.1343–1366.