IRJCS: International Research Journal of Computer Science
Volume 11, Issue 07, July 2024
ISSN: 2393-9842
https://www.irjcs.com/archives

Deep Learning Based Network Intrusion Detection System: A Deep Abstract Networks (DANets) Model Approach

Shahir Kottilingal*
Department of Artificial Intelligence:
Wakeb Data Co., KSA
shahirkottilingal@gmail.com

Publication History
Manuscript Reference No: IRJCS/RS/Vol.11/Issue07/JYCS10080
Research Article | Open Access | Double-Blind Peer-Reviewed
Article ID: IRJCS/RS/Vol.11/Issue06/JYCS10080 | Received: 24, June 2024, Revised: 05, July 2024 | Accepted: 17 July 2024
Published Online: 20, July 2024 http://www.irjcs.com/volumes/Vol11/iss-07/01.JYCS10080.pdf
Article Citation:Shahir(2024). Deep Learning Based Network Intrusion Detection System: A Deep Abstract Networks (DANets) Model Approach. IRJCS: International Research Journal of Computer Science, Volume 11, Issue 07 of 2024 pages 539-544 **doi:>** https://doi.org/10.26562/irjcs.2024.v1107.01
BibTeX Shahir@2024Deep

Abstract: The need for frequent updates on Network Intrusion Detection Systems (NIDS) is crucial for safeguarding enterprise data against cyberattacks due to the advancement in the attackers' arsenal resulting from technological advancements. Conventional signature-based NIDS often struggle to keep up with newer types of intrusions due to their sophisticated nature. This has forced the advancement of machine learning-based NIDS, which have shown promising results. There is much recent research on machine learning and deep learning-based network intrusion detection system. Also, there is few important research datasets like UNSW-NB15, KDD CUP dataset, etc published for the use of research community. In this paper, we aimed to assess the effectiveness of a Deep Abstract Networks (DANets) model in network intrusion detection. We used UNSW-NB15 dataset. The accuracy and F1 score of the proposed model were very promising.

Keywords: Deep Abstract Network, Deep Leaning, Network Intrusion Detection System, Cyber Security, Binary Classification

I. INTRODUCTION

The digital revolution has given rise to a new wave of cyber threats and skilled hackers, posing significant challenges to network security. The development of digital communications has led to the emergence of vulnerabilities that can negatively impact both individuals and organizations. It is crucial to have a robust network security system in place to safeguard confidentiality, integrity, and accessibility[1]. Network Intrusion-Detection Systems (NIDSs) are an important cybersecurity tool for network protection from the intrusion. The growth of network traffic and technological advancements in the digital space have also provided opportunities for cyber attackers. The act of exploitation involves attempting to breach network security remotely and compromise aspects such as confidentiality, integrity, and accessibility. Exploitation has a detrimental effect on the victim while simultaneously benefiting the perpetrator by providing an advantage that can be utilized for financial profit [1]. NIDS, or network intrusion detection system, is designed to keep a constant eye network traffic to identify any behaviour that could potentially violate the security policy and compromise the confidentiality, integrity, and availability of the system [2]. All the incoming and outgoing network traffic are mirrored to NIDS for performing passive traffic monitoring to detect intrusions. Conventional NIDSs are based signature and rule-based methods. These types of network intrusion detection systems encounter specific challenges and limitations. Conventional NIDSs require ongoing updates to attack signature databases to remain ahead of the constantly evolving threats. Developing sophisticated detection methods is necessary to tackle these challenges, utilizing the power of machine learning and deep learning algorithms to enhance the accuracy and efficiency of intrusion detection systems. Machine learning possesses significant potential to tackle the intricate and continuously evolving difficulties in the field of cybersecurity, as it has the capacity to adapt and learn from new data[1]. Over the past ten years, researchers have proposed various machine learning (ML) and deep learning (DL) solutions to enhance the efficiency of network intrusion detection systems (NIDS) in detecting malicious attacks. However, the substantial increase in network traffic has presented numerous challenges for NIDS systems in detecting malicious intrusions effectively. Presently, the research on employing DL techniques for NIDS is not fully explored, and there is ample scope to investigate the potential of this technology in detecting intruders within the network[3]. The availabilty of high end GPUs made exploring newer deep learning model easy in NIDS research. The aim of this research is to create a promising NIDS model using Deep Abstract Networks for the purpose of tabular data classification. This model is designed to utilize the spatial and temporal characteristics of the UNSW-NB15 benchmark dataset, thereby addressing the shortcomings of existing machine learning (ML) and deep learning (DL) models.

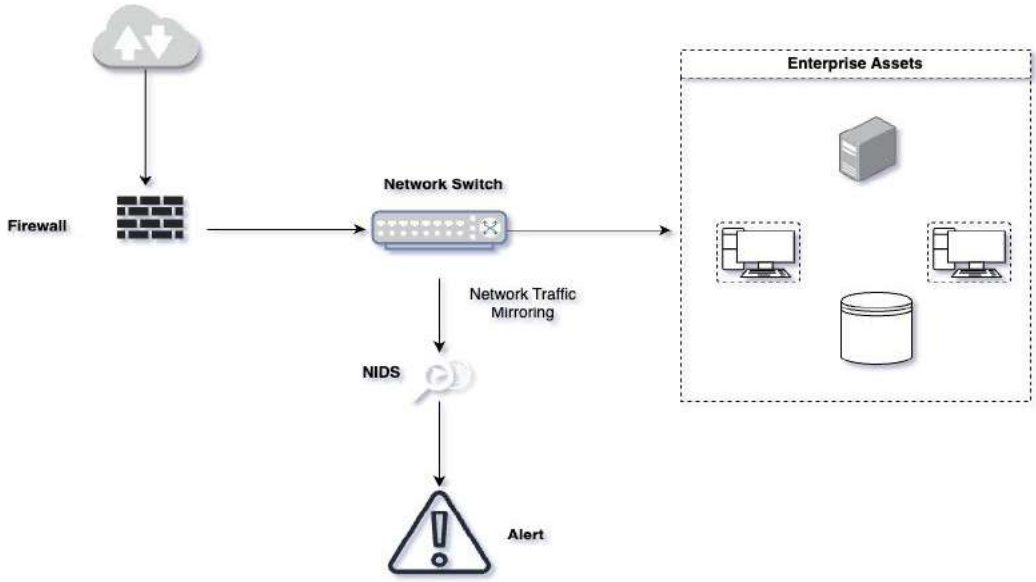


Figure 1. Network Intrusion Detection System

This research focuses on binary classification and compares its results with other benchmark machine learning models, with the aim of evaluating the proposed model's effectiveness through extensive experimentation and examining its advantages and limitations in the field of network intrusion detection. As mentioned above this study uses UNSW-NB15 dataset that is very large dataset that has a high dimension feature space. This dataset include modern low footprint attacks that was not available with previously available datsets like KDD98 and KDD99 datasets[4]. Few features of UNSW-NB15 dataset have been uncovered as contaminating to some degree. 3 features like ct_state_ttl, sttl and dttl have high contaminating effect and features like ct_dst_ltm, ct_srv_dst, ct_dst_src_ltm, ct_srv_src and ct_dst_sport_ltm have contaminating effect, but less powerful[5]. These features are excluded from the training dataset.

II. LITERATURE SURVEY

Numerous research papers offer comprehensive details on implementing NIDS with deep learning techniques. Here I will discuss about most recent research papers.

Ashiku and dagli(2021) Proposed the application of deep learning architectures to create a versatile and robust intrusion detection system (IDS) capable of detecting both familiar and unfamiliar (zero-day) network assaults. This is achieved through the use of deep neural networks (DNNs) that incorporate behavioral features, allowing the system to identify and eject intruders, thereby reducing the risk of compromise. The suggested deep learning model features a convolutional neural network (CNN) and a regularized multi-layer perceptron, rather than a fully connected feed-forward neural network (FNN). To demonstrate the effectiveness of this approach, they employed the UNSW-NB15 dataset. This dataset was used to showcase the ability of model to detect network attacks, thereby validating the proposed approach[1].

Ahmed at al.(2020) The authors elucidate the concept of IDS and subsequently present a taxonomy based on important machine learning and deep learning techniques utilized in the development of NIDS. The study offers a thorough examination of recent articles utilizing NIDS by evaluating the advantages and constraints of proposed remedies[2]. The NIDS method developed by Alsharaiah et al. (2024) integrates the capabilities of long short-term memory (LSTM) and attention mechanisms to examine the spatial and temporal characteristics of network traffic data. By employing the UNSW-NB15 network traffic dataset, their research demonstrates the advantage of integrating LSTM with attention mechanisms for improved NIDS[3].

Azam et al.(2023) investigated the most recent trends and developments in network intrusion detection systems that utilize deep learning and machine learning technologies, encompassing methodology, evaluation metrics, and data selection criteria. The objective of their work was to offer insights into constructing a highly efficient decision tree-based detection framework. They found that DL-based methods typically surpass ML-based approaches. However, implementing deep learmning in real-time NIDS can be challenging and may require substantial computational resources, including processing power and storage, which can make them difficult to implement in practice.[6].

As per Ehigiator et al(2024) AI technologies, particularly machine learning algorithms, are exceptionally adept at detecting irregular patterns that may suggest a security breach[7]. As per Bouchama and Kamal supervised learning models, such as deep neural networks, convolutional neural networks, long short-term memory, and random forests, have been shown to provide higher accuracy compared to intrusion detection systems with encoded rules[8].

Gokul et al.(2024) presented an IDS method with UNSW-NB15 and Kyoto datasets. They used SMOTE to tackle the data imbalance. The random forest model is employed to detect and classify network attack.

They concentrated on multiclass classfication problem and they achived accuracies over 99% and 98% in Kyoto and UNSW-NB15 datasets respectively [9].

Literature brings up an idea that despite the extensive efforts of researchers, Network Intrusion Detection Systems (NIDS) continue to face challenges in enhancing detection accuracy while minimizing false alarm rates and detecting new types of intrusions. As a result, machine learning (ML) and deep learning (DL)-based NIDS solutions are being increasingly deployed across networks to detect intrusions in a more efficient manner.

III. PROPOSED MODEL ARCHITECTURE

This paper concentrates on developing deep learning-based NIDS based on Deep Abstract Networks (DANets) model proposed by Chen et al. (2020). introduced Deep Abstract Networks (DANets) proposed as framework for tabular data processing. Tabular data are commonly used in various real-world applications. Despite the availability of several neural components and extensible neural networks developed by the machine learning community, only few are perfrom well with tabular data[10]. Chen et al. (2020). proposed an adaptable network model for tabular data, known as the Abstract Layer (ABSTLAY), which aims to specifically identify correlated input features and then generate higher-level features. DANETs arrange ABSTLAYs to extract higher features from the meaningful feature. Also grouping features together from different level to increase the model capability. This model designed new shortcut path that enables a high-level layer to access raw features. This shortcut path is implemented through a fundamental building block composed of ABSTLAYs. To find feature groups, ABSTLAY use feature selection functions. Next, to abstract higher-level features from groups ABSTLAY use function for feature abstracting, and finally to join all the features abstracted from various groups, it uses function for output fusion operation [Figure2].

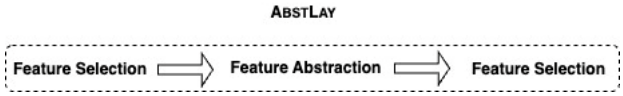
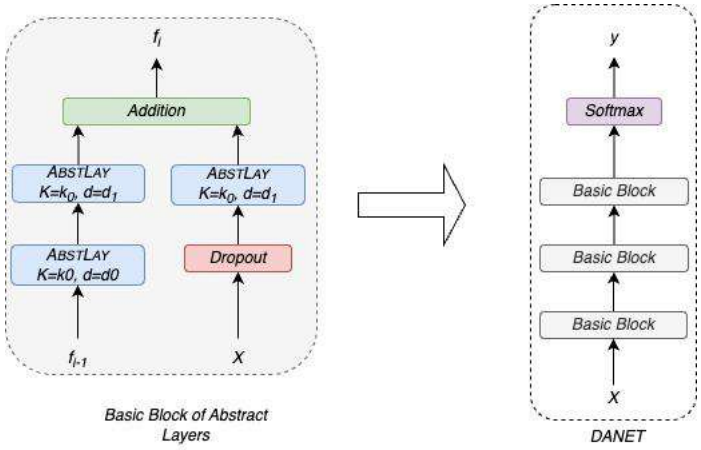


Figure 2 : Abstract Layer Functions

ABSTLAYs are used to construct the basic block, and a shortcut can be added to incorporate features extracted from the raw features to the primary model path and then stack the basic blocks in sequence to build a DANET architecture. DANETs are constructed by sequentially stacking multiple instances of this block [Figure3].



(a) (b)

Figure 3: Architecture of DANets for tabular data processing.
(a) Showing basic block specification. (b) DANets-stacking many basic blocks [10].

Results from experiments conducted on multiple public datasets have consistently demonstrated that DANets exhibit a high level of proficiency in handling tabular data for both classification and learn-to-rank tasks, displaying remarkable efficiency[10]. For classification task, Cross-entropy loss function is used with DANets. A broader and more extensive DANets can be advantageous, but it is not suggested to employ excessively deep and wide architectures because of the limited spaces of tabular features[10].

IV. METHODOLOGY

This research proposes a NIDS with deep learning framework powered with Deep Abstract Networks (DANets) using Abstract Layers (ABSTLAYs) for the purpose of classifying and to compare the effects of DANETs and the known state-of-the-art ML models. We run proposed DANets Model and other classical ML models on UNSW-NB15 binary class dataset. Hardware details: I have utilized in-house commodity hardware for training the DANets Model. Trained deep learning models with GPU. Jupyter notebook running on local machine was utilized to explore data and model training. Hardware configuration is shown in Table 1.

Table 1. Hardware Configuration

Description	Specification
Processor	11th Gen Intel Core i9-1900K @ 3.50GHZ x16
HDD	2.0 TB
RAM	128 GB
GPU	NVIDIA RTX 4090

Important steps in model development are data collection, data pre-processing, feature selection, Train and validation data split, Model Training and validation, Testing and Deployment. Detailed steps involved in model development method shown in Figure4.

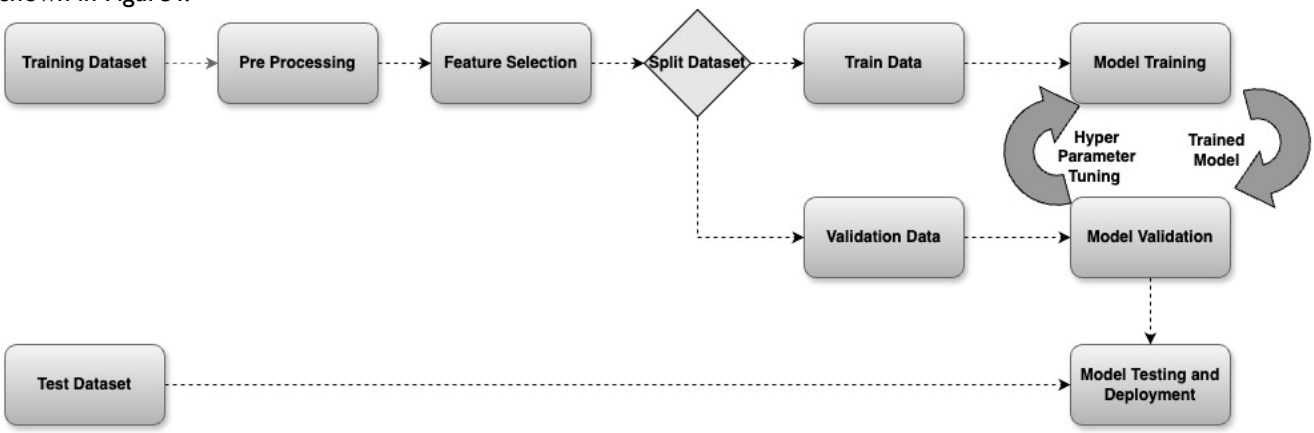


Figure 4. Steps in Training and Testing the Model

Data Collection: This study used dataset from UNSW-NB15 repository. The dataset at hand comprises a unique combination of genuine contemporary normal network traffic and synthesized modern attack activities[4]. There are training and test datasets available in the repository. Training dataset contains 175341 records and 45 columns including label. Test dataset contains 82332 records and 45 columns including label. Label column contains 0 and 1 values for normal and malicious traffic respectively. This dataset collected from UNSW repository from the below link.
https://unsw-my.sharepoint.com/personal/z5025758_ad_unsw_edu_au/_layouts/15/onedrive.aspx?ga=1&id=%2Fpersonal%2Fz5025758%5Fad%5Funsw%5Fedu%5Fau%2FDocuments%2FUNSW%2DNB15%20dataset%2FCSV%20Files%2FTraining%20and%20Testing%20Sets

Pre-processing: Pre-processing of dataset is very important in model development as it directly affect the model accuracy and performance. Pre-processing includes many steps like data cleaning, encoding of categorical variables, standardization, etc.

Feature Selection: There are 44 total number of features and label column in the dataset. Based on the paper by D'Hooge et al. (2022) I excluded features like ct_state_ttl, sttl, dttl, ct_dst_ltm, ct_srv_dst, ct_dst_src_ltm, ct_srv_src and ct_dst_sport_ltm due to its contamination impact on the model. Also, we dropped the "id" and "attack_cat" columns from the dataset. So total 34 features were there in the training dataset.

Train and Validation Split: Throughout the model development, I maintained 75% of training dataset as train data and 25% as validation data for model validation and hyper parameter tuning.

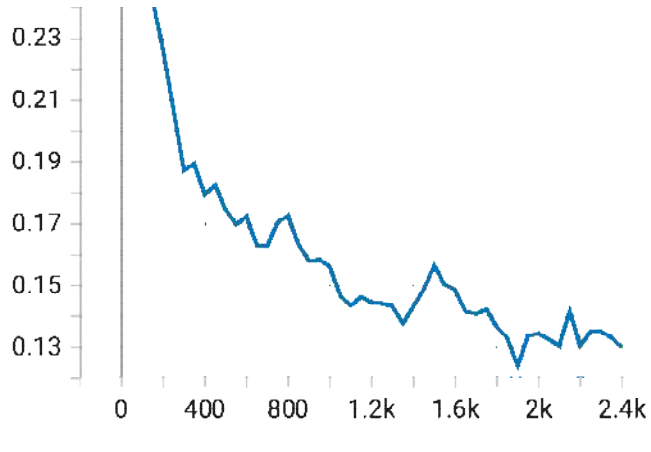


Figure 5. Training Loss

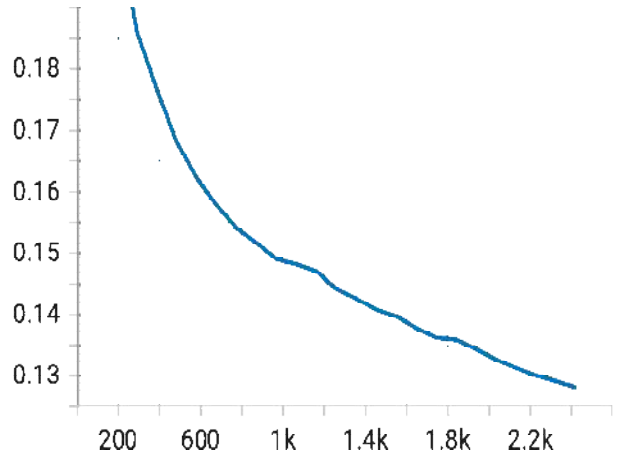


Figure 6. Validation Loss

Model Training and Validation: Proposed model implemented with pytorch tabular framework[11]. Model is binary classification model trained with DANet deep learning architecture. When designing the neural network architecture, we continuously validate the results with various combinations of hyper parameters and finally chose the appropriate number. Deep network is not recommended for tabular features[10]. I have set task parameter in model configuration as classification and learning rate as 0.001. Also set the metrics accuracy and f1 score. Training configuration hyper parameters are set as batch size = 1024, max epochs = 25, early stopping = 5, etc. Used weighted loss for managing imbalanced class as the training dataset were having 119341 records as attacks and 56000 records as normal traffic. Final trained DANets classifier model has 893K trainable parameters. Training and validation loss displayed in Figure 5 and Figure 6 respectively.

Model Testing and Deployment: The final optimized model is tested on test dataset from UNSW-NB15 dataset repository. Testing model on unseen test dataset ensures the accuracy of the model. If model was over fit or under fit on training data, it will suffer on test data. Upon verifying that the assessment criteria for the model's projections on the test dataset align with the anticipated outcomes, model can be deployed as a part of live NIDS.

V. RESULT AND DISCUSSION

The result of the model on test dataset evaluated by using the most popular matric like accuracy and F1 Score. Also evaluated by using the confusion metrics. The confusion matrix provides important data pertaining to the count of true positives, i.e., instances that are correctly classified into the appropriate category and truly belong to that class. Moreover, it takes into account true negatives, which are items that belong to another class but are accurately categorized [3]. Number of rue positives is 23924 and number of true negatives is 44383. The confusion matrix also illustrates the inaccuracy in forecasting and categorizing examples when they are mistakenly assigned to different classes rather than their actual classes. Specifically, it highlights the occurrence of false positives and negatives. The false positive is 13086 and false negative 949. Even though FPR is high, FNR rate is comparatively low. Confusion matrix in test data shown in Figure 9.

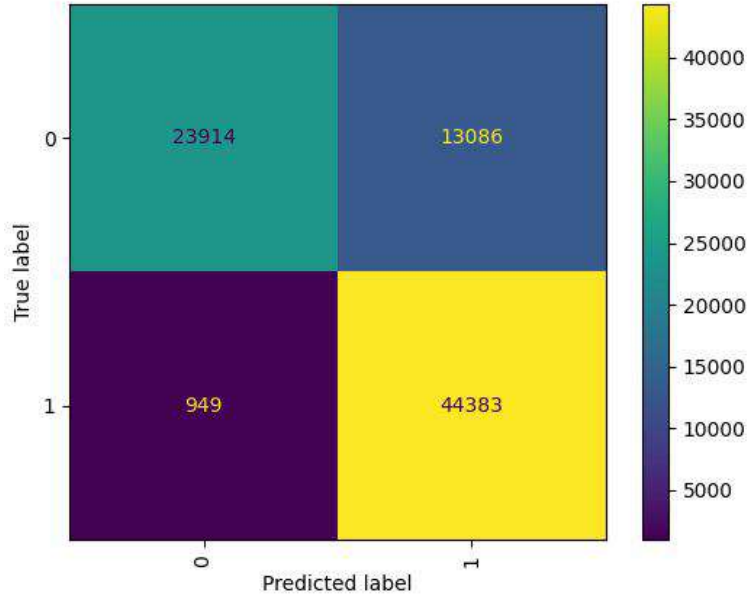


Figure 9. Confusion Metrix

Accuracy: The metric that measures the proportion of correctly classified instances to the overall number of instances is known as the classification accuracy.

$$\text{Accuracy} = TP + TN / (TP + TN + FP + FN) \quad [2]$$

F1 Score: The concept of F1 score is characterized as the harmonic mean of precision and recall, which is a statistical method for evaluating system accuracy by taking into account both precision and recall measurements.

$$\text{F1 Score} = 2(\ (\text{Precision} \times \text{Recall})/\ (\text{Precision} + \text{Recall}\ )) \quad [2]$$

The proposed model evaluated against the prediction accuracy and F1 Score of base line machine learning models like Gaussian Naïve Bayes classifier and Logistic regression to compare the efficacy of the proposed DANets classifier. The result was highly impressive as shown in Table 2. The model accuracy was highly impressive against accuracies of binary classifier models proposed previous recent research on UNSW-NB15 network traffic dataset as in Table 3.

Table 2. Comparison of Proposed Model with Base Line ML Models with the UNSW-NB15 Dataset

Model Configuration	Accuracy	F1 Score
Gaussian Naïve Bayes Classifier	73.03%	0.7884
Logistic Regression Classifier	62.27%	0.7447
Proposed DANets Classifier	82.95%	0.8295

Table 3. Comparison of Proposed Model with Benchmark Studies With the UNSW-NB15 Dataset

Model Configuration	Machine Learniung and Deep Learning algorithm	Accuracy
Alsharaiah et al. (2022)	Least Square SVM model	71.60
Alsharaiah et al. (2024)	AT-LSTM model	78.86%
Proposed Model	DANETs with Abstract Layers (ABSTLAYs)	82.95%

VI. CONCLUSION

This study investigated the NIDS with DANet model for achieving improved intrusion detection. The model's developed and was evaluated by using training and test csv dataset from the well-known UNSW-NB15 dataset repository, which is a benchmark dataset for network intrusion detection.

An extensive literature analysis was conducted to review and compare various methods and binary classification on the UNSW-NB15 dataset. The experimental results demonstrated that the proposed DANet model has superiority compared to other machine learning methods on the test data. This highlights the effectiveness of utilizing a DANet architecture in network intrusion detection system. Nevertheless, the present model's performance may yet be optimized. In future research could investigate the utility of synthetic oversampling techniques to mitigate the class imbalance challenge within the UNSW-NB15 dataset, which may lead to even better results. The research presented herein adds to the research of network intrusion detection by proposing a unique model that improves accuracy in attack detection as a binary classifier. The study highlights the promising prospect of incorporating the DANet model with other methods to develop more effective and efficient intrusion detection systems.

REFERENCE

1. Ashiku, L.; Dagli, C. Network Intrusion Detection System Using Deep Learning. Procedia Comput. Sci. 2021, 185, 239–247, https://doi:10.1016/j.procs.2021.05.025.
2. Ahmad, Z.; Shahid Khan, A.; Wai Shiang, C.; Abdullah, J.; Ahmad, F. Network Intrusion Detection System: A Systematic Study of Machine Learning and Deep Learning Approaches. Trans. Emerg. Telecommun. Technol. 2021, 32, e4150, https://doi:10.1002/ett.4150.
3. Alsharaiah, M.A.; Abualhaj, M.; Baniata, L.H.; Al-saaidah, A.; Kharma, Q.M.; Al-Zyoud, M.M. An Innovative Network Intrusion Detection System (NIDS): Hierarchical Deep Learning Model Based on Unsw-Nb15 Dataset. Int. J. Data Netw. Sci. 2024, 8, 709–722, https://doi:10.5267/j.ijdns.2024.1.007
4. Moustafa, N.; Slay, J. UNSW-NB15: A Comprehensive Data Set for Network Intrusion Detection Systems (UNSW-NB15 Network Data Set). In Proceedings of the 2015 Military Communications and Information Systems Conference (MilCIS); IEEE: Canberra, Australia, November 2015; pp. 1–6.
5. D'Hooge, L.; Verkerken, M.; Wauters, T.; Volckaert, B.; De Turck, F. Discovering Non-Metadata Contaminant Features in Intrusion Detection Datasets. In Proceedings of the 2022 19th Annual International Conference on Privacy, Security & Trust (PST); IEEE: Fredericton, NB, Canada, August 22 2022; pp. 1–11.
6. Azam, Z.; Islam, Md.M.; Huda, M.N. Comparative Analysis of Intrusion Detection Systems and Machine Learning-Based Model Analysis Through Decision Tree. IEEE Access 2023, 11, 80348–80391, https://doi:10.1109/ACCESS.2023.3296444
7. Ehigiator Towards Improved Vulnerability Management in Digital Environments: A Comprehensive Framework for Cyber Security Enhancement. Int. Res. J. Comput. Sci.
8. Bouchama, F.; Kamal, M. Enhancing Cyber Threat Detection through Machine Learning-Based Behavioral Modeling of Network Traffic Patterns. Int. J. Bus. Intell. Big Data Anal. 2021, 4, 1–9.
9. Gokul, Kavinraj,Rohith,Sangeetha(2024). Intrusion Detection using Deep Learning. IRJCS: International Research Journal of Computer Science, Volume 11, Issue 04 of 2024 pages 272-277, https://doi:10.26562/irjcs.2024.v1104.22.
10. Chen, J.; Liao, K.; Wan, Y.; Chen, D.Z.; Wu, J. DANets: Deep Abstract Networks for Tabular Data Classification and Regression. Proc. AAAI Conf. Artif. Intell. 2022, 36, 3930–3938, https://doi:10.1609/aaai.v36i4.20309.
11. Joseph, M. PyTorch Tabular: A Framework for Deep Learning with Tabular Data 2021.
12. Jayamurugan,Stanley,Kesava (2023).AI Based Spam Detection Using Logistic Regression Algorithm. IJIRAE: International Journal of Innovative Research in Advanced Engineering, Volume 10, Issue 06 of 2023 pages 307-310. https://doi.org/10.26562/irjcs.2023.v1006.06
13. Sakthivel,Srinivasan,Sathish (2023).Traffic Violation Prediction Using Deep Learning. IJIRAE:: International Journal of Innovative Research in Advanced Engineering, Volume 10, Issue 06 of 2023 pages 341-345. https://doi.org/10.26562/irjcs.2023.v1006.12
14. Hewa,Egho-Promise,Lyada,Aina,Sangodoyin,Kure (2024). The Effectiveness of a Comprehensive threat Mitigation Framework in Networking: A Multi-Layered Approach to Cyber Security. IRJCS: International Research Journal of Computer Science, Volume 11, Issue 05 of 2024 pages 529-538, https://doi.org/10.26562/irjcs.2024.v1106.03