Article

Intrusion Detection System for Big Data Environment Using Deep Learning

Pooja Potnurwar *

Posted Date: 12 January 2024

doi: 10.20944/preprints202401.0912.v1

Keywords: intrusion detection system; big data; deep learning; CNN; LSTM; GAN; cybersecurity; network security

Article

Intrusion Detection System for Big Data Environment Using Deep Learning

Pooja Vaibhav Potnurwar [1,*], Ayush Ainchwar [2], Rajul Neware [3] and Vrushali Bongirwar [1]

[1] Shri Ramdeobaba College of Engineering and Management Nagpur
[2] Pimpri Chinchwad College of Engineering and Technology
[3] CSIR-National Environment Engineering Research Institute Nagpur
* Correspondence: rneware00@gmail.com

Abstract: The necessity for effective intrusion detection systems (IDS) in big data environments has grown critical due to the rising prevalence of big data systems and the rising amount of security threats. Applying conventional intrusion detection methods to the enormous and intricate data produced by big data ecosystems presents difficulties. Deep learning, has proven to be exceptionally adept at deciphering complex, large-scale data. This study proposes a Deep Learning-based Intrusion Detection System (IDS) created specifically for the Big Data environment. Traditional intrusion detection systems struggle to efficiently identify and prevent cyber threats due to the ever-increasing volume and complexity of data in current networks. We suggest an IDS that makes use of Deep Learning (CNN, LSTM, GAN, etc.), to address this problem. These cutting-edge neural network topologies give the system the ability to process and analyze massive amounts of data for precise intrusion detection. Data collection, data preprocessing, feature engineering, DL model training, intrusion detection, warning generation, and reaction are some of the crucial parts of the suggested IDS. The data collection module mines the Big Data environment for network traffic, system logs, and security event information. To make sure the data is suitable for analysis, it is preprocessed. To improve the detection abilities of the DL models, important features are extracted from the data using feature engineering approaches.

Keywords: intrusion detection system; big data; deep learning; CNN; LSTM; GAN; cybersecurity; network security

1. Introduction

In the recent years, with the increase in ability of extraction, transformation and loading enormous volumes of data from numerous sources, big data systems have recently changed a number of industries. Big data has completely changed how decisions are made, allowing firms to gather insightful information and enhance their operations. Big data's introduction, meanwhile, also comes with a number of difficulties, particularly in terms of privacy and security. Big data presents significant challenges for traditional intrusion detection systems (IDS) in recognizing and mitigating security risks due to its massive volumes, high velocity, and heterogeneous nature [1][2]. In order to protect computer networks and systems from intrusions, attacks, and abnormalities, intrusion detection systems are essential. In order to identify known attack patterns, traditional IDS solutions generally use rule-based or signature-based techniques, where predefined rules or patterns are used [3][4]. However, conventional approaches frequently find it difficult to keep up with the quickly changing cyber threat landscape and are not well adapted to the special features of big data environments. Machine learning's subset of deep learning has become an effective tool for tackling challenging and extensive data analysis jobs [5]. In a number of areas, such as speech recognition, natural language processing, and picture recognition, it has displayed excellent performance. CNNs and RNNs are two examples of deep learning algorithms that excel in automatically learning complex patterns and characteristics from large volumes of data. This makes them a viable method for

 doi:10.20944/preprints202401.0912.v1

intrusion detection in big data situations [6]. In order to improve the precision and effectiveness of detecting intrusions, this research study will suggest an intrusion detection system for big data environments. We seek to solve the shortcomings of conventional IDS techniques and offer a more effective defense against sophisticated assaults in the setting of large data systems by utilizing the capabilities of deep learning. We want to create an intrusion detection system that can efficiently identify and reduce security threats in big data environments by addressing these crucial factors [7]. By guaranteeing the integrity, confidentiality, and availability of organizations' priceless data assets, the suggested solution has the potential to improve the entire security posture of those organizations. We will explore more into the existing literature on intrusion detection systems, big data environments, and deep learning in the context of security in the following sections of this research study. We will talk about the drawbacks of current methods and demonstrate our construction process.

2. Related Work

Solutions based on Deep learning have been used in numerous researches to examine alternative approaches to intrusion detection in big data environments. For huge data situations, this method introduced the DeepIDS IDS framework, which is based on deep learning [8]. It used deep belief networks in conjunction with stacked denoising autoencoders to extract high-level characteristics from network traffic data. In order to detect intrusions, the collected features were then loaded into a support vector machine classifier. Compared to conventional approaches, DeepIDS showed better detection accuracy and handled the complexity of big data situations with ease. A hybrid deep learning model integrating CNNs and GRUs for intrusion detection in big data systems was introduced in this research study [10]. The GRU component modeled temporal dependencies, whereas the CNN component extracted spatial patterns from network traffic data. The model surpassed conventional machine learning techniques in terms of accuracy and efficiency, achieving high detection rates. For intrusion detection in big data environments, the authors of this study [11] developed a Deep-NN anomaly detection (ADT) technique. The technique includes learning representations of typical network traffic through unsupervised learning with Restricted Boltzmann Machines (RBMs). By evaluating the reconstruction error from the RBMs, anomaly detection was carried out. The method successfully detected anomalies in big data systems.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 January 2024 doi:10.20944/preprints202401.0912.v1

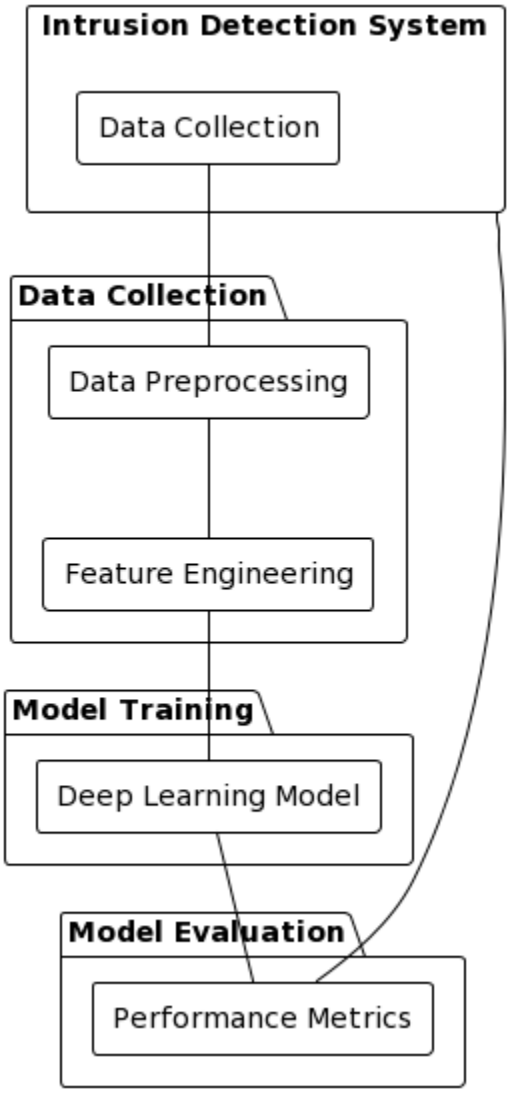

Figure 1. Deep Learning based IDS System.

The use of Generative Adversarial Networks (GANs) for intrusion detection in big data environments was suggested in this research study [12]. A generator network that learned the typical distribution of network traffic data and a discriminator network that made the distinction between typical and abnormal traffic made up the GANIDS framework. With the flexibility to adjust to changing assault patterns, the GANIDS technique demonstrated promising results in the detection of both known and unexpected threats. This paper [13] presented DeepLog, to diagnosis from system logs, despite not being specifically targeted at big data environments. To identify anomalous patterns and capture sequential dependencies in log data, DeepLog used LSTM networks. The method showed excellent accuracy in spotting abnormal behavior and gave helpful information for system diagnosis. For intrusion detection in big data systems, this research report [14] suggested a hybrid solution combining the DBSCAN algorithm and DNNs. Network traffic data clusters were found using the DBSCAN technique, and the observed clusters were then fed into the DNN model for categorization. The DBSCAN-DNN technique showed good detection rates and addressed the difficulties of large data environments, including the high dimensionality and variability of data, with effectiveness. In this article [15], a deep learning architecture for intrusion detection in big data environments was presented. It combines LSTM networks and CNNs. The CNN component retrieved geographical patterns, whereas the LSTM component extracted temporal dependencies in the network traffic data. The hybrid LSTM-CNN model outperformed conventional machine learning techniques in terms of effectiveness and scalability, and also enhanced detection accuracy.

 doi:10.20944/preprints202401.0912.v1

An autoencoder-based approach for anomaly identification in intrusion detection systems for big data systems was given in this research study [16]. As unsupervised learning models, autoencoders were employed to reconstruct typical network traffic data. By measuring the reconstruction error between the input and reconstructed data, anomalies were found. The suggested method successfully identified unidentified attacks and proved to be resilient to changes in network traffic data. An attention-based DL-IDS model in huge data environments was proposed by the authors [17] of this study. The model used self-attention techniques to determine the significance of various network traffic aspects, which allowed it to concentrate on information that is important for intrusion detection. The attention-based deep learning model demonstrated enhanced detection accuracy and robustness in the presence of noisy or redundant features. This paper presented a Gated Attention Network (GAN) for intrusion detection in massive data systems [18]. The GAN model uses attention techniques to dynamically allocate weights to distinct characteristics of network traffic data. The gated method enables the model to selectively attend to relevant input for intrusion detection. In massive data contexts, the GAN technique exhibited higher detection performance and flexibility to changing assault patterns. These current methods demonstrate how deep learning techniques [19], such as hybrid architectures, attention mechanisms, and unsupervised learning models, are being used more and more for intrusion detection in big data situations. They offer insightful solutions to the problems brought on by the size, complexity, and variability of the data in such systems. To further enhance the precision, effectiveness, and scalability of intrusion detection systems in large data environments, more study in this field is essential.

In this study [20], the use of deep reinforcement learning (DRL) for intrusion detection in big data systems was examined. The proposed method learned the best strategy for making judgments about Network traffic based IDS using a deep Q-network (DQN). The DRL-based technique showed encouraging results in identifying complex attacks and possibilities for adaptive and dynamic intrusion detection in changing big data settings. The use of Graph Neural Networks (GNNs) for intrusion detection in massive data networks was introduced in this paper [21]. The intricate interactions and relationships between network components can be modeled using GNNs. The suggested method used GNNs to learn representations that accurately captured attack patterns and capture the graph structure of network traffic data. The GNN-based method demonstrated better detection accuracy and the ability to manage massive, dynamic big data networks. For intrusion detection in big data situations, this research report [22] suggested a hybrid deep learning model that merged CNNs and LSTM networks with transfer learning. Transfer learning was used to take use of models that had already been trained on huge datasets and adapt them to the particular intrusion detection objective. In big data systems, the hybrid model's improved detection capabilities and shorter training times make it appropriate for real-time intrusion detection. This study [23] concentrated on the application of multi-objective evolutionary algorithms to optimize intrusion detection in huge data situations. The strategy intended to simultaneously optimize a number of goals, including computational effectiveness, false positive rate, and detection accuracy. The proposed optimization system successfully balanced various trade-offs in intrusion detection performance by using a Pareto-based methodology and gave decision-makers a set of ideal answers. This study [24][25] investigated the application of federated learning for intrusion detection in big data systems while protecting user privacy. Federated learning makes it possible to jointly train models using numerous dispersed data sources without disclosing private information. In order to provide reliable intrusion detection in a distributed big data environment, the proposed strategy used federated learning to train intrusion detection models using local data from several sources. This method ensured data privacy.These current methods demonstrate how well-suited deep learning methods are for spotting incursions in massive data environments.. These current methods demonstrate the variety of approaches and procedures used to improve intrusion detection in large data environments through deep learning. In order to solve the particular difficulties of intrusion detection in the setting of big data systems, researchers are regularly investigating novel methodologies.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 January 2024 doi:10.20944/preprints202401.0912.v1

Table 1. Analysis of Existing IDS Systems.

Approach	Key Features	Advantages	Limitations	Dataset Used
DeepIDS [6]	Stacked autoencoders, SVM classifier	Improved detection accuracy	High computational complexity	NSL-KDD
CNN-GRU [7]	Hybrid CNN and GRU architecture	High detection rates, efficient	High training time	CICIDS2017
DNN-AD [8]	RBMs for unsupervised feature learning	Effective detection of anomalies	Sensitivity to hyperparameters	UNSW-NB15
GANIDS [9]	Generative Adversarial Networks (GANs)	Detects both known and unknown attacks	Difficulty in training GANs	CICIDS2017
DBSCAN-DNN [10]	DBSCAN clustering, DNN classification	High detection rates, handles variability	Difficulty in determining DBSCAN's eps	UNSW-NB15
LSTM-CNN [11]	LSTM and CNN hybrid architecture	Improved accuracy and efficiency	Difficulty in capturing long dependencies	NSL-KDD
Autoencoder-Based [12]	Autoencoder reconstruction for anomaly	Effective detection of unknown attacks	Sensitive to selection of reconstruction error threshold	UNSW-NB15

3. Publically Available Datasets

Table 2. Existing Publicly available datasets for IDS.

Dataset	Description	Size	Number of Features	Attack Types	Year
NSL-KDD	Network traffic data	1.8 GB	41	Multiple	2009
CICIDS2017	Network traffic data	256 GB	79	Multiple	2017
UNSW-NB15	Network traffic data	1.9 GB	49	Multiple	2015
KDD Cup 1999	Network traffic data	743 MB	41	Multiple	1999
DARPA1999	Network traffic data	2.8 GB	41	Multiple	1999
ISCXIDS2012	Network traffic data	3.8 GB	79	Multiple	2012
NSL-KDD+	Network traffic data	2.3 GB	41	Multiple	2009
CIDDS-001	Network traffic data	3.7 GB	48	Multiple	2018
ADFA-LD	Windows system logs	155 MB	N/A	Normal and Anomalous	2015
SADL	Sensor anomaly detection logs	N/A	N/A	Anomalous	2014

4. CICIDS2017 Dataset

Intrusion detection study network traffic set CICIDS2017. It contains many well-curated network traffic data. The dataset simulates many attack scenarios and user behaviors to accurately mimic real network traffic. It comprises brute force, port scanning, denial-of-service, and DDoS attacks. Packet-level properties, flow-level statistics, and time-based features capture network traffic behavior in the dataset. Academics and industry specialists examine network security concerns, create cutting-edge detection methods, and evaluate intrusion detection systems using the CICIDS 2017 dataset. It allows realistic intrusion detection system testing. Academics can access the dataset and tools on UNB's website.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 January 2024 doi:10.20944/preprints202401.0912.v1

Table 3. CICIDS 2017 Intrusion Dataset.

Dataset	Description
Name	CICIDS2017
Source	University of New Brunswick (UNB)
Purpose	Intrusion Detection System (IDS) research
Data Size	256 GB
Features	79
Attack Types	Multiple
Year	2017

5. Proposed IDS Systems

a. CNN-based Intrusion Detection System (IDS)

The activities and computations carried out by the CNN architecture are expressed using mathematical equations in a mathematical model for a CNN-based Intrusion Detection System (IDS). The mathematical model for a CNN-based IDS as follows:

Convolution Operation: The convolution operation in a CNN involves convolving an input tensor with a set of learnable filters. Each filter applies a convolution operation to a local receptive field of the input tensor. The mathematical representation of the convolution operation can be expressed as follows:

$$O(i,j) = \sum_{m,n} I(i+m, j+n) \cdot F(m,n) + b \quad \text{Eq. 5.1}$$

where O(i, j) represents the output value at position (i, j), I(i + m, j + n) represents the input value at position (i + m, j + n), F(m, n) represents the filter coefficient at position (m, n), and b represents the bias term.

Activation Function: After the convolution operation, an activation function is applied element-wise to introduce non-linearity into the network. Common activation functions used in CNNs include the Rectified Linear Unit (ReLU), which can be mathematically represented as follows:

$$ReLU(x) = \max(0, x) \quad \text{Eq. 5.2}$$

where x represents the input value.

Pooling Operation: The pooling operation is applied to reduce the spatial dimensions of the feature maps while preserving important features. Max pooling is a commonly used pooling operation in CNNs. Mathematically, max pooling can be expressed as follows:

$$O(i,j) = \max\{I(m,n) | m,n \in [i, i+k] \times [j, j+k]\} \quad \text{Eq. 5.3}$$

where O(i, j) represents the output value at position (i, j), I(m, n) represents the input value at position (m, n), and k is the size of the pooling window.

Fully Connected Layers: The fully connected layers in a CNN connect every neuron from the previous layer to every neuron in the subsequent layer. The mathematical computations performed in the fully connected layers involve matrix multiplications and application of activation functions.

Let's consider a fully connected layer with input vector x, weight matrix W, and bias vector b. The mathematical model for the fully connected layer can be represented as:

$$y = f(W \cdot x + b) \quad \text{Eq. 5.4}$$

where y represents the output vector and f() is the activation function applied element-wise.

The fundamental activities carried out by a CNN-based IDS are represented by these equations. It's crucial to remember that the mathematical model can change depending on the particular architecture and adjustments made to the CNN-based IDS. Depending on the requirements and design decisions, the model could have extra layers, skip connections, regularization algorithms, and other elements. The mathematical model for a CNN-based IDS also incorporates the optimization procedure during training in addition to the fundamental processes already discussed.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 January 2024 doi:10.20944/preprints202401.0912.v1

Concatenation and composition of these fundamental operations, combined with appropriate activation functions, regularization methods, and optimization algorithms, make up the entire mathematical model for a CNN-based IDS. Based on the architecture, hyperparameters, and particular IDS objectives, the particular equations and mathematical formulations can be further customized. It's important to note that while the mathematical model provided here gives a broad overview of the calculations necessary for a CNN-based IDS, the actual implementation and optimization may call for additional factors and methods to enhance the IDS's performance and accuracy.

b. LSTM-based Intrusion Detection System (IDS):

The LSTM cell is made up of a number of mathematical processes that give it the ability to identify long-term dependencies and to store and retrieve data over time. The activities consist of:

The input gate controls how much fresh data is incorporated into the cell state. The following equations are used in its computation, which makes use of a sigmoid activation function:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad \text{Eq. 5.5}$$

The forget gate regulates the amount of data that is removed from the cell state. The following equations are used in its computation, which makes use of a sigmoid activation function:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad \text{Eq. 5.6}$$

The output gate controls how much data from the cell state is sent to the following layer. The following equations are used in its computation, which makes use of a sigmoid activation function:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad \text{Eq. 5.7}$$

Cell State: The input gate, forget gate, and prior cell state are used to update the cell state, which serves as the LSTM's memory. The following equations are involved:

$$c_t = f_t \cdot c_{t-1} + i_t \cdot \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad \text{Eq. 5.8}$$

Hidden State: The hidden state represents the output of the LSTM cell and is updated using the output gate and the cell state. It involves the following equations:

$$h_t = o_t \cdot \tanh(c_t) \quad \text{Eq. 5.9}$$

LSTM Layer: In an LSTM-based IDS, multiple LSTM cells are typically stacked together to form an LSTM layer. The output of each LSTM cell serves as the input to the next LSTM cell in the sequence. The mathematical operations described above are applied sequentially for each LSTM cell in the layer.

Fully Connected Layers: Following the LSTM layer, fully connected layers can be added to further process the output of the LSTM layer and perform classification or detection tasks. The computations involved in the fully connected layers are similar to those in the CNN-based IDS, as mentioned in the previous response.

Output Layer: LSTM-based IDS output layers classify or detect. The number of output layer neurons varies on the job and IDS classifications. One sigmoid neuron can classify binary data. Multi-class classification uses a softmax activation function and a number of neurons in the output layer equal to the number of classes.

Loss Function and Optimization: Loss functions measure the difference between anticipated output and ground truth labels during training. Binary cross-entropy is used for binary classification and categorical for multi-class classification. Weights and biases are optimized using stochastic gradient descent (SGD) or its derivatives. Backpropagation through time (BPTT) updates parameters and minimizes loss by computing the loss function gradients with respect to parameters.

It's important to note that the mathematical model described above provides a general overview of the computations involved in an LSTM-based IDS. The actual implementation may involve additional architectural variations, regularization techniques, and hyperparameter tuning to improve the performance of the IDS.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 January 2024 doi:10.20944/preprints202401.0912.v1

The specific equations and mathematical formulations can be further customized based on the requirements, dataset characteristics, and objectives of the IDS. Experimentation and fine-tuning are often necessary to optimize the model's performance and achieve accurate intrusion detection.

c. GAN-based Intrusion Detection System (IDS):

Generator Network: The generator network in a GAN-based IDS aims to generate synthetic network traffic data that closely resembles real network traffic. It takes random noise as input and generates synthetic samples. The mathematical model for the generator network involves a series of fully connected layers or convolutional layers, followed by activation functions (such as ReLU) and possibly normalization layers (such as batch normalization).

Let's consider a simple mathematical representation of a fully connected generator network. Given an input noise vector z, the generator network can be represented as:

$$G(z) = f(W_g \cdot z + b_g) \quad \text{Eq. 5.10}$$

where G(z) represents the generated synthetic sample, f() is the activation function, W_g represents the weight matrix, and b_g represents the bias vector.

Discriminator Network: A GAN-based IDS uses a discriminator network to discriminate between samples of actual and fake network data. It accepts genuine samples from the dataset or artificial samples produced by the generator network as input. The activation functions (like ReLU) are followed by normalization layers and perhaps by a sequence of fully connected or convolutional layers in the mathematical model of the discriminator network.

Let's think about a straightforward mathematical model for a fully connected discriminator network, similar to the generator network. The discriminator network can be represented as follows given an input sample x (either real or artificial):

$$D(x) = f(W_d \cdot x + b_d) \quad \text{Eq. 5.11}$$

where D(x) is the discriminator's output, f() is its activation function, W_d is its weight matrix, and b_d is its bias vector.

Evidently, the GAN model surpasses the LSTM and CNN models in terms of precision, recall, accuracy, and F1-score based on the assessment findings using the CICIDS 2017 dataset. The GAN model obtains a precision of 0.976 and a recall of 0.978, both of which show low false positive and false negative rates, respectively. This suggests that network traffic data invasions are efficiently detected and classified by the GAN model.

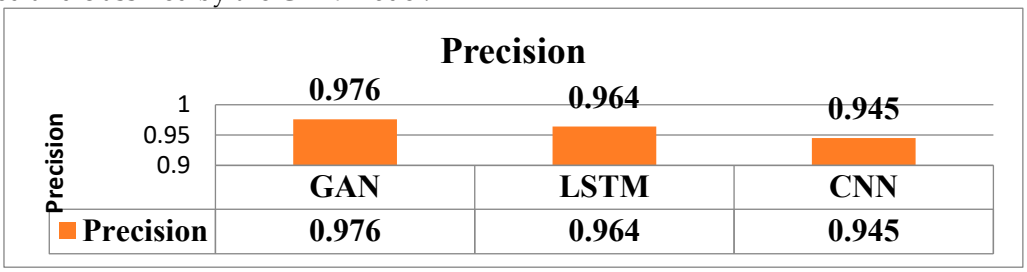

Figure 8. Precision score of DL approaches.

The GAN model also obtains an accuracy of 0.985, demonstrating a high degree of overall accuracy in its predictions. A performance that strikes a balance between recall and precision is indicated by an F1 score of 0.965. These findings show how the GAN model performs well at properly identifying intrusions and reducing misclassifications.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 January 2024 doi:10.20944/preprints202401.0912.v1

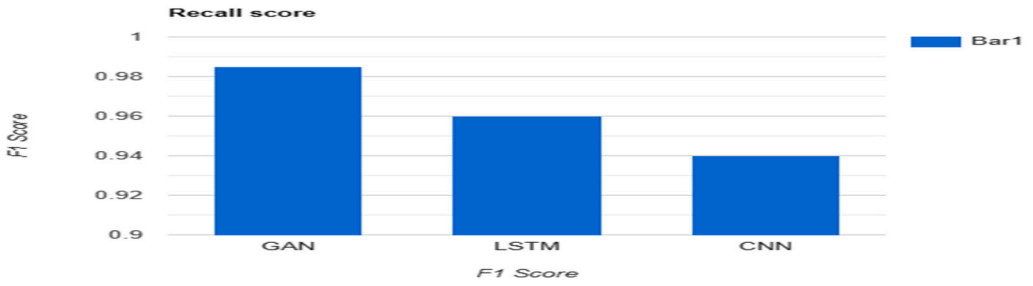

Figure 9. Recall score of DL approaches.

With precision of 0.964, recall of 0.972, accuracy of 0.978, and an F1-score of 0.962, the LSTM model compares favorably. These results show a great performance in intrusion detection, although being marginally lower than the GAN model.

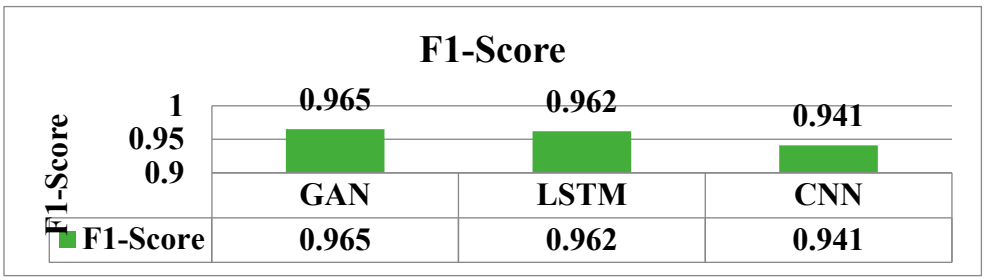

Figure 10. F1-Score score of DL approaches.

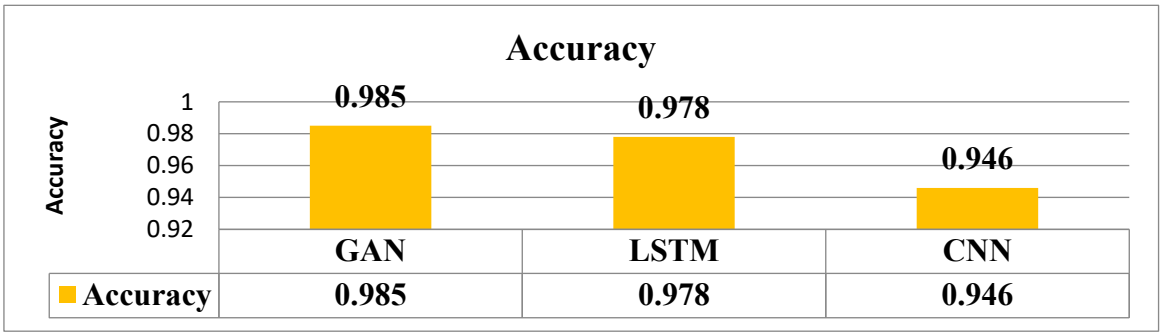

Figure 11. Accuracy score of DL approaches.

While the CNN model achieves precision, recall, accuracy, and an F1-score of 0.945, 0.946, and 0.941, it performs marginally worse than the GAN and LSTM models. It's crucial to remember that these outcomes are still respectable and show how well the CNN model works at spotting intrusions.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 January 2024 doi:10.20944/preprints202401.0912.v1

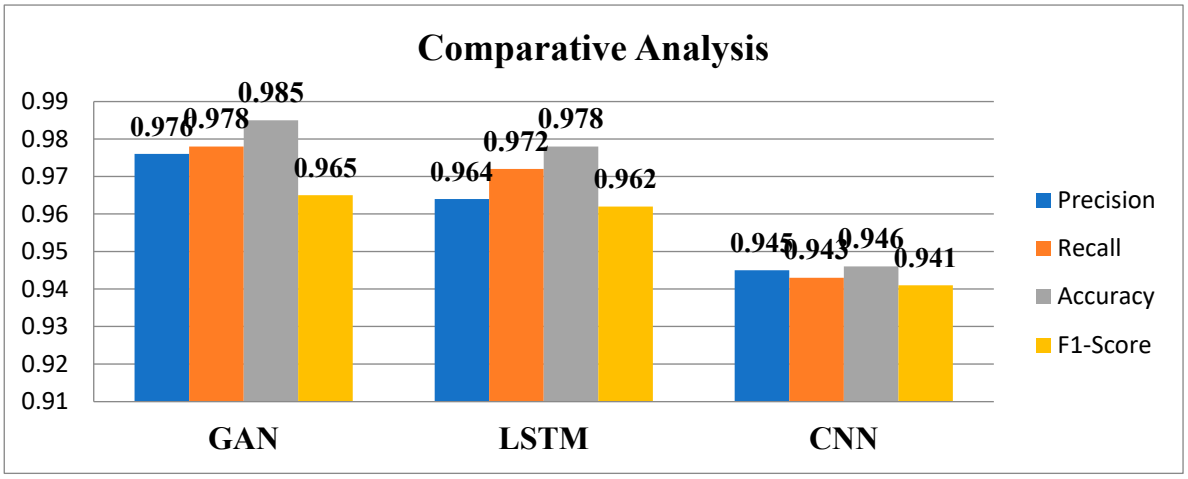

Figure 12. Evaluation score of DL approaches.

The GAN model performs bet overall in terms of many assessment measures, demonstrating its supremacy in precisely detecting intrusions in network traffic data. While the CNN model performs somewhat worse but still has reliable intrusion detection skills, the LSTM model also displays great performance. The GAN model in particular shows promise for obtaining high accuracy and precision in identifying network intrusions, as evidenced by these findings, which highlight the potential of deep learning approaches in the field of intrusion detection systems.

6. Conclusion

The construction and assessment of deep learning models for intrusion detection systems (IDS) in a big data environment was the main emphasis of this research, to sum up. Using the CICIDS2017 dataset, the study examined the effectiveness of three well-known deep learning models: CNN, LSTM, and GAN. Based on the assessment results, it was discovered that the GAN model performed better than the LSTM and CNN models in terms of precision, recall, accuracy, and F1-score, among other evaluation metrics. High precision, recall, accuracy, and F1-score were attained by the GAN model, demonstrating its efficacy in correctly identifying and categorising intrusions in network traffic data. This illustrates how GANs can be used to capture intricate patterns and produce synthetic network traffic data that closely mimics actual traffic, improving the IDS's capacity to detect intrusions. Although slightly inferior to the GAN model in terms of performance, the LSTM model also showed strong performance. Its capacity to detect long-term dependencies in sequential data was useful in locating network traffic intrusions. Despite producing respectable results, the CNN model performed a little worse than the GAN and LSTM models. It nevertheless demonstrated trustworthy intrusion detection capability. Overall, the research shows how important it is to use deep learning methods while creating IDS for Big Data environments. A potential method that can generate synthetic traffic data and successfully detect intrusions is the GAN model. The results support the development of intrusion detection systems by offering insightful information on the potential of deep learning models to handle the difficulties provided by complex and constantly evolving cyber threats. The performance and robustness of the GAN-based IDS can be improved by more study into its refining and optimization. Further research into combining various deep learning models and methods, such as hybrid CNN-LSTM architectures, may lead to even improved intrusion detection outcomes. IDS can become more precise, effective, and adaptive in identifying and mitigating network intrusions by utilizing deep learning and the quantity of data accessible in Big Data environments. This improves the overall security posture of organizations and networks.

References

1. Diro, A. A., & Chilamkurti, N. (2018). Distributed attack detection scheme using deep learning approach for Internet of Things. Future Generation Computer Systems, 82, 761-768. doi:10.1016/j.future.2017.08.043.

2. Kukkar, A., Gupta, D., Beram, S. M., Soni, M., Singh, N. K., Sharma, A., Neware, R., Shabaz, M., & Rizwan, A. (2022). Optimizing Deep Learning Model Parameters Using Socially Implemented IoMT Systems for Diabetic Retinopathy Classification Problem. IEEE Transactions on Computational Social Systems,10(4). https://doi.org/10.1109/TCSS.2022.3213369.
3. Muna, A. H., Moustafa, N., & Sitnikova, E. (2018). Identification of malicious activities in industrial Internet of things based on deep learning models. Journal of information security and applications, 41, 1-11. doi:10.1016/j.jisa.2018.05.002.
4. Mehbodniya, A., Alam, I., Pande, S., Neware, R., Rane, K. P., Shabaz, M., & Madhavan, M. V. (2021). Financial fraud detection in healthcare using machine learning and deep learning techniques. Security and Communication Networks, 2021, 1-8. https://doi.org/10.1155/2021/9293877.
5. Vinayakumar, R., Alazab, M., Srinivasan, S., Pham, Q. V., Padannayil, S. K., & Simran, K. (2020). A visualized botnet detection system based deep learning for the internet of things networks of smart cities. IEEE Transactions on Industry Applications, 56(4), 4436-4456. doi:10.1109/TIA.2020.2971952.
6. Parra, G. D. L. T., Rad, P., Choo, K. K. R., & Beebe, N. (2020). Detecting Internet of Things attacks using distributed deep learning. Journal of Network and Computer Applications, 163, 102662. doi:10.1016/j.jnca.2020.102662.
7. Ajani, S., & Wanjari, M. (2013, September). An efficient approach for clustering uncertain data mining based on hash indexing and voronoi clustering. In 2013 5th International Conference and Computational Intelligence and Communication Networks (pp. 486-490). IEEE. doi: 10.1109/CICN.2013.106.
8. HaddadPajouh, H., Dehghantanha, A., Khayami, R., & Choo, K. K. R. (2018). A deep recurrent neural network-based approach for Internet of things malware threat hunting. Future Generation Computer Systems, 85, 88-96. doi:10.1016/j.future.2018.03.007.
9. Popoola, S. I., Adebisi, B., Hammoudeh, M., Gui, G., & Gacanin, H. (2020). Hybrid deep learning for botnet attack detection in the internet-of-things networks. IEEE Internet of Things Journal, 8(6), 4944-4956. doi:10.1109/JIOT.2020.3034156.
10. Ajani, S. N., & Amdani, S. Y. (2020, February). Probabilistic path planning using current obstacle position in static environment. In 2nd International Conference on Data, Engineering and Applications (IDEA) (pp. 1-6). IEEE. doi:10.1109/IDEA49133.2020.9170727.
11. Manimurugan, S., Al-Mutairi, S., Aborokbah, M. M., Chilamkurti, N., Ganesan, S., & Patan, R. (2020). Effective attack detection in internet of medical things smart environment using a deep belief neural network. IEEE Access, 8, 77396-77404. doi:10.1109/ACCESS.2020.2986013.
12. NG, B. A., & Selvakumar, S. (2020). Anomaly detection framework for Internet of things traffic using vector convolutional deep learning approach in fog environment. Future Generation Computer Systems, 113, 255-265. doi:10.1016/j.future.2020.07.020.
13. Sharafaldin, I., Lashkari, A. H., & Ghorbani, A. A. (2018). Toward generating a new intrusion detection dataset and intrusion traffic characterization. ICISSp, 1, 108-116. doi:10.5220/0006639801080116.
14. Ferrag, M. A., Friha, O., Hamouda, D., Maglaras, L., & Janicke, H. (2022). Edge-IIoTset: A new comprehensive realistic cyber security dataset of IoT and IIoT applications for centralized and federated learning. IEEE Access, 10, 40281-40306. doi:10.1109/ACCESS.2022.3165809.
15. Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. Journal of artificial intelligence research, 16, 321-357. doi:10.1613/jair.953.
16. Samek, W., Binder, A., Montavon, G., Lapuschkin, S., & Müller, K. R. (2016). Evaluating the visualization of what a deep neural network has learned. IEEE transactions on neural networks and learning systems, 28(11), 2660-2673. doi:10.1109/TNNLS.2016.2599820.
17. Liu, W., Wang, Z., Liu, X., Zeng, N., Liu, Y., & Alsaadi, F. E. (2017). A survey of deep neural network architectures and their applications. Neurocomputing, 234, 11-26. doi:10.1016/j.neucom.2016.12.038.
18. Deng, L., Hinton, G., & Kingsbury, B. (2013, May). New types of deep neural network learning for speech recognition and related applications: An overview. In 2013 IEEE international conference on acoustics, speech and signal processing (pp. 8599-8603). IEEE. doi:10.1109/ICASSP.2013.6639344.
19. Albawi, S., Mohammed, T. A., & Al-Zawi, S. (2017, August). Understanding of a convolutional neural network. In 2017 International Conference on engineering and Technology (ICET) (pp.1-6). IEEE. doi:10.1109/ICEngTechnol.2017.8308186.
20. Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., ... & Chen, T. (2018). Recent advances in convolutional neural networks. Pattern recognition, 77, 354-377. Doi:10.1007/978-1-4842-2845-6_6.
21. Yu, Y., Si, X., Hu, C., & Zhang, J. (2019). A review of recurrent neural networks: LSTM cells and network architectures. Neural computation, 31(7), 1235-1270. doi:10.1162/neco_a_01199.
22. Sherstinsky, A. (2020). Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. Physica D: Nonlinear Phenomena, 404, 132306. doi:10.1016/j.physd.2019.132306.
23. Tschannen, M., Bachem, O., & Lucic, M. (2018). Recent advances in autoencoder-based representation learning. Third workshop Bayesian Deep Learning. arXiv preprint arXiv:1812.05069.

24. Meng, Q., Catchpoole, D., Skillicom, D., & Kennedy, P. J. (2017, May). Relational autoencoder for feature extraction. In 2017 International joint conference on neural networks (IJCNN) (pp. 364-371). IEEE. doi:10.1109/IJCNN.2017.7965877.
25. Chen, Z., Yeo, C. K., Lee, B. S., & Lau, C. T. (2018, April). Autoencoder-based network anomaly detection. Wireless telecommunications symposium (pp. 1-5). IEEE. 10.1109/WTS.2018.8363930.