Deep learning based Cyber Attacks Detection using Hybrid CNN-AE Model

Katikam Mahesh[1]
Assistant Professor in Tirumala Engineering College (Autonomous) Jonnalagadda Narasaraopet India 522601

Abstract ***Thanks to the internet and an extensive number of digital devices, life is quite comfortable these days. As with all excellent there are disadvantages, and the digital world of today is certainly not an exemption. While the internet has transformed our lives in the present, securing personal data remains an immense task. It is it that leads to cyber-attacks The detection of intrusions is keyConventional methods like denial-of-service attacks, phishing scams, and malicious software attacks are poorly detected by techniques like DT, SVM, and ANN. This study presents a unique CNN-Autoencoders Model to automatically improve detection attacks using the publicly available UNSW NB-15 input dataset, thereby enhancing accuracy.***

Index **Terms-Classification, Detection, Cyber Attacs, Convolution Neural Networks [CNN], Auto Encoders [AE], denial-of-service [DoS].**

I. INTRODUCTION

ConvNets, short for Convolutional Neural Networks, are a particular kind of deep learning algorithm that are mostly used for tasks requiring object recognition, such as picture categorization, detection, and segmentation. CNNs are used in many real-world applications, including security camera systems and driverless cars, among others. Based on convolutional neural networks, the method can be classified as semi-supervised data driven approaches (CNN). Using standard network hyperparameters and data obtained from an attack-free system, the recommended method chooses a proper CNN architecture and thresholds for online intrusion detection on its own. A kind of deep neural network model called as an autoencoder [AE]is. The fact that the inputs and outputs of this model are similar is its most noticeable feature. The middle of the hidden layers narrows down in comparison to the other layers, which is another noticeable trait. This buried layer of constricted neurons in the autoencoder deep neural network model is known as described as the "bottleneck."

II. RELATED WORKS

It is implied that the conventional optimization techniques are getting less reliable for handling difficulties due to the exponential increase in the number and complexity of optimization problems [1],[2]. A. Milani, M. Bialetti, and V. Santucci. An algebraic structure for swarm and evolutionary algorithms in combinatorial optimization. Algorithms using metaheuristics [3–4]. Marius Portmann, Nour Moustafa, Siamak Layeghy, and Sarhan Mohanad. NetFlow Datasets for Network Intrusion Detection Assisted by Machine Learning. But they are inaccurate in their detection. H. Fu, C. Fu, H. Chen, J. Su, H. Xu, and Q. Cao. Utilizing Support Vector Machines An autoencoder model has two main parts. Encoders are components identified in the first part. Up to the layer designated as the bottleneck, the encoder process makes sure that the data from the inputs are encoded. In comparison to other hidden layers, the bottleneck layer has fewer neurons. It can also be thought of in this sense as a region where data that heads towards the bottleneck gets compacted.

III. PROPOSED METHDOLOGY

The core component of the architecture of the attack detection approach is the security system's constant surveillance of network system data traffic in order to recognize and categorize different kinds of security attacks. In the attack detection approach, a hybrid Convolutional Neural Network (CNN) and Auto encoders are used to detect cyberattacks in network systems that are introduced into the system by the attackers. Through opposing network attacks, attackers can exploit system data and get unwanted access to private network information. The selection

of significant features lowers the computational complexity of the CNN-AE model while increasing its performance efficiency. Cyberattack detection is taught to the CNN-AE model.

III.I Data Collection

UNSW-NB 15 was a dataset pertaining to internet intrusions. It has unprocessed network packets. There are 175,341 records in the original training collection and 82,332 records in the testing set. They exist in several forms, such as attack and regular.

ID	Feature	ID	Feature	ID	Feature
1	attack_cat	16	dloss	31	response_body_len
2	dur	17	sinpkt	32	ct_srv_src
3	proto	18	dinpkt	33	ct_state_ttl
4	service	19	sjit	34	ct_dst_ltm
5	state	20	djit	35	ct_src_dport_ltm
6	spkts	21	swin	36	ct_dst_sport_ltm
7	dpkts	22	stcpb	37	ct_dst_src_ltm
8	sbytes	23	dtrcpb	38	is_ftp_login
9	dbytes	24	dwin	39	ct_ftp_cmd
10	rate	25	tcprtt	40	ct_flw_http_mthd
11	sttl	26	synack	41	ct_src_ltm
12	dttl	27	ackdat	42	ct_srv_dst
13	sload	28	smean	43	is_sm_ips_ports
14	dload	29	dmean		
15	sloss	30	trans_depth		

Table 1: Features of UNB-NW15

III.II Data Preprocessing

The data is preprocessed in order to eliminate redundant and uncertainties once it has been unified into one dataset. For the sake of data consistency, numerous uncertainties—such as managing missing values and reducing duplicates—are filtered out. preparing the data for future analysis or modeling by preprocessing it allows for the dataset's integrity and quality. In this phase, basic exploration tasks like displaying the first and last rows to understand the data structure, verifying the shape (number of rows and columns), looking up column names to identify features, verifying that the data is not null, and examining the distribution of the target column ('Label') to understand class distribution are all carried out as part of an exploratory data analysis (EDA).

III.III Auto Encoders for Attack Detection

Two primary components make up an autoencoder model. Encoders are components found in the first section. Up to the layer known as the bottleneck, the encoder process makes sure that the data from the inputs are encoded. In comparison to other hidden layers, the bottleneck layer has fewer neurons. It can also be thought of in this sense as a region where data that is approaching the bottleneck gets compressed. The autoencoder model's second component, referred known as the encoder, begins at the bottleneck layer and works its way up to outputs. The encoded hidden layer data can be decoded thanks to this section. For autoencoders, the forward propagation and backpropagation procedures that work for deep learning models also work.

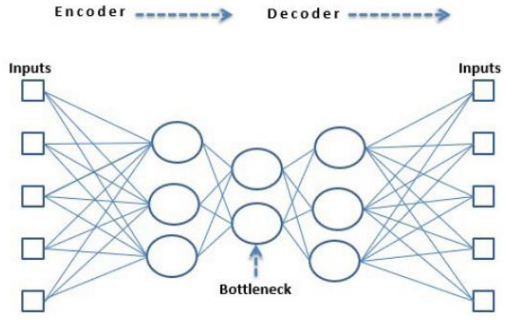

Fig 1: Architecture of Auto Encoders

III.IV CNN UNIT

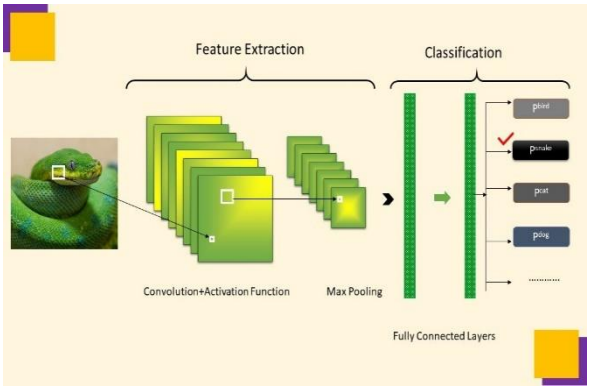

Fig2: CNN Model

III.V

ALGORITHM OF AN AUTOENCODERS

Start:

1. Import TensorFlow and other libraries.
2. Load the dataset.
3. First example: Basic autoencoder.
4. Second example: Image denoising. Define a convolutional autoencoder.

5. Third example: Anomaly detection.
6. Overview. Load ECG data. Build the model. Detect anomalies.
7. Next steps.

Stop:

IV.EXPERIMENT RESULTS

FitAutoencoderThe variables (train, x_train, epochs=50, batch size=256, shuffle=True, validation data= (x_test, test))

epochs = 5

batch size = 128

history = autoencoder**.** Fit (Train, Train, batch size=batch size, epochs=epochs, verbose=1, validation data= (Test, Test))

Train on 60000 samples, validate on 10000 samples

Epoch 1/5

60000/60000 [==============================] - 2s - loss: 0.0441 - valyls: 0.0225

Epoch 2/5

60000/60000 [==============================] - 2s - loss: 0.0174 - valyls: 0.0130

Epoch 3/5

60000/60000 [==============================] - 2s - loss: 0.0110 - valyls: 0.0087

Epoch 4/5

60000/60000 [==============================] - 2s - loss: 0.0078 - valyls: 0.0066

Epoch 5/5

60000/60000 [==============================] - 2s - loss: 0.0062 - valyls: 0.0055

encoded_imgs = conv_encoder. predict (Test) convenor = Model (x, h)

For I in range(n), n = 10 Pl. Figure (fig size= (20, 8)):

Pl. Subplot (1, n, i+1) = a

plimsol (images encoded[I]). reconfigure (4, 16). T) Pl. Gray ()

Geotaxis() in ax.set_visible (None)

Get_Yaxis() in ax.plt.show() set_visible(False)

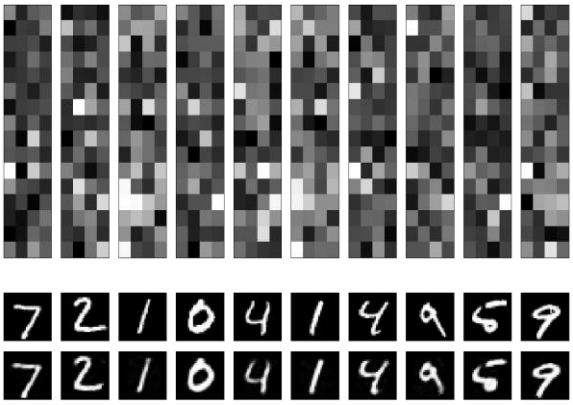

Fig3: Detect using Auto Encoders

V. CONCLUSION

The internet and a plethora of digital gadgets have made living very easy in the modern world. Like everything good, there are drawbacks, and the modern digital world is no different. Even if the internet has changed our lives, protecting personal information is still a very difficult issue. That's what causes cyberattacks. Finding intrusions is essential. Techniques like DT, SVM, and ANN are not very good at detecting conventional tactics like denial-of-service attacks, phishing schemes, and malicious software attacks. Using the publicly accessible UNSW NB-15 input dataset, this study provides a novel CNN-Autoencoders Model to automatically improve detection attacks and increase accuracy.

ACKNOWLEDGMENT

Research Group has played a significant role in creating an atmosphere that encourages creativity and teamwork. We extend our sincere gratitude to our friends and colleagues who offered insightful criticism, supportive encouragement, and insightful insights throughout the writing process. Their knowledge and viewpoints have tremendously enhanced our job.

REFERENCE

[1] Pant M, Kumar S (2022) Particle swarm optimization and intuitionistic fuzzy set-based novel method for fuzzy time series forecasting. Granule Compute 7(2):285–303

[2] Santucci, V.; Bialetti, M.; Milani, A. An algebraic framework for swarm and evolutionary algorithms in combinatorial optimization. Swarm Evil. Compute. 2020, 55, 100673.

[3] Sarhan, Mohanad, Siamak Laight, Nour Moustafa, and Marius Portmann. NetFlow Datasets for Machine Learning-Based Network Intrusion Detection Systems. In Big Data Technologies and Applications: 10th EAI International Conference, BDTA 2020, and 13th EAI International Conference on Wireless Internet, Wiconi 2020, Virtual Event, December 11, 2020, Proceedings (p. 117). Springer Nature.

[4] Xu, H.; Cao, Q.; Fu, H.; Fu, C.; Chen, H.; Su, J. Application of Support Vector Machine Model Based on an Improved Elephant Herding Optimization Algorithm in Network Intrusion Detection; Springer: Singapore, 2019; pp. 283–295.

[5] Prasad, C., Subramanian, K. and Sujatha, P. (2019). Cost–benefit analysis for optimal DG placement in distribution systems by using elephant herding optimization algorithm. Renewables: Wind, Water, and Solar, 6(1).

[6] Cassetto, O. What Is UBA, UEBA, & SIEM? Security Management Terms Defined. Exabeam, 13 July 2017. Available online: https://www.exabeam.com/siem/uba-ueba-siem-security-management-terms-defined-exabeam/(accessed on 8 June 2023).

[7] Rajasekaran, A.S.; Maria, A.; Rajagopal, M.; Lorincz, J. Blockchain Enabled Anonymous Privacy-Preserving Authentication Scheme for Internet of Health Things. Sensors 2022, 23, 240. [CrossRef] [PubMed]

[8] Warner, J. User Behavior Analytics (UBA/UEBA): The Key to Uncovering Insider and Unknown Security Threats.

[9] Shashanka, M.; Shen, M.-Y.; Wang, J. User and entity behavior analytics for enterprise security. In Proceedings of the 2016 IEEE International Conference on Big Data (Big Data), Washington, DC, USA, 5–8 December 2016. [CrossRef]

[10] Eberle, W.; Graves, J.; Holder, L. Insider Threat Detection Using a Graph-Based Approach. J. Appl. Secur. Res. 2010, 6, 32–81. [CrossRef].