INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN MULTIDISCIPLINARY EDUCATION

ISSN (print): 2833-4515, ISSN (online): 2833-4531

Volume 03 Issue 12 December 2024

DOI: 10.58806/ijirme.2024.v3i12n19, Impact factor- 5.138

Page No. 1981-1989

Leveraging Deep Learning Models for Intrusion Detection Systems for Secure Networks

Anuj Kumar Gupta[1], Achuthananda Reddy Polu[2], Bhumeka Narra[3], Dheeraj Varun Kumar Reddy Buddula[4], Hari Hara Sudheer Patchipulusu[5], Navya Vattikonda[6]

[1]Oracle ERP Senior Business Analyst ,Genesis Alkali
[2]Senior SDE, Cloudhub IT Solutions
[3]Sr Software Developer, Statefarm
[4]Software Engineer, Elevance Health Inc
[5]Senior Software Engineer, Walmart
[6]Business Intelligence Engineer, International Medical Group Inc

ABSTRACT: Real-time cyber threat identification and mitigation depend heavily on intrusion detection systems (IDS) for secure networks. Network systems utilize analysis to spot malicious events unauthorized access and system vulnerabilities in network traffic. Machine learning forms the foundation of these security implementations. Through improved network security measures, IDS protect business assets by ensuring both data availability and uncompromised security. After enhancing the effectiveness of intrusion detection, this study suggests a strong method based on DL. A CNN model is developed and evaluated against traditional models, including KNN, Autoencoders (AE), and DNN that are trained on the NSL-KDD dataset. CNN delivers remarkable performance when used in network threat detection with success metrics of 98.63% accuracy alongside 98.45% precision, 98.98% recall, and an F1-score of 98.72%, proving its efficiency for threat recognition. Visualization and comparative performance analysis further prove the model's effectiveness, paving the way for its possible use in safe network settings. The benefits of using DL frameworks to improve IDS systems are highlighted in this paper.

KEYWORDS: Cybersecurity, Intrusion Detection System, Secure Networks, Network Security, deep learning, NSL-KDD.

I. INTRODUCTION

Modern global communication services and essential infrastructure depend on networks as their central structural element during the current digital transformation period. Society's increasing dependence on connected systems has increased the critical need for information protection while ensuring digital security has escalated dramatically. Every person, business entity and nation experiences substantial danger from cyber threats that cover both basic data breaches alongside complex malware and advanced ransomware attacks [1]. Digital asset security now stands as a top organizational priority which propels advancements in cybersecurity techniques and technology development.

A broad variety of procedures, tools, and techniques are collectively known as cybersecurity measures, and their purpose is to protect computer systems, networks, and data from intrusion, abuse, and destruction [2]. IDS stand as crucial monitoring tools which inspect network traffic to discover questionable behaviors. IDSs operate as specialized tools which identify unexpected behavior while reporting unauthorized attempts to access systems thereby providing crucial security protection for current defensive systems [3]. The combination of system activity monitoring and real-time alert generation from IDS systems effectively prevents cyberattacks and facilitates vulnerability detection while repelling cybercriminals and creating records for future threat mitigation efforts.

Traditional IDS methodologies encounter clear disadvantages due to the expansion of sophisticated and large-scale cyberattacks. Traditional static rule-based IDS approaches frequently face difficulties with new security threats so AI and ML integration becomes vital for IDS performance enhancement [4]. Machine learning has revolutionized IDS capabilities through its automation of known and unknown threat detection. The emerging technology DL presents machine learning with exceptional capabilities of handling extensive high-dimensional information[5]. Automatic feature extraction together with threat pattern discovery and real-time threat adaptation serve DL models as essential components for developing resilient IDS systems. IDSs powered by DL techniques will detect threats with greater accuracy while minimizing mistakes and delivering instant reaction capabilities against suspicious events [6].

A. *Aim and contribution*

This paper's objective is to design a secure network IDS that makes good use of ML methods for intrusion detection and classification. The detection performance optimization research employs the NSL-KDD dataset to implement robust machine learning models alongside feature selection strategies and progressive preprocessing techniques to provide better cyber threat protection. The main contributions are:

- Develop effective DL Models for IDS using NSL-KDD dataset.
- Identified significant features from the NSL-KDD dataset through a systematic feature selection process.
- Introduced advanced preprocessing steps, including Z-score normalization and encoding, to handle outliers, normalize data, and reduce dimensionality, ensuring data consistency.
- Implemented and compared multiple machine learning models, including CNN, KNN, AE, and DNN, providing insights into their effectiveness in intrusion detection.
- The proposed IDS framework demonstrated effective network traffic classification and anomaly detection through evaluation using F1-score, recall, accuracy and precision metrics..

B. *Structure of the paper*

The study is structured as follows: Section II presents relevant work for IDS. Section III details the procedures, methodology and materials used. Section IV presents the experimental findings and results in analysis and a discussion of the proposed system. Section V presents the conclusion and future work.

II. LITERATURE REVIEW

This section outlines the previous research on the IDS for Secure Networks employing ML and DL methods and techniques.

In this study, Aboueata et.al (2019) developed ML models and evaluated their efficacy using ANN and SVM methods. An UNSW-NB-15 dataset was utilized for both training and testing the models. They have also reduced the training duration and complexity of the ML models while determining the ideal collection of features through feature engineering and parameter tuning. While using the right features, SVM and ANN methods achieve 91% and 92% accuracy in anomaly identification, respectively [7].

In This study, aims, Atefi, Hashim and Kassim et.al (2019) using anomaly analysis for IDS classification with the most recent dataset, CICIDS-2017, which may be utilized for IDS evaluation. An MCC for MLandDL-based classification performance is presented in one of the outcomes. With a score of 0.9293%, DNN is clearly a superior classifier, while KNN only manages a 0.8824% score[8].

In this study, Pattawaro and Polprasert et. al's (2018) model achieves an accuracy of 84.41%, a detection rate of 86.36%, and a FAR of 18.20% for the KDD Test dataset. In addition, feature selection allows us to achieve the same level of performance as models trained with the full set of features, even though our proposed model only makes use of 75 out of 122 characteristics (61.47%)[9].

In this study, Sezari, Moller and Deutschmann et al. (2018) a network IDS that relies on anomaly detection and uses DL to identify instances of aberrant behavior based on a model of typical system behavior. After a train and evaluating our model, they utilized the DARPA dataset that was utilized in the 1999 KDD Cup. Compared to other studies that used this dataset, our model outperformed the competition with a low FAR and a very accurate detection rate [10].

In this study, Zhang et.al (2017) suggest an architecture for intrusion detection that uses semi-supervised ML effectively. In particular, the framework enhances performance by using an information gain-based feature selection strategy and a LapSVM as its training model. A 97.8 percent accuracy rate with a 2% false-positive rate was achieved by our framework in the NSL-KDD experiments[11].

In this study, Sen, Sen and Chattopadhyay et.al (2014) the present situation call for cutting-edge research to develop more advanced IDS. In this research, they provide a BPNN architecture that may be used to build an anomaly-based IDS that is both accurate and fast. The architecture is developed in this context using the KDD'99 data set[12].

Below Table I shows the literature review summary of intrusion detection in secure networks with different papers, methods, datasets used, their key findings and their limitations and future work.

TABLE I. SUMMARY OF LITERATURE REVIEW FOR INTRUSION DETECTION IN SECURE NETWORKS USING ML AND DL METHODS

Authors	Methodology	Dataset	Key Findings	Limitations & Future Work
Aboueata et.al.	ANN and SVM	UNSW-NB-15	With optimal feature sets, achieved anomaly detection accuracy91% (SVM) and 92% (ANN).	Requires optimization to reduce training time and model complexity.

Atefi, Hashim and Kassim et.al.	KNN and DNN	CICIDS-2017	DNN achieved superior performance in anomaly classification with an MCC score0.9293% compared to KNN's 0.8824%.	No specific limitations or future work were mentioned.
Pattawaro and Polprasert et al.	Recurrent Neural Network (RNN)-based DNN	KDDTest	Achieved 84.41%accuracy, 86.36%detection rate, and 18.20% FAR, using 75 out of 122 features.	Further improvement is needed in accuracy, detection rate, and false alarm reduction.
Sezari, Moller and Deutschmann et.al.	Deep Learning	DARPA dataset (KDD 1999 Cup)	Achieved high accuracy and low FAR in detecting known and unknown network intrusion attacks.	No specific limitations or future work mentioned.
Zhang et.al.	Laplacian Support Vector Machine (LapSVM)	NSL-KDD	Achieved 97.8% accuracy with a 2% FP rate using a semi-supervised ML framework.	Future work should focus on refining the model for further improved performance.
Sen, Sen and Chattopadhyay et al.	Back Propagation Neural Network (BPNN)	KDD'99	An effective BPNN design was suggested for anomaly-based IDS, which achieved a high detection rate and accuracy.	Advanced research is required to develop more sophisticated IDS solutions.

III. METHODOLOGY

To develop Intrusion Detection Systems (IDS) for secure networks, a systematic methodology is employed, beginning with the selection of the NSL-KDDdataset as the primary data source. The following figure 1 shows all the steps of implementation. After selecting the dataset, preprocessing steps are applied, including handling null values through imputation or removal, normalizing features using Z-score normalization for uniformity, and encoding categorical attributes with LabelCount encoders to manage outliers and reduce dimensionality. Next, the preprocessed data is divided into a training set (75%) and a testing set (25%) to facilitate effective model training and evaluation. Various ML models, including CNN, KNN, Autoencoders (AE), and Deep Neural Networks (DNN), are implemented to identify patterns and anomalies. Subsequently, performance is measured using evaluation metrics like F1-score, recall, accuracy, and precision, derived from a confusion matrix, ensuring the IDS effectively distinguishes between normal and malicious activities, enhancing network security.

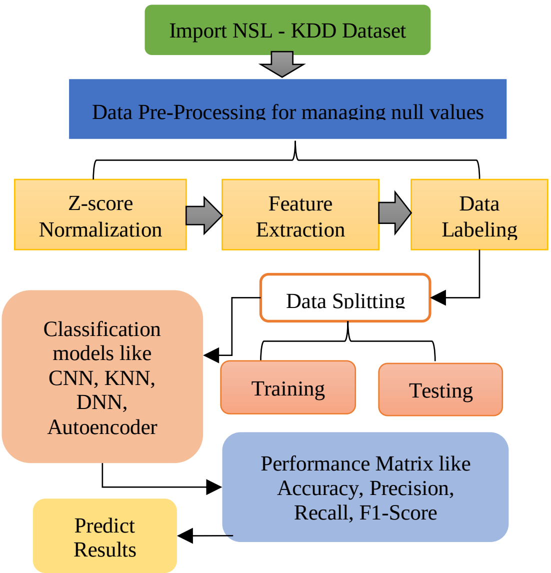
Import NSL - KDD Dataset
Data Pre-Processing for managing null values
Z-score Normalization
Feature Extraction
Data Labeling
Data Splitting
Classification models like CNN, KNN, DNN, Autoencoder
Training
Testing
Performance Matrix like Accuracy, Precision, Recall, F1-Score
Predict Results

Fig. 1. Flowchart for IDS secure networks

The whole steps of development that are shown in Figure 1 are explained below:

A. Data collection

T The NSL-KDDdataset has been developed and upgraded from the KDDCup99dataset. Using the feature selection strategy, we were able to extract 12 features from NSL-KDD, out of a total of 41. For network data, NSL-KDD is a useful simulator. The NSL-KDD data visualization graphics are provided below:

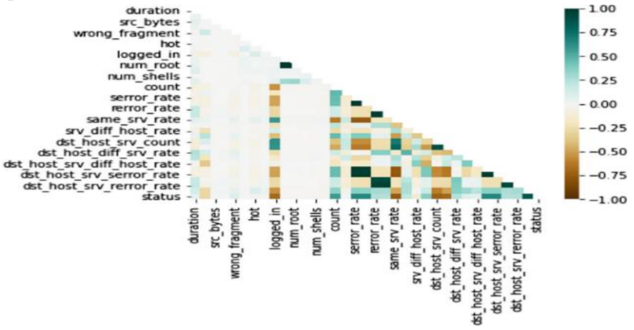

Fig. 2. Heatmap for NSL-KDD Dataset

Figure 2 visualizes the NSL-KDD dataset, highlighting features like duration, protocol type, logged_in, and connection rate metrics. Darker colors indicate higher values, offering insights into feature relationships and patterns. This compact representation aids in understanding dataset characteristics and selecting relevant features for network intrusion detection model development.

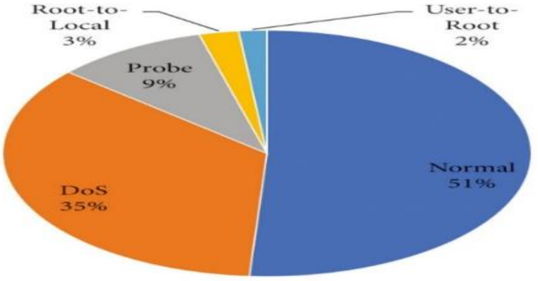

Fig. 3. Pie chart of NSL-KDD dataset distribution

In Figure 3, the pie chart illustrates the distribution of attack and normal classes in the NSL-KDD dataset. The majority of the data represents normal traffic (51%), followed by Denial of Service (DoS) attacks at 35%. Other categories include Probe (9%), Root-to-Local (3%), and User-to-Root (2%), reflecting the dataset's focus on diverse intrusion types.

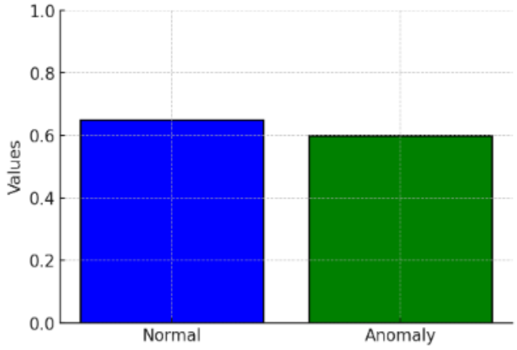

Fig. 4. Distribution of NSL-KDD Dataset

Figure 4 illustrates the comparison of values for normal and anomaly categories in the context of IDS analysis. The blue and green bars represent the respective data for the categories, highlighting a slightly higher value for the normal category (0.65) compared to the anomaly category (0.60). The y-axis corresponds to the normalized values, ranging from 0 to 1, while the x-axis denotes the respective categories like normal and anomaly.

B. Data Preprocessing

The total system's efficiency is directly impacted by the pre-processing procedures used [13]. In most cases, a mix of normalization, feature extraction, and numericalization will be used during pre-processing. A goal of this study is to improve the model's core subsystem's detection rates. Hence, this section of the research is crucial. The following pre-processing steps are given below:

- **Managing null values:** Imputing the mean, median, or mode for missing values or eliminating rows with null values are strategies that assist in keeping the dataset consistent.

C. *Z-score Normalization*

The values of the attributes are normalized using Z-score normalization. By applying this normalization to the attribute value, we can see that the standard deviation is now one, and the mean is zero. The z-score is another name for zero mean normalization, which describes this normalization quality. The equation for it is provided (1) [14].

$$a(i) = \frac{a(i) - mean(A)}{std(A)} \qquad (1)$$

The i^{th} value of A, denoted as a(i), is going to be updated using the preceding equation, and A is an attribute.

D. *Feature Extraction*

Feature extraction is commonly used to decrease computation. Common methods for feature extraction include data packet encoding and correlation-based feature selection. One approach employs a correlation function to reduce data size by selecting subsets of the data.

E. *Data labeling*

The numerical representation of the categorical features should be transformed into a vector in Euclidean space: When encoding the three categorical attributes "protocol type," "service," and "flag," we use a LabelCount encoder that arranges the categories according to the feature's frequency of each category. LabelCount excels in reducing dimensionality and being insensitive to outliers, which are particularly useful when dealing with features that have a high number of categories.

F. *Data splitting*

The preprocessed data is used to build two sets: one for training and one for testing. In order to train the model, 75% of the data is taken up by the training set, and to assess its performance, 25% of the data is taken up by the testing set.

G. *Classification of CNN model*

Neurons in CNNs learn to optimize themselves, much like in classic ANNs. The foundation of innumerable ANNs remains the same: each neuron will take an input and execute an operation (like a scalar product followed by a non-linear function, for example). All the way from the first raw image vector input to the final class score, the network will only provide one perceptual scoring function, the weight. Loss functions linked to classes will be located in the last layer, and the usual strategies employed with conventional ANNs are still relevant.

Because there are fewer connections between convolutional layers, the pooling layers lessen the computational load. Additionally, pooling layers enhance the receptive field of succeeding convolutional layers and improve translation invariance features[15]. For training purposes, a loss function is utilized to measure the mistakes, and one or more fully connected layers are typically added to the network's convolutional stream.

A collection of n kernelsW={w1,w2,…,wn} and their biasesB={b1,b2,…, bn}, are convolved with inputdata at each CNN layer. A resultant feature map xk is generated by convolutioning the data with each kernel. The definition of transformation for each convolutional layer l is given by (2):

$$x_k^1 = \sigma(w_k^{l-1} * x^{l-1} + b_k^{l-1} \qquad (2)$$

CNN learns by sliding a small window across the inputs; variables such as bias and weights inside this window can be optimized based on properties of the input data regardless of their location in the data.

H. *Performance metrics*

A measure of how well the CNN-based IDS model performs according to intrusion detection performance [16]. The numerical expression of classification accuracy was achieved using a confusion matrix. Among the many popular machine learning methods, the confusion matrix compiles data on the real and anticipated classes produced

by a classification algorithm. The actual and expected classes make up the two dimensions of the confusion matrix. In contrast to the projected class states shown in the columns, the actual class examples in the rows are represented by the data. The confusion matrix has four columns: TP, TN, FP, and FN. What follows is an explanation of the study's evaluation metrics:

Accuracy: The fraction of records properly classified out of all records in Equation (3):

$$Accuracy = \frac{TP+TN}{TP+Fp+TN+FN} \square \qquad \square\square\square$$

Precision: A precision of a model is a measure of how well it makes good predictions. The accuracy rate is defined as the percentage of positive predictions (TP and FP combined) that were actually realized. The equation is shown in (4):

$$Precision = \frac{TP}{TP+FP} \square \qquad \square\square\square$$

Recall: The TPR, or recall, is a measure of how well a model can detect all positive events in a dataset. The equation is shown in (5):

$$Recall = \frac{TP}{TP+FN} \qquad \square\square\square$$

F1-score: A well-rounded assessment of a model's efficacy is provided by the F1 score, which is the harmonic mean of recall and precision. An equation shown in (6):

$$F1 - Score = 2\,\frac{(P*R\,)}{P + R} \qquad \square\square\square$$

The DL models are determined by these matrices.

IV. RESULT ANALYSIS AND DISCUSSION

The following experiments are conducted on a desktop with 8 GB of RAM, Intel Core i3, Python 3.7 is used for programming. The experiment outcomes of a CNN model trained on a NSL-KDD dataset are provided in Table 2. The following CNN model is compared with existing models like KNN[17], DNN[18] and Autoencoder (AE)[19] based on performance matrices like precision, recall, accuracy, and f1-score are given in Table II.

TABLE II. OUTCOME OF CNN MODEL FOR INTRUSION DETECTION SYSTEM ON NSL-KDD DATASET

Performance Measures	Convolutional Neural Network (CNN)
Accuracy	98.63
Precision	98.45
Recall	98.98
F1-score	98.72

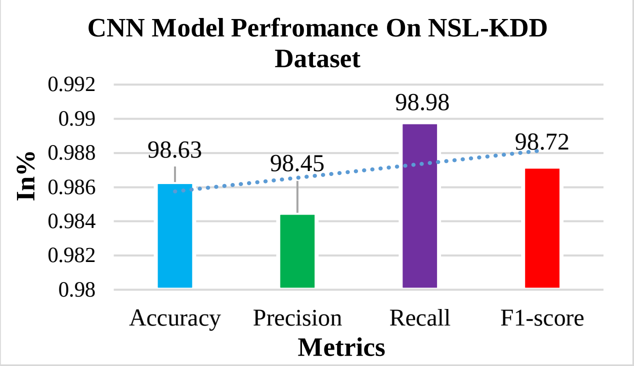

Fig. 5. Bar Graph for CNN Model Performance

Table II and Figure 5 shows the results of the CNN's intrusion detection performance. An impressive 98.63% accuracy rate shows that the algorithm is quite good at correctly identifying network intrusions, demonstrating its remarkable potency. The model's recall of 98.98% and precision of 98.45% demonstrate its capacity to detect almost all intrusion cases accurately, while the latter makes clear how well it minimizes false positives. Furthermore, the CNN model has strong performance in detecting and reducing network threats in this benchmark dataset, as evidenced by the F1-score of 98.72%, which shows a balanced trade-off among recall and precision.

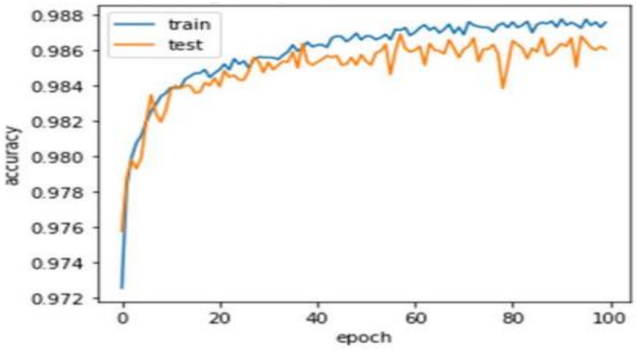

Fig. 6. Plot of accuracy curve for CNN model

The Figure 6 displays a plot of an accuracy curve for a CNN model during training (blue line) and testing (orange line) over 100 epochs. The training accuracy starts around 0.9729 and gradually increases to around 0.9868 by the end of the 100 epochs. The test accuracy starts around 0.9767 and fluctuates between 0.9767 and 0.9843 throughout the training process.

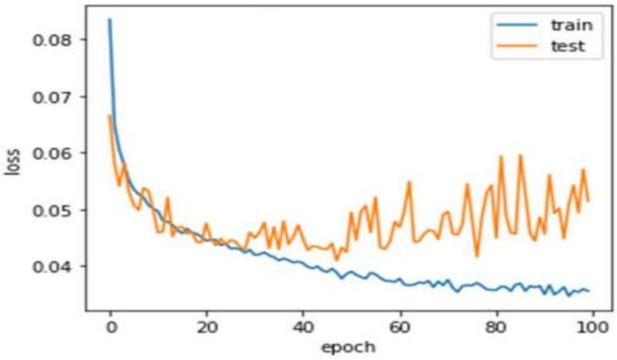

Fig. 7. Plot of loss curve for CNN model

The Figure 7 shows the loss curve for a CNN model during training (blue line) and testing (orange line) over 100 epochs. The training loss starts around 0.077 and gradually decreases to around 0.028 by the end of the 100 epochs. The test loss starts around 0.062 and fluctuates between 0.05 and 0.07 throughout the training process.

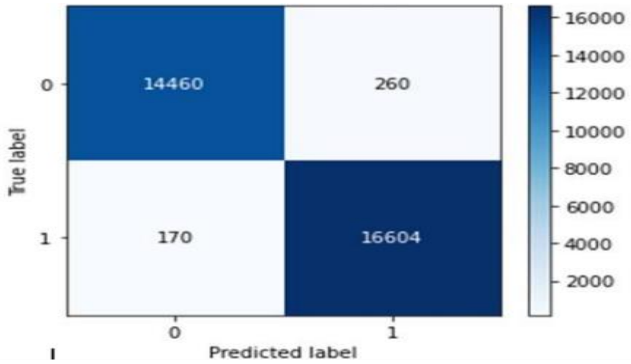

Fig. 8. Confusion Matrix of CNN model

Figure 8 illustrates a confusion matrix presented that evaluates performance of a binary classification model. The model got 16,604 positive cases and 14,460 negative examples (both with true labels 0 and 1). But it got 170 positives and 260 negatives mixed up, and it thought the negative class was positive. These results indicate high overall accuracy across both classes.

TABLE III. COMPARISON BETWEEN ML AND DL MODELS PERFORMANCE ON NSL-KDD DATASET

Models	Accuracy	Precision	Recall	F1-score
CNN	98.63	98.45	98.98	98.72
KNN	91	92.6	90.5	91.5
AE	87.2	84.6	92.8	80.7
DNN	79.74	82.22	79.74	76.47

The above Table III presents the performance of DL models across performance parameters. Among these models, the CNN demonstrates superior performance, achieving an accuracy of 98.63%, a precision of 98.45%, a recall of 98.98%, and a F1-score14 of 98.72%. The KNN model follows, with an accuracy91%, a precision of 92.6%, a recall of 90.5%, and an F1-score of 91.5%. The Autoencoder model attains an accuracy87.2%, with a precision84.6%, a recall92.8%, and an F1-score80.7%. Lastly, the DNN model exhibits the lowest performance, with an accuracy of 79.74%, a precision of 82.22%, a recall of 79.74%, and an F1-score of 76.47%. Overall, the CNN achieves the best accuracy in comparison to other DL models.

V. CONCLUSION

Building an IDS for secure networks, using a behavioral approach, is a dynamic field where researchers continue to encounter challenges and propose innovative solutions. In this study, a DL-based approach for IDS in secure networks was developed and evaluated using the NSL-KDD dataset. This study presents a deep learning-based IDS using a CNN, which outperforms traditional ML models like KNN, Autoencoders (AE), and DNN in terms of F1-score, precision,recall, and accuracy. The CNN model achieved an impressive accuracy of 98.63%, precision 98.45%, recall 98.98%, and an F1-score 98.72%, highlighting its strong ability to correctly classify network intrusions and minimize false positives. The results confirm the model's robustness in identifying and mitigating various network threats. However, the study is limited by the use of the NSL-KDD dataset, which may not reflect real-time or evolving network traffic patterns. Future research should explore the integration of more diverse datasets, real-world attack scenarios, and advanced techniques like transfer learning and ensemble models to further enhance IDS performance and adapt to the continuously changing landscape of network security.

REFERENCES

1. H. Liu and B. Lang, "Machine learning and deep learning methods for intrusion detection systems: A survey," 2019. doi: 10.3390/app9204396.
2. K. Kim and M. E. Aminanto, "Deep learning in intrusion detection perspective: Overview and further challenges," in *Proceedings - WBIS 2017: 2017 International Workshop on Big Data and Information Security*, 2017. doi: 10.1109/IWBIS.2017.8275095.
3. R. Rama Devi and M. Abualkibash, "Intrusion Detection System Classification Using Different Machine Learning Algorithms on KDD-99 and NSL-KDD Datasets - A Review Paper," *Int. J. Comput. Sci. Inf. Technol.*, 2019, doi: 10.5121/ijcsit.2019.11306.
4. J. Jabez and B. Muthukumar, "Intrusion detection system (ids): Anomaly detection using outlier detection approach," in *Procedia Computer Science*, 2015. doi: 10.1016/j.procs.2015.04.191.
5. V. K. Rahul, R. Vinayakumar, K. Soman, and P. Poornachandran, "Evaluating Shallow and Deep Neural Networks for Network Intrusion Detection Systems in Cyber Security," in *2018 9th International Conference on Computing, Communication and Networking Technologies, ICCCNT 2018*, 2018. doi: 10.1109/ICCCNT.2018.8494096.
6. S. Pandey, "MODERN NETWORK SECURITY: ISSUES AND CHALLENGES," *Int. J. Eng. Sci. Technol.*, vol. 3, 2011.
7. N. Aboueata, S. Alrasbi, A. Erbad, A. Kassler, and D. Bhamare, "Supervised machine learning techniques for efficient network intrusion detection," in *Proceedings - International Conference on Computer Communications and Networks, ICCCN*, 2019. doi: 10.1109/ICCCN.2019.8847179.
8. K. Atefi, H. Hashim, and M. Kassim, "Anomaly analysis for the classification purpose of intrusion detection system with K-nearest neighbors and deep neural network," in *Proceeding - 2019 IEEE 7th Conference on Systems, Process and Control, ICSPC 2019*, 2019. doi: 10.1109/ICSPC47137.2019.9068081.
9. A. Pattawaro and C. Polprasert, "Anomaly-Based Network Intrusion Detection System through Feature Selection and Hybrid Machine Learning Technique," in *International Conference on ICT and Knowledge Engineering*, 2018. doi: 10.1109/ICTKE.2018.8612331.
10. B. Sezari, D. P. F. Moller, and A. Deutschmann, "Anomaly-Based Network Intrusion Detection Model Using Deep Learning in Airports," in *Proceedings - 17th IEEE International Conference on Trust, Security and Privacy in Computing and Communications and 12th IEEE International Conference on Big Data Science and Engineering, Trustcom/BigDataSE 2018*, 2018. doi: 10.1109/TrustCom/BigDataSE.2018.00261.
11. X. Zhang, P. Zhu, J. Tian, and J. Zhang, "An effective semi-supervised model for intrusion detection using feature selection based LapSVM," in *IEEE CITS 2017 - 2017 International Conference on Computer, Information and Telecommunication Systems*, 2017. doi: 10.1109/CITS.2017.8035323.
12. N. Sen, R. Sen, and M. Chattopadhyay, "An effective back propagation neural network architecture for the development of an efficient anomaly-based intrusion detection system," in *Proceedings - 2014 6th International Conference on Computational Intelligence and Communication Networks, CICN 2014*, 2014. doi: 10.1109/CICN.2014.221.
13. R. Thomas and D. Pavithran, "A Survey of Intrusion Detection Models based on NSL-KDD Data Set," in *ITT 2018 - Information Technology Trends: Emerging Technologies for Artificial Intelligence*, 2018. doi: 10.1109/CTIT.2018.8649498.
14. B. Ingre and A. Yadav, "Performance analysis of NSL-KDD dataset using ANN," in *International Conference on Signal Processing and Communication Engineering Systems - Proceedings of SPACES 2015, in Association with IEEE*, 2015. doi: 10.1109/SPACES.2015.7058223.
15. C. L. T. S. L. C. Y. C. C. Mohammadpour Leila, "A Convolutional Neural Network for Network," *A Convolutional Neural Netw. Netw. Intrusion Detect. Syst.*, p. 16, 2018.
16. Y. Ding and Y. Zhai, "Intrusion detection system for NSL-KDD dataset using convolutional neural networks," in *ACM International Conference Proceeding Series*, 2018. doi: 10.1145/3297156.3297230.
17. R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep Learning Approach for Intelligent Intrusion Detection System," *IEEE Access*, 2019, doi: 10.1109/ACCESS.2019.2895334.
18. X. Chen Chongzhen Zhang, Fangming Ruan, Lan Yin, "A deep learning approach for network intrusion detection based on NSL-KDD dataset," *IEEE*, 2019.
19. S. Gurung, M. K. Ghose, and A. Subedi, "Deep Learning Approach on Network Intrusion Detection System using NSL-KDD Dataset," *Int. J. Comput. Netw. Inf. Secur.*, 2019, doi: 10.5815/ijcnis.2019.03.02..
20. Routhu, K., Bodepudi, V., Jha, K. M., & Chinta, P. C. R. (2020). A Deep Learning Architectures for Enhancing Cyber

Security Protocols in Big Data Integrated ERP Systems. *Available at SSRN 5102662*.

21. Chinta, P. C. R., & Katnapally, N. (2021). Neural Network-Based Risk Assessment for Cybersecurity in Big Data-Oriented ERP Infrastructures. *Neural Network-Based Risk Assessment for Cybersecurity in Big Data-Oriented ERP Infrastructures*.
22. Katnapally, N., Chinta, P. C. R., Routhu, K. K., Velaga, V., Bodepudi, V., & Karaka, L. M. (2021). Leveraging Big Data Analytics and Machine Learning Techniques for Sentiment Analysis of Amazon Product Reviews in Business Insights. *American Journal of Computing and Engineering*, *4*(2), 35-51.
23. Karaka, L. M. (2021). Optimising Product Enhancements Strategic Approaches to Managing Complexity. *Available at SSRN 5147875*.
24. Chinta, P. C. R., & Karaka, L. M. AGENTIC AI AND REINFORCEMENT LEARNING: TOWARDS MORE AUTONOMOUS AND ADAPTIVE AI SYSTEMS.
25. Boppana, S. B., Moore, C. S., Bodepudi, V., Jha, K. M., Maka, S. R., & Sadaram, G. AI And ML Applications In Big Data Analytics: Transforming ERP Security Models For Modern Enterprises.
26. Chinta, P. C. R., Katnapally, N., Ja, K., Bodepudi, V., Babu, S., & Boppana, M. S. (2022). Exploring the role of neural networks in big data-driven ERP systems for proactive cybersecurity management. *Kurdish Studies*.
27. Chinta, P. C. R. (2022). Enhancing Supply Chain Efficiency and Performance Through ERP Optimisation Strategies. *Journal of Artificial Intelligence & Cloud Computing*, *1*(4), 10-47363.
28. Sadaram, G., Sakuru, M., Karaka, L. M., Reddy, M. S., Bodepudi, V., Boppana, S. B., & Maka, S. R. (2022). Internet of Things (IoT) Cybersecurity Enhancement through Artificial Intelligence: A Study on Intrusion Detection Systems. *Universal Library of Engineering Technology*, (2022).
29. Moore, C. (2023). AI-powered big data and ERP systems for autonomous detection of cybersecurity vulnerabilities. *Nanotechnology Perceptions*, *19*, 46-64.
30. Chinta, P. C. R. (2023). The Art of Business Analysis in Information Management Projects: Best Practices and Insights. *DOI*, *10*.
31. Chinta, P. C. R. (2023). Leveraging Machine Learning Techniques for Predictive Analysis in Merger and Acquisition (M&A). *Journal of Artificial Intelligence and Big Data*, *3*(1), 10-31586.
32. Krishna Madhav, J., Varun, B., Niharika, K., Srinivasa Rao, M., & Laxmana Murthy, K. (2023). Optimising Sales Forecasts in ERP Systems Using Machine Learning and Predictive Analytics. *J Contemp Edu Theo Artific Intel: JCETAI-104*.
33. Maka, S. R. (2023). Understanding the Fundamentals of Digital Transformation in Financial Services: Drivers and Strategic Insights. *Available at SSRN 5116707*.
34. Routhu, KishanKumar & Katnapally, Niharika & Sakuru, Manikanth. (2023). Machine Learning for Cyber Defense: A Comparative Analysis of Supervised and Unsupervised Learning Approaches. Journal for ReAttach Therapy and Developmental Diversities. 6. 10.53555/jrtdd.v6i10s(2).3481.
35. Chinta, Purna Chandra Rao & Moore, Chethan Sriharsha. (2023). Cloud-Based AI and Big Data Analytics for Real-Time Business Decision-Making. 36. 96-123. 10.47363/JAICC/2023.
36. Krishna Madhav, J., Varun, B., Niharika, K., Srinivasa Rao, M., & Laxmana Murthy, K. (2023). Optimising Sales Forecasts in ERP Systems Using Machine Learning and Predictive Analytics. *J Contemp Edu Theo Artific Intel: JCETAI-104*.
37. Bodepudi, V. (2023). Understanding the Fundamentals of Digital Transformation in Financial Services: Drivers and Strategic Insights. *Journal of Artificial Intelligence and Big Data*, *3*(1), 10-31586.
38. Jha, K. M., Bodepudi, V., Boppana, S. B., Katnapally, N., Maka, S. R., & Sakuru, M. Deep Learning-Enabled Big Data Analytics for Cybersecurity Threat Detection in ERP Ecosystems.