An Ensemble Learning Based Block-Chain Framework-Enabled Collaborative Intrusion Detection

[1*] Krishna Bihari Dubey, [2] Dr. Mukta Goyal
[1*] Research Scholar in the Department of Computer Science & Engineering and Information Technology,
Jaypee Institute of Information Technology, Noida, India
[2] Associate Professor in the Department of Computer Science & Engineering and Information Technology,
Jaypee Institute of Information Technology, Noida, India

*Correspondencing Author E-mail**:** kbd1979@gmail.com

Abstract— Significant research has been done on combining intrusion detection with blockchain to increase data privacy and identify ongoing and upcoming cyberattacks. These method makes use of learning-based ensemble methods to assist in the identification of complex hazardous events while at the same time maintaining the confidentiality of the data. These models may also be used to provide additional privacy and security assurances during the live migration of virtual machines (VMs) to the cloud. As a result of this, virtual machines could be moved between data centres or cloud service providers in a safe and prompt manner. In this paper, a Deep Block-chain Framework (DBF) is suggested. The goal of the DBF is to create a privacy-based blockchain that includes smart contracts & security-based centralized intrusion detection. When dealing with sequential network data utilizing the UNSW-NB15 datasets, an ensemble learning approach is used for the purpose of detecting intrusions.

I. Introduction

The development of secure decentralized apps is significantly hampered by a lack of faith in the shared virtualization technology that is used in cloud environments. Malicious cyberattacks, including such Distributed Denial of Service (DDoS) and ransomware, are launched against cloud-based computer systems. To carry out appropriate mitigation and reduce any harm done to cloud services, it is essential to be able to recognize and react to such assaults[1]. Intrusion prevention and detection systems, also known as IDSs and IPSs, are often used in order to monitor & identify complex attacks that originate through endpoints of cloud networks. Even now, cloud systems are vulnerable to complex attack scenarios, which are becoming worse as block chain technology develops[2]. For example, in June 2018, numerous block chain crypto currencies (such as Bit coin Gold and Mona Coin) were the targets of 51% assaults, which resulted in the loss of tokens valued at roughly 18 million. Due to this flaw, attackers were able to double-spend transactions, which had an impact on the network's overall integrity. Additionally, utilizing block chain technology and smart contracts, an unidentified attacker was able to steal more than 3.6 million ethers[3].

In a variety of contexts, blockchain technology has been used to improve trust and data privacy[4]. The use of blockchain technology is not limited to the domain of digital currencies and financial services. Vote counting, the electricity industry, the Internet of Things (IoT), supply chain and manufacturing, pharmaceuticals in addition to healthcare, big data, cyber security, and government services are just a few examples of applications that could be developed using this technology. There are many other potential applications as well. IDS and block chain are complementary technologies that can work together to detect and prevent cyber-attacks on Internet of Things (IoT) and cloud computing networks[5]. The

deployment sites of cloud-based intrusion detection systems allow for the differentiation into host- and network-based systems. A host-based IDS (HIDS) is used to monitor and examine operating system audit data[6]. If the Host Intrusion Detection System (HIDS) discovers malicious behaviors emanating from a certain host or virtual machine (VM), the source IP would be flagged as permitting access to the whole network. Because of this, it will be impossible for user-to-root attacks to hop from one virtual machine to the next and acquire access to that VM. The network infrastructure of cloud networks makes use of a Network-based Intrusion Detection System

(NIDS) in order to monitor the network traffic of all of the associated machines. It can detect backdoor, port-scanning, direct and indirect floods, as well as suspicious malware activity[7].

Scalable and affordable collaborative IDSs (CIDSs) are used to examine different cloud nodes. The cloud presents a number of basic challenges, one of the most essential of which is the ability to keep data security & trust management across a number of different cloud service providers. The public, distributed, and decentralized nature of the cloud system might make building trust difficult since different components are under the authority of various parties[8]. Due to worries about data security and privacy, cloud companies are often unwilling to share data or disclose breach instances. Insider threats such as collusion as well as betrayal attacks, in which malicious nodes collaborate together to expanded misleading information & decrease the efficiency of alarm aggregation, present yet another significant challenge. These types of attacks involve rogue nodes working together just to spread misleading information. In addition to the detection of attack events using CIDSs in the cloud, confidentiality methods are commonly used to convert, modify, or obscure the original material in to safeguard it from unwanted access. This is done in order to prevent the content from being compromised. In contrast to the identification of assault occurrences, this is carried out as well. The technologies of block chains and smart contracts are becoming more popular as methods of protecting users' privacy in the cloud. These technologies also offer cloud components additional authentication and integrity.

Through encryption and consensus procedures, block chain technology overcomes the absence of security, trust, and accountability[9]. In a distributed crypto currency system where all transactions are handled independently from third parties or centralized organizations, Bit coin is regarded as one of the first successful implementations. This protects the authenticity and integrity of the data. Numerous peer-reviewed cryptographic hash methods are used to defend the system architecture of crypto-currencies[10]. Ethereum is a crypto currency and a decentralized computing platform that enables programmers to create autonomous agents that may function as smart contracts on a block chain network. However, there is presently a lack of a trustworthy paradigm of trust for digital smart contracts that are executed systematically in block chains. Numerous security weaknesses and assaults in block chain and related technologies like bit coin and Ethereum have been recently brought to light through publications[11]. This paper proposes a Deep Block-chain Framework-enabled Collaborative Intrusion Detection based on Ensemble Learning.

Many of the authors discussed about collaborative intrusion detection system using different methods and algorithms. Some of them are discussed below.

[12] proposed a novel Generative Adversarial Networks (GAN) based Block Chain enabled secured Routing Protocol (GBCRP) and also proposed a new intrusion detection system using GAN. [13] had proposed a brand-new collaborative intrusion detection (CID) technique using block chain for MMG systems in smart grid. A case study on an MMG system is used to illustrate the efficiency of the suggested strategy. [14] This research suggested a distributed edge device-based training model offloading cooperative intrusion detection technique. [15] In this article, a deep block chain framework (DBF) was put out as a solution for IOT networks' need for privacy-based block chain with smart contracts and distributed intrusion detection that is based on security.

Contribution:

1. To identify cyber-attacks from network data movement in cloud systems, an intrusion detection approach employing an ensemble learning algorithm is presented.
2. To assess the usefulness of the suggested system before deploying it to the cloud, Its performance is analysed and contrasted with that of a number of other intrusion detection methods.

Organization of the study

The work may be broken down into the following sections: Background of the proposed methods are provided in Section 2, an explanation of the methodology that underpins the recommended algorithm is provided in Section 3, Section 4 delivers the findings of the study, and Section 5 provides a conclusion.

II. BACKGROUND

In this section, background of algorithms such as random forest, Gaussian Naïve Bayes, Logistic Regression are described.

A. Random Forest

During training, random forests (RF) produce a number of different decision trees. For regression or classification, the final prediction is combined from the mean prediction and the mode of the classes. Because they employ a range of findings to reach a conclusion, ensemble methods are so called for this reason. Here is how the random forest categorization process works: Assuming there are just two child nodes (binary tree), each decision tree in Scikit-learn utilizes Gini Importance to evaluate a node's significance:

$$im_j = wsn_j mp_j - wsn_{left(j)} mp_{left(j)} - wsn_{right(j)} mp_{right(j)}$$

Where

$im_j \rightarrow$ the significance of the node j protocol

$wsn_j \rightarrow$ the number of weighted samples that made it to node j.

$mp_j \rightarrow$ a measure of the level of impurity in node j

$left(j) \rightarrow$ The child node on the left branched out on node j.

$right(j) \rightarrow$ child node on node j coming from the right split.

Following that, the relevance of each characteristic on a decision tree is assessed as follows: $fi_i = \frac{\sum_{j:node\ j\ splits\ on\ feature\ i} ni_j}{\sum_{k \in all\ nodes} ni_k}$

fi sub(i)= the importance of feature i
ni sub(j)= the importance of node j
These may then be divided by the sum of all feature significance values to normalize to a range of 0 to 1:

$$normfi_i = \frac{fi_i}{\sum_{j \in all\ features} fi_j}$$

The average of a characteristic over all trees, at the Random Forest level, determines its final importance. The sum of the feature significance values for all the trees is divided by the total number of trees: $RFfi_i = \frac{\sum_{j \in all\ trees} normfi_{ij}}{T}$

RFfi sub(i)= I computed the feature's relevance using data from all the trees in the Random Forest model.
normfi sub(ij)= the normalized feature importance for i in tree j
T = total number of trees

B. Gaussian Naïve Bayes

When dealing with continuous data, it is common practice to make the assumption that the continuous values related to each class follow a normal distribution. This is because it is easier to analyses the data this way. The possibility of the qualities is seen as-

$$P(x_i \mid y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right)$$

Sometimes assume variance

- is independent of Y (i.e., σi),
- or independent of Xi (i.e., σk)
- or both (i.e., σ)

C. Logistic Regression

Logistic regression is one of those machine learning (ML) algorithms that is not a black box since we understand precisely what it works. Black box models are often more advanced variants, such as an extremely deep neural network (DNN). In binomial logistic regression, the result may be either 0 or 1. One of those machine learning (ML) techniques that is not a mystery is logistic regression since we know exactly how it works. Black box models are often more powerful iterations, such an incredibly deep neural network (DNN). When using binomial logistic regression, the outcome might either be 0 or 1. Calculating whether a certain event will result in a 1 or a 0 is the next step. Given that p is the probability of a 1, 1-p represents the likelihood that it won't. This is an example of the Bernoulli distribution, a particular kind of Binomial distribution. Using a modified linear regression model, logistic regression seeks to reflect the problem.

$$\hat{y} = \beta_0 + \beta_1 x_1 + ... + \beta_n x_n$$

The sigmoid function could be:

$$P = \frac{1}{1+e^{-\hat{y}}}$$

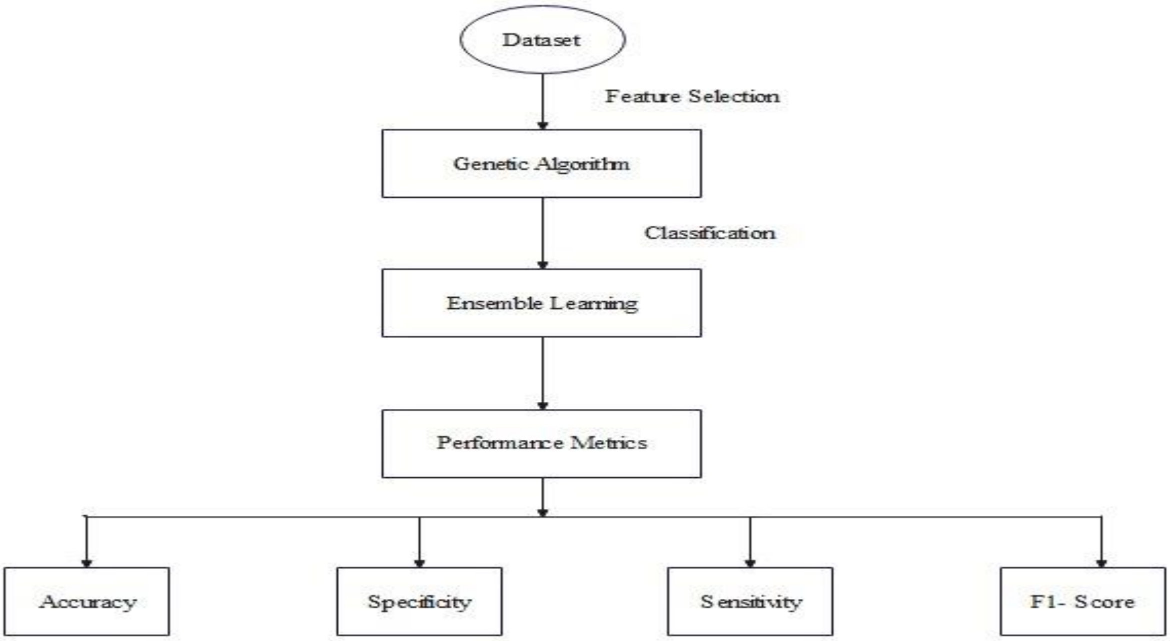
From this point, a single-layered neural network (NN) with a sigmoid activation function may be used to represent logistic regression. In other words, by minimizing the cross-entropy loss function given by the typical stochastic gradient descent (SGD) technique, we may estimate the weight parameters:

$$l(p, y) = -y\log(p) - (1-y)\log(1-p)$$
$$where \quad y \in [0,1]$$

Therefore, the multinomial logistic regression can only include adding more sigmoid neurons to the layer. Additionally, we might utilize the SoftMax to maintain the output as a probability distribution rather than the logistic function as the activation function. The loss is converted into the log likelihood loss in the multinomial logistic regression scenario with a SoftMax.

III. Methodology

In this study, Deep Block-chain Framework (DBF) is proposed, which aims to provide privacy-based block-chain with smart contracts and security-based distributed intrusion detection and also implemented a novel machine learning techniques i.e., ensemble learning to detect the Denial-of-Service attacks in cloud computing. The features are extracted by a nature-inspired algorithm called GA. And these extracted features will be trained to the proposed algorithm for detecting DDoS attacks. Following block diagram depicts the flow of proposed methodology.



The general description of Genetic Algorithm is described below.

A. *Genetic Algorithm (GA)*

The Genetic Algorithm, sometimes known as GA, is a search-based optimization method that is founded on the theories of genetics and natural selection. It is widely used to identify optimum or near-optimal solutions to challenging problems, the resolution of which would normally require a lifetime of effort. In addition to being used for the aim of resolving optimization-related challenges, it is also often employed in the research & machine learning domains.

In order to calculate the results, the genetic algorithm is taken into account the genetic structure as well as the behaviors of the chromosomes that make up the population. The following ideas are the foundation upon which genetic algorithms are built. There are a variety of solutions to the issue that may be found on each chromosome. As a consequence of this, the population has examples of all the chromosomes.

Every individual in the population may be characterised by making use of a fitness function. Consequently, improving one's degree of physical fitness is the solution to this problem. The healthiest and most capable individuals of the population are chosen to take on the role of parents to the children of the generation that comes after them. This guarantees that the population will continue to flourish in the future.

Because of the mutation, the offspring who are produced will have features that are a combination of those of both their parents. A change in the structure of a gene, regardless of how little it may be, is referred to as a mutation.

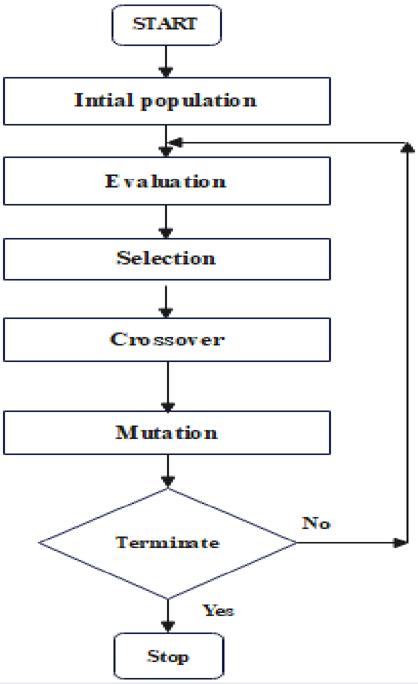


Figure 1 Flowchart for Genetic Algorithm

After extract useful features from dataset by using genetic algorithm, the selected features are further assigned for data analysis to train the proposed method.

B. Data Analysis

In order to extract useful features from the dataset, GA first select some of its features, which are then trained using the Ensemble-based classification method. The suggested approach is compared with various ones already in use, including logistic regression, random forest, and Gaussian naive bayes.

1) Ensemble Learning

To solve a particular computational intelligence problem, ensemble learning involves the systematic generation and combination of numerous models, such as classifiers or experts. In order to improve a model's performance or reduce the likelihood of someone unintentionally choosing a poor model, ensemble learning is often utilized. In addition to its usage for error correction, ensemble learning is put to use in order to improve prediction performance. This is accomplished by using ensemble learning in order to achieve results including such low regression error and high classification accuracy. The term "ensemble learning" refers to a set of methods that include data fusion, incremental learning, non-stationary learning, and presenting the model's decision with a confidence. By mixing several models, ensemble learning enhances the output of machine learning. Comparing this strategy to a single model, it produces predictions that are better. The fundamental concept is to educate a group of professionals (classifiers) before letting them cast a vote.

The predictions made by each model are seen as "votes" in the competition. The forecasts that most models provide serve as the foundation for the final forecast. In addition to other things, averaging may calculate probabilities in classification problems and make predictions in regression situations. Voting is one of the simplest ways to combine predictions from many machine learning algorithms. The voting classifier is a container for a variety of classifiers that are trained and assessed concurrently to make advantage of the distinct features of each method. A majority vote utilizing one of two processes decides what the final result of a prediction will be.

Hard voting/ majority voting: Voting by show of hands, sometimes known as "hard voting," is the most basic form of majority voting. In this particular circumstance, the class that was selected will be the one that garnered the most votes totaling Nc (yt). We are able to forecast the class label y by using the votes of each classifier in which the majority prevailed.

$$\hat{y} = \arg\max(N_c(y_t^1), N_c(y_t^2), ..., N_c(y_t^n))$$

Consider combining three classifiers that divide a training sample into the following categories:

- classifier 1 -> class 0
- classifier 2 -> class 0
- classifier 3 -> class 1

y^=mode {0,0,1} =0

Using a majority vote, we would categorize the sample as “class 0”. Soft Voting: In this illustration, the probability vectors for each predicted class are merged and averaged. This procedure is called soft voting. The winner will be determined by the class that has the greatest value.

To know the performance of the proposed algorithm, we evaluated the performance parameters like accuracy, sensitivity, and specificity and F1 score by comparing it with existing algorithms by considering two scenarios with and without feature selection.

The recommended algorithm's quality is evaluated using the performance metrics listed below.

C. *Performance metrics*

Accuracy, sensitivity, precision, and the F1-score are some of the metrics that are used in the process of evaluating an algorithm's performance in relation to the confusion matrix's performance measurements.

Accuracy: It is the proportion of individuals that have been properly identified relative to the total number of subjects.

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

Sensitivity: The percentage of labels that our computer properly identifies as positive is known as recall, also known as sensitivity.

$$Sensitivity = \frac{TP}{TP + FN}$$

Precision: By factoring in the overall number of precise predictions, it is feasible to determine the accuracy of an outlook. This idea also goes by the name of predictive value.

$$\Pr ecision = \frac{TP}{TP + FP}$$

F1-Score: A metric that takes into account both recall and accuracy is the F1-score.

$$F1 - score = 2 * \frac{\Pr ecisison * \text{Re}\, call}{\Pr ecision + \text{Re}\, call}$$

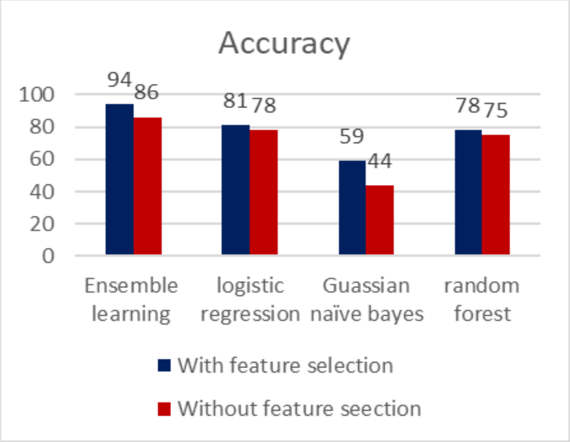
Specificity: The negative has been accurately labelled as specificity by the system.

$$Specificity = \frac{TN}{TN + FP}$$

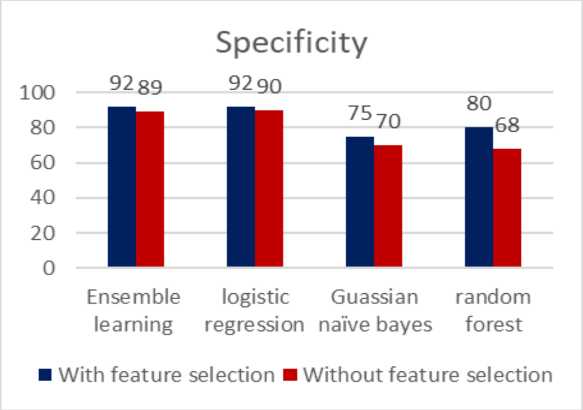
Where,
TP= True Positive
TN= True Negative
FP= False Positive
FN= False Negative

IV. Results

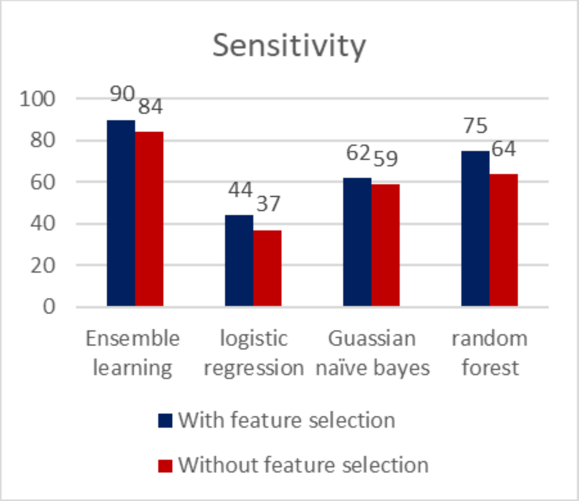
In order to demonstrate the efficiency of the proposed algorithm, performance measurements are used to compare it to other classification algorithms, such as Logistic regression, Gaussian Naive Bayes, and Random Forest algorithms. This comparison is done in order to demonstrate the algorithm's effectiveness.



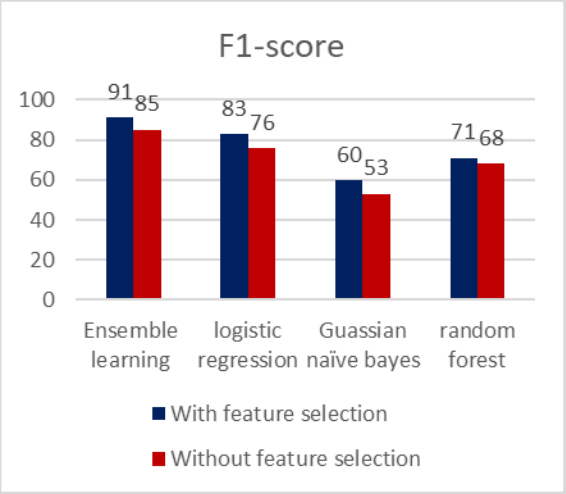
The above figure represents the accuracy analysis of four algorithms (with and without feature selection) with the x-axis being the proposed and existing algorithm and the y-axis being the accuracy value. Ensemble learning Accuracy without feature selection is 86% with feature selection is 94%. It can be concluded that Ensemble learning with feature selection produces effective results when compared with other algorithms.



The above figure represents the specificity analysis of four algorithms (with and without feature selection) with the x-axis being the proposed and existing algorithm and the y-axis being the accuracy value. Ensemble learning specificity without feature selection is 89% with feature selection is 92%. It can be concluded that Ensemble learning with feature selection produces effective results when compared with other algorithms.



The above figure represents the sensitivity analysis of four algorithms (with and without feature selection) with the x-axis being the proposed and existing algorithm and the y-axis being the accuracy value. Ensemble learning sensitivity without feature selection is 84% with feature selection is 90%. It can be concluded that Ensemble learning with feature selection produces effective results when compared with other algorithms.



The above figure represents the F1-score analysis of four algorithms (with and without feature selection) with the x-axis being the proposed and existing algorithm and the y-axis being the accuracy value. Ensemble learning F1-score without feature selection is 85% with feature selection is 91%. It can be concluded that Ensemble learning with feature selection produces effective results when compared with other algorithms.

V. Conclusion

In this study, a novel technique is used to discover for collaborative intrusion detection in cloud computing systems. i.e., Deep Block-chain Framework (DBF), which aims to provide privacy-based block-chain with smart contracts and security-based distributed intrusion detection and also implemented a novel machine learning techniques i.e., Ensemble learning. The actionable features are extracted from the set of features by using GA based nature optimization algorithm. Within the extracted features, a sample of features is selected and trained to the proposed model to detect the collaborative intrusion. The proposed algorithm is compared with other algorithms like other algorithms following performance parameters to prove the efficiency. It is observable from the GA-based feature selection that high prior features are selected. The performance parameters like accuracy, sensitivity, and specificity and F1 score are compared for proposed and existing algorithms by considering two scenarios with and without feature selection. The proposed algorithm with feature selection is with an accuracy of

94% and without feature selection 86% which is higher than the existing algorithms with a feature selection accuracy score. All values of the proposed algorithm are higher than existing algorithms values proving that the proposed algorithm is highly efficient. Hence it can be concluded that the GA-based ensemble learning algorithm is highly accurate in detecting collaborative intrusion in cloud computing systems.

VI. Acknowledgment

I would like to thank my Ph. D. guide Dr. Mukta Goyal for their motivational support and supervision in writing this article.

VII. Conflict of Interest

The authors certify that they have no affiliations with or involvement in any organization or entity with any financial interest, or non-financial interest in the subject matter or materials discussed in this manuscript

References:

[1] W. Li, S. Tug, W. Meng, and Y. Wang, "Designing collaborative blockchained signature-based intrusion detection in IoT environments," *Futur. Gener. Comput. Syst.*, vol. 96, pp. 481–489, 2019, doi: 10.1016/j.future.2019.02.064.

[2] W. Li, W. Meng, and L. F. Kwok, "Investigating the influence of special on-off attacks on challenge-based collaborative intrusion detection networks," *Futur. Internet*, vol. 10, no. 1, pp. 1–16, 2018, doi: 10.3390/fi10010006.

[3] J. Arshad, M. A. Azad, M. M. Abdellatif, M. H. Ur Rehman, and K. Salah, "COLIDE: A collaborative intrusion detection framework for Internet of Things," *IET Networks*, vol. 8, no. 1, pp. 3–14, 2019, doi: 10.1049/iet-net.2018.5036.

[4] N. Alexopoulos, E. Vasilomanolakis, N. R. Ivánkó, and M. Mühlhäuser, "Towards blockchain-based collaborative intrusion detection systems," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 10707 LNCS, pp. 107–118, 2018, doi: 10.1007/978-3-319-99843-5_10.

[5] B. Hu, C. Zhou, Y. C. Tian, Y. Qin, and X. Junping, "A Collaborative Intrusion Detection Approach Using Blockchain for Multimicrogrid Systems," *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 49, no. 8, pp. 1720–1730, 2019, doi: 10.1109/TSMC.2019.2911548.

[6] J. Hong and C. C. Liu, "Intelligent Electronic Devices with Collaborative Intrusion Detection Systems," *IEEE Trans. Smart Grid*, vol. 10, no. 1, pp. 271–281, 2019, doi: 10.1109/TSG.2017.2737826.

[7] A. Patel, H. Alhussian, J. M. Pedersen, B. Bounabat, J. C. Júnior, and S. Katsikas, "A nifty collaborative intrusion detection and prevention architecture for Smart Grid ecosystems," *Comput. Secur.*, vol. 64, pp. 92–109, 2017, doi: 10.1016/j.cose.2016.07.002.

[8] W. Meng, E. W. Tischhauser, Q. Wang, Y. Wang, and J. Han, "When intrusion detection meets blockchain technology: A review," *IEEE Access*, vol. 6, pp. 10179–10188, 2018, doi: 10.1109/ACCESS.2018.2799854.

[9] Y. I. Ll. Lucio, K. Marceles Villalba, and S. A. Donado, "Adaptive Blockchain Technology for a Cybersecurity Framework in IIoT," *Rev. Iberoam. Tecnol. del Aprendiz.*, vol. 17, no. 2, pp. 178–184, 2022, doi: 10.1109/RITA.2022.3166857.

[10] S. Teng, N. Wu, H. Zhu, L. Teng, and W. Zhang, "SVM-DT-based adaptive and collaborative intrusion detection," *IEEE/CAA J. Autom. Sin.*, vol. 5, no. 1, pp. 108–118, 2018, doi: 10.1109/JAS.2017.7510730.

[11] N. Kolokotronis, S. Brotsis, G. Germanos, C. Vassilakis, and S. Shiaeles, "On blockchain architectures for trust-based collaborative intrusion detection," *Proc. - 2019 IEEE World Congr. Serv. Serv. 2019*, no. July, pp. 21–28, 2019, doi: 10.1109/SERVICES.2019.00019.

[12] S. Rajasoundaran, S. V. N. S. Kumar, M. Selvi, S. Ganapathy, R. Rakesh, and A. Kannan, "Machine learning based volatile block chain construction for secure routing in decentralized military sensor networks," *Wirel. Networks*, vol. 27, no. 7, pp. 4513–4534, 2021, doi: 10.1007/s11276-021-02748-2.

[13] B. Hu, C. Zhou, Y. C. Tian, Y. Qin, and X. Junping, "A Collaborative Intrusion Detection Approach Using Blockchain for Multimicrogrid Systems," *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 49, no. 8, pp.

1720–1730, 2019, doi: 10.1109/TSMC.2019.2911548.

[14] H. Liu *et al.*, “Blockchain and Federated Learning for Collaborative Intrusion Detection in Vehicular Edge Computing,” *IEEE Trans. Veh. Technol.*, vol. 70, no. 6, pp. 6073–6084, 2021, doi: 10.1109/TVT.2021.3076780.

[15] O. Alkadi, N. Moustafa, B. Turnbull, and K. K. R. Choo, “A Deep Blockchain Framework-Enabled Collaborative Intrusion Detection for Protecting IoT and Cloud Networks,” *IEEE Internet Things J.*, vol. 8, no. 12, pp. 9463–9472, 2021, doi: 10.1109/JIOT.2020.2996590.