International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Autoencoder-Boosted Lightweight Dense Net for Dimensionality Reduction and DOS Attack Classification in WSN

Sarkunavathi A.[*1], Dr. Srinivasan V.[2], Dr. Ramalingam M.[3]

Submitted: 06/02/2024 **Revised**: 14/03/2024 **Accepted**: 20/03/2024

Abstract: Wireless Sensor Networks (WSNs) are liable to Denial of Service (DoS) attacks, which can be easily executed in this context. This study presents a comparative analysis of five prominent deep learning architectures, namely AlexNet, VGGNet, ResNet, DenseNet, and Lightweight DenseNet, for their efficacy in classifying Denial of Service (DoS) attacks in Wireless Sensor Networks (WSNs). The evaluation is conducted using labeled instances of different types of DoS attacks from the WSN-DS and IOTID20 datasets. Various evaluation metrics including F1-score, recall, precision and accuracy computational efficiency are employed to discern the suitability of these architectures for real-time WSN applications. Experimental results from training and testing on the WSN-DS and IOTID20 datasets provide insights into the performance of each architecture, aiding in the selection of optimal models for DoS attack classification in WSNs.

Keywords: *Deep learning, DoS attacks, Lightweight DenseNet, Autoencoder*

1. Introduction

Wireless Sensor Networks (WSNs) face significant vulnerabilities to Denial of Service (DoS) attacks[1-3], which exploit the network's characteristics to disrupt normal operations. DoS attacks commonly take two forms: flooding the network with excessive traffic, causing congestion, or exploiting protocol vulnerabilities to disrupt legitimate traffic flow. Both types can severely impair WSN functionality, hindering data transmission and processing. Machine learning methods offer a proactive approach to detect and prevent such attacks, classifying traffic into normal flow, DoS flooding attacks, and other flows. These techniques, particularly classification engines, are dynamic, capable of identifying new attack types, and known for their adaptability, scalability, and reliability.

Deep learning systems possess the ability to derive rich insights from data, extending beyond their explicit training domains. This versatility makes them well-suited for diverse tasks such as image recognition, natural language comprehension, and speech analysis [4-5].

Comprising interconnected layers of nodes, these models collaborate to refine predictions and classifications. Through complex, nonlinear transformations across these layers, deep learning models generate statistical outputs from input data. Iteratively refined until reaching satisfactory accuracy levels, this process, characteristic of deep learning, operates through multiple layers or depths, hence the term "deep."

[1] *Department of Information Technology, Annamalai University, Annamalai Nagar , 608002, India*
[2] *Department of Information Technology, Annamalai University, Annamalai Nagar , 608002, India*
[3] *Department of Information Technology, Mailam Engineering College,Mailm,604304, India*
** Corresponding Author Email: sarkuna.bala@gmail.com*

One notable advantage of deep learning lies in its automatic feature extraction capability, a departure from old-style machine learning methods that often count on manual feature engineering. In deep learning, neural networks unconventionally discern relevant features right from raw data, obviating the need for human-designed features. This automation renders deep learning models highly adaptable to diverse tasks and data types, as they adeptly discover and leverage pertinent features during training.

In the context of WSNs and DoS attack classification, autoencoders [6-7] can be employed to condense the dimensionality of the input data, effectively capturing the essential characteristics of network traffic patterns. Using existing networks such as Dense Convolutional Network (DenseNet) [8-9] and Residual Network (ResNet) [10] allows for feature extraction from convolutional layers. If DenseNet is selected as the feature extractor and recurrent layers are excluded for efficiency purposes.

By leveraging autoencoders for dimensionality reduction and lightweight DenseNet architectures [11-14] for classification, this approach aims to enhance the accuracy and efficiency of DoS attack detection in WSNs. The program is less trustworthy if the face location is partially veiled, facing in any other direction, or if the light is insufficient.

This paper investigates the effectiveness of this combined approach and evaluates its performance in terms of classification accuracy, computational efficiency, and real-time applicability. The results of this research have significant implications for the security and reliability of WSNs in various practical scenarios. The reported metrics for the Autoencoder-Lightweight DenseNet architecture utilized in this work are as follows: Training Loss of 8.7%

and Training Accuracy of 99.7%, alongside Validation Loss of 12.11% and Validation Accuracy of 94.84%.

2. Related Work

Muhammad et al [15] proposed scheme which uses a hybrid feature selection approach and a deep neural network-(DNN-) based classifier that secures network data from various cyberattacks, combining feature selection and classification techniques to achieve higher F1-score, recall, precision and accuracy compared to existing methods. Notably, their model mitigates overfitting by removing redundant features, reducing time and computational complexity. Their future work will involve testing the model on other datasets and exploring additional feature selection techniques for further improvement.

Zhang et al. [16] introduced a novel convolutional block called Lightweight Dense Block (LDB) and proposed a lightweight character recognition network named CDensenet-U. The CDensenet-U framework incorporates weight compression strategies, including LDB utilization, depth separable convolution, and a 1*1 kernel convolution with a scale factor to reduce input channels. Although CDenseNet-U demonstrated promising results, there are still areas to explore, such as devising more efficient techniques to reduce computational costs and minimize the weight sizes of dense blocks while retaining crucial feature information.

Wu et al. [17] analyzed network traffic data considering both spatial and temporal characteristics. They introduced LuNet, a novel hierarchical neural network merging Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs). LuNet processes input traffic data, capturing intricate details in spatial and temporal features simultaneously. Experimentation on NSL-KDD and UNSW-NB15 datasets showcased LuNet's remarkable detection capabilities with a reduced false-positive alarm rate. However, LuNet faced challenges in effectively classifying attacks with limited samples in the training dataset, like Backdoors and Worms.

Hussain et al. [18] developed a method to convert network traffic data into images and trained a sophisticated CNN model, ResNet, with impressive results. They achieved 99.99% accuracy in detecting DoS and DDoS attacks and 87% precision in identifying eleven attack types, outperforming existing techniques by 9%.

However, ResNet faced scalability challenges as the network complexity grew with dataset size. Ongoing research focuses on addressing these scalability issues while sustaining high performance in network intrusion detection.

In their research, Zhang et al. [19] presented RANet, a model built upon group gating convolutional networks. Utilizing an overlapping approach in the final max-pooling layer, they carried out evaluations across five publicly available datasets. Their model demonstrated impressive accuracy rates across these datasets: 83.23% on NSL-KDD Test (+), 69.04% on NSL-KDD (21), 99.78% on KDDCUP99, 97.55% on Kyoto, and 96.73% on CICIDS2017 datasets

Khan et al. [20] developed a two-stage deep learning (TSDL) model for efficient detection of intrusion over networks. Combining stacked autoencoder architecture with a soft-max classifier, the model automatically extracts key features from unlabeled data, facilitating effective classification. Experimental results on KDD99 and UNSW-NB15 datasets demonstrated superior performance compared to existing methods, achieving recognition rates of up to 99.996% and 89.134%, respectively.

Huang et al. [21] developed a new lightweight hybrid neural network for classifying medical images, especially useful with limited training data. It combines a modified PCANet with a simplified DenseNet, overcoming limitations of the original PCANet and achieving accurate classification with fewer adjustable weights than traditional DenseNet. Tests on various datasets show our hybrid network outperforms popular models like ResNet , VGG, AlexNet and DenseNet in accuracy ,sensitivity, and specificity. Future work will focus on refining the PCANet and exploring new ways to combine features during training.

Corin et al. [22] proposed a lightweight approach of Convolutional Neural Network for quick attack detection. Unlike other methods, it doesn't require threshold configuration or extensive feature engineering, making deployment easy. Their unique traffic preprocessing aids efficient DDoS attack detection. Evaluation shows LUCID performs as well as top methods, with consistent results across datasets.

3. Methodology

Over the past decade, advancements in technology, such as IoT and 5G networks, have transformed computer networks, facilitating extensive data exchange but also introducing vulnerabilities to attacks. In Wireless Sensor Networks (WSNs), feature extraction plays a crucial role in sensing and vindicating these threats. The proposed model combines DenseNet with an autoencoder for feature extraction in WSNs[23-24], offering several benefits which is depicted in Fig 1 :

• DenseNet's dense connectivity efficiently extracts features from sensor data, capturing intricate patterns indicative of different attacks.

• When combined with an autoencoder, DenseNet enhances representation learning by compressing data into a lower-dimensional space and extracting higher-level features.

• Joint learning of hierarchical features enables the system

to learn compact representations and complex features, aiding in anomaly detection.

• By leveraging DenseNet and autoencoder capabilities, the system can effectively identify anomalies or attacks within WSN data, even adapting to new attack patterns.

• Autoencoders utilize unlabeled data for representation learning, beneficial when labeled attack data is limited, enabling the system to capture various attack scenarios.

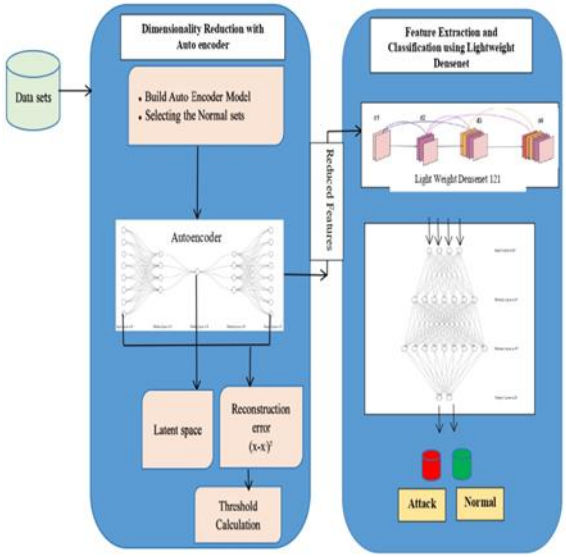

Fig.1. Proposed Architecture

3.1. Dimensionality Reduction using Deep Auto Encoder

Dimensionality reduction[27-30] is a crucial process aimed at decreasing the number of dimensions in a dataset, either by excluding less informative features (Feature Selection) or transforming the data into a lower-dimensional space (Feature Extraction). This reduction helps mitigate overfitting, where a model becomes overly tuned to training data and performs poorly on unseen real-world data. Autoencoders (AEs) are neural networks consisting of an encoder, a hidden unit, and a decoder, aiming to generate output resembling the input data through backpropagation. AEs primarily focus on feature extraction within the encoder architecture, enabling data transformation into a dimensionally reduced representation. Their ability to train on unlabeled data makes AEs [25-26]effective at identifying unknown attacks. The proposed approach leverages higher reconstruction loss for anomalous traffic flows, surpassing benchmark unsupervised algorithms in perceiving Denial-of-Service (DoS) attacks according to numerical experiments.

A deep autoencoder consists of two mirrored deep-belief networks, each with four to five shallow layers. One network encodes the data, while the other decodes it. These networks have more layers than a basic autoencoder, enabling them to capture complex features. Each layer is built using restricted Boltzmann machines, which are fundamental units in deep-belief networks.

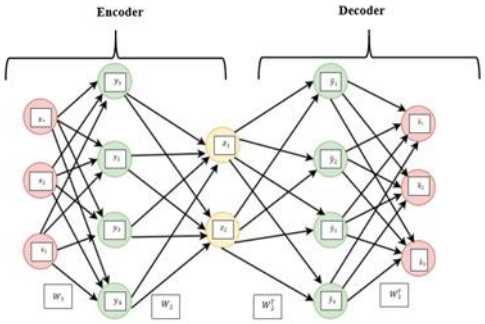

Fig.2. Deep AutoEncoder

The mathematical construction of a deep autoencoder with a hidden layer is encapsulated by specifying the encoder function as α , and the decoder function as $\widehat{X}$.

The encoding function α can be represented as ,

$$\alpha = f(Wx + b) \qquad (3.1)$$

and the decoding function $\widehat{X}$ can be represented as ,

$$\widehat{X} = f'(W'\alpha + b') \qquad (3.2)$$

Where f and f ‘ are the nonlinear activation function , W $\in R^{l x m}$ and $W' \in R^{m x l}$ are the weight matrices, $b \in R^{l}$ and $b' \in R^{m}$ are the bias vectors and $\alpha \in R^{l}$ is the hidden layer output .

The reconstruction error can be computed by providing a set of inputs $\{x_i\}_{i=1}^{n}$ as,

$$\sum_{i=1}^{n} \| \widehat{x_i} - x_i \|^2 \qquad (3.3)$$

From the Fig 2 represents the Deep Auto encoder architecture. The x_1,x_2 and x_3 represent the input and $\widehat{x_1}$, $\widehat{x_2}$, $\widehat{x_3}$ represents the output with two central nodes z_1 and z_2 which is arranged in a way symmetric in three hidden layers. The deep autoencoder's purpose is to curtail the difference between output $\hat{x}$ and input x.

The reconstruction error function $\sum_{i=1}^{n} \| \widehat{x_i} - x_i \|^2$ is minimized by using a deep autoencoder by learning the weight matrices W_1, W_T and bias vector b_1, b_T^1 to accomplish the self learning objective. Consequently, the goal of a deep autoencoder can be restated as the following optimization challenge.

$$W1, b_1^{min}, W_1^T, b_1^T \sum_{i=1}^{n} \| \widehat{x_i} - x_i \|^2 \quad (3.4)$$

3.2. Feature Extraction and Classification using Light Weight Densenet

The decision to use the DenseNet framework instead of ResNet is mainly based on considerations of computational efficiency and addressing the vanishing gradient problem. While ResNet can address the vanishing gradient issue to some extent, its architecture becomes computationally

intensive as the network depth increases, resulting in exponential parameter growth.A Lightweight DenseNet architecture retains the benefits of DenseNet while reducing computational complexity and model size, making it ideal for resource-constrained environments like Wireless Sensor Networks (WSNs). The combination of learned representations from the autoencoder and hierarchical features from DenseNet aids in detecting deviations from normal patterns, indicating potential attacks or abnormal behavior.

A dense cluster in DenseNets is made up of n cells with identical characteristics. Each cell receives an inhibitory spike train at a rate of x, determined by stochastic connections between the soma and cytosol. The activation likelihood of a cell, denoted as q, depends on the cluster size, specifically the number of cells it contains. To numerically determine q, prior studies' methodologies are employed to derive a solution in the following manner:

$$q = \varsigma(x) = -\frac{(c-nx)+\sqrt{4p(\lambda^{-}+x)(n-1)d-(c-nx)^{2}}}{2p(\lambda^{-}+x)(n-1)} \qquad (3.5)$$

with

$$c = rp+\lambda^{+}p-r-\lambda^{-}n-npr-\lambda^{+}pn, \qquad (3.6)$$
$$d = n\lambda^{+}$$

where

p - The probability of repeated firing occurring when a cell attempts to fire.

r - rate of cell firing

The rate at which a cell fires is denoted by the symbol r, whereas excitatory and inhibitory spikes are denoted externally by the symbols + and -respectively. The main reason for using $\zeta(\cdot)$ for vectors and matrices is to make the way they are written easier to understand.

The basic structure of DenseNet is depicted in Fig 3, which includes dense blocks, transition layers, convolutional layers, and fully connected layers.

The dense block shown in Fig.4 is made up of tightly connected dense units that have convolutional operations, Batch Normalization (BN), Rectified Linear Unit (ReLU), and other nonlinear mapping functions..

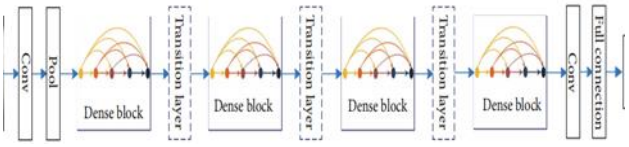

Fig.3. Basic Structure of DenseNet

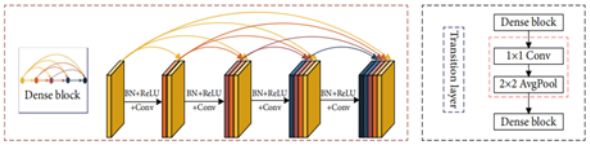

Fig.4. Dense Block and Transition Layer

The Transition layer, placed amid adjacent dense blocks. It consists of a 1×1 convolution followed by a 2×2 average pooling operation. The main function of this layer is to compress the input from the dense block, retaining all mined feature information. This compression reduces the size and dimensionality of the feature maps, controlling the number of parameters within the dense block and preventing overfitting in the network.

The Fig. 5 illustrates the concept of a simplified dense block, where the original dense block is replaced by several simplified dense blocks arranged sequentially. This substitution aims to decrease the number of dense units within each block. By employing multiple simplified dense blocks in succession, the original dense blocks are replaced, leading to a reduction in the output feature map dimensions. Despite this reduction, the process maintains feature reuse within the model architecture.

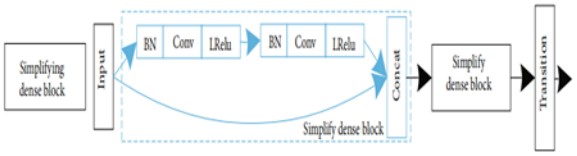

Fig.5. Simplified Dense block for Light weight Densenet

The Fig.6 illustrates the "one-time aggregation" strategy, where the outputs from all dense units in the dense block are aggregated towards the end, instead of the interconnected nature of pairs of dense units. To cut down on inference time and energy use, this lightweight network design technique focuses on two main factors: model size and Floating Point Operations (FLOPs). It also considers GPU computational efficiency and Memory Access Cost (MAC) as important factors.

Fig.6. Simplified Dense block for Light weight Densenet

Densely connected aggregations of intermediate features can indeed generate robust features with fewer parameters and activations. However, this approach can also result in significant memory access overheads. To address these challenges in the DenseNet detector architecture, Lee et al. [31] introduced a fast and efficient architecture called

VoVNet. VoVNet incorporates the concept of "one-time aggregation" (OSA) to mitigate these issues.

The Fig.7 illustrates the DenseNet utilizing the dense aggregation method, where all preceding features are aggregated at each subsequent layer. This results in an increase in the input channel size as the network progresses through its layers, despite only a few new outputs being generated.

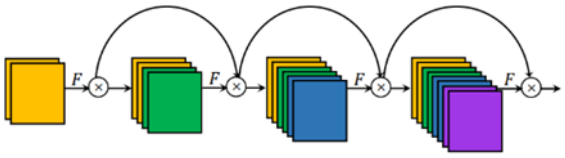

Fig.7. Dense Aggregation Densenet

In contrast, Fig .8 represents the One-Shot Aggregation used in VoVNet. This method aggregates or concatenates all features just once, in the last feature map. Unlike DenseNet's method, which leads to a linear increase in input channel size, the One-Shot Aggregation method ensures a constant input size throughout the network. By concatenating the features only in the final feature map, this approach enables the enlargement of the new output channel.

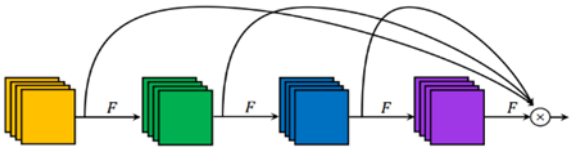

Fig.8. One time aggregation of VoVNet

3.3. Datasets and Feature Extraction

The WSN-DS dataset[32] underwent preprocessing to generate 23 features representing individual sensor states and simulating five types of Denial of Service (DoS) attacks: Flooding, Blackhole, Normal, TDMA, and Grayhole. Tailored for intrusion detection, this Wireless Sensor Network (WSN) dataset enables the application of machine learning and deep learning techniques for DoS attack identification and categorization. It comprises 365,788 records with 19 unique attributes.

The IoTID20 dataset initially consists of 86 columns and 625,783 rows, with each row associated with a specific network activity. Preprocessing focused on enhancing the accuracy of label, category, and sub-category features to improve classification precision. However, the primary focus of the work lies in binary classification of label features, distinguishing between normal and anomaly, while category features encompass five classifications: MITM attack , DoS attack , Scan attack, Mirai attack and normal.

4. Results and Discussion

4.1. Hyperparameter Settings

This study acknowledges several critical model hyperparameters that have a significant impact on the training process and the overall effectiveness of the developed model.

Table 1 Hyperparameter Settings

Hyper parameter	**Value**
Epoch	10, 25
Activation Function	ReLU
Loss Function	Sparse Categorical Cross Entropy
Optimization algorithm	Adam
Learning rate	0.001
Verbose	1

The parameters detailed in Table 1 cover crucial aspects such as activation function, epochs (number of training iterations), learning rate (controls weight updates), verbosity (level of output information during training), patience (stopping criterion for training), choice of optimization technique (algorithm for updating parameters), and selection of a loss function (measures disparity between predicted and actual values). Each of these hyperparameters plays a vital role in shaping the model's learning process and its ability to achieve optimal performance.

4.2. Evaluation Metrics and Results

In this proposed method, we assess and compare performance metrics such as F1-score, accuracy, recall and precision,to evaluate its effectiveness.

Accuracy (ACC) gauges the proportion of correctly classified data samples. A higher accuracy suggests effective learning on a balanced test dataset. However, in situations with imbalanced test datasets, solely depending on accuracy can mislead about the model's performance. Figure 9 illustrates that the proposed Lightweight DenseNet 121 exhibits superior accuracy compared to other deep learning models.

$$\textbf{Accuracy} = (TP + TN)/(TP + TN + FP + FN)$$

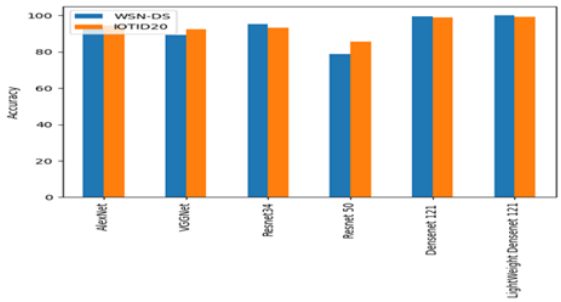

Fig.9. Comparison of the overall prediction accuracy of the proposed Lightweight Densenet

The ratio of properly predicted samples of a class to the total number of occurrences of the same class is estimated by recall, commonly referred to as the true positive rate.A machine learning model that performs well is indicated by a higher recall value between 0 and 1. According to Figure 10, the proposed Lightweight DenseNet 121 demonstrates superior recall compared to other deep learning models.

Recall = TP/(TP + FN)

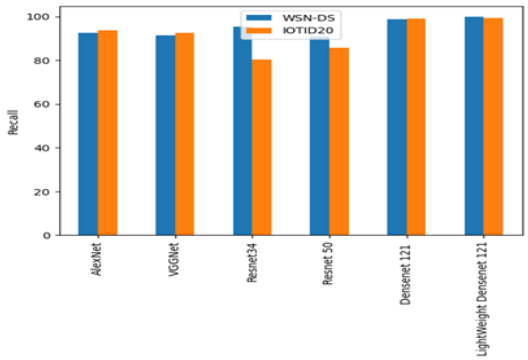

Fig.10. Comparison of the of Recall value of the Proposed model with WSN-DS and IOTID20 Dataset

By computing the ratio of successfully predicted samples to all predicted samples for a given class, precision evaluates the accuracy of correct predictions. To assess model performance, it is frequently assessed in conjunction with recall. However, a complete measure such as the F1-score is desirable, especially for imbalanced test datasets, when accuracy and recall clash. As illustrated in Figure 11, the suggested Lightweight DenseNet 121 outperforms other deep learning models in terms of precision.

Precision = TP/(TP + FP)

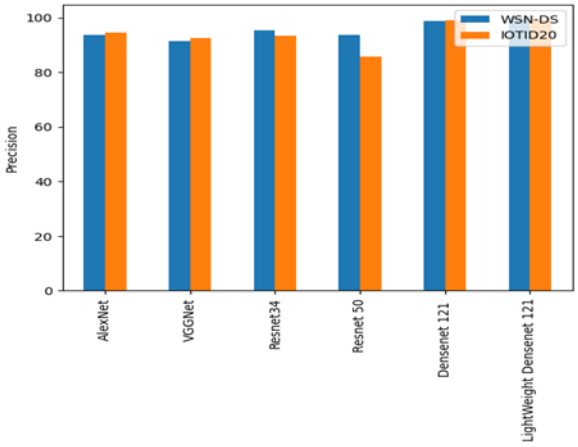

Fig.11. Comparison of the of Precision value of the Proposed model with WSN-DS and IOTID20 Dataset

The precision versus recall trade-off is calculated using the F1-score. It is the harmonic mean of memory and precision .The Fig.12 shows that the proposed Lighweight Densenet 121 has higher F1 Measure when compared to other deep learning models

F1 Measure = 2 × (Precision × Recall)/(Precision + Recall)

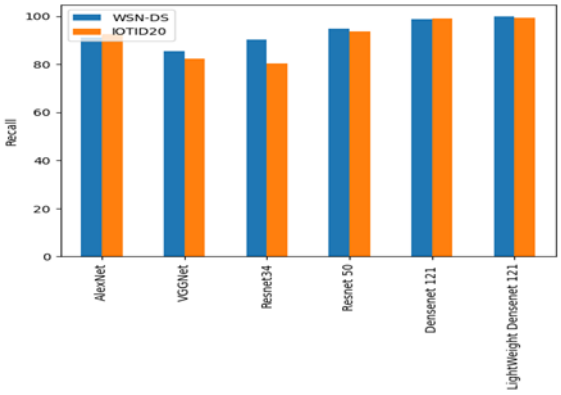

Fig.12. Comparison of the of F1-measure value of the Proposed model with WSN-DS and IOTID20 Dataset

The Autoencoder-Lightweight DenseNet model showed promising results in detecting attacks on the WSN dataset. It achieved a training accuracy of 99.4% after 10 epochs and 99.7% after 25 epochs, using a learning rate of 0.001 and Relu activation function for binary classification. The model's computational efficiency was demonstrated with execution times of 805 seconds for 10 epochs and 1103 seconds for 25 epochs.During the training phase, performance metrics were recorded after 10 epochs. About 96.07% training accuracy and 99.44% validation accuracy were attained by the model. The training loss, indicating the difference between predicted and actual values during training, was 25.7%, while the validation loss, measuring performance on unseen data, was recorded at 7.01%..

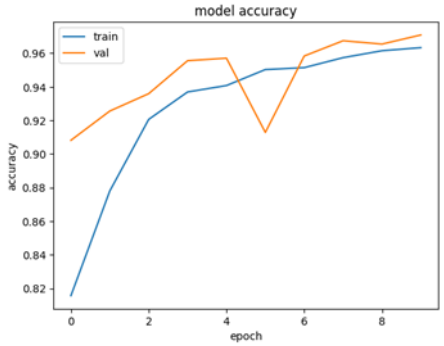

Fig.13. Model Accuracy with 10 Epochs

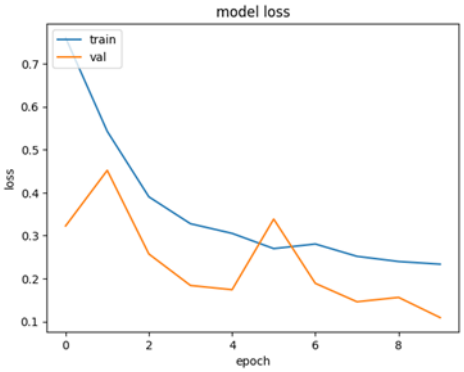

Fig.14. Model Loss with 10 Epochs

Overall, these metrics suggest a well-performing model, with relatively high accuracy and low losses on both training and validation sets after 10 epochs of training which is depicted in Fig. 13 and Fig. 14 .

During the training of an intrusion detection model using the Autoencoder-Lightweight DenseNet architecture, certain performance metrics were achieved after 25 epochs. The reported metrics are as follows: Training Loss: 8.7%, Training Accuracy: 99.7%, Validation Loss: 12.11%, Validation Accuracy: 94.84%. These metrics were monitored and recorded across the 25 training epochs, as depicted in Fig .15 and Fig.16.

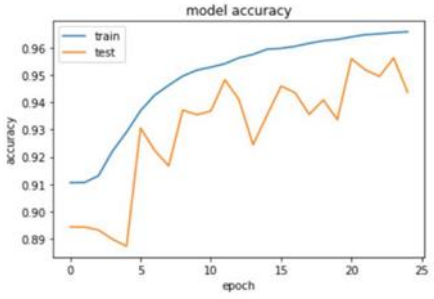

Fig.15. Model Accuracy with 25 Epochs

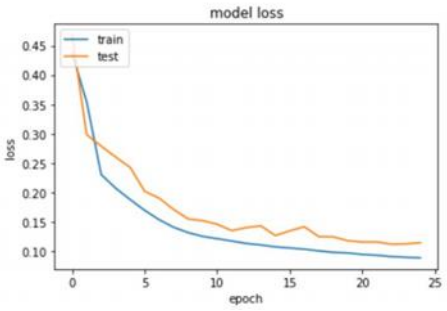

Fig.16. Model Loss with 25 Epochs

5. Conclusion and Future Work

In conclusion, this study conducted a comparative analysis of five prominent deep learning architectures for classifying Denial of Service (DoS) attacks in Wireless Sensor Networks (WSNs). Through evaluation using labeled instances from the WSN-DS and IOTID20 datasets, various metrics such as accuracy, precision, recall, F1-score, and computational efficiency were employed to assess the efficacy of these architectures for real-time WSN applications.The experimental results provided insights into the performance of each architecture in classifying DoS attacks, thereby aiding in the selection of optimal models for WSN security. The findings indicate that the Lightweight DenseNet architecture showed promising results, demonstrating high accuracy and efficiency in detecting DoS attacks in WSNs.

For future work, further exploration and refinement of the Lightweight DenseNet architecture could be conducted to enhance its performance in detecting various types of DoS attacks. Additionally, extending the evaluation to include more diverse and challenging datasets, as well as real-world deployment scenarios, would provide a more comprehensive understanding of the architectures' capabilities and limitations. Moreover, investigating techniques for improving the computational efficiency of deep learning models in resource-constrained WSN environments would be beneficial for practical deployment. Overall, continued research in this area holds the potential to advance the development of robust and efficient solutions for securing Wireless Sensor Networks against DoS attacks.

References

[1] Islam, Mohammad Nafis Ul & Fahmin, Ahmed & Hossain, Md Shohrab & Atiquzzaman, Mohammed. Denial-of-Service Attacks on Wireless Sensor Network and Defense Techniques. Wireless Personal Communications. 116. 1-29. 10.1007/s11277-020-07776-3,2021.

[2] Gavrić, Ž., & Simic, D.B. Overview of DOS attacks on wireless sensor networks and experimental results for simulation of interference attacks. Revista Ingenieria E Investigacion, 38, 130-138,2018.

[3] Stankovic, J.A., & Wood, A.D. A Taxonomy for Denial-of-Service Attacks in Wireless Sensor Networks. Handbook of Sensor Networks,2004.

[4] [4] Sarker, I. Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions. SN COMPUT. SCI. 2, 420,2021. https://doi.org/10.1007/s42979-021-00815-1

[5] [5] Francesco Piccialli, Fabio Giampaolo, Edoardo Prezioso, Danilo Crisci, and Salvatore Cuomo. Predictive Analytics for Smart Parking: A Deep Learning Approach in Forecasting of IoT Data. ACM Trans. Internet Technol. 21, 3, Article 68 (August 2021), 21 pages,2021.

https://doi.org/10.1145/3412842

[6] Mirsky, Y.; Doitshman, T.; Elovici, Y.; Shabtai, A. Kitsune An Ensemble of Autoencoders for Online Network Intrusion Detection.In Proceedings of the 25th Annual Network and Distributed System Security Symposium, NDSS 2018, San Diego, CA, USA,18–21 February 2018.

[7] Zavrak, S.; ˙Iskefiyeli, M.(2020) Anomaly-Based Intrusion Detection From Network Flow Features Using Variational Autoencoder. IEEE Access 2020, 8, 108346–108358.

[8] Huang, G., Liu, Z., Van Der Maaten, L. and Weinberger, K.Q. Densely Connected Convolutional Networks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, 21-26 July 2017, 4700-4708. https://doi.org/10.1109/CVPR.2017.243

[9] Hemalatha J, Roseline SA, Geetha S, Kadry S, Damaševičius R. An Efficient DenseNet-Based Deep Learning Model for Malware Detection. Entropy (Basel). 2021 Mar 15;23(3):344. doi: 10.3390/e23030344. PMID: 33804035; PMCID: PMC7998822,2021.

[10] Rezende, E.; Ruppert, G.; Carvalho, T.; Ramos, F.; De Geus, P.Malicious software classification using transfer learning of resnet-50deep neural network. In Proceedings of the 2017 16th IEEE International Conference on Machine Learning and Applications(ICMLA), Cancun, Mexico, 18–21 December 2017; pp. 1011–1014.

[11] He, Yi. A New Lightweight DenseNet Based on Mix-Structure Convolution. IOP Conference Series: Materials Science and Engineering.790,2020.

[12] Jingdong Yang, Lei Zhang, Xinjun Tang, Man Han, CodnNet: A lightweight CNN architecture for detection of COVID-19 infection,Applied Soft Computing,Volume 130,109656,ISSN1568-4946 2022.

[13] Din, Sadia & Paul, Anand & Ahmad, Awais. Lightweight deep dense Demosaicking and Denoising using convolutional neural networks. Multimedia Tools and Applications. 79. 10.1007/s11042-020-08908-4,2020.

[14] Huang, L., Ren, K., Fan, C., and Deng, H., A Lite Asymmetric DenseNet for effective object detection based on convolutional neural networks (CNN), Optoelectronic Imaging and Multimedia Technology VI, vol. 11187,2019. doi:10.1117/12.2538755.

[15] Muhammad Naveed, Fahim Arif, Syed Muhammad Usman, Aamir Anwar, Myriam Hadjouni, Hela Elmannai, Saddam Hussain, Syed Sajid Ullah, Fazlullah Umar, "A Deep Learning-Based Framework for Feature Extraction and Classification of Intrusion Detection in Networks", Wireless Communications and Mobile Computing, vol. 2022, Article ID 2215852, 11 pages,. https://doi.org/10.1155/2022/2215852

[16] Zhang, Z., Tang, Z., Wang, Y., Zhang, H., Yan, S., & Wang, M. Compressed densenet for lightweight character recognition. arXiv preprint arXiv:1912.07016,2019.

[17] P. Wu and H. Guo, "LuNet: A Deep Neural Network for Network Intrusion Detection," 2019 IEEE Symposium Series on Computational Intelligence (SSCI), Xiamen, China, 2019, pp. 617-624, doi: 10.1109/SSCI44817.2019.9003126.

[18] F. Hussain, S. G. Abbas, M. Husnain, U. U. Fayyaz, F. Shahzad and G. A. Shah, IoT DoS and DDoS Attack Detection using ResNet, 2020 IEEE 23rd International Multitopic Conference (INMIC), Bahawalpur, Pakistan, 2020, pp. 1-6, doi: 10.1109/INMIC50486.2020.9318216.

[19] Zhang et al. Zhang X, Yang F, Hu Y, Tian Z, Liu W, Li Y, She W. RANet: network intrusion detection with group-gating convolutional neural network. Journal of Network and Computer Applications. 2022;198(2):103266. doi: 10.1016/j.jnca.2021.103266

[20] F. A. Khan, A. Gumaei, A. Derhab and A. Hussain, A Novel Two-Stage Deep Learning Model for Efficient Network Intrusion Detection, in IEEE Access, vol. 7, pp. 30373-30385, 2019, doi: 10.1109/ACCESS.2019.2899721.

[21] Z. Huang, X. Zhu, M. Ding, and X. Zhang . Medical image classification using a light-weighted hybrid neural network based on PCANet and DenseNet, IEEE Access, vol. 8,pp. 24697–24712, 2020.

[22] R. Doriguzzi-Corin, S. Millar, S. Scott-Hayward, J. Martínez-del-Rincón and D. Siracusa,Lucid: A Practical, Lightweight Deep Learning Solution for DDoS Attack Detection, IEEE Transactions on Network and Service Management, vol. 17, no. 2, pp. 876-889, June 2020, doi: 10.1109/TNSM.2020.2971776.

[23] Albahli S, Nazir T, Mehmood A, Irtaza A, Alkhalifah A, Albattah W.AEI-DNET: A Novel DenseNet Model with an Autoencoder for the Stock Market Predictions Using Stock Technical Indicators. Electronics. 11(4):611,2022. https://doi.org/10.3390/electronics11040611

[24] Pintelas E, Livieris IE, Pintelas PE. A Convolutional Autoencoder Topology for Classification in High-Dimensional Noisy Image Datasets. Sensors (Basel). 2021 Nov 20;21(22):7731. doi: 10.3390/s21227731.

[25] Lopez-Martin, Manuel & Carro, Belén & Sanchez-Esguevillas, Antonio & Lloret, Jaime. Conditional Variational Autoencoder for Prediction and Feature Recovery Applied to Intrusion Detection in IoT. Sensors. 2017.

[26] Ieracitano, Cosimo & Adeel, Ahsan & Morabito, Francesco & Hussain, Amir. A Novel Statistical Analysis and Autoencoder Driven Intelligent Intrusion Detection Approach. Neurocomputing,2019. 387. 10.1016/j.neucom.2019.11.016.

[27] [27]Yasi Wang, Hongxun Yao, Sicheng Zhao,Auto-encoder based dimensionality reduction,Neurocomputing,Volume 184,Pages232-242,ISSN0925-2312,2016.

https://doi.org/10.1016/j.neucom.2015.08.104.

[28] R. K. Keser and B. U. Töreyin,Autoencoder Based Dimensionality Reduction of Feature Vectors for Object Recognition," 2019 15th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS), Sorrento, Italy, 2019, pp. 577-584, doi: 10.1109/SITIS.2019.00097.

[29] Zamparo, L., & Zhang, Z. Deep Autoencoders for Dimensionality Reduction of High-Content Screening Data. ArXiv, abs/1501.01348,2015.

[30] Wang, J., He, H., & Prokhorov, D.V. A Folded Neural Network Autoencoder for Dimensionality Reduction. International Neural Network Society Winter Conference,2012.

[31] Y. Lee, J. W. Hwang, S. Lee, Y. Bae, and J. Park,An Energy and GPU-Computation Efficient Backbone Network for Real-Time Object Detection," Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 752–760, Long Beach,USA,2019.

[32] Almomani, I., Al-Kasasbeh, B. and Al-Akhras, M.,. WSN-DS: A dataset for intrusion detection systems in wireless sensor networks. Journal of Sensors,2016.