Enhancing Cloud Security with Deep Learning - An ANN Approach for Cyber Threat Detection

Mothe Sathyam Reddy[1,*], Gattu Thanuja[2,**], Bojja Sobharani[3,***], Thalapaneni Suma Durga[4,****], Dr. Rama Krishna Eluri[5,†], and Dodda Venkat Reddy[6,‡]

[1]Assistant Professor, Department of CSE, Narasaraopeta Engineering College, Narasaraopeta, Palanadu, Andhra Pradesh, India
[2]Department of CSE, Narasaraopeta Engineering College, Narasaraopeta, Palanadu, Andhra Pradesh, India
[3]Department of CSE, Narasaraopeta Engineering College, Narasaraopeta, Palanadu, Andhra Pradesh, India
[4]Department of CSE, Narasaraopeta Engineering College, Narasaraopeta, Palanadu, Andhra Pradesh, India
[5]Professor, Department of CSE, Narasaraopeta Engineering College, Narasaraopeta, Palanadu, Andhra Pradesh, India
[6]Assistant Professor, Department of CSE, Narasaraopeta Engineering College, Narasaraopeta, Palanadu, Andhra Pradesh, India

Abstract. As cloud computing becomes more and more fundamental to digital infrastructure, safety features are important to state the cyber threats. This paper presents an technique to improving cloud protection using deep gaining knowledge mainly Artificial Neural Networks (ANNs). We leverage ANN algorithms along with Levenberg-Marquardt, Scaled Conjugate Gradient, and Bayesian Regularization to increase an advanced hazard detection gadget able to figuring out complicated attack patterns in cloud environments. Our research focuses on detecting various cyber threats, along with malware, phishing, and Distributed Denial of Service (DDoS) attacks.Finally, the incorporation of ANNs and other deep learning techniques can be advantageous to cloud security frameworks as they offer a novel approach to cyberthreat identification. Our study demonstrates how ANN-based models for quick and efficient threat identification can improve cloud security.
Keywords: Cloud computing, cyber threat detection, Deep Learning, Levenberg-Marquardt algorithm, scale conjugate gradient, Bayesian Regularization, Artificial Neural Networks (ANNs).

1 INTRODUCTION

The emergence of cloud computing has fundamentally altered how people and organizations store, manage, and access information. It provides previously unheard of scale and flexibil-

[*]e-mail: sathyamreddym@gmail.com
[**]e-mail: gattuthanuja23@gmail.com
[***]e-mail: bojjasobha321@gmail.com
[****]e-mail: sumadurgatalapaneni@gmail.com
[†]e-mail: ramakrishnaephd7@gmail.com
[‡]e-mail: doddavenkatareddy@gmail.com

ity[1]. Because cloud systems are centralised, they are attractive targets for cybercriminals, who are always coming up with more complex and varied ways to take advantage of weaknesses.As a result, new adaptive, intelligent security responses that can catch and neutralize threats in real time are not readily available. Ability to improve cloud system cyber security. In particular, artificial neural networks (ANNs) have shown remarkable ability to detect and identify complex patterns in large data sets[2].They are particularly adept at both known and unknown threats because of their ability to adapt and learn from past attacks[2]. This paper investigates the application of deep mastering strategies, that specialize in ANNs[2], to decorate cloud safety through superior cyber threat detection.

2 LITERATURE SURVEY

In addition to providing flexibility, scalability, and performance comparable to standard not exceptional performance, cloud computing also adds security risks such as virus attacks, insider threats, and data breaches[9]. Unauthorised access to sensitive cloud records is one kind of a data breach that can cause harm to one's finances and reputation. Malicious software programs infiltrate cloud infrastructure, jeopardising agency availability and information integrity. This is the cause of malware attacks. Insider threats originate from criminal clients who abuse their access to steal data or interfere with business operations. Traditional protection capabilities, like firewalls and antivirus software, are regularly inadequate to cope with those evolving threats as they will be inclined to be reactive and might not discover superior or zero-day attacks in real-time [1].
Deep mastering, a subset of device studying, shows great capability in improving cybersecurity. Models like Artificial Neural Networks (ANNs) can pick out complex styles in massive datasets, making them well-suitable for detecting anomalies and predicting functionality threats. Recurrent Neural Networks (RNNs),[7] together with Long Short-Term Memory (LSTM) networks, excel at reading facts sequences, consisting of network site visitors through the years. Convolutional Neural Networks (CNNs), historically used for photo and video evaluation, also are effective in detecting malicious activities in network traffic. Studies show that deep getting to know models normally outperform traditional gadget mastering and rule-primarily based structures, reaching higher detection prices and less false positives [11]. Despite their promise, implementing deep learning in cybersecurity faces disturbing situations. These models require big amounts of top notch statistics for training, which can be difficult to attain and label. Training is computationally in depth, necessitating effective hardware and optimized algorithms[2]. While deep mastering can lessen fake positives, retaining a balance amongst detection prices and faux positives is essential. Staying updated with the modern developments in cyber dangers calls for ordinary updates and retraining. In order to fully realise their potential in cloud safety, future research need to focus on improving records amassing and preprocessing[4], refining deep reading algorithms, and incorporating such designs into whole cybersecurity frameworks [10].

3 DATA COLLECTION

Data can be collected from datasets available on Kaggle.com. The datasets contain data on cyber threats, system logs, network traffic, malware detection, intrusion detection, and anamoly detection.CICIDS 2017[4],this dataset covers a extensive range of community attack situations. UNSW-NB15[4],this dataset also provides well known records, which is to differentiate malicious packages from malicious programs.NSL-KDD[4],this is a received dataset name of intrusion detection,it is optimized to eliminate unnecessary records,making it a reliable method of testing.

4 PREPROCESSING

Preprocessing can be done in two steps i.e, handling null values and normalization. During the normalization process, we must choose which category data to convert to numerical data utilizing label or one hot encoding[5].

4.1 Label Encoding

If a categorical feature (Xj) has (n) unique categories, label encoding maps each category to a unique integer value. This can be represented as:

$$X_j^{(\text{encoded})} = f(X_j) \tag{1}$$

Where f is a bijective function mapping categories to integers.

4.2 Feature Scaling

For feature scaling using standardization (z-score normalization), the formula applied to each feature Xj is:

$$X_i^{(\text{scaled})} = \frac{X_i - \mu_i}{\sigma_i} \tag{2}$$

Where:

- μ_i is the mean of feature i in the training set.
- σ_i is the standard deviation of feature i in the training set.

5 PROPOSED ANN MODEL

To decorate cloud protection the use of deep getting to know through an Artificial Neural Network (ANN) method.This begins with the cautious selection and instruction of facts from the significant CICIDS 2017 and UNSW-NB15 datasets. Given the large quantity of records, sampling 10percent of the preprocessed dataset. The data must be divided into training, testing, and validation sets. Thirty percent of the data is for testing and validation, while the remaining seventy percent is for training.The ANN model can be trained by using the three algorithms: Levenberg-Marquardt,Scale conjugate gradient and Bayesian Regularization algorithms.

5.1 Levenberg-Marquart algorithm:

This optimization method combines gradient descent and Gauss-Newton method dynamically determines teaching Productivity.It can be defined as:

$$\theta_{k+1} = \theta_k - \left[J^T(\theta_k)J(\theta_k) + \lambda I\right]^{-1} J^T(\theta_k)r(\theta_k) \tag{3}$$

Where:

- θ_k represents the parameters at iteration k

- $J(\theta_k)$ is the Jacobian matrix of partial derivatives
- $r(\theta_k)$ is the residual vector
- λ is the damping factor
- I is the identity matrix

Table 1. Comparing the ANN model with other models

SNo	Model	Accuracy
1	ANN	90.20
2	CNN	86.58
3	RNN	87.84
4	LSTM	79.00
5	AutoEncoders	86.01
6	SVM (Original)	84.74
7	SVM (Encoded)	84.95

The table-1 shows about the different model accuraries.The ANN model gives the best accuracy when compared with the other models like CNN, RNN, LSTM, AutoEncoders, SVM(Original), SVM(Encoded).

5.2 Bayesian Regularization:

Bayesian Regularization employs a probabilistic framework around the model training procedure and is useful when dealing with neural networks, mainly to avoid model overfitting. The cost function E in Bayesian Regularization can be represented as:

$$E = \alpha E_D + \beta E_W \tag{4}$$

where:

- E is the total cost function.
- E_D is the data error term.
- E_W is the weight regularization term.
- α and β are regularization parameters controlling the trade-off between the error term and the regularization term.

5.3 Scale Conjugate Gradient Algorithm:

This technique is a variant of the conjugate gradient technique, scaled to improve convergence pace and stability. The key formula representing the Scaled Conjugate Gradient (SCG) algorithm in a single expression for updating the weights in a neural network is:

$$\mathbf{W}_{k+1} = \mathbf{W}_k + \lambda_k \mathbf{P}_k \tag{5}$$

where:

- $\mathbf{W}_k$ is the current weight vector at iteration k.
- λ_k is the step size (scaling factor) determined by the algorithm.
- $\mathbf{P}_k$ is the search direction.

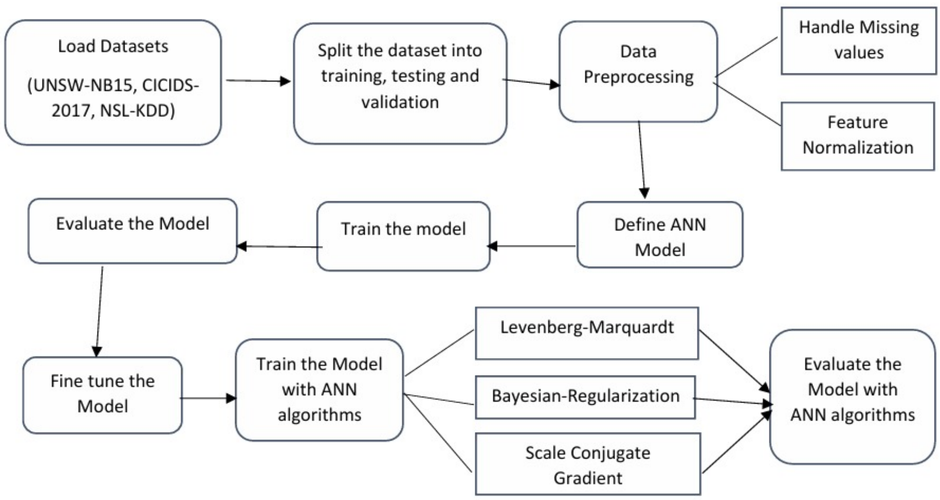

Figure 1. Proposed Artificial Neural Networks Model Architecture

5.4 Feed-forward Neural Network:

A feed-forward neural network with L layers can be represented as a series of matrix multiplications and activation functions.For each layer l from 1 to L:

$$c^{[l]} = g^{[l]}(W^{[l]} \cdot c^{[l-1]} + d^{[l]}) \tag{6}$$

where:

- $c^{[l]}$ is the activation of layer l,
- $W^{[l]}$ and $d^{[l]}$ are the weight matrix and bias vector for layer l,
- $g^{[l]}$ is the activation function (e.g., ReLU, sigmoid).

6 Algorithm for the Proposed Model

6.1 Preparation of Information:

1. The Collection of Data Set and its separation into training, test, and validation subsets are explained in the first subsection. Bring in the datasets for CICIDS 2017 and UNSW-NB15.

2. Missing Data: Complete the incomplete data on the data set, usually by using the mean in the numeric columns.

3. Categorical features: Convert categorical values into numerical values using label encoding and other such features.

6.2 Model Definition:

The Articulation of the Neural Network:

1. Input Layer: The number of features in the dataset corresponds to the number of input nodes.

2. Hidden Layers: Initialize one or several hidden layers, specifying how many neurons and which activation function will be applied (for example ReLU).

3. Output Layer: In the case of a binary classification problem, the output layer is made up of a single neuron and has sigmoid activation.

6.3 Model Compilation:

1. Optimizer: Apply any of the optimizing algorithm to find the minima of the loss, for instance Adam optimizer.

2. Loss Function: In the case of binary classification problems, the loss function will employ the use of binary cross entropy.

3. Metrics: Select the accuracy as the intended performance metric for the model evaluation.

6.4 Model Training:

1. Fit the Model: Execute the training of the artificial neural networks by the training set for a diverse number of epochs and batch size. In the course of this training, a hold-out sample is employed to measure the efficiency of the model on an unseen sample.

6.5 Model Evaluation:

1. Evaluate the Model: Conduct the evaluation of the constructed model on the defining test set to assess the degree of accuracy in the tested model.

6.6 Model Output:

1. Display the Accuracy: The accuracy is expressed as a percentage where correctly classified instances make up a fraction of the test set.

6.7 Save the Model:

1. Saving the trained model: Saving the complete model comprising architecture, defined loss functions and optimizer states makes comprehensive use of the Keras framework.

6.8 Display the Model Architecture:

1. Summary of the Model: Features of the model architecture are summarized suggesting the layers, output shapes and number of parameters.

6.9 Fine-Tune the Model:

1. Modify Hyperparameters: Model performance optimization can be done by changing the learning rate, optimizer type, number of neurons, number of layers, or activation functions among others.
2. Retrain Specific Layers: The entire model can also be retrained or modified by re-configuring selectively some layers in relation to fine-tuning.
3. Recompile and Retrain: If alteration of the model has occurred, it is recompilated and is re-trained on the available data set.
4. Evaluate Post Fine-Tuning: Finally, the model is once again compared to the test set with the intention of verifying how much improvement has been achieved.

7 Performance Evaluation

The following graphs and tables are the performance metrics used in our ANN model.

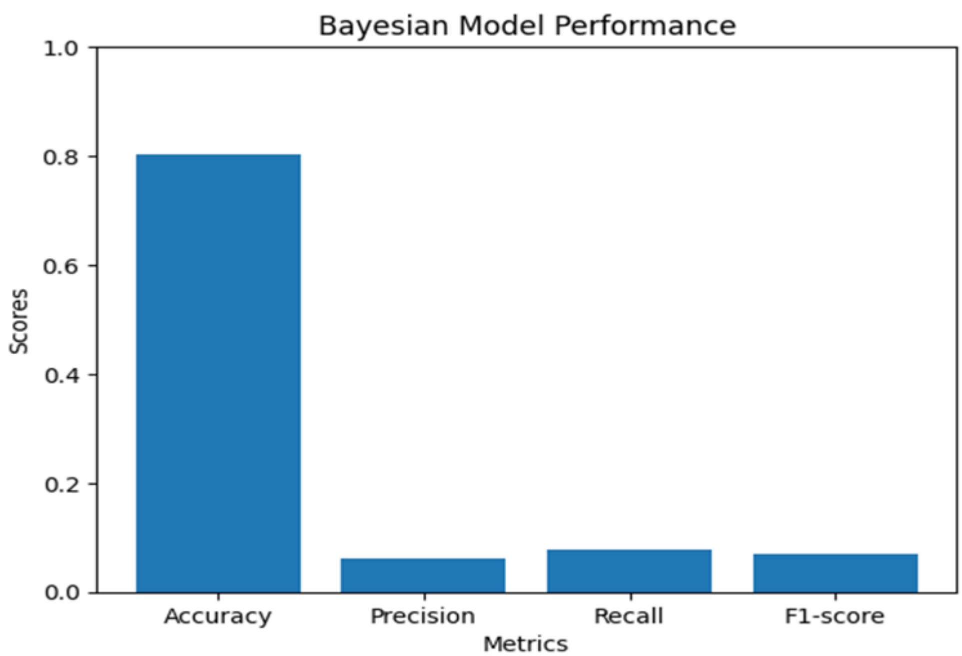

Figure 2. Performance metrics for ANN Model

7.1 Binary Conversion

The model's output is a probability value, which is converted to a binary prediction based on a threshold of 0.5. It is defined as:

$$y_{\text{pred\_binary}} = \begin{cases} 1 & \text{if } y_{\text{pred}} > 0.5 \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

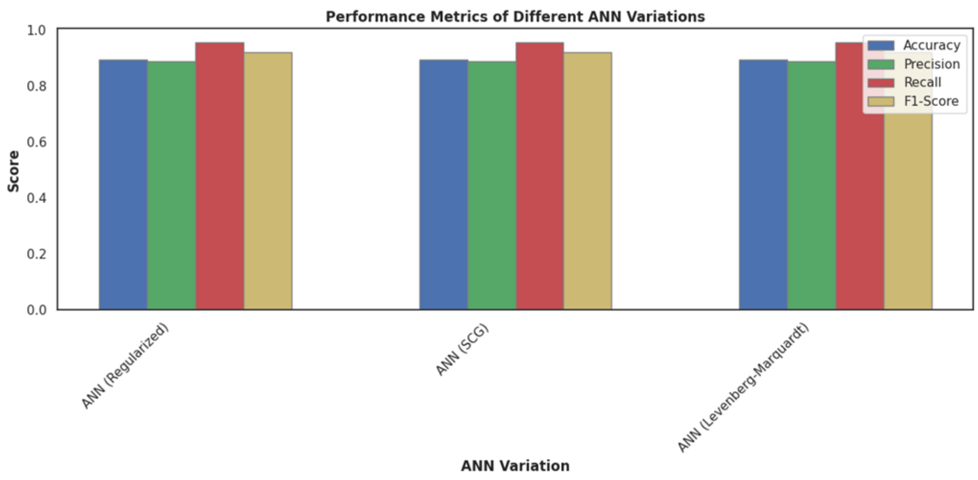

Figure 3. Metrics for the algorithms used in ANN Model

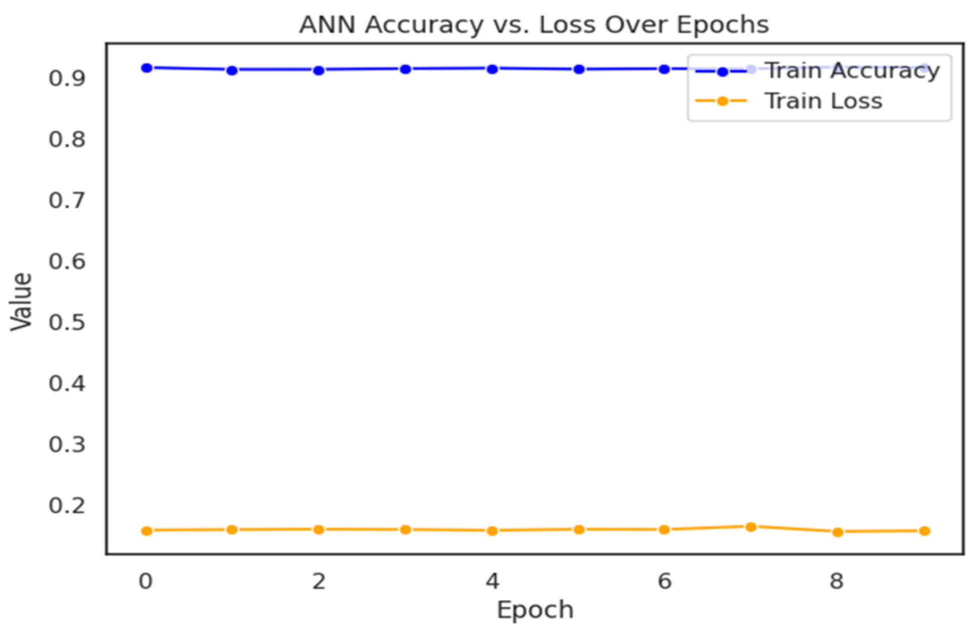

Figure 4. UNSW-NB15 accuracy versus Loss

where

- y_{pred} is the predicted probability.

The Table 2 shows the amount of data taken in the UNSW-NB15 dataset. It includes the training data, testing data, and validation data.

We can compare with the other models ANN gives best results than other models. ANN model contains the training parameters of 13,253 and total number of optimizer params is

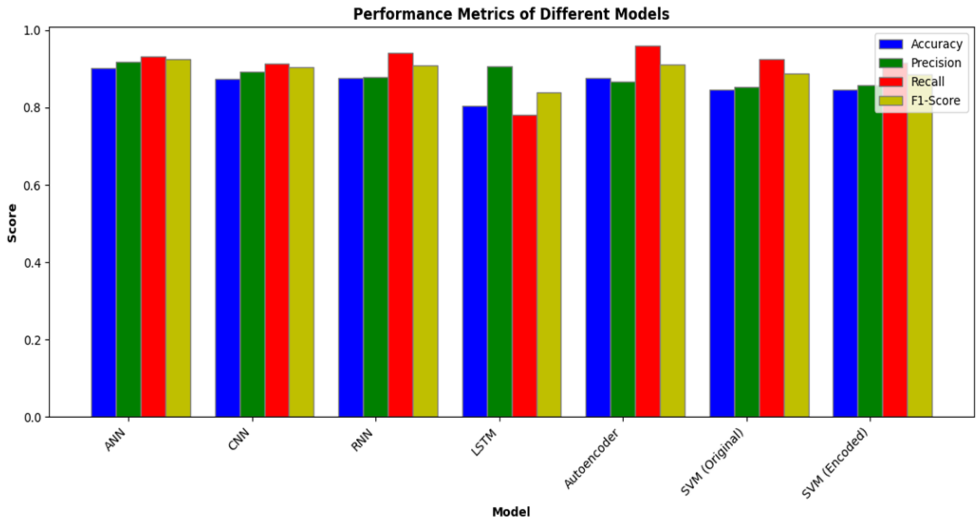

Figure 5. Performance metrics for the models

Table 2. Dataset Information (UNSW-NB15)

SNo	Data	Total Records
1	Training data	18036 (70%)
2	Testing data	3866 (15%)
3	Validation data	3865 (15%)
4	Total records	25767 (100%)

Table 3. Information about the models in UNSW-NB15

Model	ANN	CNN	RNN	LSTM
Total Params	13,253	99,269	19,013	57,029
Trainable Params	4,417	33,089	6,337	19,009
Non-trainable params	0	0	0	0
Optimizer params	8,836	66,180	12,676	38,020
Accuracy	90.12	87.30	87.69	80.44

8,836 is shown in table-3[6]. This comparison highlights the ANN's superior effectiveness in this particular task, while also underscoring the varying performance of different models based on their parameter configurations[6].In table-3,it describes that the number of training records are 18000 and testing and validation records are 3000 each[6]. Now both the datasets are trained with the ANN model after that we compare with the other models.

8 Comparative Analysis

Comparing our ANN model with other models like CNN,RNN, LSTM, Auto encoders.our model gives the best results compared with other models.

The above table-4 represents the how both the datasets gives the model accuracies.By using the ANN model both datasets has to be trained.

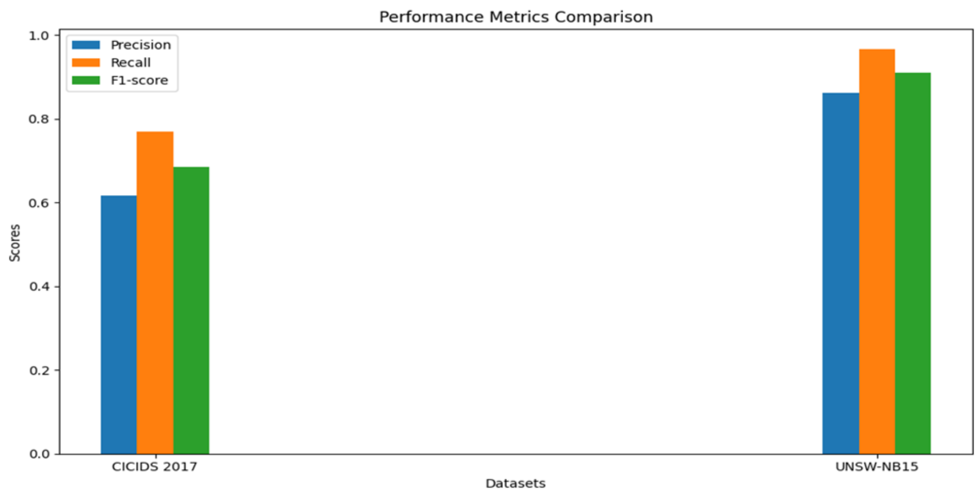

Figure 6. Comparsion of performance metrics for the both datasets UNSW and CICIDS

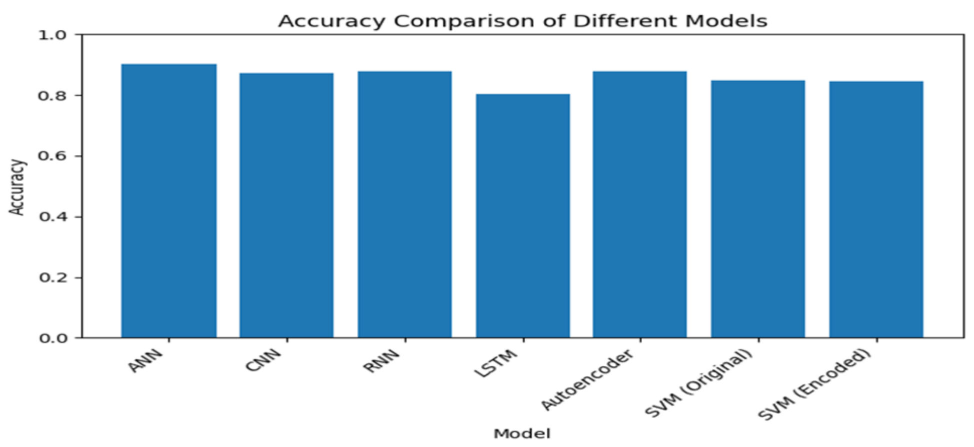

Figure 7. Comparing ANN with the other models

Table 4. Accuraries of both datasets

SNo	Dataset	Model Accuracy
1	UNSW-NB15	90.12
2	CICIDS-2017	80.18

To conclude, the study demonstrated that deep learning techniques especially, an improved variation of artificial neural networks employing algorithms like SCG ,LM and BR, can be used to advance cloud security G2 and adjustments of hyperparameters and datasets integration (UNSW-NB15 and CICIDS 2017), the model reached an impressive threat detection accuracy, indicating its suitability for real-time threat detection and embedding into multilayer cloud security systems[6].

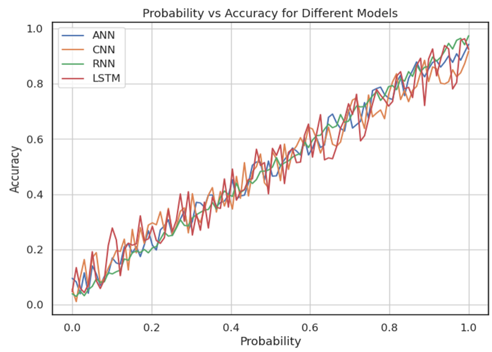

Figure 8. Probability vs Accuracy for different models

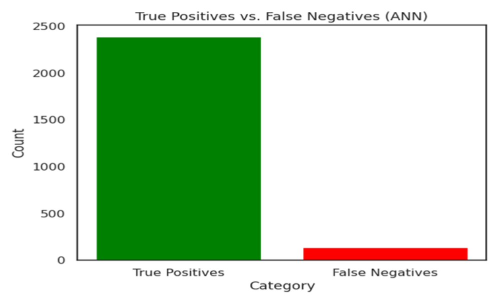

Figure 9. True Positives vs False Negatives for the ANN model

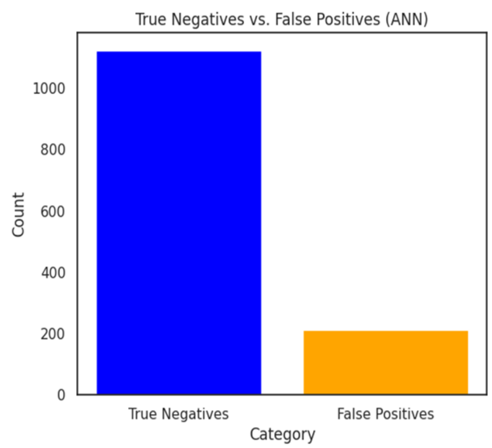

Figure 10. True Negatives vs False Positives for the ANN model

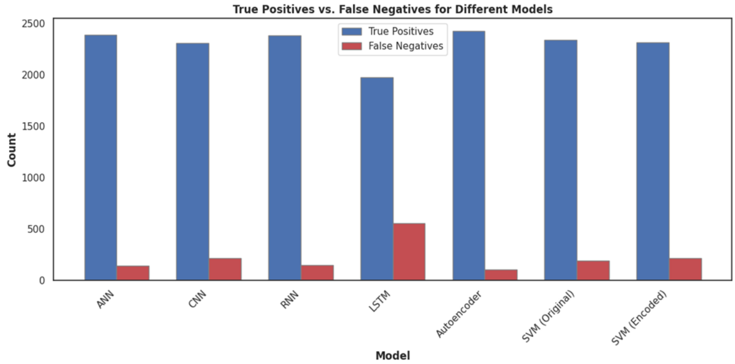

Figure 11. Tp vs FN of ANN model with other models

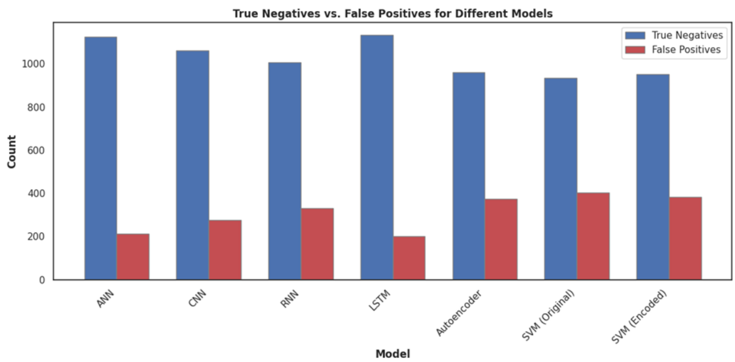

Figure 12. TN vs FP of ANN model with other models

9 Result

This paper developed an effective intelligence based model to enhance cloud security through the detection of cyber attacks. By using advanced training techniques such as Levenberg-Marquardt, scale conjugate gradient and Bayesian Regularization, the study was able to achieve improvements in detection of accuracy when validated on UNSW-NB15 and CICIDS 2017 datasets. The paper stressed the importance of algorithm selection and hyper-parameter tuning while aiming at enhancing the performance of the ANN architecture. In addition, further comparative analysis with other models have as well confirmed the ability of the ANN in performing real-time threat detection in clouds. This study shows that the construction of ANNs in today's environment, with all the tools and equipment of cloud security, will be able to effectively cope with the new type of cyber threats. It constitutes a flexible and effective partial solution in the fighting of future advanced threats targeting largely adopted cloud services.

References

[1] Kale, V. (2014). Guide to cloud computing for business and technology managers: from distributed computing to cloudware applications. CRC Press.

[2] L. Hasimi, D. Zavantis, E. Shakshui, and A. Yasar, "Cloud Computing Security and Deep Learning: An ANN approach," in *14th International Conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN 2023)*, Nov. 7-9, 2023, Almaty, Kazakhstan. doi: 10.1016/j.procs.2023.12.155.

[3] A. Gupta and M. Kalra, "Intrusion Detection and Prevention system using Cuckoo search algorithm with ANN in Cloud Computing," in *2020 Sixth International Conference on Parallel, Distributed and Grid Computing (PDGC)*, Nov. 2020, pp. 66–72. doi: 10.1109/PDGC50313.2020.9315771.

[4] (https://www.Kaggle.Com/datasets/chethuhn/community-intrusion-dataset),(https://www.Kaggle.Com/datasets/dhoogla/unswnb15), (https://www.kaggle.com /datasets/ hassan06/nslkdd).

[5] W. Etaiwi and G. Naymat, 'The Impact of applying Different Preprocessing Steps on Review Spam Detection', presented at the 8TH INTERNATIONAL CONFERENCE ON EMERGING UBIQUITOUS SYSTEMS AND PERVASIVE NETWORKS (EUSPN 2017) / 7TH INTERNATIONAL CONFERENCE ON CURRENT AND FUTURE TRENDS OF INFORMATION AND COMMUNICATION TECHNOLOGIES IN HEALTHCARE (ICTH-2017) / AFFILIATED WORKSHOPS, E. Shakshuki, Ed., 2017, pp. 273–279. doi: 10.1016/j.procs.2017.08.368.

[6] E. K. Subramanian and L. Tamilselvan, "A focus on future cloud: machine learning-based cloud security," SOCA, vol. 13, no. 3, pp. 237–249, Sep. 2019, doi: 10.1007/s11761-019-00270-0.

[7] R. Zarai, M. Kachout, M. A. G. Hazber, and M. A. Mahdi, "Recurrent Neural Networks and Deep Neural Networks Based on Intrusion Detection System," *OALib*, vol. 07, no. 03, pp. 1–11, 2020, doi: 10.4236/oalib.1106151.

[8] A. B. Nassif, M. A. Talib, Q. Nasir, H. Albadani, and F. M. Dakalbab, "Machine Learning for Cloud Security: A Systematic Review," *IEEE Access*, vol. 9, pp. 20717–20735, 2021, doi: 10.1109/ACCESS.2021.3054129.

[9] P. E.. Rauber, S. G. Fadel, A. X. Falcão, and A. C. Telea, "Visualizing the Hidden Activity of Artificial Neural Networks," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 1, pp. 101–110, Jan. 2017, doi: 10.1109/TVCG.2016.2598838.

[10] N. Srikanth and T. Prem Jacob, "An Real Time Cloud Security System and Issues comparison using Machine and Deep Learning," in *2021 Fifth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, Palladam, India: IEEE, Nov. 2021, pp. 523–529. doi: 10.1109/ISMAC52330.2021.9640650.

[11] G. B. Humphrey *et al.*, "Improved validation framework and R-package for artificial neural network models," *Environmental Modelling & Software*, vol. 92, pp. 82–106, Jun. 2017, doi: 10.1016/j.envsoft.2017.01.023.

[12] C. Fan, M. Chen, X. Wang, J. Wang, and B. Huang, "A Review on Data Preprocessing Techniques Toward Efficient and Reliable Knowledge Discovery From Building Operational Data," *Frontiers in Energy Research*, vol. 9, 2021, Accessed: Jun. 27, 2023. [Online]. Available:https://www.frontiersin.org/articles/10.3389/fenrg.2021.652801.

[13] M. A. Ferrag, O. Friha, D. Hamouda, L. Maglaras, and H. Janicke, "Edge-IIoTset: A New Comprehensive Realistic Cyber Security Dataset of IoT and IIoT Applications for Centralized and Federated Learning," *TechRxiv*, Jan. 27, 2022. doi: 10.36227/techrxiv.18857336.v1.

[14] M. U. Sana, Z. Li, F. Javaid, H. B. Liaqat, and M. U. Ali, “Enhanced Security in Cloud Computing Using Neural Network and Encryption,” *IEEE Access*, vol. 9, pp. 145785–145799, 2021, doi: 10.1109/ACCESS.2021.3122938.