Extended Sparse Transient Search Deep Transfer Learning Based Intrusion Detection System

Gavini Sreelatha [1]

[1] Research Scholar, Lincoln University College, Kaula Lampur, Malaysia. Email id:sreelathaprince13@gmail.com

Dr.A.Vinaya Babu [2]

[2] Professor, Stanley College of Engineering and Technology for Women, Abids, India Email id:avb1222@gmail.com

Dr.Divya Midhunchakkarvarthy[3]

[3] Associate Professor, Lincoln University College, Kaula Lampur, Malaysia, Email id:divya@lincoln.edu.my

Abstract:

Intrusion detection system (IDS) is typically responsible for tracking and identifying fraudulent behaviours in any operating network. This paper propose an extended sparse transient search deep transfer learning-based intrusion detection system (STSDTL-IDS) that overcomes the limitations and classifies complex attacks accurately and finely. A feature selection method, namely, chimp optimization algorithm (COA) is used to eliminate the unwanted features and it is used to detect assaults by identifying relevant aspects in the dataset with high precision. The proposed method is a hybrid method which includes sparse transient auto encoder with deep transfer learning, and extended Transient Search Optimization algorithm to improve the cloud IDS efficiency. The performance of the sparse deep transfer learning algorithm is improved using the extended transient search optimization. The python tool is used and the UNSW-NB15 and CICIDS2017 datasets are used. The experimental result shows that the proposed method outperforms when compared to the existing methods.

Keywords: Intrusion detection, Cloud security, sparse deep transfer learning, adaptive transient search optimization, chimp optimization.

1. Introduction

Cloud computing is a highly versatile and efficient platform that offers pay-per-use access to the computer system infrastructure, data management, and computational power on demand. Cloud computing's distributed nature makes it a convenient target for invaders who are constantly leveraging its flaws with new attacks. Standard attacks such as Distributed Denial of Service (DDoS), Denial of Service (DoS), IP spoofing, and others have been discovered to be prone to cloud computing [1]. Additionally, there is a significant risk of insider attacks, in which approved users may initiate attacks within the tenant network, resulting in the system's complete failure. These types of attacks negatively impact the cloud's security, availability and integrity. Various tools for protecting cloud networks against various threats, such as user authentication, access control mechanisms, and firewalls, have already been established by organizations over the last decade. While these solutions prohibit outsiders from gaining unauthorized access, they are not resistant to insider attacks. As a result, the IDS [2] were created as a second line of protection to prevent data loss due to intruders. Various IDS identification techniques, including Host IDS [3] and Network IDS [4], have been used in the past several years. The IDS, when used in conjunction with access control lists, firewall rules, and data protection methods, could provide cloud security.

An IDS [5] is a proactive intrusion detection method that detects and classifies intrusions, threats, and abuses of security protocols at the host and network level infrastructure in real time. Intrusion detection is divided into two types based on disruptive behaviors: network-based intrusion detection systems (NIDS) and host-based intrusion detection systems (HIDS) [6, 7]. Network behaviors gathered using network infrastructure through mirroring by networking devices such as routers, switches, and network taps, and then analysis is done to find attacks and potential threats hidden within the network traffic. HIDS is an IDS system that detects attacks by using system activities in the context of numerous log files operating on the remote host machine.

Several traditional machine learning approaches, such as artificial neural network (ANN), K-nearest neighbor (KNN), support vector machine (SVM) [8], random forest (RF) [9] decision tree (DT) [10], have already been researched for intrusion detection over the last several years. In recent years, several DL-driven IDS approaches have been created, including deep neural networks (DNN) [11], recurrent neural networks (RNN) [12], self-taught learning, and others. Even though these techniques work well, they are limited in their capacity to adjust to shifts in attack patterns. Adaptability refers to an IDS's ability to respond to any potential change in the environment's attack patterns, as well as its ability to recognize them with high precision.

Motivation: An IDS is a network traffic monitoring system that detects malicious activity and sends out warnings when it is found. Any malicious activity or infringement is usually reported to administrators or gathered centrally using a security information and event management (SIEM) [13] system. Since recognizing and responding to anomalous attacks is possible, a NIDS [14] is important for network security. The main advantage of IDS is that it alerts IT personnel whenever an intrusion or security breaches is suspected. The majority of existing frameworks have security vulnerabilities, making it possible for influential insiders to exploit them. Not every form of intrusion is known. Intrusion detection that occurs quickly can assist in identifying intruders and limiting damage.

Contribution

To create an efficient cloud IDS based on feature selection and classification approaches that will improve cloud protection. The proposed framework uses methods to identify useful features and distinguish attacks.
COA is used to identify important features in the dataset with high accuracy for detecting the attacks.
An extended STSDTL-IDS is implemented to effectively identify traffic as normal and attack data with a high accuracy.
The paper is organized as follows: In section 1, a brief introduction about the topic along with the motivation and contribution is given. Section 2 provides the works related to the proposed algorithm and section 3 gives the proposed methodology. Section 4 provides the result of the work. Finally, the conclusion is given in the fifth section.

2. Relative Works

Sahoo et al. [15] proposed a strategy to reduce irrelevancy in the IDS system that employs the Whale Pearson hybrid feature selection wrapper. The binary Whale Optimization Algorithm (WOA) had been improved with the Whale Pearson hybrid wrapper. The WOA is a form of swarm intelligence algorithm that is based on humpback whale behavior.

Almiani et al. [16] introduced artificially fully automated IDS for Fog protection against the cyber-attacks. The suggested framework employed multi-layered recurrent neural networks that are built to be deployed for Fog computing protection near end-users and for IoT applications. A balanced version of the demanding dataset, NSL-KDD was utilized, to explain the proposed model.

Chen et al. [17] proposed a network intrusion detection method for cloud computing environments. The decision tree's C4.5 algorithm was utilized to develop the intrusion detection model. The intrusion detection model was built using random forest algorithm, and network traffic data of various network levels of each cloud server was collected in real time using tcpdump tool and the data mining technology.

A HIDS is presented in this paper by Besharati et al. [18] for securing virtual machines in the cloud world. To accomplish this, the most significant characteristics of each category were first identified using logistic regression, and then these features were enhanced using the regularization technique. The bagging algorithm was then used to classify attacks using a set of three classifiers: decision tree, linear discriminate analysis, and neural network for each class.

IDS are used to address these security concerns. An adequate training for IDS is needed to detect all intrusions accurately and instantly. The presence of a trivial feature set in training data increases memory space and reduces training time. Ghosh et al. [19] introduced a novel CS-PSO-based IDS to quickly and easily distinguish attacks. NSL-KDD dataset was picked to illustrate the IDS's capabilities.

Jaber and Rehman [20] introduced a novel intrusion detection method that uses a fuzzy c means clustering in conjunction with a support vector machine to enhance detection accuracy in a cloud computing system. The presented scheme was put into action and compared to current systems. Experiments were performed using the NSLKDD dataset.

Problem statement: The cloud storage is distributed because it is too large to be stored on a single storage unit. Users frequently choose to save their personal data and sensitive information in the cloud. As a result, data protection is the most serious issue in cloud security. An IDS is commonly used to detect improper use, and manipulation of structures. Despite the fact that large amounts of data contain various irrelevant or jobless attributes, this may reduce classification accuracy in cloud environments and consume a large amount of computing resources. An IDS should be able to detect and respond to all anomalous patterns and traffic within the system by monitoring, detecting, and responding to illegal actions. IDS, on the other hand, has a complete data processing challenge with its large and imbalanced datasets. It is extremely difficult and time intensive to distinguish between intrusion and typical network traffic activity. To determine the sequence of intrusion on the network connection, an analyst must evaluate all of the data that huge and wide. As a result, it requires a method to identify network intrusion and reflect current network activity. In cloud IDS, an efficient feature selection (FS) method and an extended STSDTL model are used to tackle these issues.

3. Proposed methodology

The current state of cloud protection highlights the importance of intrusion detection and classification of attacks. For that, Extended STSDTL technique is utilized. The dataset collected undergoes feature selection algorithm to select optimal features. This is done using COA to make the classification more exact and precise. The selected features then undergo classification for classifying the attacks. For the purpose of classification, sparse deep transfer learning is used. The output of the classifier is given to the decision module which takes decision and alerts the VM by using alert module. This novel type of method leverages and identifies the intrusion with the detection of different types of benign with new attacks by the utilization of the knowledge from the source domain, thereby leading to the protection of cloud security.

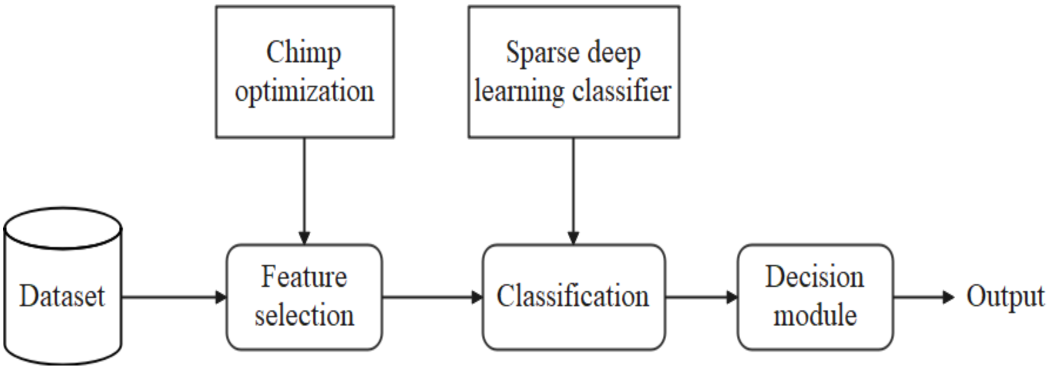


Figure 1: Block diagram of proposed methodology

3.1 *Feature selection using chimp optimization algorithm*

In IDS, the selection of feature is the major significant task. Here, the most valuable features are chosen through eliminating the unwanted and jobless features. By this process, the complexity and computation time can be reduced. Here, chimp optimization algorithm (COA) [21] is introduced to select the most relevant features from the dataset. COA is new meta-heuristic algorithm which imitates the hunting behaviour of chimpanzees. Drivers, barriers, chasers, and attackers are the four types of chimps found in Colorado. Chimps' social hunting behaviour could be separated into 2 phases: exploration and exploitation. In the following section, the mathematical models for these phases and four steps of the search are described.

3.1.1 Driving and chasing the prey

Equations (1) and (2) given below are used to model driving and chasing the prey numerically.

$$d = \left| e.y_{prey}(t) - n.y_{chimp}(t) \right| \tag{1}$$

$$y_{chimp}(t+1) = y_{prey}(t) - b \cdot s \tag{2}$$

Where number of current iteration is indicated by t, coefficient vectors denoted using b, n, e and position vector of chimp and prey represented using y_{chimp} and y_{prey} respectively. b, n, e are calculated using the equations (3), (5) and (4) respectively.

$$b = 2g \cdot (r_1 - b) \tag{3}$$

$$e = 2 \cdot r_2 \tag{4}$$

$$n = chaotic\_value \tag{5}$$

Where g is deduced from 2.5 to 0, non-linearly through iteration process. Random vectors in range [0, 1] is represented using r_1 and r_2. The variable n represents the chaotic vector calculated on the basis of various chaotic maps.

3.1.2 Exploration phase and Exploitation phase

At the very first step, there seems to be no knowledge about prey's optimal location. To address this weakness, the attacker's location is considered to be that of prey. The attacker's location can then be used to modify the driver, barrier, and chaser positions. As a result, four of the better solutions are saved, and perhaps other chimps are obliged to update their locations based on the positions of best chimps. The equations (6) to (8) characterize this approach:

$$d_{attac\ker} = \left|e_1 y_{attac\ker} - n_1 s\right| \ , \ d_{Barrier} = \left|e_2 y_{Barrier} - n_2 y\right| \\ d_{chaser} = \left|e_3 y_{chaser} - n_3 y\right| \ , \ d_{driver} = \left|e_4 y_{driver} - n_4 y\right| \tag{6}$$

$$y_1 = y_{attac\ker} - b_1\left(d_{attac\ker}\right) \ , \ y_2 = y_{Barrier} - b_2 (d_{Barrier}) \\ y_3 = y_{chaser} - b_3\left(d_{chaser}\right) \ , \ y_4 = y_{driver} - b_4\left(d_{driver}\right) \tag{7}$$

$$y(t+1) = \frac{y_1 + y_2 + y_3 + y_4}{4} \tag{8}$$

The location of a search agent in the search space is modified in relation to the locations of other chimps. The chimp's final location is determined at random inside a circle identified by the positions of attacker, barrier, chaser, and drivers. In the exploitation phase, when target stays down, the chimp strikes and ends the hunting.

3.2 *Extended sparse transient search deep transfer learning based classification*

After the selection of optimal features, the extended STSDTL-IDS approach is introduced to classify the different attacks in the cloud structure which includes the combination of sparse autoencoder (SAE), deep transfer learning based on GoogleNet and ETSO algorithm. In this proposed network, ETSO algorithm is used to enhance the performances of cloud IDS.

Autoencoder is an unsupervised deep learning network that reduces data dimensions and extracts features. Weight connections are used to map original data into the concealed layer. For data reconstruction, the hidden layer's activation value is transferred onto the output layer. To fine-tune weight and provide correct data representation, the reconstruction error is reduced. A sparse constraint is applied to restrict the number of hidden nodes by limiting the activation of its nodes. Because certain nodes in the hidden layer are active while others are inactive due to the sparse restriction, the autoencoder is changed to be SAE. y in dataset $Y = [y_1, y_2, \ldots, y_m]$ consisted of m data samples, activation of hidden layer h nodes calculated as

$$h = g\,(W^{(1)} y + c^{(1)}) \tag{9}$$

$W^{(1)}$ is the weight used to link the input and hidden layers, while $b^{(1)}$ is the bias, and the g is an activation function. The connection weights between the hidden layer and the output layer are utilized by the hidden layer to recreate the original data.

$$\tilde{y} = g\,(W^{(2)} h + b^{(2)}) \tag{10}$$

where $\tilde{y}$ denotes the reconstructed data $W^{(2)}$ is the weight between the hidden and output layers, whereas $b^{(2)}$ is the bias.

$$J\,(W, b) = \frac{1}{m} \sum_{i=1}^{m} \left(\frac{1}{2} \left\| y_i - \tilde{y}_i \right\|^2 \right) + \frac{\lambda}{2} \sum_{l=1}^{m_1 - 1} \sum_{i=1}^{s_1} \sum_{j=1}^{s_1 + 1} \left(W_{ji}^{(l)} \right)^2 \tag{11}$$

Where $J(W,b)$ is a cost function that optimizes two variables W and b. The number of layers in the network is n_l, and the serial number of the layers is l. The first term is data reconstruction error, and lowering this item might result in a more accurate data representation. The second term is a regularization to keep the weight's amplitude low and avoid network form overfitting. To fine-tune the reconstruction error and network weight, use parameter. When sparse constraints are taken into account, the cost function of SAE is rewritten as

$$J_{sparse}(W,b) = J(W,b) + \alpha \sum_{j=1}^{s_1} KL(p \| p_j) \tag{12}$$

Where α is a parameter to tune sparse penalty and $J(W,b)$. Where $KL(p \| p_j)$ is a Kullback-Leibler divergence.

$$\sum_{j=1}^{s_1} KL(p \| p_j) = \sum_{j=1}^{s_1} p \log \frac{p}{p_j} + (1-p) \log \frac{(1-p)}{(1-p_j)} \tag{13}$$

The classification of features carried out by deep transfer learning based on GoogleNet. There are 22 hidden layers in GoogleNet. The neural network has a greater depth than AlexNet. The network accurately classifies the data more effectively as a result of the increased depth. This network also identifies and extracts characteristics from input photos automatically. In this research, GoogleNet is updated to categorize binary classifications, such as malignant or benign. In general, higher-dimensional characteristics are favored since they are easier to manage in a new network and contribute more to a faster training process. This is in direct opposition to the GoogleNet architecture's function in selecting more efficient features with smaller dimensions. This GoogleNet identifies photos more effectively than previous approaches. To improve cloud IDS efficiency, the ETSO algorithm is used. When solving high-dimensional complex problems, the transient search algorithm (TSO) [22] is a modern physics-based metaheuristic algorithm that simulates the transient behaviour of switching circuits such as inductors and capacitors. However, the algorithm has slow convergence and a weak ability to overcome local optima. ETSO is proposed to fix these drawbacks. In ETSO, a weighting technique is used to increase discovery, exploitation, and convergence speeds.

A randomization within the search boundary is used to generate the position of each search agent, and the following mathematical expression is used to generate the initial population:

$$X_{ij} = lb + r \times (ub - lb) \qquad i = 0, \ldots, N \ , \ j = 0, \ldots, d \tag{14}$$

Where X_{ij} denotes the coordinates of the j^{th} dimension of the i^{th} population, N is the number of search agents, d is the dimensionality of the problem to be solved, r is a random number with a uniform distribution, and ub and lb are the upper and lower search space boundaries, respectively. The oscillation of the second-order Resistance, Inductance, Capacitance (RLC) circuit near the zero point, as well as the following mathematical expression, encourage experimentation in the search for the optimal solution.

$$X(t+1) = X^*(t) + e^{-T}[\cos(2\pi T) + \sin(2\pi T)] \left| X(t) - C_1 X^*(t) \right| \tag{15}$$

Where $X(t+1)$ denotes the current agent's position at the next moment, $X(t)$ is the current best agent's position, $X(t)$ is the current position, and T and C_1 are random coefficients. The mathematical expression for exploitation behaviour is

$$X(t+1) = X^*(t) + [X(t) - C_1 X^*(t)] \cdot e^{-T} \tag{16}$$

The inertia weights are constant for original TSO algorithms, which reduce the convergence speed. So inertia weights implemented to improve the convergence speed. Change in inertia weight according to the number of iterations

$$w(t) = a \cos^b (\ln(1 + e^{\frac{1}{T_{\max}}}) + c \tag{17}$$

Where a, b, and c are optimal parameters. In the ETSO, the exploration and exploitation behaviour are represented used following mathematical equations:

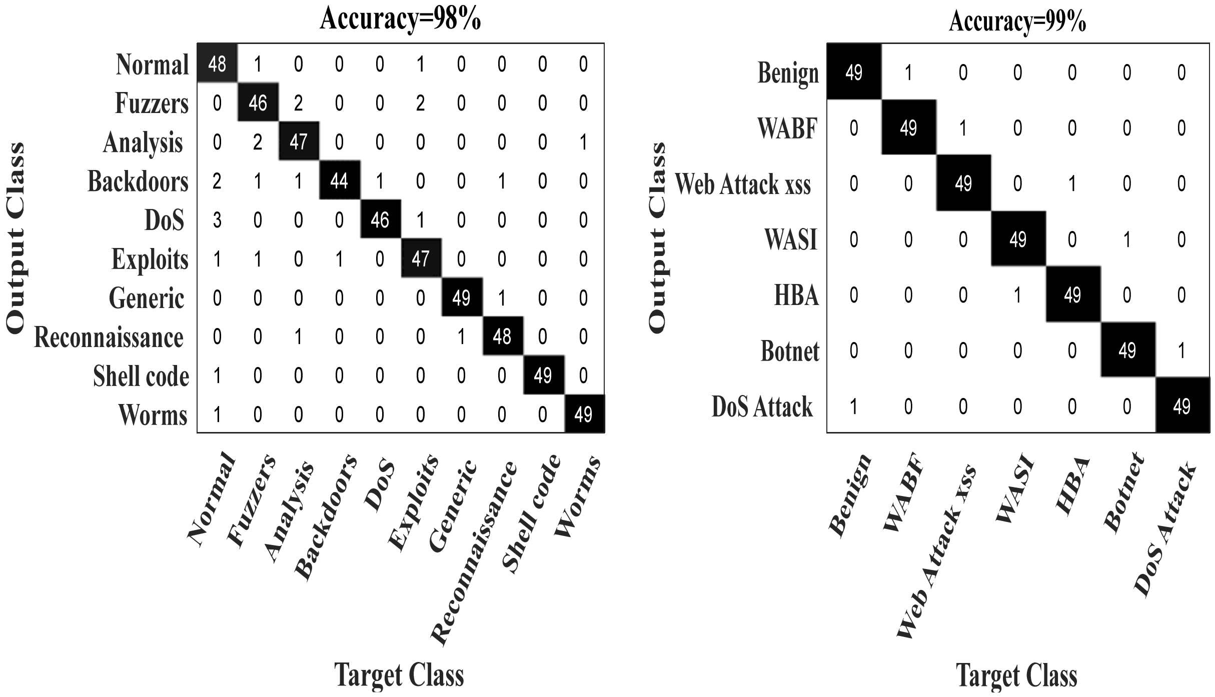
$$X(t+1) = w(t) \cdot X^*(t) + e^{-T}[\cos(2\pi T) + \sin(2\pi T)] \left| X(t) - C_1 X^*(t) \right| \tag{18}$$

$$X(t+1) = w(t) \cdot X^{*}(t) + [X(t) - C_1 \cdot X^{*}(t)] \cdot e^{-T} \tag{19}$$

ETSO algorithm shows superiority in both convergence speed and convergence precision.

4. Simulation Results

The proposed work is implemented in the python platform. The overall results show that, the proposed method is efficient compared with other existing methods. This will enable high security to the users. The terms that are proposed for performance evaluation are accuracy, precision, detection rate, and F-Score. These performance metrics of the proposed method are compared with the performance of existing methods. Two datasets are used such as CICIDS-2017 and UNSW-NB15. The CICIDS-2017 dataset [23] contains 225,745 data packages with over 80 features that span over 5 days of network operation. The CICIDS-2017 includes seven attack categories: DDoS, Dos, Infiltration, Web, Botnet, Heart bleed, and Brut power. The UNSW-NB15 dataset [24] includes 2,540,044 records, 49 features, and nine attack categories. The attacks in the UNSW-NB15 dataset include reconnaissance, shellcode, worms, generics, exploits, DoS, backdoors, analysis, and fuzzers. The confusion matrix for UNSW-NB15 dataset and CICIDS2017 dataset are represented in figure 2.



	Normal	Fuzzers	Analysis	Backdoors	DoS	Exploits	Generic	Reconnaissance	Shell code	Worms
Normal	48	1	0	0	0	1	0	0	0	0
Fuzzers	0	46	2	0	0	2	0	0	0	0
Analysis	0	2	47	0	0	0	0	0	0	1
Backdoors	2	1	1	44	1	0	0	1	0	0
DoS	3	0	0	0	46	1	0	0	0	0
Exploits	1	1	0	1	0	47	0	0	0	0
Generic	0	0	0	0	0	0	49	1	0	0
Reconnaissance	0	0	1	0	0	0	1	48	0	0
Shell code	1	0	0	0	0	0	0	0	49	0
Worms	1	0	0	0	0	0	0	0	0	49

(a) UNSW-NB15 dataset

	Benign	WABF	Web Attack xss	WASI	HBA	Botnet	DoS Attack
Benign	49	1	0	0	0	0	0
WABF	0	49	1	0	0	0	0
Web Attack xss	0	0	49	0	1	0	0
WASI	0	0	0	49	0	1	0
HBA	0	0	0	1	49	0	0
Botnet	0	0	0	0	0	49	1
DoS Attack	1	0	0	0	0	0	49

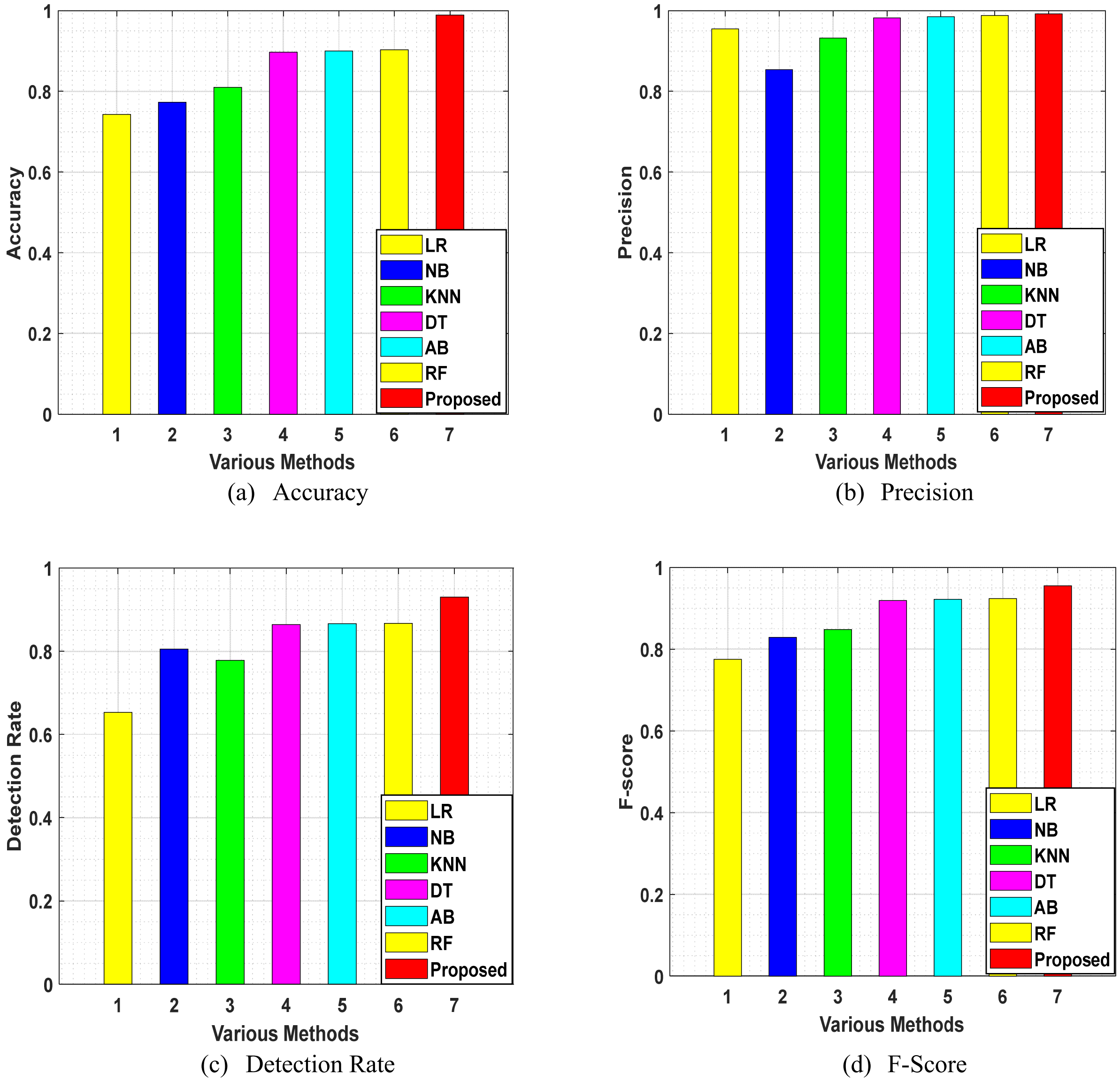
(b) CICIDS 2017 dataset

Figure 2: Confusion matrix for UNSW-NB15 and CICIDS 2017 datasets

The simulation results for UNSW-NB15 dataset in terms of simulation matrices is represented in table 1 and graphically represented in figure 3.

Table 1: Comparison results of UNSW-NB15 dataset

Method	Accuracy	Precision	Detection rate	F-Score
LR	0.743	0.955	0.653	0.775
NB	0.773	0.854	0.805	0.829
KNN	0.81	0.932	0.778	0.848
DT	0.897	0.982	0.864	0.919
AB	0.9	0.985	0.866	0.922
RF	0.903	0.988	0.867	0.924
Proposed	0.9892	0.992	0.93	0.955



(a) Accuracy (b) Precision

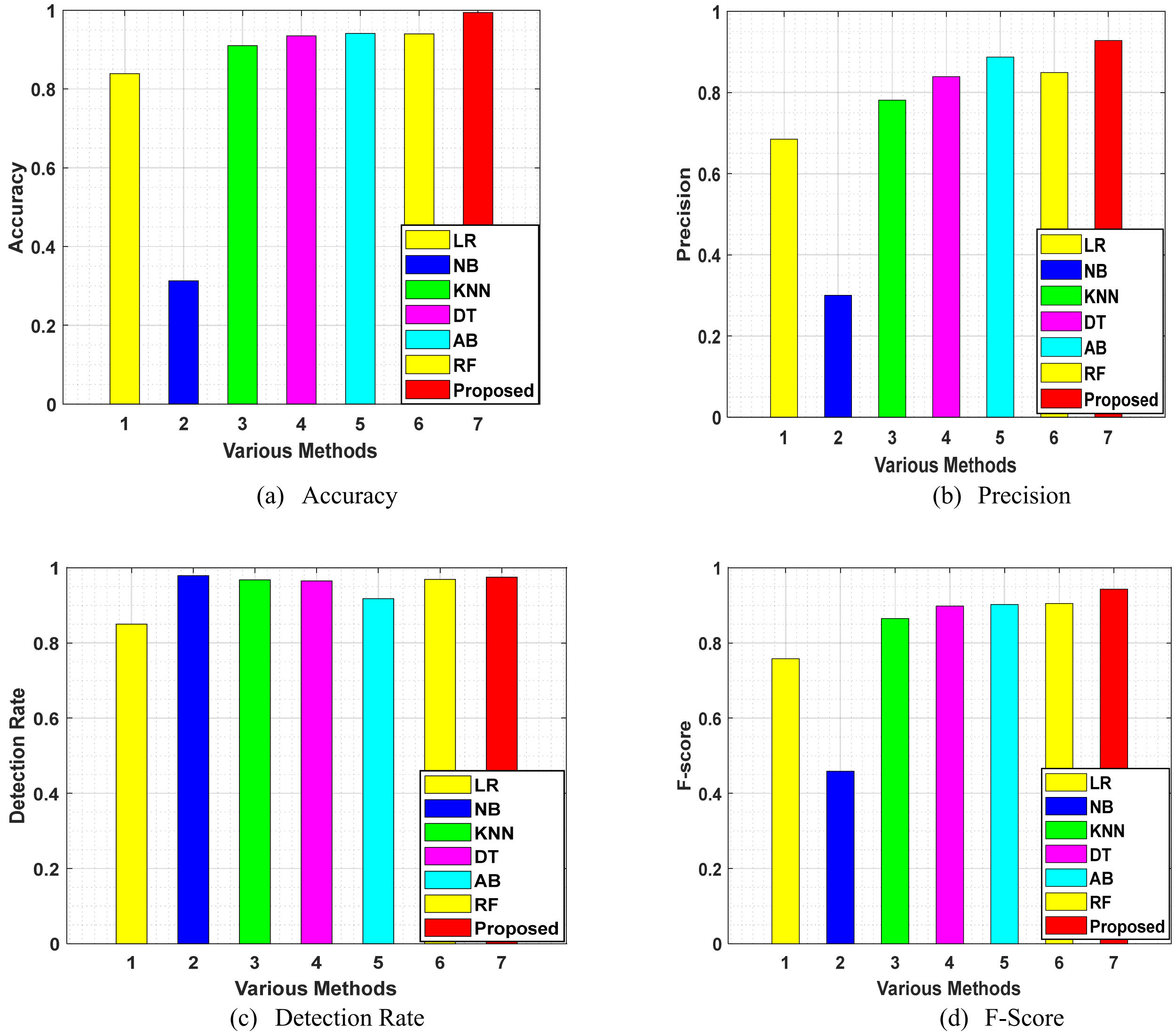
(c) Detection Rate (d) F-Score

Figure 3: Performance analysis for UNSW-NB15 dataset

The comparison of proposed method with various existing methods for UNSW-NB15 dataset is represented in figure 3. The proposed method is compared with traditional method such as LR, NB, KNN, DT, AB, and RF in terms of accuracy, precision, detection rate, and F-score. From the analysis, it is observed that the proposed method outperforms the existing method. The simulation results for CICIDS 2017 dataset in terms of performance matrices is represented in table 2 and figure 4.

Table 2: Performance comparison for CICIDS 2017 dataset

Method	Accuracy	Precision	Detection rate	F-Score
LR	0.839	0.685	0.85	0.758
NB	0.313	0.3	0.979	0.459
KNN	0.91	0.781	0.968	0.865
DT	0.935	0.839	0.965	0.898
AB	0.941	0.887	0.918	0.902
RF	0.94	0.849	0.969	0.905
proposed	0.9942	0.928	0.975	0.943



(a) Accuracy (b) Precision

(c) Detection Rate (d) F-Score

Figure 4: Performance Analysis for CICIDS 2017 dataset

The comparison of proposed method with various existing methods for CICIDS 2017 dataset is represented in figure 4. The proposed method is compared with conventional method in terms of accuracy, precision, detection rate, and F-score. From the analysis, it is observed that the proposed method is better than the existing methods.

5. Conclusion

Cloud computing is becoming more effective as technology advances, necessitating the use of IDS to protect user data. IDS assist in stopping users from engaging in cloud-based practises. In this paper, extended STSDTL-IDS is presented. The presented approach detects and classifies complex attacks accurately and finely. A feature selection method, namely, COA is used to eliminate the unwanted features and to make the classification more exact and precise. The classification accuracy improved by the sparse deep transfer learning algorithm. The performance of the sparse deep transfer learning algorithm is improved using the extended transient search optimization in which weight parameter is included to increase the performance. The tool used in this method is python, and the datasets taken here are UNSW-NB15 and CICIDS2017. The results obtained showed that the proposed algorithm is far better than the existing algorithms in terms of Accuracy, Precision, Detection rate, and F-Score. The proposed method shows higher accuracy than the research survey. The findings show that deep learning approaches may considerably increase the performance of the proposed intelligent attack detection approach, inspiring other researchers to develop stronger deep neural networks for intelligent attack detection in this direction.

References

[1] Shamshirband, S., Fathi, M., Chronopoulos, A.T., Montieri, A., Palumbo, F. and Pescapè, A., 2020. Computational intelligence intrusion detection techniques in mobile cloud computing environments: Review, taxonomy, and open research issues. Journal of Information Security and Applications, 55, p.102582.

[2] Bdair, A.H., Abdullah, R., Manickam, S. and Al-Ani, A.K., 2020. Brief of intrusion detection systems in detecting ICMPv6 attacks. In Computational Science and Technology (pp. 199-213). Springer, Singapore.

[3] Gassais, R., Ezzati-Jivan, N., Fernandez, J.M., Aloise, D. and Dagenais, M.R., 2020. Multi-level host-based intrusion detection system for Internet of things. Journal of Cloud Computing, 9(1), pp.1-16.

[4] Devan, P. and Khare, N., 2020. An efficient XGBoost–DNN-based classification model for network intrusion detection system. Neural Computing and Applications, pp.1-16.

[5] Ramaiah, M., Chandrasekaran, V., Ravi, V. and Kumar, N., 2021. An intrusion detection system using optimized deep neural network architecture. Transactions on Emerging Telecommunications Technologies, 32(4), p.e4221.

[6] Jiang, K., Wang, W., Wang, A. and Wu, H., 2020. Network intrusion detection combined hybrid sampling with deep hierarchical network. IEEE Access, 8, pp.32464-32476.

[7] Ribeiro, J., Saghezchi, F.B., Mantas, G., Rodriguez, J., Shepherd, S.J. and Abd-Alhameed, R.A., 2020. An autonomous host-based intrusion detection system for Android mobile devices. Mobile Networks and Applications, 25(1), pp.164-172.

[8] Safaldin, M., Otair, M. and Abualigah, L., 2021. Improved binary gray wolf optimizer and SVM for intrusion detection system in wireless sensor networks. Journal of ambient intelligence and humanized computing, 12(2), pp.1559-1576.

[9] Li, X., Chen, W., Zhang, Q. and Wu, L., 2020. Building auto-encoder intrusion detection system based on random forest feature selection. Computers & Security, 95, p.101851.

[10] Ahmim, A., Maglaras, L., Ferrag, M.A., Derdour, M. and Janicke, H., 2019, May. A novel hierarchical intrusion detection system based on decision tree and rules-based models. In 2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS) (pp. 228-233). IEEE.

[11] Chiba, Z., Abghour, N., Moussaid, K., El Omri, A. and Rida, M., 2019, April. A clever approach to develop an efficient deep neural network based IDS for cloud environments using a self-adaptive genetic algorithm. In 2019 International Conference on Advanced Communication Technologies and Networking (CommNet) (pp. 1-9). IEEE.

[12] Fu, Y., Lou, F., Meng, F., Tian, Z., Zhang, H. and Jiang, F., 2018, June. An intelligent network attack detection method based on rnn. In 2018 IEEE Third International Conference on Data Science in Cyberspace (DSC) (pp. 483-489). IEEE.

[13] El Hajji, S., Moukafih, N. and Orhanou, G., 2019, April. Analysis of neural network training and cost functions impact on the accuracy of IDS and SIEM systems. In International Conference on Codes, Cryptology, and Information Security (pp. 433-451). Springer, Cham.

[14] Chiba, Z., Abghour, N., Moussaid, K. and Rida, M., 2018, October. A New Hybrid Framework Based on Improved Genetic Algorithm and Simulated Annealing Algorithm for Optimization of Network IDS Based on BP Neural Network. In The Proceedings of the Third International Conference on Smart City Applications (pp. 507-521). Springer, Cham.

[15] Ravindranath, V., Ramasamy, S., Somula, R., Sahoo, K.S. and Gandomi, A.H., 2020, July. Swarm intelligence based feature selection for intrusion and detection system in cloud infrastructure. In 2020 IEEE Congress on Evolutionary Computation (CEC) (pp. 1-6). IEEE.

[16] Almiani, M., AbuGhazleh, A., Al-Rahayfeh, A., Atiewi, S. and Razaque, A., 2020. Deep recurrent neural network for IoT intrusion detection system. Simulation Modelling Practice and Theory, 101, p.102031.

[17] Chen, L., Xian, M., Liu, J. and Wang, H., 2020, August. Intrusion Detection System in Cloud Computing Environment. In 2020 International Conference on Computer Communication and Network Security (CCNS) (pp. 131-135). IEEE.

[18] Besharati, E., Naderan, M. and Namjoo, E., 2019. LR-HIDS: logistic regression host-based intrusion detection system for cloud environments. Journal of Ambient Intelligence and Humanized Computing, 10(9), pp.3669-3692.

[19] Ghosh, P., Karmakar, A., Sharma, J. and Phadikar, S., 2019. CS-PSO based intrusion detection system in cloud environment. In Emerging Technologies in Data Mining and Information Security (pp. 261-269). Springer, Singapore.

[20] Jaber, A.N. and Rehman, S.U., 2020. FCM–SVM based intrusion detection system for cloud computing environment. Cluster Computing, pp.1-11.

[21] Khishe, M. and Mosavi, M.R., 2020. Chimp optimization algorithm. Expert systems with applications, 149, p.113338.

[22] Qais, M.H., Hasanien, H.M. and Alghuwainem, S., 2020. Transient search optimization: a new meta-heuristic optimization algorithm. Applied Intelligence, 50(11), pp.3926-3941.

[23] Vinayakumar, R., Alazab, M., Soman, K.P., Poornachandran, P., Al-Nemrat, A. and Venkatraman, S., 2019. Deep learning approach for intelligent intrusion detection system. IEEE Access, 7, pp.41525-41550.

[24] Janarthanan, T. and Zargari, S., 2017, June. Feature selection in UNSW-NB15 and KDDCUP'99 datasets. In 2017 IEEE 26th international symposium on industrial electronics (ISIE) (pp. 1881-1886). IEEE.

Authors Profile

Gavini Sreelatha, SSC(2000):B.Tech from Computer Science & Engineering from JB Institute of Engineering and Technology,Hyderabad, Mtech from Computer Science & Engineering from AVN Institute of Engineering and Technology.

Curently working as Assistant Professor with a demonstrated history of working in the Education Sector and 11 years of teaching experience and 3 years of industry experience. Pursuing Ph.D in Lincoln University College, Malaysia.

Skilled in Technical Leadership, Research, Mentoring, Problem Solving, and Teaching. Research interests are Cloud Computing , Security & Machine Learning. Passion towards delivering technical content to students.

Principal, JNTUH College of Engineering, Kukatpally, Hyderabad. Director of Admissions, JNTUH, Kukatpally, Hyderabad. Director, SCDE, JNTUH, Kukatpally, Hyderabad. Head, Dept of Computer Science and Engineering, JNTU College of Engineering, Hyderabad. Convener , PG Admissions JNTUH, JNTUA, JNTUK & Kaktiya University . Presently Dr A. Vinaya Babu is associated with Methodist Group of Institutions : Stanley College of Engineering and Technology for Women (SCETW), Methodist college of Engineering & Technology and Indur institute of Engineering & Technology, Siddipet as Director Academic and planning. Achieved NBA Accreditaion to all programmes successfully in SCETW.

Dr. Divya Midhun Chakkaravarthy, Deputy Dean, Centre of Postgraduate Studies at Lincoln University College, Malaysia was conferred with the ‘Best Woman Performer of the Year Award (Overseas)’ in the International Inspirational Women Awards 2020 organised by the GISR Research Foundation on 18th January, 2020 at Hotel Park Ascent, Noida, India.