Babylonian Journal of Networking
Vol.2024, **pp**. 171–181
DOI: https://doi.org/10.58496/BJN/2024/017; ISSN: 3006-5372
https://mesopotamian.press/journals/index.php/BJN

Research Article

A Novel Deep Learning Approach for Detecting Types of Attacks in the NSL-KDD Dataset

Hadeel M Saleh [1,*], Hend Marouane [2], Ahmed Fakhfakh [2]

[1] *National School of Electronics and Telecommunications (ENET'COM), NTS'COM Laboratory Safax University, Tunisia.*
[2] *Digital and Numeric Research Center of Safax (CRNS), Tunisia.*

ARTICLE INFO

Article History

Received 05 Jun 2024
Revised 30 Jul 2024
Accepted 07 Aug 2024
Published 01 Sep 2024

Keywords

Cloud

intrusion detection system (IDS)

distributed denial of service (DDoS) attack

NSL-KDD dataset

ABSTRACT

The growing prevalence of Internet intrusions poses significant threats to the security, privacy, and reliability of systems and networks. Denial-of-service (DoS) attacks are a cause for concern as they aim to disrupt access to network resources, posing major risks. Traditional intrusion detection systems (IDS) face challenges in detecting attacks because of the evolving nature of these attacks. Therefore, advanced techniques are necessary to accomplish accurate and timely detection. This study introduces a novel approach that combines Deep learning techniques, specifically the CNN algorithm, with Principal Component Analysis (PCA) and Singular Value Decomposition (SVD) for the purpose of feature selection. The effectiveness and efficiency of our method are shown by rigorous testing on DDoS datasets. We present a novel Fast Hyper Deep Learning Model that attains a remarkable accuracy of 99%, along with perfect recall and F1-measurement scores of 100%. This model surpasses existing methodologies by a significant margin. The NSL-KDD data set allows for achieving a level of precision of100%.

1. INTRODUCTION

Security is the umbrella term for all the precautions taken to keep computer systems safe from unauthorized activity that could jeopardize data integrity or damage other resources. In today's digital world, network hacking is a serious risk, especially for businesses that depend on internet connectivity. Attackers try to block users from using network resources so they can do malicious or profitable things. Intrusion detection systems (IDS) have evolved into essential defensive tools in the information technology and communication fields in response to these worries.

These technologies are crucial for detecting and reducing different types of network breaches and act as the main line of defense against cyber threats. By identifying possible threats in real time, intrusion detection systems (IDSs) enable quick incident response. The structure of the paper is as follows: Section 2 provides a review of related work. Section 3 details the architecture of the deep learning network. Section 4 describes the research model. Section 5 assesses the proposed model through testing on simulated NSL-KDD datasets. Section 6 summarizes the contributions of the study and suggests avenues for future research.

2. RELATED WORKS

The complex structures of modern encryption algorithms, which require a great deal of computing power, have a negative impact on the efficiency of image encryption operations. As for the usage of chaotic systems in encryption methods, it is crucial to notice that although these approaches are actively investigated in academia, it has been shown that none of the encryption methods that solely uses chaotic systems can provide satisfactory security. In reference [11], a chaotic sequence was generated using a sine map. To improve system security, an elliptic curve point and dynamic permutation table were used. The study in [12] reduced the number of iterations of the hyper chaotic system from WH/4 to 2W, a significant reduction for an image with dimensions W×H. A novel post-processing technique for generating a key matrix enabled this

*Corresponding author. Email: hadealms89@gmail.com

improvement. A new approach was presented in [13], offering a low time order, high output complexity, and a simple algorithm using 3D logistic maps. Reference [14] described a new approach of encrypting medical pictures using a 2D Logistic-Gaussian hyperchaotic map. In [15], image encryption was accomplished with the aid of 3D and 4D Arnold Cat maps, and the model incorporated the secure Elliptic Curve. In [16], a unique picture encryption technique using a combination of three modified and augmented chaotic one-dimensional maps was suggested. In [17], an encryption method for 3D models was proposed using a 2D chaotic system constructed by coupling the logistic map with infinite collapse (2D-LAIC) and the semi-tensor product (STP) theory. Reference [18] introduced a bit-level permutation and hyper-chaotic system as the foundation for an encryption scheme, while [19] proposed intra bitplane scrambling for parallel image encryption. In [20], a new four-dimensional and multi-scroll hyperchaotic system was developed. In [21], a visually secure image encryption scheme was proposed by combining the adaptive-threshold sparsification compression sensing model with a novel design memristive chaotic map. In [22], a chaotic oscillator was generated using a second-order differential equation.

Novel suggestions for a key generation were proposed in [23]. The study in [24] described a collection of one-dimensional quadratic chaotic maps based on topological conjugate theory. In [25], a new framework utilizing finite precision was introduced for generating chaotic signals to improve image encryption is presented.

The encryption process was enhanced using S-BOX, an algorithm based on chaotic processes, which provided a high level of security and efficiency. In [26], the encrypted data consisting of S-Boxes generated from a chaotic logistic map was compressed before encryption. The authors in [27] suggested a 3D chaotic map using highly nonlinear S boxes for encryption, followed by a data concealing strategy based on the Lah transform. A low-dimensional chaotic scheme was employed in [28] to create an S-box with dimensions of 10 by 26. In [29] the efficacy of encryption was increased, and secure transmission was promoted they used a 3D chaotic map-based symmetric technique. In [30], the combination of various chaotic map types with an S-box was hypothesized to achieve a fast method for scrambling and encrypting colour images. In [31], the Henon map was utilized to propose new image cryptosystem key-dependent bijective S-Boxes. In [32], Combining quantum walks with the generation of cipher texts with visual meaning, a new approach of image encryption has been introduced.

This research seeks to design an efficient and secure image encryption system through simple procedures and a robust strong key. The study proposes method for encrypting images using chaotic maps and S-box algorithm. To enhance the confusion, the presented encryption method uses an S-box appended to the chaotic maps and ensures improved security while maintaining the favorable statistical characteristics of the approach.

The contribution of this work is proposing new method for key generation based on multi-stage 3D chaotic maps. The remaining parts of this work are structured as described below. Section 2 contains the existing chaotic maps used in this work. In section three, the key generation method and S-box construction are introduced, and also, we will discuss the image encryption algorithms that have been suggested. The results of the experiments and an appraisal of their effectiveness are presented in Section 4. In the end, the conclusions are discussed in the final section.

3. PROPOSED METHODOLOGY

This section describes the rapid super deep learning approach we suggested to detect various kinds of assaults on datasets. To efficiently manage the intrusion detection complexity across various dataset types, the strategy consists of feature selection, data splitting, collection, preprocessing, and model creation. The suggested techniques are: Using the NSL-KDD data set, create the Fast Hyper Deep Learning Model (FHDL) using the deep CNN technique and determine its execution times. The solution to the complexity issue is to use feature selection based on the PCA and SVD algorithms to reduce dimensionality, under fitting, and over fitting. The suggested system's methodology is depicted in Figure (1).

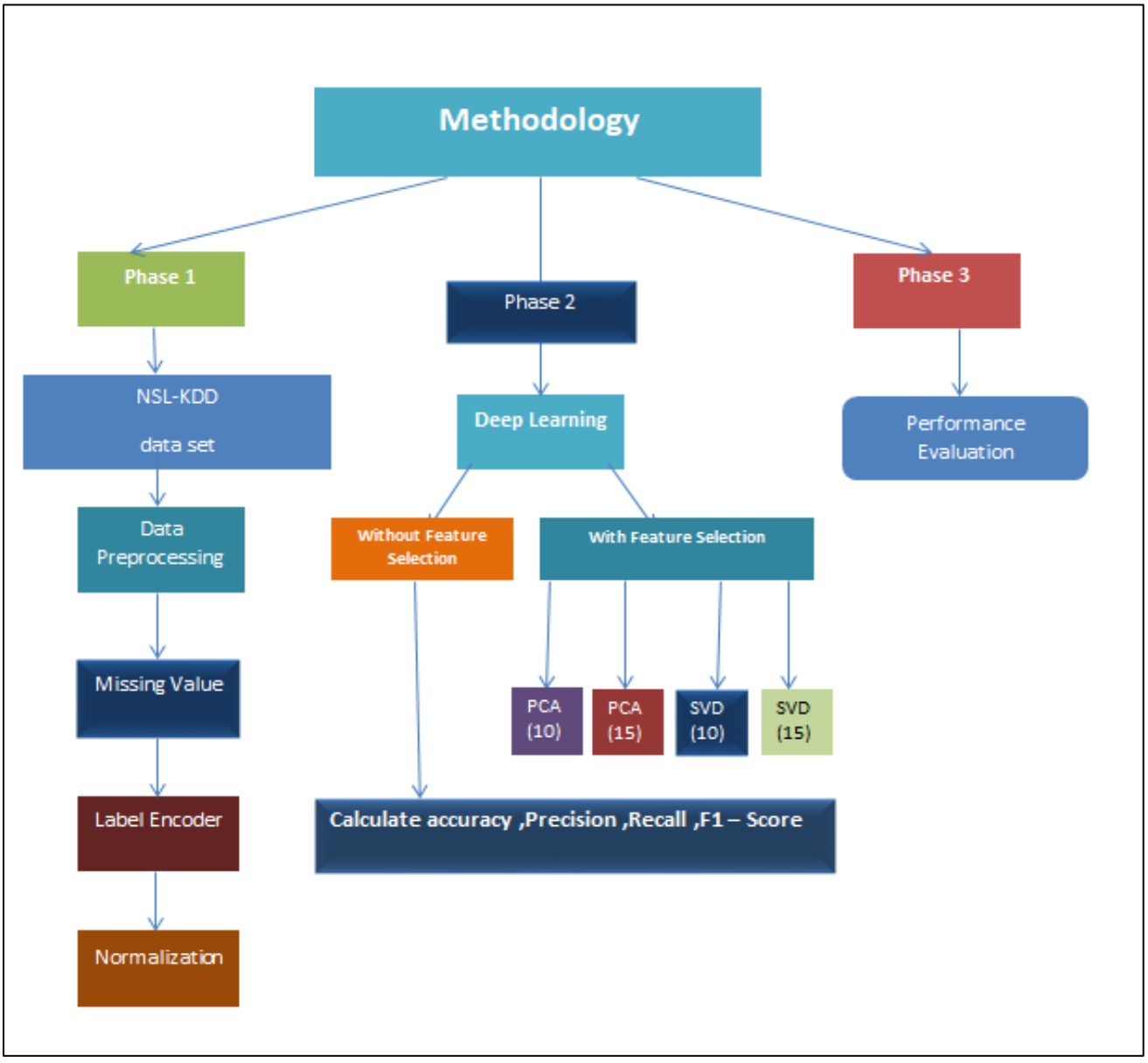

Fig .1 . Proposed Methodology

1- Data Collection

A. NSL-KDD data set

The dataset consists of 125,973 records, each including 41 distinct attributes. These records are classified into five distinct categories: normal, DoS attacks, Probe attacks, U2R attacks, and R2L attacks. It is important to recognize that the DoS class by itself comprises 7,458 pages. This study encompasses four categories of attack, each comprising several additional types of attacks, as illustrated in Figure (2). The primary objective is to distinguish between different types of Denial of Service (DoS) attacks and other forms of assaults on the NSL_KDD dataset.

The dataset used in this system was acquired from UNB. The NSL-KDD dataset was particularly developed to address some limitations identified in the KDD'99 dataset, which was acquired from Kaggle. The NSL-KDD train and test sets are sufficiently large, allowing experiments to be conducted on the entire dataset without the need to randomly select a small portion. The high prevalence of duplicate records in the KDD dataset leads to learning algorithms showing a preference for frequent records, which impedes the detection of less common but potentially more dangerous attacks such as U2R and R2L assaults. Furthermore, the presence of these repetitive records in the test set can bias the evaluation results in favor of approaches that have higher rates of detecting frequent records [17][18].

TABLE I. NSL-KDD FEATURE TYPE.

F#	Feature name	F#	Feature name	F#	Feature name
F1	Duration	F15	Su attempted	F29	Same srv rate
F2	Protocol type	F16	Num root	F30	Diff srv rate
F3	Service	F17	Num file creations	F31	Srv diff host rate
F4	Flag	F18	Num shells	F32	Dst host count
F5	Source bytes	F19	Num access files	F33	Dst host srv count
F6	Destination bytes	F20	Num outbound cmds	F34	Dst host same srv rate
F7	Land	F21	Is host login	F35	Dst host diff srv rate
F8	Wrong fragment	F22	Is guest login	F36	Dst host same src port rate
F9	Urgent	F23	Count	F37	Dst host srv diff host rate
F10	Hot	F24	Srv count	F38	Dst host serror rate
F11	Number failed logins	F25	Serror rate	F39	Dst host srv serror rate
F12	Logged in	F26	Srv serror rate	F40	Dst host rerror rate
F13	Num compromised	F27	Rerror rate	F41	Dst host srv rerror rate
F14	Root shell	F28	Srv rerror rate	F42	Class label

The NSL-KDD dataset categorizes instances as either normal or as one of 24 types of assaults, which are further classified into four categories: Denial of Service (DoS), Probe, Remote to Local (R2L), and User to Root (U2R). Figure 2 displays the four categories of Attacks found in the NSL-KDD dataset.

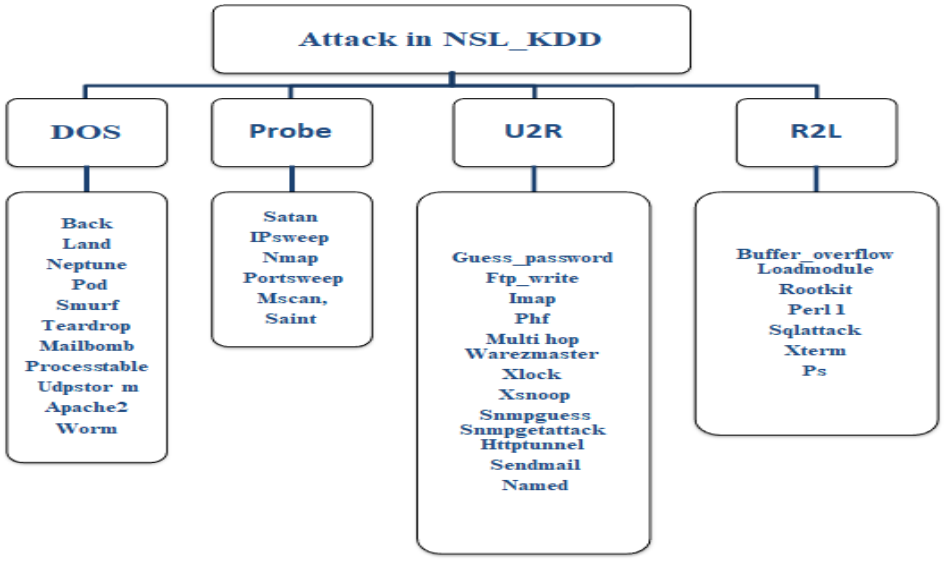

Fig .2. Types of attacks on the NSL_KDD data set

B. Preprocessing

Preprocessing is an essential and fundamental stage in the analysis of data, as well as in the fields of machine learning and data science. Data transformation is the process of converting unprocessed data into a format that is appropriate for analysis. To optimise the performance of a machine learning model and guarantee precise results, it is imperative to cleanse, standardise, and structure the data[19].

- **Label Encoder**

  The first dataset represents categorical data, such as protocols, services, and flags, in the form of strings. To incorporate these category data into our model, we employ label encoding. Label encoding is a method that converts categorical data into numerical values by assigning a unique number to each category[20].

- **Normalization**

  Normalization is a crucial preprocessing step that ensures all features are scaled in a same manner, preventing some traits from dominating the training process due to differences in size.

Our solution entails standardizing the feature values to a range of -1 to 1. Normalization is the act of rescaling the features to ensure they are within a uniform range. As stated in reference [21], this contributes to the attainment of a more stable and successful model training.

C. Data Partition

To accurately evaluate the performance of our model, we partition the NSL-KDD dataset into distinct training and testing datasets. The dataset is divided into a training set, including 70% of the data, and a testing set, comprising the remaining 30%. This division ensures that the model is trained on a sufficient amount of data while also providing a separate dataset for unbiased evaluation. By partitioning the dataset before carrying out preprocessing and applying algorithms, we guarantee that our evaluation measures accurately reflect the model's performance on unseen data.

The proposed methodology integrates data preprocessing, feature selection, and model creation to create a comprehensive solution for identifying DDoS intrusions in network security. Our approach prioritizes improving the effectiveness and durability of intrusion detection systems in practical situations [22, 23] by tackling crucial preprocessing phases and efficiently dividing the information.

D. Feature Selection

Feature selection is a method employed to pick a subset of pertinent features from the data with the aim of enhancing model performance and diminishing complexity. The three primary categories of feature selection techniques are as follows: filter methods, which utilize the statistical characteristics of the data; wrapper methods, which assess various feature sets using machine learning models; and embedded methods, which employ strategies like regularization to select features during the training process. Feature selection improves the effectiveness of training, mitigates the issue of over fitting, and promotes the interpretability of the model [24]. Our proposal involves combining deep learning architectures with feature selection techniques, such as PCA and SVD. Our hybrid strategy seamlessly integrates feature selection into the deep learning pipeline, in contrast to conventional methods that require feature selection to be conducted as a separate step prior to model training. Our objective is to improve the efficiency and efficacy of the model by combining the expressive abilities of deep learning with dimensionality reduction techniques.

E. Deep Learning Model

The classifier model comprises 27 layers of a Convolutional Neural Network (CNN). The model's design is robust and adept at properly addressing various challenges, including generalization and complexity. It enhances the precision of classification, measures the execution time, and reduces the complexity in terms of both time and space. Furthermore, the system's efficacy is demonstrated by the depiction of FHDL in figure (3) without feature selection and figure (4) with feature selection. Feature selection is an essential part of data preprocessing for machine learning and data analysis activities. It involves the use of advanced techniques such as Principal Component Analysis (PCA) and Singular Value Decomposition (SVD). Principal Component Analysis (PCA) is especially advantageous when the dataset contains several features that are highly associated with each other. This is because PCA can identify the most significant linear combinations of the original features.

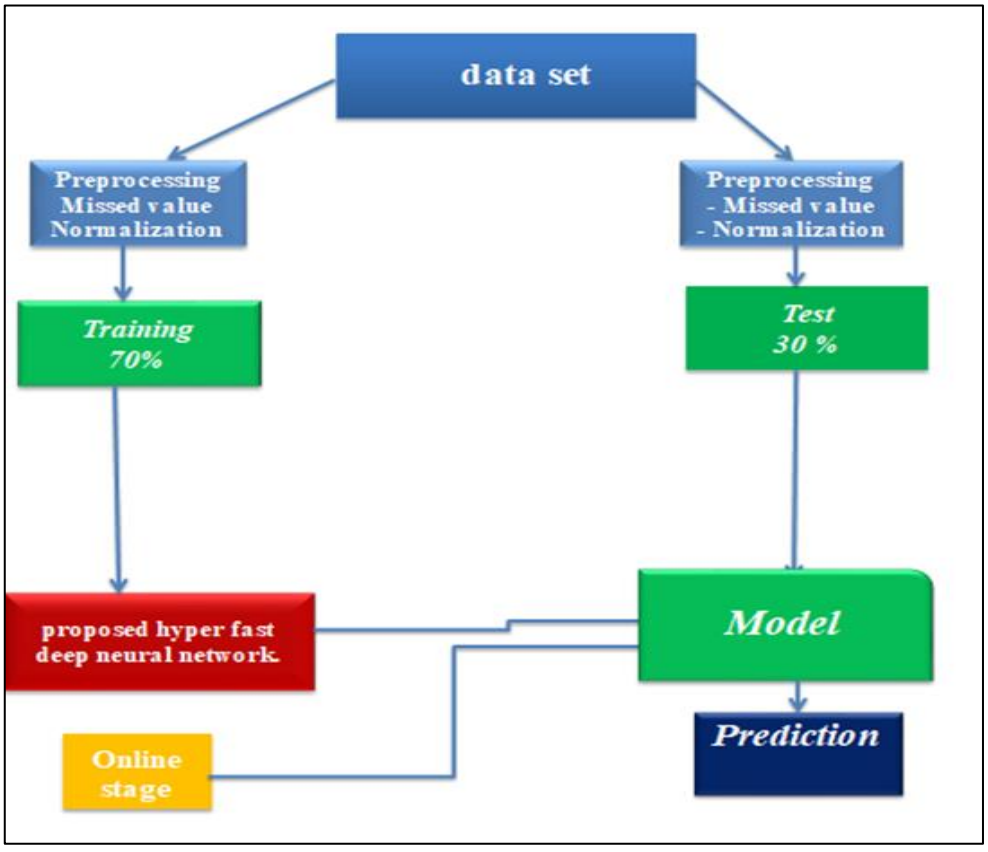

Fig .3. Deep learning-without feature selection model.

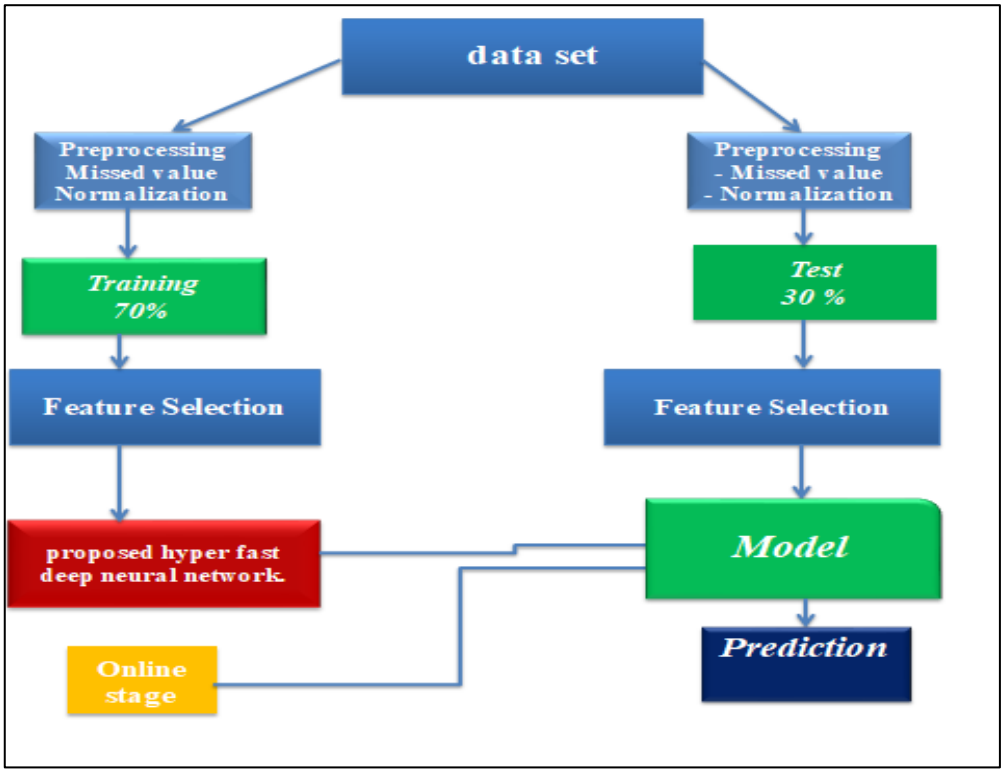

Fig 4. Deep learning-feature selection model.

Algorithm 1: Fast Hybrid Deep Learning Model for NSL-KDD Intrusion Detection without Feature Selection
Input :Raw Data .
Output: Trained intrusion detection model.
Load the Data Set a CSV file.
Normalize the feature data using Standard Scaler.
Encode the target variable using Label Binarizer.
Reshape the input data for 1D convolution.
Data Splitting:
Split the data into training (70%) and testing (30%) sets.
Model Architecture:
Create a Sequential model with the following layers:
Multiple Conv1D layers with increasing filters (16, 32, 64)
MaxPooling1D layers after
each Conv1D LeakyReLU activation functions Dense layers (128 units, 512 units)
Additional Conv1D layers
Final Dense layer with 5 units and softmax activation
Model Compilation:
Use Adam optimizer with a learning rate of 0.001 Set loss function to categorical cross entropy Use accuracy as the metric
Model Training:
Train the model for 100 epochs Use a batch size of 128 Validate on the test set during training
Prediction and Evaluation:
Make predictions on the test set Convert predicted probabilities to class labels Calculate precision, recall, and F1-score using weighted average
Output Results:
Print the calculated precision, recall, and F1-score
END

F. CNN Classifier Architecture

The architecture does not include pooling layers. Instead, a flattening layer is used to convert the 2D outputs into a 1D vector. The final dense layer of the output consists of five units and is activated by softmax, indicating its use for a five-class classification task.

This design is distinguished by the methodical utilization of LeakyReLU activation and the alternating configuration of convolutional and dense layers, which could be beneficial for capturing both local and global patterns. The model is remarkably complex, comprising a total of 27 levels, as recorded in Table 3.1. The architecture appears to be specifically built for a complex time series or sequence classification problem, with the ability to learn intricate local properties and broader general patterns in the data.

The incorporation of convolutional layers in the model enables the automated extraction of features, while the dense layers facilitate the amalgamation of these attributes to produce ultimate classification determinations, as specified in Table2. The final section of the model includes three additional convolutional layers, each having 16, 16, and 35 filters, respectively. However, in this case, there are no pooling layers present. Instead, they are substituted with a flattening layer that converts the 2D outputs into a 1D vector.

The final output dense layer comprises five units and utilizes a softmax activation function, indicating its purpose for a classification assignment involving five distinct classes. This design is distinguished by the methodical utilization of LeakyReLU activation and the alternating configuration of convolutional and dense layers, which could be beneficial for capturing both local and global patterns. The model is significantly deep, comprising a total of 27 levels, as specified in Table 2. The design appears to be custom-made for a complex time series or sequence classification problem, with the capability to capture detailed local characteristics as well as broader global trends in the data. The incorporation of convolutional layers in the model enables the autonomous extraction of features, while the dense layers facilitate the amalgamation of these attributes to generate definitive classification outcomes.

TABLE II. NSL_KDD DATA SET DEEP LEARNING RESULTS

	Precision	Accuracy	Recall	F1-Score	Time Execution
Deep	100	99	100	99	2 sec
Feature Selection PCA(10)	99	99	99	99	0S 85 US
Feature Selection PCA(15)	99	99	99	99	0s 201 us
Feature Selection SVD(10)	100	99.9	100	100	87 us
Feature Selection SVD(15)	100	99.9	100	100	0s 207 us

G. Evolution Measures

Incorporating evaluation metrics is essential for optimizing the performance of a machine learning model. Thus, selecting suitable evaluation criteria is a vital stage in distinguishing and achieving the most effective model. The user's input is represented by the text [24].

Confusion Matrix: From an academic perspective, there are several criteria that can be employed to evaluate the efficacy of particular categorization algorithms. The measurements encompass precision, recall, accuracy, and the F1-score. The calculation of these metrics relies on the calculation of a confusion matrix, which is a table that summarizes the number of accurately or wrongly predicted examples by a specific classification model, as shown in Figure 6. The explanation for each value is provided in detail below [25]:

1. A True Positive (TP) refers to instances where positive cases have been accurately categorized.
2. A False Negative (FN) occurs when positive examples are incorrectly classified as negatives.
3. A False Positive (FP) occurs when a negative event is mistakenly identified and classified as a positive one.
4. A True Negative (TN) refers to a classification model that accurately identifies situations as negative.

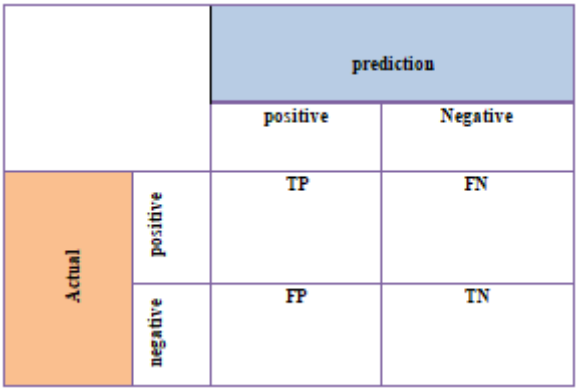

Fig .6. Confusion matrix

Accuracy: The determination is based on the proportion of accurate forecasts to the overall number of predictions made. This ratio represents the probability of a model properly forecasting occurrences. Equation (1) demonstrates the precision of a model, which is calculated by dividing the number of accurate forecasts by the total number of predictions. This ratio indicates the probability that the model will accurately predict future results.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad \dots \quad (1)$$

Precision: refers to the degree of accuracy and specificity in categorising a group of documents and defining the content of the collection. The accuracy of the class ci, denoted by the symbol (Pi), can be measured using the approach described below, as illustrated in Equation (2):

$$P_i = \frac{TP_i}{TP_i + FP_i} \quad \dots \quad (2)$$

Recall : The equation illustrates how recall quantifies the efficiency of a classifier in classifying documents. The recall of class ci, Ri, can be computed using the formula:

$$R_i = \frac{TP_i}{TP_i + FN_i} \quad \ldots (3)$$

In this case, TP_i points to a true-positive value. FP_i stands for false positives, and FN_i represents false negatives.
F1-Score : Sync rate is represented by the F1 measure. When F1 is raised, the system functions properly overall. F1 is described as follows in accordance with Equations (4) and (5):

$$F1 = \frac{2 \times precision \times recall}{precision + recall} \quad \ldots (4)$$

$$= \frac{2TP}{2TP + FP + FN} \quad \ldots (5)$$

4. LIMITATIONS AND FUTURE WORKS

Although the DEEP DDOS model has demonstrated promising results, it is essential to take into account various restrictions. The performance of the model is greatly influenced by the quality and representativeness of the training data. The presence of imbalances or biases in the dataset can impede the model's capacity to effectively identify infrequent attack patterns or make correct predictions on novel data.

Moreover, the model's significant computational complexity may pose difficulties for real-time implementation, especially in resource-constrained contexts. The model's scalability and feasibility in real-world applications may be limited due to the substantial computer resources required for its intricate structure and parameterization. Furthermore, the comprehensibility of deep learning models continues to be a persistent concern, impeding stakeholders from completely grasping and accepting the model's discoveries. Deep learning algorithms are inherently opaque, meaning that they lack transparency. This lack of transparency might hinder their acceptance in critical security applications, even when efforts are made to clarify the structure and feature representations of the model.

Rectifying the highlighted shortcomings offers multiple opportunities for further investigation. To improve the resilience and applicability of the DEEP-DDOS model, one potential strategy is to gather a wider range of comprehensive and diverse datasets. Utilizing sophisticated regularization techniques and data augmentation procedures can successfully reduce biases and enhance the performance of the model when handling novel data. To successfully use the model in real-world scenarios, it is crucial to optimize the structure and parameters of the model to minimize computing complexity while yet achieving excellent performance. Model distillation, quantization, and pruning are optimization techniques that improve the performance and enable the deployment of the model on edge devices and cloud settings.

Furthermore, improving the comprehensibility of the model by employing approaches like saliency mapping, attention mechanisms, and model-agnostic interpretability techniques can augment comprehension and foster assurance among stakeholders. Sharing actionable insights into the decision-making process of the model could foster collaboration between machine learning experts and security professionals, resulting in more efficient strategies for reducing threats. By investigating innovative methods for feature engineering, utilizing group learning strategies, and creating hybrid models that combine deep learning with conventional machine learning algorithms, it is possible to enhance the performance and robustness of the model, making it more resistant to adversarial attacks. By integrating various methodological strategies, scholars can create intrusion detection systems that are very resilient and adaptable, allowing them to efficiently counteract novel and emerging cyber threats.

5. CONCLUSION

The created model has demonstrated promising results in identifying intrusions in NSL-KDD by utilizing FHDL layers and finely tuned parameters. The model's excellent performance can be attributed to its utilization of advanced approaches such as Adaptive Moment Estimation Optimization and careful selection of hyper parameters, including kernel size and padding parameters. The model attains a precision, recall, and F-measure scores of 100 on the validation dataset, significantly reducing both false positives and false negatives, leading to an accuracy score of 99%.

These findings highlight the effectiveness of our methodology in improving the cyber security of NSL-KDD systems, which is essential for protecting sensitive information and procedures. The technique we suggest effectively reduces risks, improves the security, and guarantees the dependability of smart healthcare systems. Furthermore, the strong performance metrics confirm that the model is suitable for practical use in real-world scenarios, where rapid and precise detection of intrusions is crucial. Our approach provides a refined and efficient framework for identifying intrusions in NSL-KDD systems, effectively tackling security issues in a pragmatic manner. Additional investigation and verification of the model in other NSL-KDD

settings can contribute to the development of secure and dependable smart healthcare ecosystems. Moreover, this will improve the model's usage in the healthcare sector.

Conflicts of Interest

The absence of any financial or non-financial competing interests is mentioned in the paper.

Funding

The paper does not disclose any collaborations with funded projects or institutions, indicating the author had no external financial support.

Acknowledgment

The author would like to thank the administrative staff at the institution for their assistance and logistical support throughout the duration of this research.

References

[1] Y. Shang, "Prevention and detection of DOS attack in virtual cloud computing environment using Naive Bayes algorithm of machine learning," Measurement, vol. 143, p. 100991, 2024.
[2] A. Ahmim, F. Maazouzi, M. Ahmim, S. Namane, and I. B. Dhaou, "Distributed denial of service attack detection for the Internet of Things using hybrid deep learning model," IEEE Access, vol. 11, pp. 119862–119875, 2023.
[3] R. Uddin, S. A. Kumar, and V. Chamola, "Denial of service attacks in edge computing layers: Taxonomy, vulnerabilities, threats, and solutions," Ad Hoc Networks, vol. 152, p. 103322, 2024.
[4] Y. N. Kunang, S. Nurmaini, D. Stiawan, and B. Y. Suprapto, "Attack classification of an intrusion detection system using deep learning and hyperparameter optimization," Journal of Information Security and Applications, vol. 58, p. 102804, 2021.
[5] A. V. Kachavimath and D. G. Narayan, "A deep learning-based framework for distributed denial-of-service attacks detection in cloud environment," in Advances in Computing and Network Communications: Proceedings of CoCoNet 2020, Volume 1, Springer, 2021, pp. 605–618.
[6] P. Iyer, T. Jadhav, A. Pillai, and Samundiswary, "Analysis of modern intrusion detection algorithms and developing a smart IDS," in 2021 International Conference on Intelligent Technologies (CONIT), 2021, pp. 1–7. doi:10.1109/CONIT51480.2021.9498519.
[7] C. C. Ugwu, O. O. Obe, O. S. Popola, and A. O. Adetunmbi, "A distributed denial of service attack detection system using long short-term memory with singular value decomposition," in 2020 IEEE 2nd International Conference on Cyberspace (CYBER NIGERIA), 2021, pp. 112–118. doi:10.1109/CYBERNIGERIA51635.2021.9428870.
[8] H. A. Alamri and V. Thayananthan, "Bandwidth control mechanism and extreme gradient boosting algorithm for protecting software-defined networks against DDoS attacks," IEEE Access, vol. 8, pp. 194269–194288, 2020. doi:10.1109/ACCESS.2020.3033942.
[9] G. Kadam, S. Parekh, P. Agnihotri, D. Ambawade, and P. Bhavathankar, "An approach to reduce uncertainty problem in network intrusion detection systems," in 2020 IEEE 15th International Conference on Industrial and Information Systems (ICIIS), 2020, pp. 586–590. doi:10.1109/ICIIS51140.2020.9342634.
[10] S. Shanmuga Priya, M. Sivaram, D. Yuvaraj, and A. Jayanthiladevi, "Machine learning-based DDoS detection," in 2020 International Conference on Emerging Smart Computing and Informatics (ESCI 2020), 2020, pp. 234–237.
[11] D. Parfenov, L. Kuznetsova, N. Yanishevskaya, I. Bolodurina, A. Zhigalov, and L. Legashev, "Research application of ensemble machine learning methods to the problem of multiclass classification of DDoS attacks identification," in 2020 International Conference on Engineering and Telecommunication (EnT), 2020, pp. 1–7. doi:10.1109/EnT50437.2020.9431255.
[12] G. S. Kushwah and V. Ranga, "Voting extreme learning machine based distributed denial of service attack detection in cloud computing," Journal of Information Security and Applications, vol. 53, 2020. doi:10.1016/j.jisa.2020.102532.
[13] E. Nejati, H. Shakeri, and H. R. Sani, "Ensembling tree-based classifiers for improving the accuracy of cyber attack detection," in 8th Iranian Joint Congress on Fuzzy and Intelligent Systems (CFIS 2020), 2020, pp. 70–76. doi:10.1109/CFIS49607.2020.9238705.
[14] S. Das, D. Venugopal, and S. Shiva, "A holistic approach for detecting DDoS attacks by using ensemble unsupervised machine learning," Advances in Intelligent Systems and Computing, vol. 1130, pp. 721–738, 2020. doi:10.1007/978-3-030-39442-4_53.

[15] A. Bhardwaj, V. Mangat, and R. Vig, "Hyperband tuned deep neural network with well-posed stacked sparse autoencoder for detection of DDoS attacks in cloud," IEEE Access, vol. 8, pp. 181916–181929, 2020. doi:10.1109/ACCESS.2020.3028690.
[16] J. Liu, B. Kantarci, and C. Adams, "Machine learning-driven intrusion detection for Contiki-NG-based IoT networks exposed to NSL-KDD dataset," in Proceedings of the 2nd ACM Workshop on Wireless Security and Machine Learning (WiseML 2020), 2020, pp. 25–30. doi:10.1145/3395352.3402621.
[17] B. Mohammed and E. K. Gbashi, "Intrusion detection system for NSL-KDD dataset based on deep learning and recursive feature elimination," Engineering and Technology Journal, vol. 39, no. 7, pp. 1069–1079, 2021. doi:10.3934/mbe.2022493.
[18] A. D. Vibhute, C. H. Patil, A. V. Mane, and K. V. Kale, "Towards detection of network anomalies using machine learning algorithms on the NSL-KDD benchmark datasets," Procedia Computer Science, vol. 233, pp. 960-969, 2024.
[19] S. García, J. Luengo, and F. Herrera, Data Preprocessing in Data Mining, vol. 72, Cham, Switzerland: Springer International Publishing, pp. 59-139, 2015.
[20] D. Shah, Z. Y. Xue, and T. M. Aamodt, "Label encoding for regression networks," arXiv preprint arXiv:2212.01927, 2022.
[21] S. G. O. P. A. L. Patro and K. K. Sahu, "Normalization: A preprocessing stage," arXiv preprint arXiv:1503.06462, 2015.
[22] X. Jing, Z. Yan, and W. Pedrycz, "Security data collection and data analytics in the internet: a survey," IEEE Communications Surveys & Tutorials, vol. 21, no. 1, pp. 586–618, 2019.
[23] A. K. Soliman, C. Salama, and H. K. Mohamed, "Detecting DNS reflection amplification DDoS attack originating from the cloud," in Proceedings of the 2018 13th International Conference on Computer Engineering and Systems (ICCES 2018), 2019, pp. 145–150.
[24] D. Theng and K. K. Bhoyar, "Feature selection techniques for machine learning: A survey of more than two decades of research," Knowledge and Information Systems, vol. 66, no. 3, pp. 1575-1637, 2024.
[25] M. Hossin and M. N. Sulaiman, "A review on evaluation metrics for data classification evaluations," International Journal of Data Mining & Knowledge Management Process, vol. 5, no. 2, pp. 1, 2015.
[26] N. Ali, D. Neagu, and P. Trundle, "Evaluation of k-nearest neighbour classifier performance for heterogeneous data sets," SN Applied Sciences, vol. 1, pp. 1-15, 2019.