International Journal of AI, BigData, Computational and Management Studies
Noble Scholar Research Group | Volume 4, Issue 3, 50-60, 2023
ISSN: 3050-9416 | https://doi.org/10.63282/3050-9416.IJAIBDCMS-V4I3P106

Detecting Network Intrusions Using Big Data-Driven Artificial Intelligence Techniques in Cybersecurity

Rajiv Chalasani[1], Srikanth Reddy Vangala[2], Ram Mohan Polam[3], Bhavana Kamarthapu[4], Mitra Penmetsa[5], Jayakeshav Reddy Bhumireddy[6]
[1]Sacred Heart University.
[2]University of Bridgeport.
[3]University of Illinois at Springfield.
[4]Fairleigh Dickinson University.
[5]University of Illinois at Springfield.
[6]University of Houston.

Abstract: Cyberattacks are becoming more sophisticated, so protecting contemporary networks requires intrusion detection systems (IDS) that are both effective and intelligent. This study proposes a Convolutional Neural Network (CNN)-based model for detecting intrusions using the NSL-KDD dataset, leveraging deep learning's ability to automatically extract hierarchical features from complex network traffic patterns. The model underwent rigorous evaluation through performance metrics including precision, accuracy, recall, and F1-score. According to the results, the suggested CNN has an astonishingly high rate of accuracy of 99.9%, the model far surpasses conventional machine learning methods like Naïve Bayes and Artificial Neural Networks (ANN), and Multi-Layer Perceptron (MLP). These findings validate the strength of CNN in capturing intricate behaviors in network data, making it an attractive option for immediate and large-scale cybersecurity applications. Furthermore, the model demonstrates strong generalization, low error rates, and minimal overfitting, proving its robustness in handling diverse intrusion types. For use in future research that aims to increase detection accuracy and flexibility, they will be using hybrid models and updating Their datasets.

Keywords: Cybersecurity, NSL-KDD dataset, Network intrusion, machine learning, CNN Model.

1. Introduction

Modern life is characterized by ubiquitous global communication and networking. Thermostats and mobile phones are both linked to worldwide web These networks already pose a significant threat to security is amplified by the sheer volume of internet-connected gadgets and people [1]. To keep our information and communication systems secure, private, and available at all times, the capacity to identify and stop network threats is critical. Integral to any network or system design are intrusion detection and prevention systems (IDS/IPS), which monitor and record connection activity to spot potential assaults, either notify an administrator or thwart the attack completely [2].

There are two main types of IDS: those that operate on hosts and those that operate across networks. Specifically targeting host-based activities, host-based intrusion detection keeps tabs on things like program use and system file access. Keeping tabs on information transfer between devices and "sniffing" for suspicious network activity between various machines in the network is the main emphasis of a network intrusion detection system [3][4]. Anomaly and normal behaviour are the two overarching concepts in network theory. Depending on the amount of traffic, the applications running on the system, as well as the data types sent , a network will often act in a certain way. Two main types of network anomalies may be identified: network failures (e.g., file server outages or congestion) and network security assaults (e.g., distributed denial of service or other attacks carried out by an evil actor) [5].

Cybersecurity experts have begun to see ML as a potential replacement for conventional IDS due to its many advantages [6]. ML-enabled IDS use behavioral analysis to spot suspicious activity, which might lead to far quicker detection times and much greater accuracy [7]. Different sorts of attacks may be classified using several ML algorithms that aim to find abnormalities. ML is a kind of AI that can learn features and adjust to dynamically evolving environments. ML and statistical algorithms are very efficient in intrusion detection. Some of the pros of DL are that it has automatic feature learning, where features are extracted automatically, and the model is trained even in cases where the data is large. DL shows a better performance level compared to ML algorithms.

1.1. Motivation and Contribution

The increasing complexity and commonness of cyberattacks have enabled IDS to become an essential element in ensuring the integrity of computer networks. Complex patterns in network traffic data are not always well captured by the traditional machine

learning models, which have sub-optimally resulted in low detection accuracy. This drives the use of more elaborate deep learning methods, i.e. CNNs, capable of extracting hierarchical features automatically and providing better accuracy in classifying malicious activities. To find the best and most accurate model for intrusion detection, this project will use the NSL-KDD dataset to analyze and evaluate several prediction models, which can lead to the future creation of stronger and smarter cybersecurity systems.

The following are some of the major contributions that this research has made in the area of network security:

- Proposed a CNN-based model that is capable of learning complicated intrusion patterns in network traffic without requiring a large number of manual feature engineering.
- Data Preprocessing Pipeline: Developed efficient end-to-end preprocessing pipeline that covers null value treatment, one-hot encoding, feature selection, and normalization to have a high-quality input data to the model.
- Generalization and Stability Analysis: Analyzed the trained model performance on training/validation accuracy/loss curves and revealed good generalization and a low level of overfitting over scome epochs.
- Comprehensive Performance Analysis: Tested the model based on important classification assessment criteria that include F1-score, recall, accuracy, and precision and showed that it is suitable in real-time intrusion detection in dynamic network environments.

1.2. Justification and Novelty

The new aspect of the given methodology is the use of a CNN model to perform intrusion detection on a dataset known as NSL-KDD, which is usually built on older machines learning techniques. The CNN structure, contrary to the shallow models, can capture those complex interactions in network traffic data of both spatial and temporal domains, thus being capable of distinguishing the minute patterns that may represent different types of intrusions. Such a deep learning method avoids the laborious manual feature engineering and provides better performance, which is demonstrated by the fact that the model has reached an impressive accuracy, as well as precision, recall, and F1-scores. The loss and accuracy curves further prove that the model is stable in that the generalization is high with negligible overfitting throughout the training epochs. All these findings prove that CNN-based intrusion detection, on top of improving detection accuracy, is stable and consistent and hence a highly efficient and scalable approach to build in contemporary cybersecurity systems.

2. Literature Review

Several other major researches work on predictive modelling concerning intrusion detection have been reviewed and analyzed accordingly to help shape and give strength to the creation of this study.

Abdulhammed et al. (2019) propose different methods of dealing with imbalanced data in order to create a reliable intrusion detection system that makes use of the most current dataset available, the Coburg Intrusion Detection Dataset-001 (CIDDS-001). Using deep neural networks, random forests, voting, variational autoencoders, and stacking machine learning classifiers, the sampling method's effectiveness on CIDDS-001 is both theoretically and experimentally tested. With the balanced class distribution using fewer samples, the proposed system could identify the attacks with almost 99.99 percent accuracy, thus easier to use while dealing with data fusion problems in real time, to determine the desired data categorization [8].

Kurniawan *et al.*, (2019) offers a machine learning strategy that is expected to provide improved results for detecting intrusions in the IoT system. This strategy primarily utilizes the ensemble learning method and the synthetic minority over-sampling technique (SMOTE) algorithm. The performance of this research work shows that the suggested method can identify and categories intrusions into five types: regular, probe, dos, R2L, and U2R. The evaluation results show that compared to the base learning and the methods used in previous studies, the proposed method can improve the accuracy of intrusion detection performance to 97.02%, detection rate to 97%, and false alarm rate to 0.16%, however, have not demonstrated pleasing results in the processing time performance [9].

Rezvy *et al.* (2019) Introduce They are a good option for distinguishing between legitimate traffic and malicious network assaults because to their capacity to learn intricate patterns and behaviour. This paper used a deep autoencoder dense neural network approach to detect intrusions or attacks in 5G and IoT networks. The Aegean Wi-Fi Intrusion dataset, a standard in the field of information technology, was used to evaluate the method. A overall detection accuracy of 99.9 percent for Flooding was one of the impressive levels of performance exhibited by the studies [10].

Heng and Weise (2019) were executed by a computer security specialist with little training in machine learning and enhanced with domain intelligence. The method is taught using known attack signatures, and it presents network traffic using TCP/IP connection attributes. It assesses this method by making use of the publicly accessible NSLKDD dataset. They reach F1-scores of

96.34%, 92.34%, 99.82%, and 98.92%, respectively, for recall, accuracy, and precision. It is useful for building IDS because of its simplicity and because of these unexpectedly strong performance outcomes [11].

Pradeepthi and Kannan, (2018) make use of neuro-fuzzy classification techniques to identify botnet communications in a new way. An application was deployed to the Eucalyptus cloud and attacked using several open-source botnet simulation tools to create the dataset utilized for the studies. Achieving a 94.78% accuracy, the system was tested using 15,000 occurrences and 56 attributes. Fuzzy rules, which are integrated into the system, greatly decrease the false positive rate when compared to other similar systems [12].

Abdulhammed *et al.* (2018) proposed study separates four groups of characteristics, with 32, 10, 7, and 5 attributes extracted in each set. The classifiers outperformed earlier studies they achieved a high level of accuracy with very few FP, taking into account the number of classes and features. Random Forest with J48 achieved a 99.99% accuracy rate using the 10-fold cross-validation approach, whereas RF with supply test achieved a 99.64% accuracy rate [13].

Kumar, Viinikainen and Hamalainen, (2016) gives the idea for, and assess the performance of, an ML-based model for NIDSs. In this work, supervised ML classifiers were trained using datasets that had labelled examples of network traffic attributes produced by both good and bad apps. Because of Android's widespread usage and large percentage of mobile malware, this study focuses on Android-based malware. The findings showed that the model has a detection accuracy of up to 99.4 per cent for both known and unknown threats. Combining this ML model with more conventional IDS may help spot more sophisticated attacks while cutting down on false positives [14].

The present state of IDSs research is summarized in Table I in cybersecurity, focusing on the novel models, datasets, important results, and difficulties encountered in these studies.

Table 1: Overview of Recent Studies on Predictive Modeling of Intrusions detection in Cybersecurity

Author	Proposed Work	Dataset	Key Findings	Challenges/recommendation
Abdulhammed et al. (2019)	Evaluated sampling techniques with multiple classifiers (DNN, RF, VAE, stacking) for imbalanced data	CIDDS-001	Achieved up to 99.99% accuracy; effective handling of class imbalance	Suitable for real-time data fusion but may require optimization for scalability
Kurniawan et al. (2019)	Ensemble learning with SMOTE for IoT intrusion detection	Custom IoT dataset	97.02% accuracy, 97% detection rate, 0.16% false alarm rate	Processing time not optimal; needs improvement
Rezvy et al. (2019)	5G and Internet of Things intrusion detection using a deep autoencoded dense neural network	Aegean Wi-Fi Intrusion Dataset	99.9% accuracy in detecting Flooding attacks	Focus on specific attack type; may need generalization to more attack classes
Heng and Weise (2019)	CNN-based IDS enriched with domain knowledge	NSL-KDD	Accuracy 98.92%, Precision 99.82%, Recall 92.34%, F1-score 96.34%	Highly scalable, simple approach; effectiveness depends on feature engineering
Pradeepthi and Kannan (2018)	Neuro-fuzzy classification for botnet traffic detection	Custom botnet dataset (Eucalyptus cloud setup)	Accuracy of 94.78%, reduced false positives	Dataset is limited in scope; model dependent on fuzzy rule design
Abdulhammed et al. (2018)	Attribute reduction and ML classification (RF, J48)	KDD Cup 1999 dataset	Accuracy up to 99.99% with 10-fold cross-validation	Shows scalability with fewer attributes; effectiveness across different attacks not detailed
Kumar, Viinikainen and	ML-based NIDS for Android malware	Custom dataset from malicious/benign	Accuracy up to 99.4%, detects	Integration with traditional IDS systems recommended for enhanced

Hamalainen (2016)	detection	Android traffic	known and unknown threats	detection

3. Research Methodology

A structured pipeline is used to construct a high-performing model that can effectively identify network intrusions in the intrusion detection approach that uses the NSL-KDD dataset. To guarantee data consistency and integrity, the procedure begins with data preparation, which involves handling null values and removing duplicate entries. For the model to efficiently handle non-numerical input, one-hot encoding is used to modify categorical characteristics. Following this, feature selection is used to keep just the most important variables, which decreases dimensionality and makes computing more efficient. When all numerical characteristics are normalized using min-max, learning becomes quicker and more consistent since the features are all scaled to the same range. The ability of the model to generalize is evaluated by separating the cleaned dataset into subsets that are used for training and testing. The development and training of a convolutional neural network (CNN) requires training data, leveraging its deep learning architecture to automatically extract spatial and temporal patterns in network traffic. Metrics like F1-score, recall, accuracy, and precision are used to evaluate the model's performance, which gives a thorough picture of how well it works. The whole process is illustrated in Figure 1.

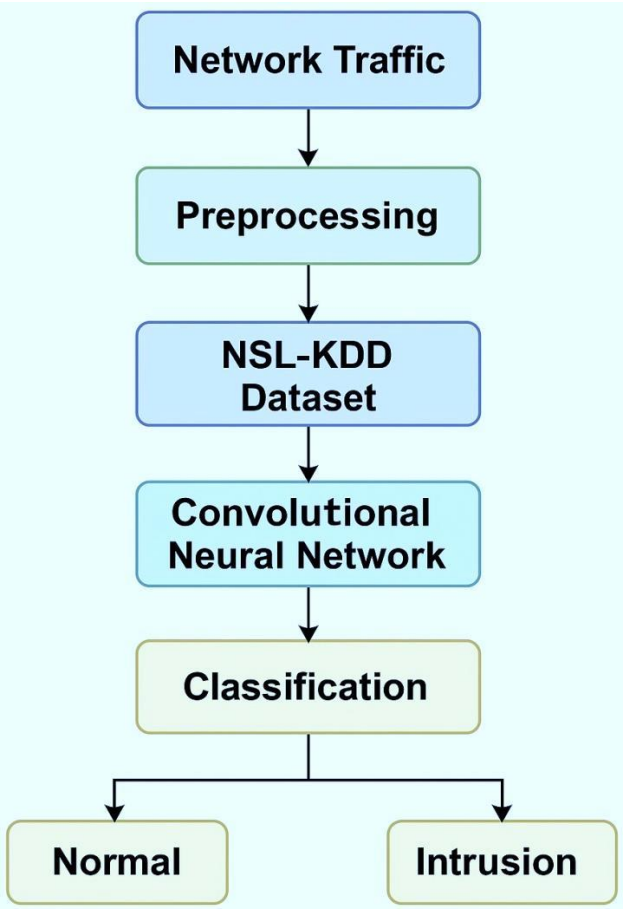

Figure 1: Proposed Flowchart for Intrusion Detection

Below is a comprehensive explanation of each step involved in the proposed flowchart for predictive modelling of intrusion detection in cybersecurity.

3.1. Data Collection

Among the several datasets available in the dataset's storage locations for NSL-KDD, the ones used for training and testing are "KDDTrain" and "KDDTest," respectively. For this dataset, 41 characteristics characterize the connection patterns, with 1 class attribute describing the sorts of attacks (normal, assault, etc.). Both types of data include numerical and symbolic properties. Inside the cell array, we transform the symbolic property into a numerical one and provide the category one independently. Data visualizations such as bar plots and heatmaps were used to examine attack distribution, feature correlations, etc., are given below:

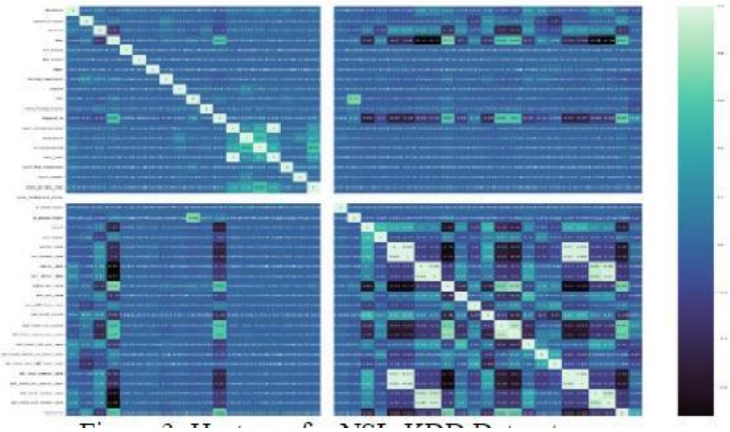

Figure 2: Heatmap for NSL-KDD Dataset

Figure 2 presents a set of study of intrusion detection using heatmaps displaying the correlation between different characteristics in the NSL-KDD dataset. In a heatmap, each block indicates a linear connection between two characteristics; greater positive correlations are shown by brighter hues, while weaker or negative correlations are indicated by darker shades. Diagonal lines with lighter colors indicate self-correlation (correlation = 1). These visualizations help identify redundant or highly correlated features, which can be optimized or removed to improve model performance and training efficiency. The diversity of patterns also highlights the complexity of interactions within the dataset.

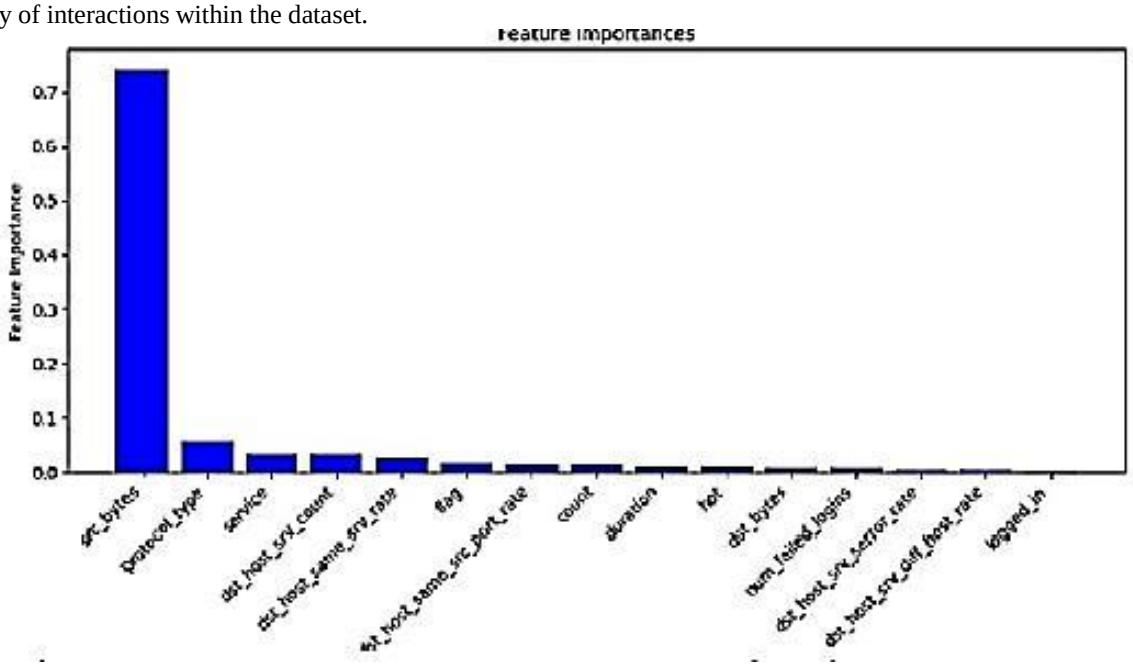

Figure 3: Feature Importance Bar Graph using the Dataset

Figure 3 illustrates the feature importance scores for various attributes in the intrusion detection model. The feature 'dst_bytes' dominates with the highest importance score, close to 0.7, indicating that it plays a critical role in predicting intrusions. Other notable features include 'protocol type', 'logged_in', 'srv_count', and 'dst_host_srv_count', though their importance scores are significantly lower, all below 0.1. The remaining features contribute minimally, with values approaching zero. This visualization helps in identifying the most influential features, guiding dimensionality reduction and improving the model's maximize output while zeroing in on critical inputs.

3.2. Data Pre-Processing

Gathering the NSL-KDD dataset, concatenating and cleaning it, and extracting pertinent characteristics were all part of the data preparation process. In machine learning, data pre-processing is a crucial step. To create a high-accuracy performance model, the pre-processing must be precise.

The following phases present an overview of the pre-processing of the NSL-KDD data, which was used in this study. The following are the important preprocessing steps:

- **Removing duplicate entries:** Duplicates may be identified and eliminated to improve the quality and integrity of data and yield more credible findings.
- **Managing null values:** The approaches such as dropping the rows containing null values or imputing the missing values with the mean maintains the consistency of the dataset.

3.3. One-Hot Encoding for Labeling

Data labelling, data tagging or data annotation is the process of assigning labels to data points in a way that ML models can more easily understand them and give accurate predictions. An alternative method to encode categorical data into a numerical format is the so-called one-hot encoding. This generates a new binary (1 or 0) column rather than assigning each category a single integer.

3.4. Data Normalization

The min--max method was adopted to normalize the records by mapping the data to a range between 0 and 1. The reason behind this was to reduce the effects of the outliers and to increase the effectiveness of the classifiers that were used.

The normalization was done based on the following mathematical formula Equation (1):

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}}$$

The original value of the feature is represented by X, while its normalized value is denoted by a. X_{min} is the minimum value of the feature, and X_{max} is the maximum value of the same.

3.5. Feature Importance

Importance of features in ML. Feature importance in machine learning is a collection of methods to estimate the relative significance of the various input variables (features) in a predictive model. It measures the impact of each attribute on the forecasts made by the model. Techniques that assign each input characteristic a numerical value regarding a specific model are called feature significance. Each trait's "importance" is indicated by the ratings. The higher the score, the more significant the feature is for the model's ability to predict a certain variable.

3.6. Data Splitting

The data was split into two groups: training and testing. 20% of the data is used for testing, while 80% is used for training.

3.7. Proposed Convolutional Neural Network (CNN) Model

CNNs learn features by implementing non-linear transformations using a stack of hierarchical layers. Tensors, which are multidimensional data arrays, comprise the data seen on the visible layer. Data typically has a grid topology; for example, time series may be seen as a 1D grid with uniform time steps, picture pixels are often arranged in a 2D grid, video files are typically in a 3D array, etc. Afterwards, a series of hidden levels retrieves different intangible attributes.

A two-dimensional kernel h calculates the 2D convolution in two dimensions using an input of x, for instance, the Equation (2) defined in given below:

$$(x * h)i, j = x[i, j] * h[i, j] = \sum n \sum m x[n, m] \cdot h[i - n][j - m]$$

the input matrix that contains their weights and the dot product of a tiny region to which they are connected.

By adding a bias term and applying a point-wise nonlinearity g to the output of the filters, a feature map is created after the convolution. When applied to a particular convolutional layer, the l-th feature map appearance is h^l, whose filters or weights are defined by the coefficients W^l,After plugging in the input x and the bias bl, the feature map h^l may be calculated using Equation (3). And (4) as follows:

$$h_{i,j}^l = g(W^l * x)_{ij} + b_l$$

where $*$ is the 2D convolution, while $g(\cdot)$ is the activation function.

Rectifier activation functions are often used in deep neural networks

$$g(x) = x^+ = max(0, x)$$

3.8. Evaluation Metrics

The suggested design's performance was evaluated using on several performance parameters. The actual values were compared to the predicted results of trained models. Based on this comparison, TP, FP, TN, and False-Negatives (FN) were estimated.

The following matrix explains F1-score, recall, accuracy, and precision:

Accuracy: A measure of how well the trained model performed relative to the whole dataset in terms of accurate predictions (input samples). It is given as Equation (5)-

$$Accuracy = \frac{\text{TP+TN}}{\text{TP+Fp+TN+FN}} \qquad (5)$$

Precision: The accuracy of a model's predictions is defined as the ratio of the number of correctly predicted positive cases to the total number of positive examples. Precision indicates. How good the classifier is in predicting the positive classes is expressed defined as Equation (6)-

$$Precision = \frac{TP}{\text{TP+FP}}$$

Recall: This metric, the ratio of events that were accurately predicted as positive to all instances that should have proved positive. In mathematical form, it is given as Equation (7)-

$$Recall = \frac{\text{TP}}{TP+FN}$$

F1 score: It is a combination of the balanced combination of accuracy and reliability l, that is, it helps to balance recall and precision. Mathematically, it is given in Equation (8)-

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

The machine and deep learning models are determined using these matrices.

4. Results and Discussion

The suggested intrusion detection system uses CNNs and is built in Python. Running the show is a 4.20 GHz Intel(R) Core (TM)2 Duo CPU T6670 paired with 4 GB of RAM. Table II displays the outcomes of assessing the proposed model using a recall-, accuracy-, precision-, and F1-score-based key performance matrix after training it on the NSL-KDD dataset. With an Accuracy of 99.9% and Precision, Recall, and F1-score all nearing 99%, the proposed CNN model performs well in terms of intrusion detection on the NSL-KDD dataset. The model's ability to identify intrusions with high accuracy and low false positive and negative rates is shown by its high degree of consistency across all performance indicators. The main advantages of the proposed CNN model include its automated feature extraction, which captures intricate patterns in network traffic data, and its robust generalization, ensuring reliable detection across diverse intrusion types. Furthermore, its superior performance compared to traditional models makes it highly suitable for real-time and large-scale IDS.

Table 2: Experiment Results of Proposed Models for of Intrusion Detection on NSL-KDD dataset

Performance matrix	Convolutional Neural Network (CNN) model
Accuracy	99.9
Precision	99
Recall	99
F1-score	99

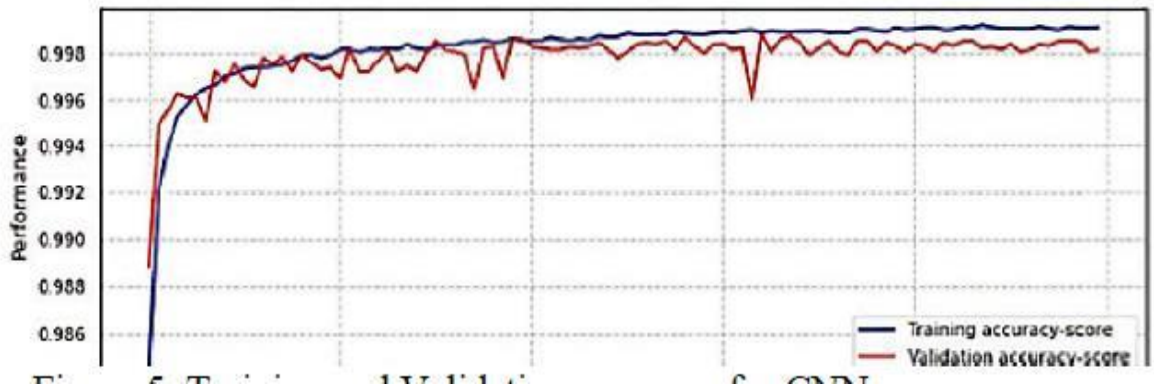

Figure 4: Accuracy curves for the CNN Model

Figure 4 illustrates the performance trend of a ML model, providing ratings for accuracy throughout training and validation over many epochs. A blue line representing the training accuracy and validation accuracy (red line) both start around 0.986 and steadily

improve, eventually stabilizing in close to 0.993. The lines remain closely aligned, indicating good generalization with minimal overfitting. Occasional minor dips in the validation accuracy suggest some fluctuations, but overall, the model maintains high and consistent performance throughout the training process.

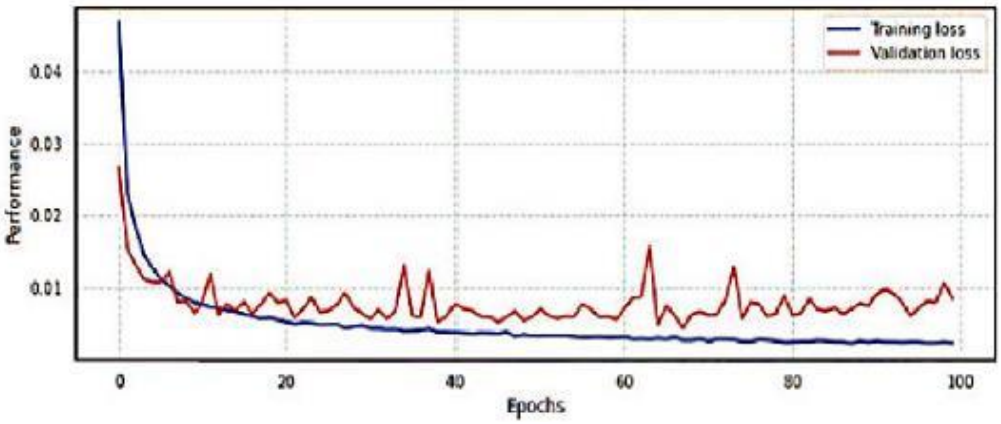

Figure 5: Loss curves for the CNN Model

Data from 100 iterations of training and validation are shown in Figure 5. Starting at a high value of about 0.42, the training loss (blue line) quickly drops to below during the first 10 epochs 0.01, eventually stabilizing around 0.005. The validation loss (red line) follows a similar initial downward trend, dropping below 0.02 early on. However, it exhibits fluctuations throughout training, with intermittent spikes reaching values between 0.01 and 0.025, particularly noticeable around epochs 40, 60, and 80. Despite the spikes, a successful model is indicated by the low validation loss, which is near to the training loss performance with slight variance in generalization.

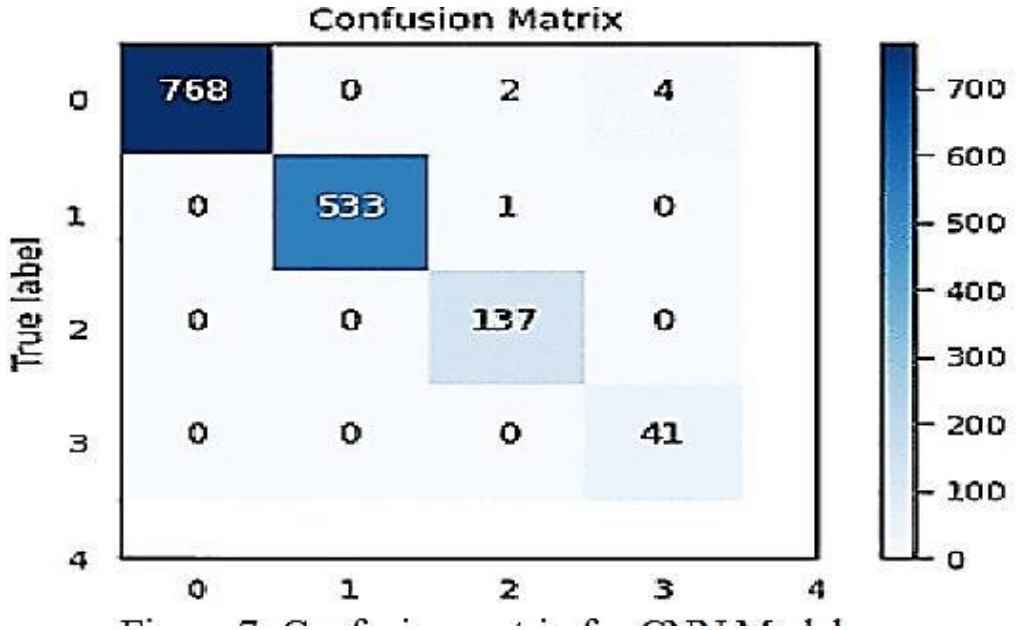

Figure 6: Confusion Matrix for CNN Model

Figure 6: The confusion matrix illustrates the classification performance of a model across four classes (0 to 3). The model shows high accuracy, especially in classifying class 0 with **768** correct predictions and class 1 with **533**. Minor misclassifications are present: class 0 had 2 samples misclassified as class 2 and 4 as class 3; class 1 had 1 mistakenly placed in class 2; there were three instances of class 2 being incorrectly placed in class (0, 1, and 2); and class 3 had all 41 instances correctly predicted. Overall, the model demonstrates strong performance with very few errors, indicating high precision and recall across all classes.

4.1. Comparative Analysis

To ensure better consequences of the proposed CNN model, a comparative accuracy examination is made with other existing models. As Table III shows, the comparison of intrusion detection on the NSL-KDD data set with different predictive models provides the following accuracy range: 93.29-96.61 per cent. Naive Bayes had the least accuracy of 75.78 per cent, ANN had 79.9 per cent accuracy, and MLP had a slightly higher accuracy of 81.6 per cent. Nevertheless, the CNN model was by far the most accurate (99.9%) among the others, thus demonstrating its greater ability to learn complicated patterns and successfully identify intrusions. This comparison has certainly shown the superiority of DL models, such as CNN, to conventional ML models in cyber defense uses.

Table 3: Comparison of different Predictive models of Intrusions detection using NSL-KDD Dataset

Models	Accuracy
Naïve Bayes [15]	75.78
ANN[16]	79.9
MLP [17]	81.6
CNN	99.9

The CNN model has a number of benefits in terms of use in intrusion detection especially when utilized on complicated data such as NSL-KDD. They lie in its capacity to automatically learn hierarchical feature representations and hence capture complex patterns in the network traffic data that a conventional model can overlook. CNNs significantly excel when dealing with large amounts of data because of the spatial awareness and parameter-sharing capabilities, which decrease the computational burden without affecting the accuracy. They are also noise resistant and can generalize effectively upon various forms of intrusions, and can thus be applied in real-time detection. Also, CNNs reduce the necessity of a large number of manual feature engineering, simplifying the model creation process and increasing the responsiveness to new cyber threats.

5. Conclusion and Future Study

IDSs are considered very useful in the protection of computer networks, but conventional IDSs have a weakness in distributed systems where the intruder may traverse among the nodes to conceal the source of attack. Distributed IDSs employ cooperative alert exchange and correlation to overcome this. Comparative analysis using the NSL-KDD dataset shows that the classic models like MLP, ANN, and NB, attained moderate accuracies of 75.78%, 79.9%, and 81.43%, respectively. Nevertheless, the suggested CNN model demonstrated an accuracy of 99.9 %, being very effective in learning complicated patterns and enhancing the detection precision. This makes CNN the best in terms of reliability among the models tested. The further direction of the research will be explainable AI (XAI), hybrid models, and testing on the other dataset to achieve better IDS performance and trust.

Further direction. Future work will involve improving the interpretability and transparency of models to develop trust in automated IDS solutions. To gain an explanation of the decisions of CNN-based models, XAI methods will be studied. Further, CNN-based models will be hybrid with RNN or attention mechanisms to learn both temporal and spatial patterns. Generalizability of the models will be further tested by expanding the evaluations to other benchmark intrusion datasets, such as UNSW-NB15 or CICIDS2017. What follows is an analysis of the system's efficiency in the practical setting, and changing cyber threats are also paramount and should be implemented and tested in real-time in varying network environments.

References

1. W. Li and H. Song, “ART: An Attack-Resistant Trust Management Scheme for Securing Vehicular Ad Hoc Networks,” *IEEE Trans. Intell. Transp. Syst.*, 2016, doi: 10.1109/TITS.2015.2494017.
2. H. Holm, “Signature based intrusion detection for zero-day attacks: (Not) A closed chapter?,” in *Proceedings of the Annual Hawaii International Conference on System Sciences*, 2014. doi: 10.1109/HICSS.2014.600.
3. Q. P. Nguyen, K. W. Lim, D. M. Divakaran, K. H. Low, and M. C. Chan, “GEE: A Gradient-based Explainable Variational Autoencoder for Network Anomaly Detection,” in *2019 IEEE Conference on Communications and Network Security, CNS 2019*, 2019. doi: 10.1109/CNS.2019.8802833.
4. S. Garg, “Predictive Analytics and Auto Remediation using Artificial Inteligence and Machine learning in Cloud Computing Operations,” *Int. J. Innov. Res. Eng. Multidiscip. Phys. Sci.*, vol. 7, no. 2, 2019.
5. F. Noorbehbahani, A. Fanian, R. Mousavi, and H. Hasannejad, “An incremental intrusion detection system using a new semi-supervised stream classification method,” *Int. J. Commun. Syst.*, vol. 30, no. 4, Mar. 2017, doi: 10.1002/dac.3002.
6. V. Kolluri, “A Pioneering Approach To Forensic Insights: Utilization Ai for Cybersecurity Incident Investigations,” *Int. J. Res. Anal. Rev. (IJRAR*, vol. 3, no. 3, 2016.
7. V. Kolluri, “An In-Depth Exploration of Unveiling Vulnerabilities: Exploring Risks in AI Models and Algorithms,” *Int. J. Res. Anal. Rev.*, vol. 1, no. 3, pp. 910–913, 2014.

8. R. Abdulhammed, M. Faezipour, A. Abuzneid, and A. AbuMallouh, "Deep and Machine Learning Approaches for Anomaly-Based Intrusion Detection of Imbalanced Network Traffic," *IEEE Sensors Lett.*, vol. 3, no. 1, pp. 1–4, Jan. 2019, doi: 10.1109/LSENS.2018.2879990.
9. A. A. Kurniawan, H. A. Santoso, M. A. Soeleman, and A. Z. Fanani, "Intrusion Detection System as Audit in IoT Infrastructure using Ensemble Learning and SMOTE Method," in *Proceeding - 2019 5th International Conference on Science in Information Technology: Embracing Industry 4.0: Towards Innovation in Cyber Physical System, ICSITech 2019*, 2019. doi: 10.1109/ICSITech46713.2019.8987524.
10. S. Rezvy, Y. Luo, M. Petridis, A. Lasebae, and T. Zebin, "An efficient deep learning model for intrusion classification and prediction in 5G and IoT networks," in *2019 53rd Annual Conference on Information Sciences and Systems, CISS 2019*, 2019. doi: 10.1109/CISS.2019.8693059.
11. L. Heng and T. Weise, "Intrusion Detection System Using Convolutional Neuronal Networks: A Cognitive Computing Approach for Anomaly Detection based on Deep Learning," in *Proceedings of 2019 IEEE 18th International Conference on Cognitive Informatics and Cognitive Computing, ICCI*CC 2019*, 2019. doi: 10.1109/ICCICC46617.2019.9146088.
12. K. V Pradeepthi and A. Kannan, "Detection of Botnet traffic by using Neuro-fuzzy based Intrusion Detection," *2018 Tenth Int. Conf. Adv. Comput.*, pp. 118–123, Dec. 2018, doi: 10.1109/ICoAC44903.2018.8939109.
13. R. Abdulhammed, M. Faezipour, A. Abuzneid, and A. Alessa, "Enhancing Wireless Intrusion Detection Using Machine Learning Classification with Reduced Attribute Sets," in *2018 14th International Wireless Communications and Mobile Computing Conference, IWCMC 2018*, 2018. doi: 10.1109/IWCMC.2018.8450479.
14. S. Kumar, A. Viinikainen, and T. Hamalainen, "Machine learning classification model for Network based Intrusion Detection System," in *2016 11th International Conference for Internet Technology and Secured Transactions (ICITST)*, 2016, pp. 242–249. doi: 10.1109/ICITST.2016.7856705.
15. B. Ingre, A. Yadav, and A. K. Soni, "Decision Tree Based Intrusion Detection System for NSL-KDD Dataset," in *Smart Innovation, Systems and Technologies*, vol. 84, no. March, 2018. doi: 10.1007/978-3-319-63645-0_23.
16. B. Ingre and A. Yadav, "Performance analysis of NSL-KDD dataset using ANN," *Int. Conf. Signal Process. Commun. Eng. Syst. - Proc. SPACES 2015, Assoc. with IEEE*, no. July, pp. 92–96, 2015, doi: 10.1109/SPACES.2015.7058223.
17. C. Ieracitano *et al.*, *Statistical Analysis Driven Optimized Deep Learning System for Intrusion Detection*, vol. 10989 LNAI. Springer International Publishing, 2018. doi: 10.1007/978-3-030-00563-4_74.
18. Polu, A. R., Buddula, D. V. K. R., Narra, B., Gupta, A., Vattikonda, N., & Patchipulusu, H. (2021). Evolution of AI in Software Development and Cybersecurity: Unifying Automation, Innovation, and Protection in the Digital Age. *Available at SSRN 5266517*.
19. 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*.
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. Kalla, D. (2022). AI-Powered Driver Behavior Analysis and Accident Prevention Systems for Advanced Driver Assistance. *International Journal of Scientific Research and Modern Technology (IJSRMT) Volume*, *1*.
24. 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.
25. Kuraku, D. S., Kalla, D., & Samaah, F. (2022). Navigating the link between internet user attitudes and cybersecurity awareness in the era of phishing challenges. *International Advanced Research Journal in Science, Engineering and Technology*, *9*(12).
26. 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).
27. Karaka, L. M. (2021). Optimising Product Enhancements Strategic Approaches to Managing Complexity. *Available at SSRN 5147875*.
28. Polu, A. R., Vattikonda, N., Buddula, D. V. K. R., Narra, B., Patchipulusu, H., & Gupta, A. (2021). Integrating AI-Based Sentiment Analysis With Social Media Data For Enhanced Marketing Insights. *Available at SSRN 5266555*.
29. 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.

30. Kalla, D., Smith, N., Samaah, F., & Polimetla, K. (2022). Enhancing Early Diagnosis: Machine Learning Applications in Diabetes Prediction. *Journal of Artificial Intelligence & Cloud Computing. SRC/JAICC-205. DOI: doi. org/10.47363/JAICC/2022 (1)*, *191*, 2-7.
31. Kalla, D., Kuraku, D. S., & Samaah, F. (2021). Enhancing cyber security by predicting malwares using supervised machine learning models. *International Journal of Computing and Artificial Intelligence*, *2*(2), 55-62.
32. Katari, A., & Kalla, D. (2021). Cost Optimization in Cloud-Based Financial Data Lakes: Techniques and Case Studies. *ESP Journal of Engineering & Technology Advancements (ESP-JETA)*, *1*(1), 150-157.
33. Kalla, D., Smith, N., Samaah, F., & Polimetla, K. (2021). Facial Emotion and Sentiment Detection Using Convolutional Neural Network. *Indian Journal of Artificial Intelligence Research (INDJAIR)*, *1*(1), 1-13.