International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 3
DOI: https://doi.org/10.17762/ijritcc.v11i3.6328
***Article Received**: 18 December 2022 **Revised**: 22 January 2023 **Accepted**: 10 February 2023*

An Intelligent Optimal Secure Framework for Malicious Events Prevention in IOT Cloud Networks

Majjaru Chandra Babu[1], Senthilkumar K[2*]
[1]Research Scholar, School of Information Technology and Engineering,
VIT University,Vellore,
Tamil Nadu, India.
Email: majjaru.chandrababu2017@vitstudent.ac.in
[2*]Associate Professor, School of Computer Science Engineering,
VIT University,Vellore,
Tamil Nadu, India.
[*]Email: senthilkumark@vit.com

Abstract: The intrusion is considered a significant problematic parameter in Cloud networks. Thus, an efficient mechanism is required to avoid intrusion and provide more security to the cloud system. Therefore, the novel Artificial Bee-based Elman Neural Security Framework (ABENSF) is developed in this article. The developed model rescales the raw dataset using the pre-processing function. Moreover, the artificial bee's optimal fitness function is integrated into the feature extraction phase to track and extract the attack features. In addition, the monitoring mechanism in the developed model provides high security to the network by preventing attacks. Thus, the tracking and monitoring functions avoid intrusion by eliminating known and unknown attacks. The presented work was designed and validated with an NSL-KDD dataset in python software. Finally, the performance parameters of the presented work are estimated and verified with the existing techniques in a comparative analysis. The comparative performance shows that the developed model has earned better outcomes than others.

Keywords: Attack prevention; Security framework; Cloud computing; Neural Network; Internet of Things; Cloud storage

I.INTRODUCTION

The cloud computing concept has emerged from the distributed software architecture [1]. Cloud computing is like traditional computing; it is developed to foresee resource attainability across various categories [2]. The Because of immersive virtualization, cloud computing has become the actual computing platform permitting dynamic, scalable, and elastic reconfiguration of computing assets according to one's necessity [3]. Cloud computing is increasing the resources without the need for depth knowledge in new systems, without training new employees, and without developing new software [4].

Cloud service providers (CSP) provide software computing, infrastructure, and platform, i.e., Platform as a Service (PaaS), Software as a Service (SaaS), Infrastructure as a Service (IaaS), [5]. It can locate models were community, hybrid private, and public [6]. IaaS gives attractive offers to High-performance computing (HPC)[7]. It authorizes cloud users to select all the specifications of their resources on the cloud. The leaders of the IaaS were Amazon and Microsoft Azure cloud [8].

Platform as a Service (PaaS) is used to locate, control, and scale new applications. It was related to the manage storage place scheme, per data services, and a physically powerful ecology for locating and running new applications [9]. In Software as a Service, application service is delivered through cloud infrastructure and is supplied and metered on a basis [10]. With wireless system-aided computing, many authentic plots may have been gathered and kept in the cloud [11]. The cloud's capacity is utilized to explore the effective solution for these original plots. The similarities were separated from the machine learning mechanism [12]. Data will be forwarded to the base station to allocate radio efficiency efficiently. When a base station was installed in a new place, there were no materials about the original plots [14]. The new base station emerged into services, the real-time scenario collected from the system and later used for ancient material for learning [15]. The historical data contained more attributes, user numbers, Channel state information, and International mobile subscriber identification numbers (IMSIs). IMSIs users might be irrelevant to the allocation of resources. It can be extracted without losing much data quality [16]. However, there may be some defects in the data measurements, transferral, depot, and partial and equivalent characteristic vectors [17]. 70 to 90% of the aspect vectors were allocated to the training set [18].

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 3
DOI: https://doi.org/10.17762/ijritcc.v11i3.6328
***Article Received**: 18 December 2022 **Revised**: 22 January 2023 **Accepted**: 10 February 2023*

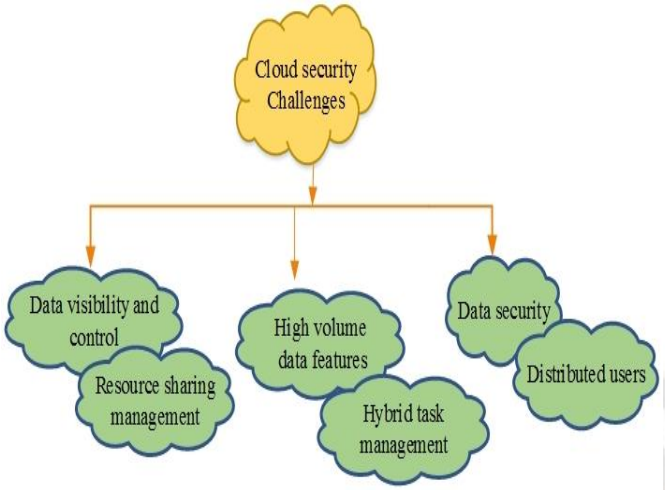

Figure 1. Security challenges in Cloud computing

The erect productivity model and connected solutions were transferred to the base stations [19]. At the base station, the measured data created the new characteristic binary. The new vector temporarily collected the data, and then the base station forwarded it to the cloud [20]. The primary issue for cloud computing security was data accessed devoid of authorization among the virtual strategy operations at the same server. Sometimes the cloud examination might be inaccessible for a long time because of mistakes and crashes [21, 22]. Several security models were implemented in the past, such as neural networks [24] and boosting mechanisms [25]. Still, the security issues are not solved because of data complexity. So, the present study aims to design an optimal solution for predicting malicious activities in trained cloud IoT networks.

II. RELATED WORK

Some of the few recent pieces of literature on cloud computing using machine learning.

Cloud computing is a great innovation technology due to its widespread aspects, such as online storage, high scalability, and ease of accessibility. It plays a significant role in the workforce. But it also has some drawbacks; with the increasing usage of cloud systems, security issues are proportional. So Mohammad et al.[25] said that the machine learning-based cloud computing mechanism to examine the big data used to improve the security and data transmitting rates. Big data can control a large amount of data in cloud systems. Big data analysis has high accuracy, performance, computational time, and effective data management. But it contains service tax is high and unexpected outages.

Cloud computing attained an essential position in the healthcare facility because of the condition to boost healthcare facility presentation. Anyhow, the optical choice of practical equipment which approaches the medical demand expresses a vital summons. For that, Abdelaziz et al. [26] explained that the recent model of the healthcare facility is based upon cloud computing through Parallel Particle Swarm optimization development used for selecting the virtual machines. Virtual machines model submits the fresh model chronic kidney infection analysis and forecast. It is executed using two methods; neural network (NN) and linear regression (LR). Several firms moved to cloud data centers as they generally used IT services. It is essential for cloud systems. However, there is so much data breaking, and it takes high time to verify the data.

One of the main issues in constructing future sustainable smart cities is expertise control and development. So Iatrellis, Omiros et al. [27] explained that a highly cloud-based IT approach merges the learning capability of management of personnel working and technical workers. The cloud-based IT approach connects to a highly skilled and expert system to contribute suitable competencies to the smart city. After achieving efficient competence management, the semantic model has been transmitted to a relational database. However, the data can be destroyed, and data may be used illegally.

The modern and rapidly developed mechanism termed the Internet of things (IoT) evolves in the separation of network and telecommunications with particular discussion to a contemporary place of wireless schemes. Stergiou et al. [28] said that security problems faced IOT, as well as cloud computing, by using the technology of wireless telecommunication systems. The primary goal of the relation and collaboration among the objects and things transmitted along with the wireless networks is to complete the purpose of the joint individual, for attaining the best surroundings for the usage of big data. However, it was power dependence, high cost, and technical complexity.

The key contribution of this present study is detailed as follows,

- Initially, the IoT cloud intrusion data was gathered and imported into the python environment.
- Then, a novel ABENSF has been designed with the required features extraction and malicious events forecasting parameters
- Moreover, the present noise features in the data are filtered in the initial phase pre-processing function
- After noise filtering, the present features were analyzed, and malicious features were classified.
- Once the malicious features are recognized, they are neglected by the network cloud environment.
- Finally, the detection robustness has been measured in terms of precision, accuracy, F-measure, recall, and error rate.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 3
DOI: https://doi.org/10.17762/ijritcc.v11i3.6328
***Article Received**: 18 December 2022 **Revised**: 22 January 2023 **Accepted**: 10 February 2023*

III. SYSTEM MODEL AND PROBLEM STATEMENT

Securing the cloud networks is an important factor in preserving user data privacy in the wireless network environment. Several security models have been introduced in the past to maintain the privacy of the cloud system. But the unstructured behavior of each user's data has raised the complexity range in securing the user's data. The fixed security behaviors are not supported for the unstructured user data for offering security functions. Hence, a less confidential score has been recorded for the cloud-based IoT system. These issues have motivated this present study to implement the optimal solution as the security mechanism.

Less security and a lower rate of the confidential score were considered the problem in the existing models. So to overcome those issues, the proposed model was developed with a higher security rate. In these existing models, a high amount of data were stored in the cloud environment, but a very lower rate of security was provided to those data. To increase security, the proposed model was developed. To obtain more security here, IoT data sets were chosen for implementation. Subsequently, the system model with the problem is illustrated in Fig. 2.

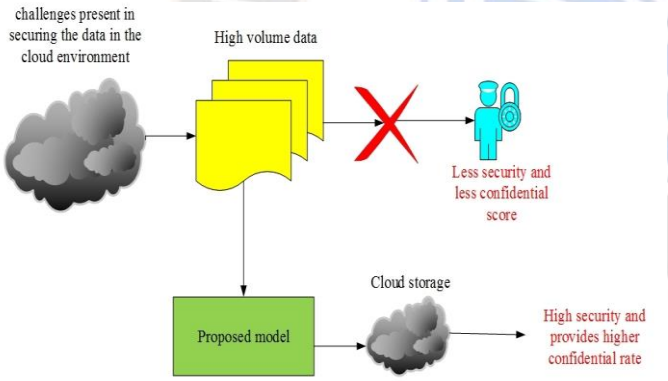

Figure 2. System model with problem

IV. PROPOSED METHODOLOGY

The novel artificial bee-based Elman neural security framework (ABENSF) has been planned for implementation in the IoT cloud wireless networks. Here, the motive of this presented model is to detect and prevent known and unknown attacks. Incorporating the artificial bee function has afforded the finest attack prediction and neglecting outcomes. Moreover, intrusion-based wireless cloud data has been considered to validate the model. Also, to check the robustness of the presented model, a few unknown attacks were launched, and security parameters were noted. The proposed architecture is described in Fig. 3.

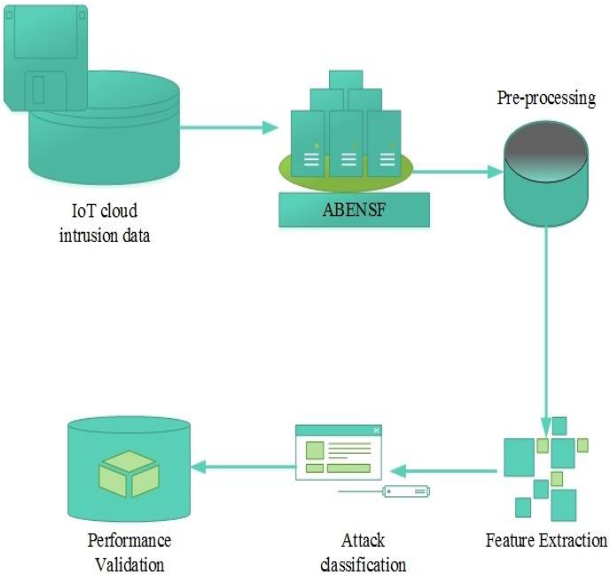

Figure 3. ABENSF Framework

In this research, IoT cloud intrusion data were chosen for the process. The collected data were given into the proposed model (ABENSF). Initially, pre-processing was done to remove the noise features. After removing the noise, feature extraction was done on the pre-processed data. Here the unwanted and meaningless features were extracted, and the good features were used for further process. In this investigation, artificial bee optimization processes were used for better attack prediction and neglection processes. Moreover, attack classifications happened after extracting the features. Consequently, the presentation of the implemented design was validated based on precision, accuracy, F-measure, recall, and error rate.

A. Design of ABENSF layers

The proposed model was designed based on the features of artificial bee optimization and the Elman neural network. The proposed model has five layers: the input layer, Hidden layer, classification layer, optimization layer, and output layer. At the input layer, the IoT-based intrusion data set were initialized; after that, the pre-processing was done in the hidden layer. Pre-processing was done to remove the noise data. Moreover, feature extraction was done in the classification layer. In that, the meaningless features were removed, and the better features were chosen and then compared with the fitness of the artificial bee. Based on the features, the outcome was displayed in the output layer of the model. Subsequently, Fig. 4 shows the layers of ABENSF.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 3
DOI: https://doi.org/10.17762/ijritcc.v11i3.6328
***Article Received**: 18 December 2022 **Revised**: 22 January 2023 **Accepted**: 10 February 2023*

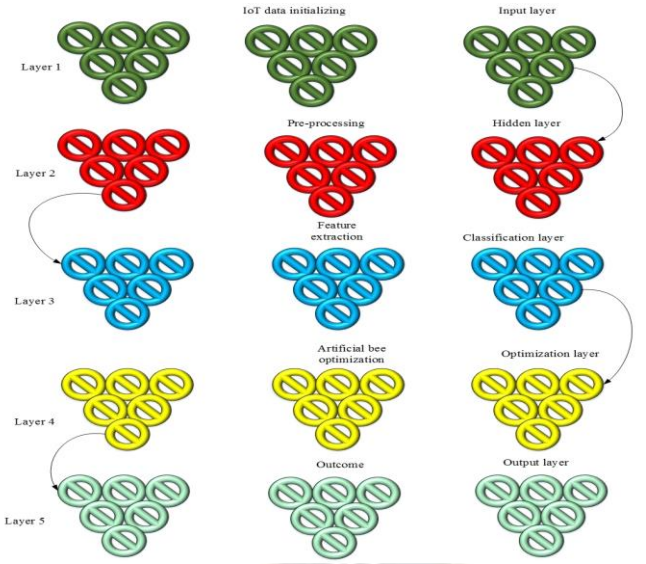

Figure 4. Layers of ABENSF

1) Pre-processing: Pre-processing was done to remove the noise data. However, pre-processing helps to reduce the computational complexities as well as to decrease the execution time. The noise data make the system slow and needs more time to execute the outcome. Before pre-processing, initially, the data set was initialized. Moreover, the data initialization of the proposed model was expressed in Eqn. (1),

$$D(i) = n(1,2,3.....j) \tag{1}$$

where $D(i)$ represents the initializing data function of the proposed model, n corresponds to normal data presented in the data set, and j defines the number of data present in the data set. After initialization, the system trained the data automatically. Moreover, pre-processing was done after initializing the data set. The initialized data set contains both the normal and the noise data. However, pre-processing of the proposed model was declared through Eqn. (2),

$$\rho(d) = D(i) - D^{*}(i) \tag{2}$$

Here, the parameter $\rho(d)$ defines the pre-processing function of the proposed model, $D^{*}(i)$ which was considered as the noise data presented in the initialized data. Consequently, $D(i)$ refers to the initialized data.

2) Feature Extraction: After pre-processing, feature tracking was happening within the present features. Moreover, the features, including normal and malicious activities, were detected initially. After that, the malicious features were neglected from the data. The feature tracking expression of the proposed model was declared in Eqn. (3),

$$T(f) = \delta[g, m(d)] \tag{3}$$

where $T(f)$ defines the feature tracking function of the proposed design moreover g corresponds to normal features without malicious activities and $m(d)$ refers to the features including malicious activities and δ was considered as the best fitness parameter of the artificial bee. After feature tracking, feature extraction was done to extract meaningful features. Subsequently, the extracted features were compared with the fitness function of an artificial bee. The feature extraction of the proposed model was declared through Eqn. (4),

$$E(f) = D(i) - z^{*} = \gamma_{ef} \tag{4}$$

where $E(f)$ it defines the feature extraction function of the proposed model, $D(i)$ represents the initialized data set, and z^{*} denotes the meaningless features. However, γ_{ef} denotes the extracted meaningful features.

3)Attack Detection and Prevention: The extracted features contain both the normal features as well as the malicious. For better security purposes, the malicious features were removed, and the normal features were chosen for further processing. For predicting the attack, the attack prediction expression was given in Eqn. (5),

$$\mathrm{A}(\gamma_{\mathrm{ef}}) = \begin{cases} \text{if } (\gamma_{\mathrm{ef}} = 0) & \text{Benign} \\ \text{if } (\gamma_{\mathrm{ef}} \neq 0) & \text{malicious} \end{cases} \tag{5}$$

Here, $A(\gamma_{ef})$ defines the attack prediction function of the model and γ_{ef} the extracted features. Moreover, the malicious features caused the system to slow down, and better results were not attained. So, here only normal features were chosen for the process. If the attack were presented in the features, then the system classifies that attack based on the types. So, at the classification phase predicted attack was classified. In the proposed model, the attack was classified based on the features, and the classification of attacks was such as DoS, Probe, R2L, and U2r. Subsequently, the classification of attack in the proposed model was expressed in (6),

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 3
DOI: https://doi.org/10.17762/ijritcc.v11i3.6328
Article Received: 18 December 2022 Revised: 22 January 2023 Accepted: 10 February 2023

$$\lambda_C = \begin{cases} if\left(A_f = -1\right) \; ; DoS \\ if\left(A_f = 1\right) \quad ; \; \Pr obe \\ if\left(A_f = 2\right) \quad ; R2L \\ if\left(A_f = -2\right) \; ; U2r \end{cases} \tag{6}$$

where λ_C refers to the classification function of the model and A_f represents the identified attack features. If the identified attack feature equals -1, it is classified as a DoS attack. If the attack features are equal to 1, it is predicted as a probing attack. Similarly, if the attack feature equals 2 and -2, it is classified as an R2L attack and U2r attack, respectively.

4) Monitoring Module

In the developed model, a monitoring mechanism is integrated to provide better security. The monitoring function continuously tracks malicious events in the system. Furthermore, it provides avoidance and prevention by neglecting the attacks from the system. The monitoring mechanism is expressed in Eqn. (7).

$$M_{nm} = \hbar\left(d_f - A_f\right) \tag{7}$$

Here, M_{nm} indicates the monitoring function, $\hbar$ denotes the monitoring variable and d_f represents the dataset features. Thus, the monitoring mechanism in the developed model removes the attack features present in the dataset. After known attack classification, an unknown attack is launched into the system to validate the security performance of the developed model. Here, a brute force attack is launched on the system to check the performance of the presented work. The tracking mechanism in the developed model continuously monitors and provides avoidance by neglecting the attacks (unknown attacks). Moreover, the developed model neglects unknown attacks and provides better security. Furthermore, the results are estimated in dual cases for performance validation.

Algorithm 1 ABENSF

Start
{
$D(i) = n(1,2,3.....j)$;
// initialize the IoT intrusion data set
Pre-processing ()
{
$\rho(d)$, $D(i), D^*(i)$
//initializing the pre-processing variables
$\rho(d) = D(i) - D^*(i)$
// training noise has been removed, here $\rho(d)$ *was considered as a pre-processing parameter of the proposed model*
}
Feature extraction ()
{
$\text{int}\, T(f), \delta, g, m(d)$
// initialize the feature extraction parameters
Feature tracking= $T(f) \leftarrow m(d)$
// Meaningless features were neglected, and the meaningful features were extracted
$\delta = g, m(d)$
// Consequently, meaningful better features were extracted.
}
Attack detection()
{
int $A(\gamma_{ef})$, γ_{ef}
//initialize the attack detecting parameters
If(γ_{ef} *=0)*
{
Benign
//here, benign function represents the normal features
} ***else (malicious)***
//hence, the attacks were predicted
Classification module ()
{
$if\,(A_f = 1)$
{
DoS attack; //if the attack features are equal to 1, it is predicted as a DoS attack
}
if ($A_f = -1)$
{
Probe attack; //if the attack features are equal to -1, it is predicted as a Probe attack
}
if ($A_f = 2)$
{
R2L attack; //if the attack features are equal to 2, it is predicted as an R2L attack
}
if ($A_f = -2)$
{
U2r attack; //if the attack features are equal to -2, it is predicted as a U2r attack
}
//Moreover, the attacks were classified
Attack prevention ()
{
$M_{nm} \xrightarrow{\hbar} \left(d_f - A_f\right)$
//Neglecting the attack features present in the dataset
}
}
//Unknown attack is launched to validate the performance

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 3
DOI: https://doi.org/10.17762/ijritcc.v11i3.6328
***Article Received**: 18 December 2022 **Revised**: 22 January 2023 **Accepted**: 10 February 2023*

of the system
Performance validation
}
stop

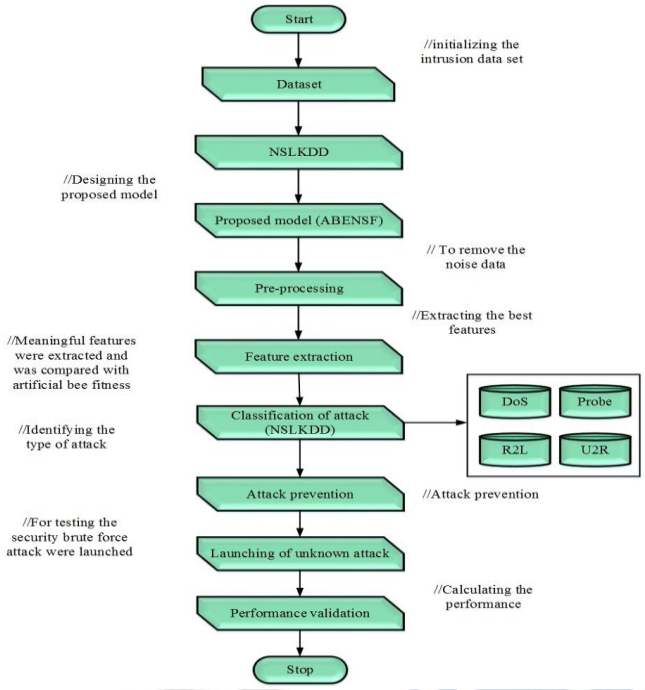

Figure 5. Workflow of ABENSF

Subsequently, the working process of the proposed model was explained in the pseudo-code format and was represented in algorithm1. Then, the workflow illustration of the newly developed model is shown in Fig. 5. At this work, the intrusion IoT data set was used. So, here is the NSLKDD data set that was chosen for this present research. After initializing the data set, the proposed model was designed to process the functions. Pre-processing was done to remove the noise features presented at the initialized data set. After that, feature extraction was done. From that, the meaningless features were extracted, and the meaningful features were used for further process. Subsequently, attack prediction was happening through the extracted meaningful features. If the attack was presented in those features, then the attacks were classified. Moreover, after classification, the attack was prevented for better outcomes and to increase security.

V. RESULT AND DISCUSSION

Here, the result of the implemented model was discussed. The proposed model was developed on a security basis. The presentation rate of the proposed model was measured as well as a comparison was made. The performance rate of the proposed model was high when compared to other existing models. The proposed model was designed in the python platform, running in a windows 10 environment. For designing, **the** NSLKDD dataset was used. Here, the dataset is split into 67% for training and 33% for testing. Through this research, the malicious features were removed, as well as provided better security to the system. The parameter required for execution was tabulated in the table 1.

TABLE 1. EXECUTION PARAMETERS

Parameter	specification
OS	Windows 10
Platform	Python
version	3.10
Application	Network
Datasets	NSL-KDD

A. Case Study

In the case study, the working procedure of the proposed model was explained in detail. The collected data set was pre-processed; after that, the meaningful features were extracted. To provide more security, attacks were detected with the extracted features. If the attack was presented in the features, then those were removed, and the attack was classified. Moreover, after classifying the attack, prevention of attack was done to avoid attack as well as to provide security. However, at last, one of the unknown attacks was launched to check the security. In this research, brute force attacks were launched to validate the security rate of the design. Consequently, the overall presentation of the system was measured in terms of accuracy, precision, recall, f-measure, and error rate. In this proposed model, three different protocols were obtained. A higher rate of people used the icmp protocol; more than 280000 users used the icmp protocol. More than 190000 users used the TCP protocol. A lower rate of users used the UDP protocol. UDP protocol was used by below 25000 users. Consequently, Fig. 6 depicts the different protocols used by the user. Many files related to testing and training functions were enclosed in the NSLKDD data set. In this research, the NSLKDD data set was used because for measuring the strength as well as to predict the intrusion characters.

In this proposed model, "0" was considered as the benign node, a normal node that does not contain any malicious nodes. The value "1" denotes the malicious nodes. Moreover, a Lower amount of malicious nodes were presented in the initialized data set. After pre-processing feature extraction, the rate of malicious nodes decreased, and the system provided more security to the files in the cloud storage. Node identification of the implemented design is shown in Fig. 7.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 3
DOI: https://doi.org/10.17762/ijritcc.v11i3.6328
***Article Received**: 18 December 2022 **Revised**: 22 January 2023 **Accepted**: 10 February 2023*

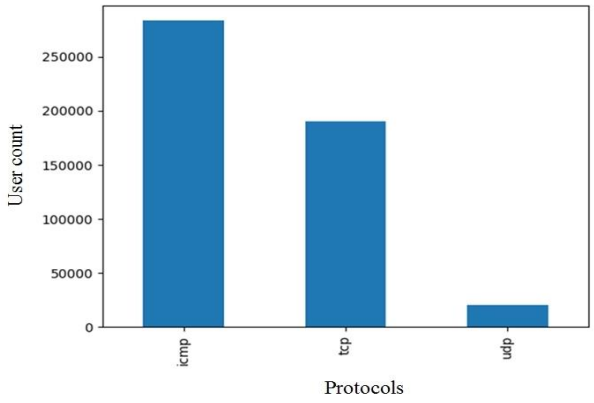

Figure 6. User count in different protocols

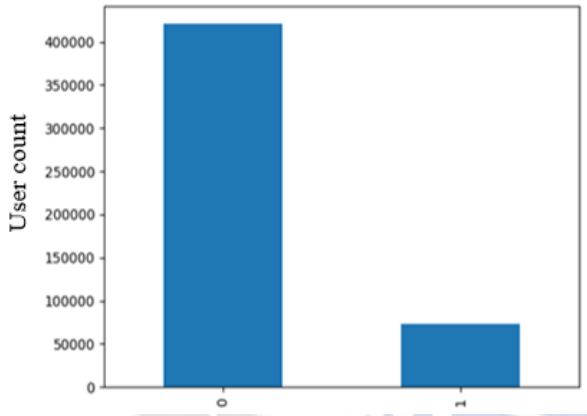

Figure 7. Identification of nodes (0-benign, 1-malicious)

The proposed model classifies the different types of attacks present in the NSLKDD data set. Moreover, the attacks were classified based on the features presented in the data set. The attacks were classified into DoS, normal, probe, r21, and u2r. The attack classification of the initialized data set is illustrated in Fig. 8.

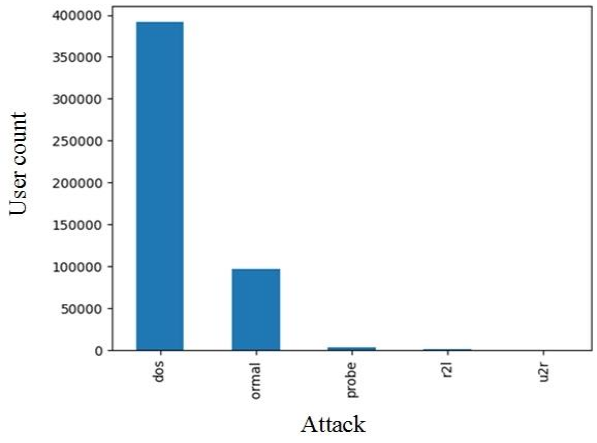

Figure 8. Attack classification

Minimizing the length of the data set leads to decreasing the validation loss. Here, the validation loss and the validation accuracy were calculated based on the iteration. Validation accuracy was increased up to 100% at 100 iterations. Moreover, the validation loss was increased initially after that decreasing and reached 0 levels. Subsequently, the validation loss and accuracy were illustrated in Fig. 9.

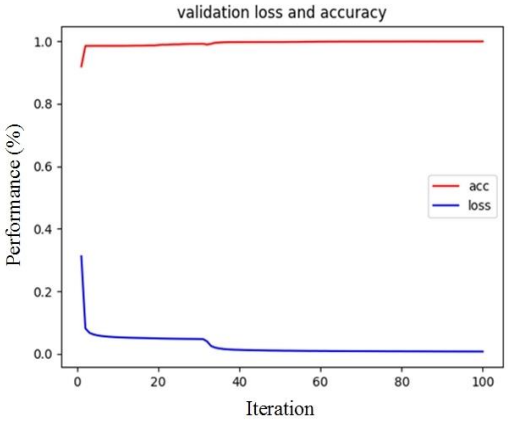

Figure 9. Validation loss and accuracy

The confusion matrix of the proposed design was used to validate the classification process in every predicted phase. The confusion matrix was validated for classifying the user's file type. Moreover, the obtained confusion matrix of the NSL-KDD data is shown in Fig. 10.

Figure 10. Confusion matrix

The correlation matrix was used to recognize the exact range of attacks present in the large data set. Fig. 11 depicts the correlation matrix of the proposed model. The overall features presented in the dataset were correlated in the correlation matrix. The higher correlated features were neglected from the correlation matrix.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 3
DOI: https://doi.org/10.17762/ijritcc.v11i3.6328
***Article Received**: 18 December 2022 **Revised**: 22 January 2023 **Accepted**: 10 February 2023*

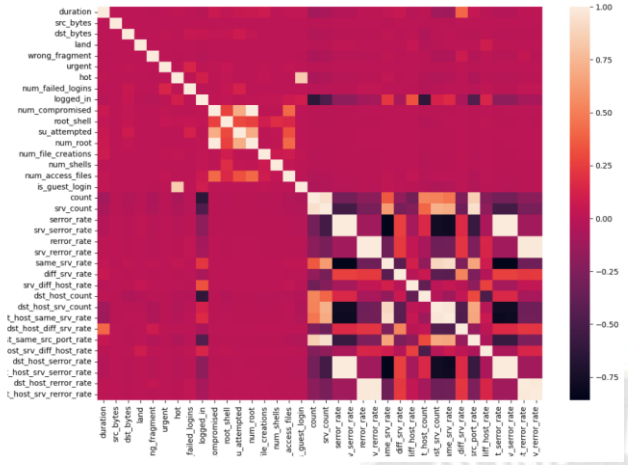

Figure 11. Correlation matrix

Moreover, the overall presentation of the implemented design before and after the attack is shown in Fig. 12. The graph shows the increased performance rate of the proposed model in terms of recall, accuracy, precision, and f-measure and error rate. A higher rate of performance was attained before attack launching, and slight variations arose after attack launching

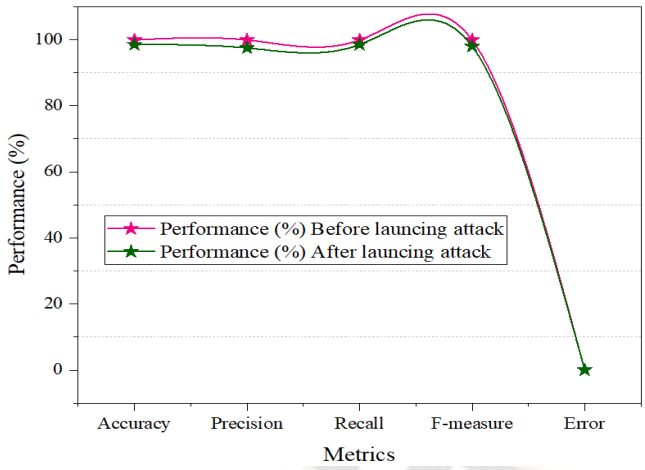

Figure 12. Overall performance of ABENSF

B. Performance & Comparison Analysis

In comparison analysis, the performance rate of the developed design was compared to the other existing models. Moreover, here the accuracy, recall, precision, and f-measure rate of the implemented design were compared to existing models. The accuracy rate of the implemented model was compared to the existing model such as Convolution Neural Network + Long Short Term Memory (CNNLSTM) [27], Bi-directional Long Short Term Memory (BLSTM) [27], Ensemble [27], Classification and Regression Tree (CART) [27], Multi-Layer Perception (MLP) [27]. Subsequently, the precision, recall as well as F-measure rate of the implemented model were compared to the existing models such as Support Vector Machine (SVM) [28], perception [28], Association Rule Mining (ARM) [28], Decision Tree (DT) [28], Nave Bayes (NB) [28]. Among them, the proposed model gives a better rate of parameters.

1)Accuracy comparison: Among them, the proposed model attained 99.86% accuracy before and 98.44% after the attack. It was high compared with the existing models. Subsequently, in this proposed model, the accuracy rate depends on security. The proposed model provides more security. Thus, the accuracy rate is also high. The accuracy of the proposed model was expressed in Eqn. (8),

$$a = \frac{\mathrm{A}+\mathrm{B}}{\mathrm{A}+C+\mathrm{B}+D} \tag{8}$$

where a refers to the accuracy parameter of the proposed model. The amount of true positive was termed as, A and true negative was B . Moreover, the false positive rate was defined as C the parameter D corresponding to a false negative. The accuracy comparison of the proposed model is shown in Fig. 13. Accuracy rate of the implemented model was calculated based on the number of positives and negatives.

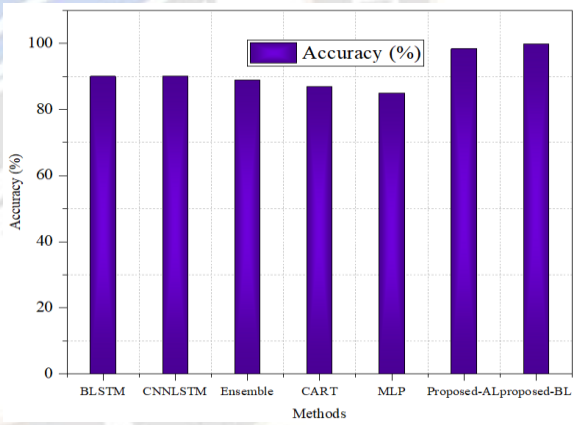

Figure 13. Accuracy comparison

The amount of correctness was referred to as accuracy, and here the accuracy rate of the proposed model was compared to other existing techniques such as BLSTM, CNNLSTM, Ensemble, and CART.

2) Recall: Based on the rate of true positive and false negative, rate recall was calculated. In the proposed model, the recall was based on the security rate. Subsequently, the proposed model provides better security so that the system provides better recall and accuracy rates. The recall rate of the proposed model was expressed in Eqn. (9),

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 3
DOI: https://doi.org/10.17762/ijritcc.v11i3.6328
***Article Received**: 18 December 2022 **Revised**: 22 January 2023 **Accepted**: 10 February 2023*

$$r = \frac{\mathrm{A}}{\mathrm{A} + D} \tag{9}$$

where r corresponds to the recall parameter of the proposed model. The recall rate of the implemented design was about 98.44% after launching the attack and attaining 99.86% of recall before the attack launching. Subsequently, the recall of the proposed model was compared with the existing models such as SVM, ARM, DT, and NB. Among them, the proposed model attains a better recall rate.

The recall rate of the SVM model was about 88%, the recall rate of the ARM model was 89%, the DT model achieved 94% of recall, and the Nave Bayes model reached a recall rate was about 92%. The proposed model attains a better recall rate when compared with the above-mentioned model. A recall comparison of the implemented model is shown in Fig. 14.

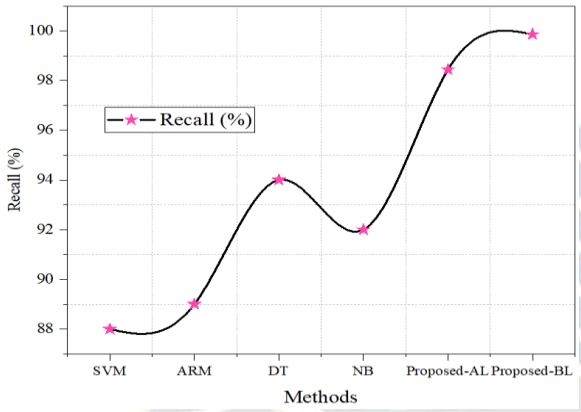

Figure 14. Recall comparison

3) Precision: The detection rate was termed precision. Moreover, the attack detection of the proposed model was mentioned as a precision rate. The rate of precision was calculated based on the true and false prediction rates of the proposed model. The precision of the proposed model was expressed in Eqn. (10),

$$p = \frac{\mathrm{A}}{\mathrm{A} + C} \tag{10}$$

Here, the parameter p defines the precision rate of the model. Here for measuring the precision rate, the true positive and false positive rate was chosen. Here the precision rate of the proposed model was about 99.85% before attack launching, and 97.45% of accuracy was achieved after launching the attack. The proposed model's precision rate was compared with the existing models, such as perception, ARM, DT, and NB. Moreover, a higher rate of precision was attained through the proposed model. A precision comparison of the model is illustrated in Fig. 15.

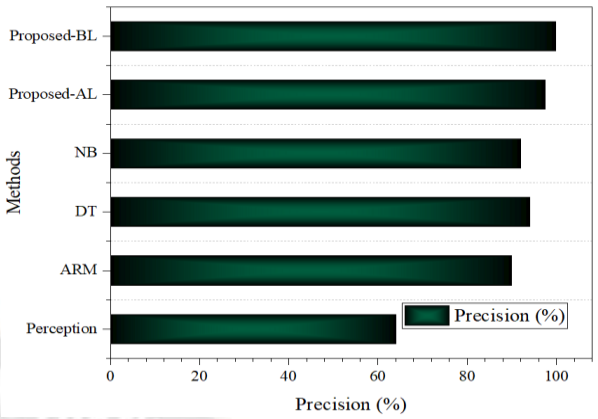

Figure 15. Precision comparison

4)F-measure: F-measure was calculated based on the precision and recall rates of the developed design. However, the proposed model attained a better f-measure rate before the attack, about 99.85%, and reached a 97.92% f-measure rate after launching the attack. The F-measure of the proposed model was expressed in Eqn. (11),

$$\lambda_f = 2 \times \frac{p \times r}{p + r} \tag{11}$$

Here, λ_f defines the parameter used to calculate the f-measure rate. In this proposed model, the f-measure rate of the proposed model was compared to other existing models, such as SVM, ARM, DT, and NB models. Among them, the proposed model achieves a better f-measure rate than the existing models. Fig.16 shows the f-measure analysis of the developed design.

B. Discussion

In the discussion, the presentation rates of the implemented design were discussed. The proposed model attained a better rate of recall, accuracy, precision, and f-measure, and a lower rate of error was attained. Hence, the overall presentation of the proposed model before and after launching the brute force attack is shown in table 2.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 3
DOI: https://doi.org/10.17762/ijritcc.v11i3.6328
***Article Received**: 18 December 2022 **Revised**: 22 January 2023 **Accepted**: 10 February 2023*

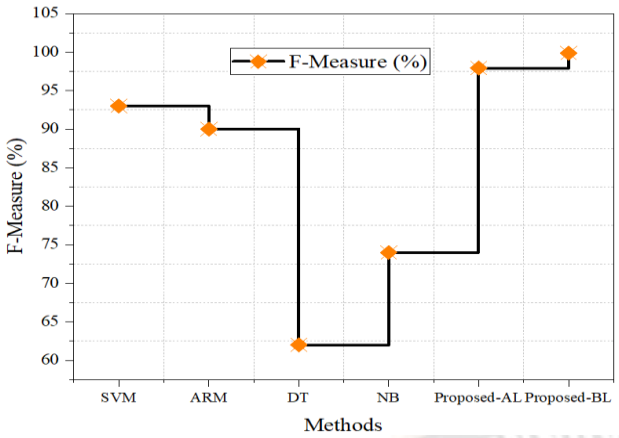

Figure 16. Comparison of f-measure

TABLE 2. OVERALL PERFORMANCE OF ABENSF

Parameters	Performance (%) Before launching a brute force attack	Performance (%) After launching a brute force attack
Precision	99.85	97.45
F-Measure	99.85	97.92
Accuracy	99.86	98.44
Recall	99.86	98.44
Error	0.0009	0.015

VI. CONCLUSION

In this research, ABENSF intrusion detection was developed for security purposes. The presented work was validated with an intrusion detection dataset (NSL-KDD). Initially, the dataset was imported into the system and filtered using the pre-processing function. Consequently, feature extraction was performed to extract the attack features. The artificial bee fitness function in the developed model tracks and extracts the attack features optimally. Moreover, a monitoring mechanism was incorporated into the presented work to prevent attacks and provide intrusion avoidance. Furthermore, the robustness of the developed model was verified by launching an unknown attack (Brute Force Attack). Finally, the outcome parameters were estimated in dual cases and validated with a comparative assessment. Moreover, the parameter improvement score was also estimated in the comparative analysis. It is observed that in the developed model, the accuracy is improved by 4%, the f-measure is enhanced by 4%, the recall percentage is increased by 4%, and the precision is enhanced by 5%. Consequently, the developed model has attained a very low error rate of 0.0009%. Thus, the developed model improves network security and provides intrusion avoidance by neglecting the attacks.

ACKNOWLEDGEMENT

None.

REFERENCES

[1] B. Alouffi, M. Hasnain, A. Alharbi, W. Alosaimi, H. Alyami, and M. Ayaz, "A Systematic Literature Review on Cloud Computing Security: Threats and Mitigation Strategies," IEEE Access, Vol. 9, pp. 57792-57807, 2021. https://doi.org/10.1109/ACCESS.2021.3073203.

[2] F. Muheidat, and L. Tawalbeh, "Mobile and Cloud Computing Security", In: Y. Maleh, M. Shojafar, M. Alazab, and Y. Baddi, (eds.) "Machine Intelligence and Big Data Analytics for Cybersecurity Applications," Studies in Computational Intelligence, vol. 919, 2021. Springer, Cham. https://doi.org/10.1007/978-3-030-57024-8_21.

[3] L.M. Dang, M.J. Piran, D. Han, K. Min, and H. Moon, "A Survey on Internet of Things and Cloud Computing for Healthcare," Electronics, Vol. 8, pp. 768, 2019. https://doi.org/10.3390/electronics8070768.

[4] F. Nadeem, "Evaluating and Ranking Cloud IaaS, PaaS and SaaS Models Based on Functional and Non-Functional Key Performance Indicators," IEEE Access, Vol. 10, pp. 63245-63257, 2022. https://doi.org/10.1109/ACCESS.2022.3182688.

[5] F. Shokri Habashi, S. Yousefi, and B. Ghalebsaz Jeddi, "Resource allocation mechanisms for maximizing provider's revenue in infrastructure as a service (IaaS) cloud," Cluster Computing, Vol. 24, pp. 2407–2423, 2021. https://doi.org/10.1007/s10586-021-03262-y.

[6] A. Olmsted, "Platform As A Service Development Cost & Security," International Journal of Intelligent Systems, Vol. 10, pp. 3-4, 2017.

[7] C. Hanane, A. Battou, and O. Baz, "Performance Security in Distributed System: Comparative Study," International Journal of Computer Applications, Vol. 179, pp. 15, 2018.

[8] R. Gupta, S. Verma, and K. Janjua, "Custom Application Development in Cloud Environment: Using Salesforce," 2018 4th International Conference on Computing Sciences (ICCS), pp. 23-27, 2018. https://doi.org/10.1109/ICCS.2018.00010.

[9] I. Odun-Ayo, M. Ananya, F. Agono, and R. Goddy-Worlu, "Cloud Computing Architecture: A Critical Analysis," 2018 18th International Conference on Computational Science and Applications (ICCSA), pp. 1-7. https://doi.org/10.1109/ICCSA.2018.8439638.

[10] J.B. Wang, J. Wang, Y. Wu, J.Y. Wang, H. Zhu, M. Lin, and J. Wang, "A Machine Learning Framework for Resource Allocation Assisted by Cloud Computing," IEEE Network, Vol. 32, No. 2, pp. 144-151, 2018. https://doi.org/10.1109/MNET.2018.1700293.

[11] Z. Wang, S. Yang, X. Xiang, A. Vasilijević, N. Mišković, and D. Nađ, "Cloud-based mission control of USV fleet: Architecture, implementation and experiments," Control Engineering Practice, Vol. 106, pp. 104-657, 2021. https://doi.org/10.1016/j.conengprac.2020.104657.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 3
DOI: https://doi.org/10.17762/ijritcc.v11i3.6328
***Article Received**: 18 December 2022 **Revised**: 22 January 2023 **Accepted**: 10 February 2023*

[12] L. Hong-tan, K. Cui-hua, B. Muthu, and C.B. Sivaparthipan, "Big data and ambient intelligence in IoT-based wireless student health monitoring system," Aggression and Violent Behavior, pp.101-601, 2021. https://doi.org/10.1016/j.avb.2021.101601.

[13] C. Chen, B. Liu, S. Wan, P. Qiao, and Q. Pei, "An Edge Traffic Flow Detection Scheme Based on Deep Learning in an Intelligent Transportation System," IEEE Transactions on Intelligent Transportation Systems, Vol. 22, Ch. 3, pp. 1840-1852, 2020. https://doi.org/10.1109/TITS.2020.3025687.

[14] T.M. Ghazal, "Positioning of UAV Base Stations Using 5G and Beyond Networks for IoMT Applications," Arabian Journal for Science and Engineering, https://doi.org/10.1007/s13369-021-05985-x.

[15] K. Yu, L. Lin, M. Alazab, L. Tan, and B. Gu, "Deep Learning-Based Traffic Safety Solution for a Mixture of Autonomous and Manual Vehicles in a 5G-Enabled Intelligent Transportation System," IEEE Transactions on Intelligent Transportation Systems, Vol. 22 Ch. 7, pp. 4337-4347, 2020. https://doi.org/10.1109/TITS.2020.3042504.

[16] J.B. Wang, J. Wang, Y. Wu, J.Y. Wang, H. Zhu, M. Lin, and J. Wang, "A Machine Learning Framework for Resource Allocation Assisted by Cloud Computing," IEEE Network, Vol. 32, Ch. 2, pp. 144-151, 2018. https://doi.org/10.1109/MNET.2018.1700293.

[17] O. Azeroual, M. Jha, A. Nikiforova, K. Sha, M. Alsmirat, and S. Jha, "A Record Linkage-Based Data Deduplication Framework with DataCleaner Extension," Multimodal Technologies and Interaction, Vol. 6, Ch. 27, 2022. https://doi.org/10.3390/mti6040027.

[18] B. Richhariya, M. Tanveer, and A.H. Rashid, "Diagnosis of Alzheimer's disease using universum support vector machine based recursive feature elimination (USVM-RFE)," Biomedical Signal Processing and Control, Vol. 59, pp. 101-903, 2020. https://doi.org/10.1016/j.bspc.2020.101903.

[19] D. Han, S. Li, Y. Peng, and Z. Chen, "Energy Sharing-Based Energy and User Joint Allocation Method in Heterogeneous Network," IEEE Access, Vol.8, pp. 37077-37086, 2020. https://doi.org/10.1109/ACCESS.2020.2975293.

[20] S. Singh, and H.S. Saini, "Intelligent Ad-Hoc-On Demand Multipath Distance Vector for Wormhole Attack in Clustered WSN," Wireless Personal Communications, Vol. 122, pp. 1305–1327, 2022. https://doi.org/10.1007/s11277-021-08950-x.

[21] P.A. Abdalla, and A. Varol, "Advantages to Disadvantages of Cloud Computing for Small-Sized Business," 2019 7th International Symposium on Digital Forensics and Security (ISDFS), pp. 1-6, 2019. https://doi.org/10.1109/ISDFS.2019.8757549.

[22] P.M. Khilar, V. Chaudhari, and R.R. Swain, "Trust-Based Access Control in Cloud Computing Using Machine Learning", In: H. Das, R. Barik, H. Dubey, and D. Roy, (eds.) "Cloud Computing for Geospatial Big Data Analytics," Studies in Big Data, vol. 49, 2019. Springer, Cham. https://doi.org/10.1007/978-3-030-03359-0_3.

[23] S. Gill, "Predicting the growth and trend of COVID-19 pandemic using machine learning and cloud computing," Internet of Things, Vol. 11, pp.100-222, 2020. https://doi.org/10.1016/j.iot.2020.100222.

[24] A. Abdelaziz, M. Elhoseny, A.S. Salam, and A.M. Riad, "A machine learning model for improving healthcare services on cloud computing environment," Measurement, Vol. 119, pp. 117-128, 2018. https://doi.org/10.1016/j.measurement.2018.01.022.

[25] J. Gao, H. Wang, and H. Shen, "Machine Learning Based Workload Prediction in Cloud Computing," 2020 29th International Conference on Computer Communications and Networks (ICCCN), pp. 1-9, https://doi.org/10.1109/ICCCN49398.2020.9209730.

[26] C.L. Stergiou, A.P. Plageras, K.E. Psannis, and B.B. Gupta, "Secure Machine Learning Scenario from Big Data in Cloud Computing via Internet of Things Network", In: B. Gupta, G. Perez, D. Agrawal, and D. Gupta, (eds.) Handbook of Computer Networks and Cyber Security, Springer, Cham. https://doi.org/10.1007/978-3-030-22277-2_21.

[27] B. Jothi, and M. Pushpalatha, "WILS-TRS — a novel optimized deep learning based intrusion detection framework for IoT networks," Personal and Ubiquitous Computing, 2021, https://doi.org/10.1007/s00779-021-01578-5.

[28] N. Koroniotis, N. Moustafa, and E. Sitnikova, "A new network forensic framework based on deep learning for Internet of Things networks: A particle deep framework," Future Generation Computer Systems, Vol. 110, pp. 91-106, 2020. https://doi.org/10.1016/j.future.2020.03.042.