International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Deep Learning Inspired Intelligent Framework to Ensure Effective Intrution Detection in Cloud

Vadetay Saraswathi Bai[1]*, Prof T. Sudha[2]

Submitted: 08/12/2023 **Revised**: 19/01/2024 **Accepted**: 30/01/2024

Abstract: In the context of cloud forensics, the exponential growth in data generation due to the widespread use of cloud computing has created substantial issues for protecting the safety and privacy of cloud-based infrastructure. Many security solutions, such as intrusion detection systems (IDS), based on artificial intelligence (AI) have been created to deal with these threats. Intrusion detection systems (IDS) are a paradigm for assessing the safety of network traffic in the cloud. This research uses principle component analysis (PCA) and singular value decomposition (SVD) to minimize the dimensionality of the feature space, which in turn improves the accuracy of IDS in cloud forensics. In order to evaluate the efficacy of these techniques, they are applied to the UNSW-NB15 data set, which serves as a standard for IDS testing in cloud forensics. Our results show that Classifier, an algorithm based on deep learning, is superior to other approaches when it comes to detecting fraudulent data streams in cloud-based networks. This provides support for the hypothesis that integrating PCA, SVD, and deep learning techniques into cloud forensics analysis might provide fruitful results. To guard against cyber threats and guarantee the dependability and security of cloud-based systems, this work adds to the continuing efforts to enhance the accuracy and efficiency of IDS systems in cloud forensics.

Keywords: *Cloud Forensics, Data Generation, Cloud Computing, Artificial Intelligence (AI), Principal Component Analysis (PCA), Singular Value Decomposition (SVD)*

1. Introduction

Cyberspace has become an integral part of our lives. We use it for work, school, entertainment, and communication. However, cyberspace is also a dangerous place. Cyberattacks are on the rise, and they can have a devastating impact on individuals, businesses, and governments. Machine learning is a branch of artificial intelligence that allows computers to learn without being explicitly programmed. It is a powerful tool that can be used to solve a variety of problems, including those in cybersecurity. Machine learning algorithms can be used to detect intrusions, identify malware, and analyze data from computer systems to identify evidence of crime. They can also be used to predict future attacks and to develop new security measures. Machine learning is a promising tool for cybersecurity, but it is important to note that it is not a silver bullet. Machine learning algorithms can be fooled by attackers, and they can make mistakes. It is important to use machine learning algorithms in conjunction with other security measures to protect computer systems and networks.

[1*]*Research Scholar, Dept. of CSE, School of Engineering and Technology, Sri Padmavati Mahila Visvavidyalayam Email id: saihumika9@gmail.com*
[2]*Dept. of Computer Science, Sri Padmavati Mahila Visvavidyalayam, Email id: thatimakula_sudha@yahoo.com*

Role of machine learning in cybersecurity:

- **Intrusion detection**: Machine learning algorithms can be used to detect unauthorized access to computer systems. They can do this by analyzing network traffic, system logs, and other data.
- **Malware detection:** Machine learning algorithms can be used to detect malicious software. They can do this by analyzing the code of the software, the behavior of the software, and the files that the software creates.
- **Forensic analysis:** Machine learning algorithms can be used to analyze data from computer systems to identify evidence of crime. They can do this by looking for patterns in the data that are indicative of criminal activity.
- **Predictive analytics:** Machine learning algorithms can be used to predict future attacks. They can do this by analyzing historical data to identify patterns that are associated with attacks.
- **Security automation:** Machine learning algorithms can be used to automate security tasks. They can do this by identifying security risks and then taking steps to mitigate those risks.

Machine learning is a powerful tool that can be used to improve cybersecurity. However, it is important to note that machine learning algorithms are not perfect. They can make mistakes, and they can be fooled by attackers. It is important to use machine learning algorithms in

conjunction with other security measures to protect computer systems and networks. Cybersecurity and cyber forensics are two important fields that are constantly evolving where machine learning is a powerful tool that can be used to improve cybersecurity and cyber forensics, but it is important to be aware of the challenges and limitations of using machine learning algorithms.

One of the challenges of using machine learning algorithms for cybersecurity and cyber forensics is that the data sets are often very large and complex. This can make it difficult to train machine learning algorithms to accurately detect and classify malicious activity. Another challenge is that machine learning algorithms can be fooled by attackers. Attackers can use techniques such as adversarial machine learning to create malware that is designed to evade detection by machine learning algorithms. Finally, machine learning algorithms can be biased. This means that they may be more likely to detect or classify certain types of malicious activity than others. This can lead to false positives and false negatives.

Despite these challenges, machine learning is a promising tool for cybersecurity and cyber forensics. Machine learning algorithms can be used to detect intrusions, identify malware, and analyze data from computer systems to identify evidence of crime. They can also be used to predict future attacks and to develop new security measures.

It is important to use machine learning algorithms in conjunction with other security measures to protect computer systems and networks. Machine learning algorithms can be a valuable tool for cybersecurity and cyber forensics, but they are not a silver bullet.

Developing effective cybersecurity measures is a difficult challenge. Signatures are employed extensively in most modern detection methods. In order to function properly, a signature-based detection system requires constant human monitoring and signature updates. Due to its reliance on the signature database, the signature-based approach is inadequate in the face of evolving malware and novel cyber threats [5,6]. Artificial intelligence research, especially in the field of deep learning, is intended to address the shortcomings of current approaches to cyber security.

As mentioned in [7,26], feature reduction (RF) is a method for choosing the best subset of features to use in describing patterns that belong to various classes. Features were reduced using principal component analysis (PCA) and singular value decomposition (SVD). major findings of the research are summarized concisely as follows:

This research assesses the cutting edge of network IDS by merging IDS methodologies with deep learning methods.

- Predictability in intrusion detection systems that use deep learning should improve. Networks of Neural Convolutions
- CNN may be used to perform a variety of operations on a dataset, including the extraction of features, the prediction of future intrusions, and the acquisition of more accurate detection findings.
- Reduce the number of features in the proposed model using principal component analysis and singular value decomposition to make it more effective.
- Applying the UNSW-15 datasets, assess and evaluate the proposed NIDCNN.

2. Literature Review

To improve upon the multilayer perceptron's design, deep neural networks include more hidden layers in between the input and output layers. Deep neural networks are able to learn very complicated and extensive data representations because to advancements in recent optimization and regularization approaches (Kingma and Ba, 2014). The Convolutional Neural Network and the Recurrent Neural Network are two examples of supervised learning models that have emerged from the study of Deep Neural Networks. We may classify the Boltzmann machine, Self-Organizing maps, and Autoencoders as unsupervised learning architectures. The generative models are a significant subset of deep neural networks (Goodfellow, 2016).

The goal of a generative model is to take an original training sample and use it to generate synthetic data that superficially matches the original sample but is wholly fake while maintaining the same underlying structure (Goodfellow, 2016). In terms of architecture and design requirements, these generative models are quite close to the supervised and unsupervised models we've already covered. The generating abilities of most generative models may be easily adjusted by modifying the training cost function and the training process. Non-standard training approaches are used by deep generative models like Generative Adversarial Networks (GANs) to directly estimate a density function over a data distribution. A GAN consists of two adversarial networks, the generator and the discriminator, which are taught to engage in a minimax game against one another. According to Goodfellow et al. (2014a, b), it is the responsibility of the generator to come up with plausible fakes that will fool the discriminator, and it is the responsibility of the discriminator to determine which images are real and which are fake within the given

sample. It is an innovative and straightforward methodology to estimate the density function using the minimax method on the original sample photographs (Goodfellow, 2016). State-of-the-art realistic generation challenges that GANs have been used to include video frame prediction (Lotter et al. 2016), picture resolution enhancement (Ledig et al. 2016), generative image alteration (Zhu et al. 2016), and image to text translation (Isola et al. 2017). When it comes to creating high-quality, lifelike images, GANs have proven to be the gold standard for all of these and many more uses. In this research, we take a look at a novel use of GANs in the field of cyber security.

Cybersecurity is an umbrella word for the research and development of safeguards for internet-connected computing infrastructure. In this summary, we'll look at research on employing GANs to either defend against or launch assaults against an adversarial system. The word "Cybersecurity" is used to cover a wide range of topics since, in the modern world, almost every digital system is accessible online and must include some kind of cybersecurity.

In the context of cybersecurity and adversarial assaults on system security, this section provides the literature on generative adversarial networks. The study is split into two parts, with the first covering research on the use of GANs to mitigate adversarial attacks and improve the resilience of security systems to such assaults (Sect. 3.1). Section 3.2 then reviews research on the potential of GANs as adversarial threats to security systems by looking at cases where they were utilized to design adversarial systems.

One of the primary study topics for the use of GANs ever since its inception in 2014 has been the use of the technology for adversarial studies. This is in response to research showing that neural networks and other machine learning models struggle when the feature space in which learning takes place is altered (Malhotra, 2018). Adversarial systems may use the difficulty of generalization to fool machine learning algorithms into making incorrect conclusions (Goodfellow et al., 2014a, b). Knowing what kinds of assaults to anticipate is the first step in strengthening a system's defenses against them. Poisoning attacks are a kind of adversarial learning used to enhance the probability of misclassification in supervised classification tasks by inserting intentionally misleading samples into training data. On the other side, evasion attempts include tampering with and distorting input data in an effort to fool a trained classifier (Szegedy et al., 2013; Biggio and Roli, 2018). Training machine learning models to be ready for such assaults is one approach that academics have sought to enhance performance against poisoning and evasion adversarial attacks (Apruzzese et al. 2018; Lin et al. 2018). To do this, potential toxic samples are included in the training dataset (Anderson et al., 2016).

Generative adversarial networks (Yin et al., 2018) are used to create these adversarial instances from the original training examples. Given that the machine learning model has learnt certain instances of probable adversarial assaults inside a specific domain, the consequence is an improvement in resilience against adversarial attacks (Grosse et al., 2017). Using GAN-generated adversarial examples in training sets has been shown to significantly improve malware detection (Kim et al., 2018) and steganography for securely encoding data (Shi et al., 2017). To this point, we have shown works where the primary focus of the study was on creating adversarial samples for use as training data in another machine learning model, such as a classifier. The generator network, after learning an adversarial embedding space, has been successfully used to a variety of security tasks, including stenography and cryptography (Abadi and Andersen, 2016). For example, in (Tang et al. 2017), the authors simulate the competition between steganography using additive distortion and steganalysis using deep learning to construct a GAN-based stenographic method. Using a learned distortion function, the generator in the Automatic Stenographic Distortion Learning (ASDL-GAN) system creates steganographic pictures, and the steganalytic discriminator determines whether or not the generated images match the cover. A possible adversarial disruption to normally normal picture data might be provided via the steganographic deformation embedding space, as in research such as (Zügner et al., 2018). As an alternative, the authors of (Tang et al., 2017) teach the generator to exploit this distortion space for steganographic purposes. Finally, we wrap off this part by briefly summarizing data privacy research that makes use of comparable methodologies to those we just reviewed. Integrating noise where it counts is key to ensuring data privacy and usefulness. To let the GAN learn privatization mechanisms from the dataset without needing access to the dataset statistics, Huang et al. (2018) introduce a method they call Generative Adversarial Privacy (GAP), a novel context-aware model of privacy.

Wu, Ping, and Guo, Hui [8] A deep neural network was suggested as a means of detecting network breaches on a large scale. LuNet uses convolutional neural networks (CNN) to extract spatial information and recurrent neural networks (LSTM) to extract temporal features from traffic data. In order to minimize data loss caused by training inconsistencies between CNN and RNN, the two networks are coordinated to get inputs with same

granularity. The design also includes batch normalization to speed up the learning process. The UNSW-NB15 datasets are rated at 94.98% correctness.

Azizjon, M., et al. [9] An effective IDS was created using the DL (1D-CNN) component. Testing on the UNSW NB15 IDS dataset, the performance of a one-dimensional convolutional neural network (CNN) employing supervised approaches showed an 88.93% accuracy rate in classifying and labeling (normal and attack) data.

These methods, developed by L. Ashiku et al. [10], use deep learning to build a network-based intrusion detection system (IDS). The usefulness of the model was proved using the UNSW-NB15 dataset, and the emphasis is on how deep learning, or (DNNs), may give flexible IDS by being taught to discern distinguished from unique or zero-day network behavioral patterns. A performance accuracy of 95.6% was found during experimental testing.

A foundation for a 1D-CNN was built by M.Hooshmand and D. Hosah alli [11]. The suggested method separates out the (TCP), (UDP), and OTHER protocols from the rest of the net flow data before processing it. Before constructing a model, the attributes are selected using a Chi-squared test. as a means of oversampling, the synthetic minority oversampling technique is used.The authors' method has a 76.3% success rate. Using the UNSWNB15 dataset, we verify the accuracy of the model.

3. Background

3.1.Deep Learning CNN Techniques

Our research is on developing many supervised deep-learning classifiers for use in a network intrusion detection system. In this study, we'll look at how well PCA and SVD-based intrusion detection classifiers perform [16].The weights in a Convolutional Neural Network (CNN) are shared locally, so they are applied uniformly over the whole input. A filter is formed by many weights all connected to the same output. To help deep architectures learn non-linear representations of the input data, CNN layers include a point-wise non-linearity, like the logistic function, and a pooling operator, which aggregates the statistics of the features at close locations to reduce the spatial size of the picture. An output layer with all connections follows the final convolutional layer [17].

3.2. Reduction of Data Features

To reduce the size of data that is a linear combination of the important dimensions, linear methods are applied. Nonlinear methods are geared on reducing data dimensionality by non-linear combination of the output. (LFE) is the process of developing the new feature space's variables using linear extraction from the input feature space [18,19]. Although several studies have been conducted, PCA and SVD remain the most well-liked linear feature extraction methods.In order to minimize the amount of error introduced into the reconstruction process while maintaining a high level of feature reduction, PCA is used in this work [20].

3.2.1. Principal Component Analysis (PCA)

Dimensionality reduction describes such a technique. Reducing the number of options and settings available to a computer makes it work more efficiently. Selection and extraction merit are the processes in feature reduction.Different feature reduction strategies[21] exist. PCA Multivariate signals, spectra from many sources, physicochemical data, hyperspectral pictures, and other high-dimensional datasets with hidden information need this technique as the primary means of analysis. PCA is included in the vast majority of analysis methodologies software packages and is often employed automatically as a recommended practice. Steps of principal component analysis [16,22]:

1-XrepresentaPCAinput matrix with just an n-vector and an m-dimensional data collection.

2-Using Eq.1 determine the average data (X) for every dimension:

$$\overline{X} = \frac{1}{n}\sum_{i=1}^{n} X_i \qquad \dots 1$$

n samples number,

X_i item values i.

3- Use the given Eq. 2to determine the covariance matrix (C x):

$$C_x = 1/n - 1 \sum_{i=1}^{n}(X_i - \overline{X})\,(X_i - \overline{X})^{T} \qquad \dots 2$$

4-Using Eq. 3, determine the eigen-values , eigen-vectors (v m) of the covariance matrix:

eigen-values indicate m

eigen-vectors indicate v m

$$C_x v_m = \lambda_m v_m \qquad \dots 3$$

5- The eigenvalues put in descending order..

6- A set of eigenvectors known as a principal component (PC) corresponds to the arranged eigenvalues from step 5.

3.2.2. Singular Value Decomposition (SVD)

It is an extension of the eigen decomposition of rectangular matrices that is useful for scientific investigation. The primary goal of singular value decomposition (SVD) is to decompose a rectangular matrix into two simple matrices (a two-dimensional matrix and a diagonal matrix) [23]. Like principal component analysis (PCA), singular value decomposition (SVD) generates a lower rank matrix of the same dimensions that provides a least square estimate of a data set. The goal of the SVD technique, as shown in Eq.4, the objective of the SVD approach is to diagonalizable the data matrix ((X∈ $R^{p\times q}$)) into three matrices.

$$X = LSR^T = [l_1 \dots \quad l_p] \begin{bmatrix} s_1 & 0 & 0 & 0 \\ 0 & s_2 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & s_q \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} -r_1^T \\ -r_2^T \\ \vdots \\ -r_q^T \end{bmatrix} \qquad \dots 4$$

L(p p) represents the left singular vectors, R(q q) represents the right singular vectors, and S(p q) is a diagonal matrix that lists the singular values from largest to smallest, with the largest single value being in the upper-left index of S. It is easy to leave out the less important parts of such an explanation in order to build an approximate description with the right number of dimensions [25-27]. The left and right singular matrices, abbreviated L and R, are the orthonormal bases [24]. For each given X, there exists an m-n matrix. The best k single values are calculated using the following idea.

1. U has m x k stacks and is an orthonormal stake matrix. Each stake in this matrix is a unit vector, and any two stakes have the same dot product.

2. V has n k stakes and is an orthonormal stake model. The modified form of the orthonormal rows of VT is represented by V. The stakes are listed in descending order of importance.

The diagonal matrix

3.S has k× k entries. Nothing exists that is not a part of the primary diagonal. Singular values of X are represented by S elements.

4.A large matrix X can be split into three large matrices by using the SVD components U, S, and V

$$X_{m\times n} = U_m \times kSk \times k(V_n \times k)T$$

…5

A k-low dimensionality is retrieved from one of the input matrices, X, via the SVD notion, which is demonstrated in Eq. (6). U, S, and VT are shortened forms of U, S, and VT, respectively. Only the top k individual values are kept in Y in this case

$$Y = U_k \times S_k \times V_k^T \qquad \dots 6$$

4. NID CNN Classification Paradigm

Here, we'll describe the framework of our proposal, The goals of the thesis can't be achieved without the procedures of data preprocessing, data splitting, feature reduction, classification, and assessment that make up the core of the suggested system design. This part provides a high-level summary, while the following subsections go into further depth on each stage. Standardization is the first stage of data preparation. Second, principal component analysis and singular value decomposition are employed to reduce the number of features. These methods help choose the most important characteristics while reducing the number of features needed for the classification stage. In the third phase, the Network Intrusion Detection Convolutional Neural Networks (NIDCNN) model is used to determine whether the data flow inside the network is typical or not. The results of the proposed method were finally evaluated using many scales.

Phase 1: Load UNSW-NB15 dataset

In order to provide a mix of real-world, modern-day operations and synthetic, current attack characteristics, the UNSW Canberra Cyber Range Lab's IXIA Perfect Storm application was utilized to generate the incoming network packets for the UNSW-NB 15 dataset. collected traffic (in the form of Pcap files) totaled 100 GB after being collected using the tcpdump program. There are nine distinct kinds of attacks included in this data collection, including fuzzers, analyses, backdoors, denial-of-service attacks, exploits, generic assaults, reconnaissance attacks, shellcode, and worms. Twelve algorithms are used by Argus and Bro-IDS to generate the class label and the forty-nine characteristics. UNSW NB15 Characteristics A CSV file explains what each feature is.

The four CSV files (1,2,3,4.csv UNSW-NB15) include a combined total of 540,044 entries (the regression coefficients table file and the GT.csv event list file). separating these records for the UNSW NB15. csv into a

testing set and a training set. Thus, csv was divided into a training set and an evaluation set. The first group has 175,341 records, while the second group contains 82,332 files from the assault and regular categories.

Phase 2: Segmentation of the UNSW-NB15 Dataset

To guarantee accurate generalization and prevent overtraining, hold-out-validation was used. A 70% training set and a 30% test set were split out from the original UNSW-NB15 dataset. There is a detailed explanation of this technique in:-

Algorithm 1

1. Segmentation the data set
2. UNSW-NB15 dataset as inputs
3. Show the values by splitting the dataset into practice and examination sets (70/30).
4. begin by outlining the evaluation of sets of model parameter values.
5. any group coefficient resulting from all iterations of repetition and sampling
6. Conclusion: After excluding a specific sample, use the model on the remaining information.
7. Learn about the recalcitrant samples.
8. pause for
9. It is important to assess how hold-out estimates typically perform.
10. pause for
11. Choose the best possible combination of parameters.
12. Using the idealistic parameter group, fit the final model to the entire set of workout data.
13. End

Phase 3:Preprocessing UNSW-NB15 Dataset

In the second phase of the proposed system, called "preprocessing," an effort is made to normalize the raw dataset. The primary goal of this laborious procedure is to provide a reliable and suitable dataset for deep learning algorithms. To complete this level, the tried-and-true scaler tactic is used. It was used as a pretreatment of the data after the data was divided in the previous step, as explained in procedure (2). There are two instances when this process (training and testing) is involved.

Algorithm 2

1. Standardization of Data
2. Data input: dividing dataset
3. Data output: standardized
4. begin
5. Calculate $\text{Mean}(\bar{X}) = \frac{\sum_{i=1}^{n} X_i}{n}$ then set the result as μ.
6. Calculate $\text{Variance} = \frac{\sum_{i=1}^{n}(X_i - \bar{X})^2}{n-1}$
7. Calculate standard deviation $(\text{SD}) = \sqrt{\frac{\sum_{i=1}^{n}(X_i - \bar{X})^2}{n-1}}$, then make the outcome as σ.
8. Calculate the stander scaler $Z_{Scaled} = \frac{(X-\mu)}{\sigma}$.
9. End

Stage 4: Using algorithm (3) PCA and algorithm (4) SVD, reduce features.

Algorithm 3

1. Data input: standardized (SD)
2. Reduced characteristics as output (RF)
3. start
4. Establish a data matrices with each of the values for the parameters in the columns and then each row represents a distinct item in the row.
5. Compute the covariance matrix from the data matrix using Eq. $C_x = 1/n-1 \sum_{i=1}^{n}(X_i - \overline{X})(X_i - \overline{X})^T$.
6. Determined the covariance matrix's eigenvalues and eigenvectors by Eqs.
7. $C_x = 1/n-1 \sum_{i=1}^{n}(X_i - \overline{X})(X_i - \overline{X})^T$ and $C_x v_m = \lambda_m v_m$
8. To decrease the data's dimension, use the eigenvalues.
9. Return (decreased Features).
10. Finish.

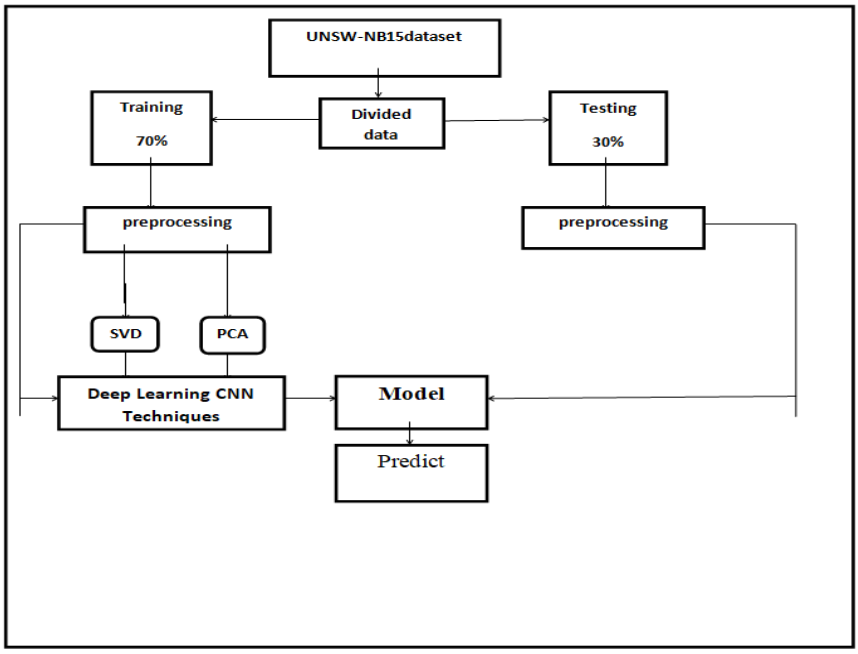
Algorithm 4

Input: Standardized data

Output: Reduced features

1. begin
2. find the value of VV^T and V^TV.
3. find the value of X matrix $X_{m\times n} = U_m \times kSk \times k(V_n \times k)T$
4. find the value of Y matrix $Y = U_k \times S_k \times V_k^T$.
5. Diagonalizable the data matrix.
6. Return Reduced Features
7. End.

Stage 5:Construct classification NIDCNN model with twenty seven layers In figure (1) and algorithm (5) illustrated the proposed classification model



Algorithm 5

1. The suggested classification model for NIDCNN
2. UNSW-NB15 dataset as input
3. output: The NIDCNN classification model's accuracy
4. Begin
5. Load UNSW-NB15 dataset.
6. Divided dataset, calling the Algorithm 1.
7. Standardized dataset, calling the Algorithm 2.
8. Reduce features firstly PCA, calling the Algorithm 3, then SVD, calling the Algorithm 4.
9. classification of a structure Modeling NIDCNN with the twenty seven subsequent layers:
10. Nine-layer Convolutional Neural Network (CNN).
11. Maximum Pooling: six layers.
12. Eight layers of leaky ReLU.
13. Make one layer flat.
14. Three Dense layers
15. Return accuracy
16. End.

5. Experimental Results and Discussions

We evaluate PCA and SVD-based deep learning CNN algorithms on the new gold standard for intrusion detection, the UNSW-NB15dataset. These algorithms were tested on a computer with a 1.80 GHz or 1.99 GHz Intel(R) Core(TM) i7-8565U CPU, 10 GB of RAM, a 64-bit operating system, and Python 3.6 for their programming language.

5.1 Performance Evaluation

We employed a number of evaluation techniques to test the suggested NIDCNN strategy., A classifier's accuracy, F-Score, precision, and sensitivity (Recall) are measured using these metrics[26,27].

$$Accuracy = (TP + TN)/(TP + TN + FP + FN) \quad (7)$$

$$F_1 = 2 * \frac{precision*recall}{precision+recall} \quad (8)$$

$$Precision = \frac{TP}{TP+FP} \quad (9)$$

$$Recall = \frac{TP}{TP+FN} \quad (10)$$

Where, TP and TN true positive and true negative, respectively, while FP and FN false positive and false negative acronyms, respectively

5.2 Result and discuss

On the renowned UNSW-15 dataset, the suggested NIDCNN was assessed. Two algorithms were compared to the suggested NIDCNN's evaluation: PCA and SVD,aid in reducing the number of features needed for the classification step and helping to choose the most crucial ones. so when using proposed model with 42 features without feature reduction techniques, results of the measurements were an accuracy of 100 %, a precision of 100 %, a sensitivity 31%, and an F-score of 48% in time 0.532 sec., table (1) displays these results and the chart explained figure(2).

Table (1). classification pure model

Method%	**Pure model**
Accuracy	100
Precision	100
Recall	30
F-score	47
Time in sec	0.53

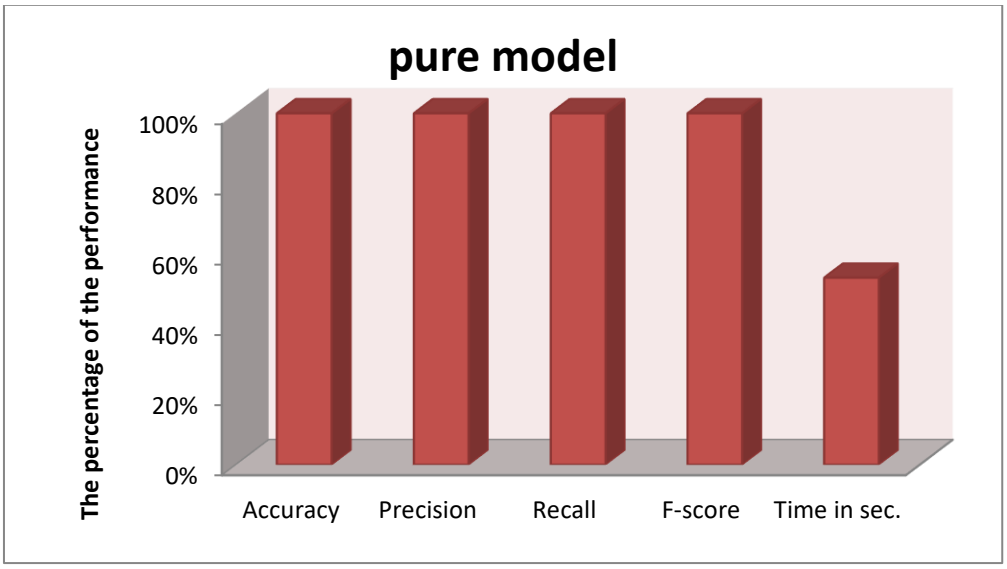


Fig. (2) Pure Model

Using the PCA Features Reduction Technique are shown as two methods, the first is when applying PCA-10, table (2) get the evaluation results, and figure (3) explains the PCA Technique.

Table (2). PCA-10 features reduction.

Method%	**PCA-10**
Accuracy	**100**
Precision	**100**
Recall	**62**
F-score	**77**
Time in sec	**0.102**

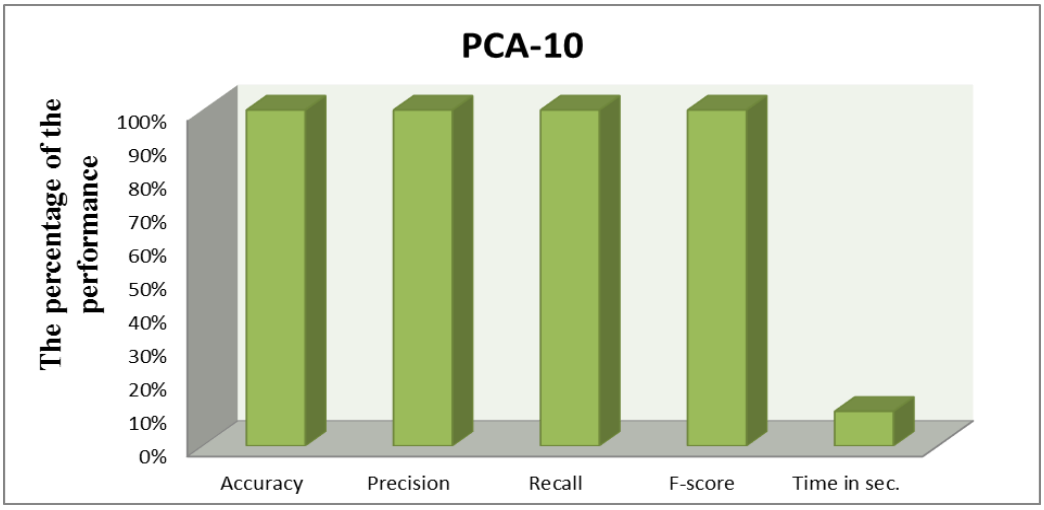


Fig.(3) PCA-10 feature reduction the second is when applying PCA-15, table (3) get the evaluation results, and figure (4) explains the PCA Technique.

Table (3). PCA-15 features reduction.

Method%	PCA-15
Accuracy	100
Precision	100
Recall	63
F-score	77
Time in sec	0.26

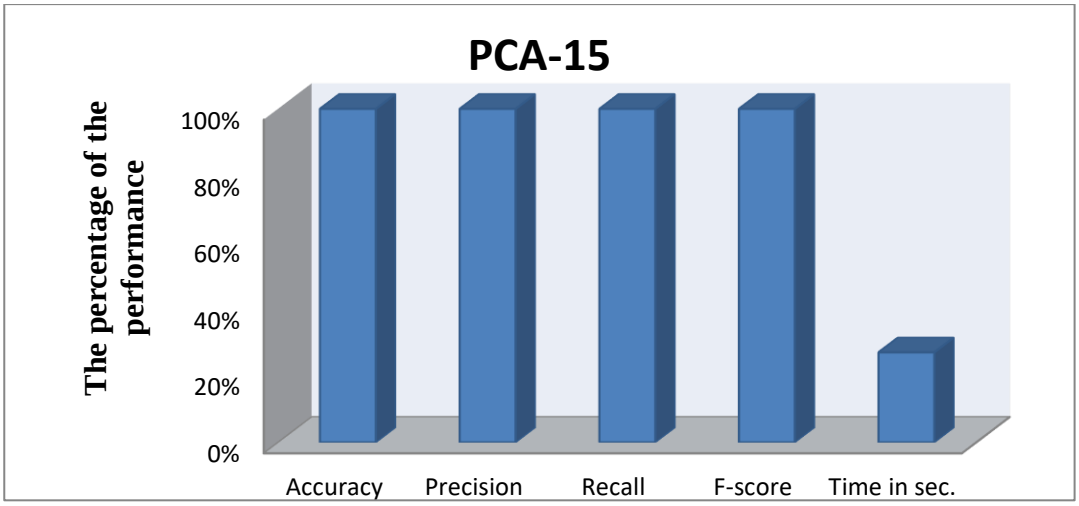


Fig.(4) PCA-15 feature reduction to reduce features using SVD in two ways, the first SVD-10, Table (4) shows figure (5).

Table (4). SVD-10 features reduction.

Method%	SVD-10
Accuracy	100
Precision	100
Recall	63
F-score	77
Time in sec	0.102

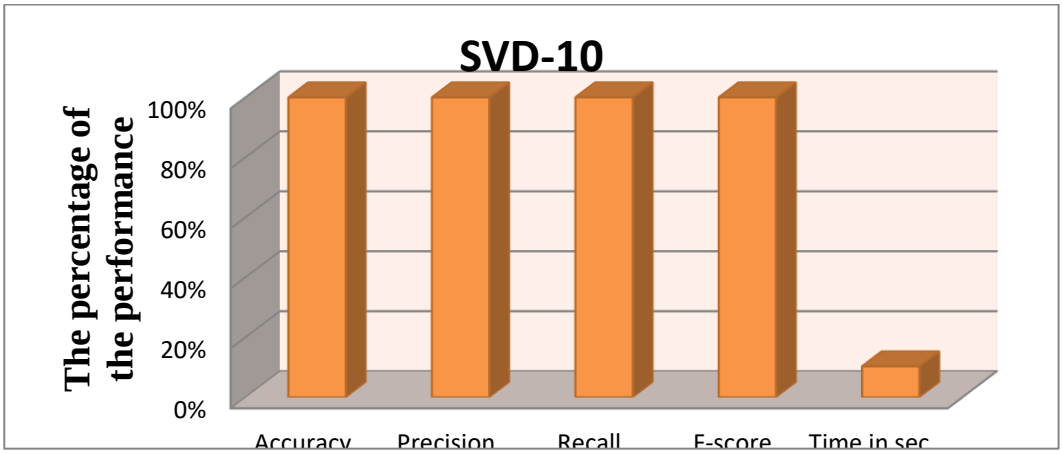


Fig.(5) SVD-10 *feature reduction* second using SVD-15, Table (5) shows figure (6).

Table (5). SVD-15 features reduction.

Method%	SVD-15
Accuracy	**100**
Precision	**100**
Recall	**63**
F-score	**77**
Time in sec	**0.26**

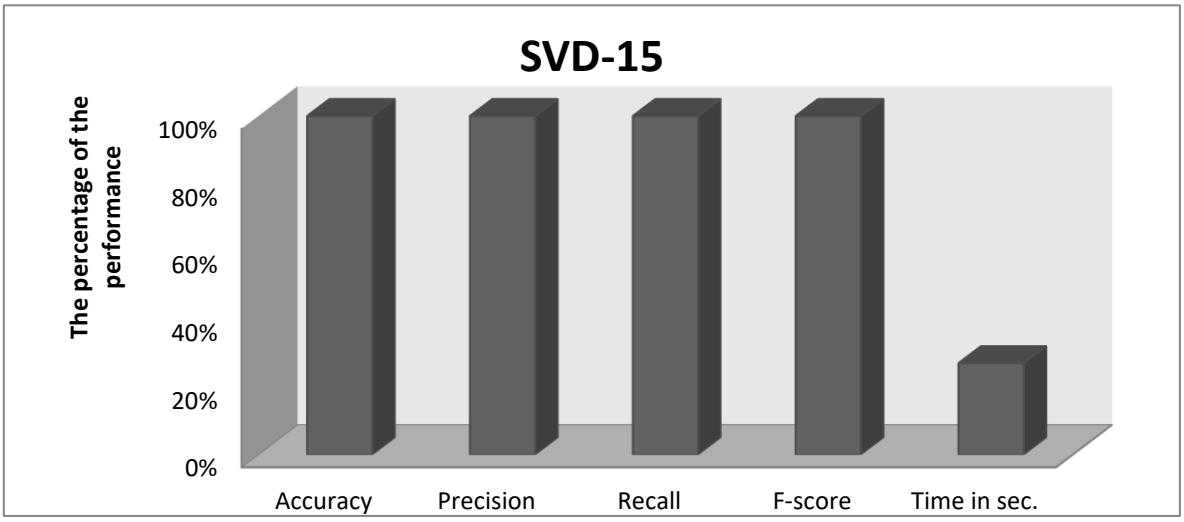


Fig. (6) SVD-15 feature reduction

Table (1) to (5) shows that employing the suggested 1D-CNN with both PCA-15 and SVD-15-based approaches in time 0.26, we obtained 100% testing accuracy, 100% precision, 63% recall, and a 77% F-score. The results are remarkably equivalent when utilizing 1D-CNN with both PCA-10 and SVD-10-based techniques in time of 0.102 to achieve testing accuracy, precision, recall, and F-score of 100%,100%,63%, in time 0.102 and77% respectively. In pure model 100% accuracy, 100% precision, 31% recall, and 48% F-score, in time 0.532,. Table (6) shows what was explained in the previous tables:

Table (6).suggested model without and with the feature reduction

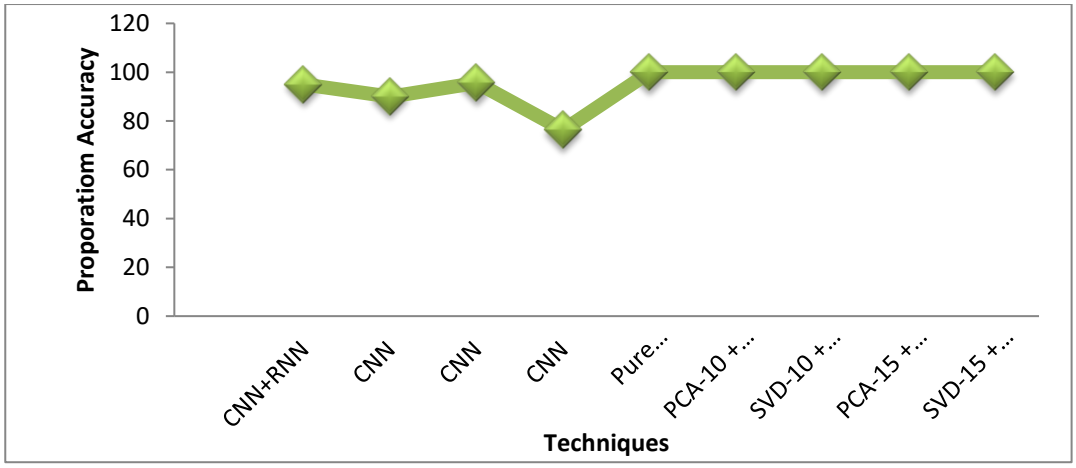
Method %	Deep learning CNN Pure	CNN with PCA 10	CNN with PCA 15	CNN with SVD 10	CNN with SVD 15
Accuracy	**100**	**100**	**100**	**100**	**100**
Precision	**100**	**100**	**100**	**100**	**100**
Recall	**30**	**63**	**63**	**63**	**63**
F-score	**47**	**77**	**77**	**77**	**77**
Time in sec	**0.53**	**0.102**	**0.26**	**0.102**	**0.26**

5.3 Comparative Findings from Relevant Studies

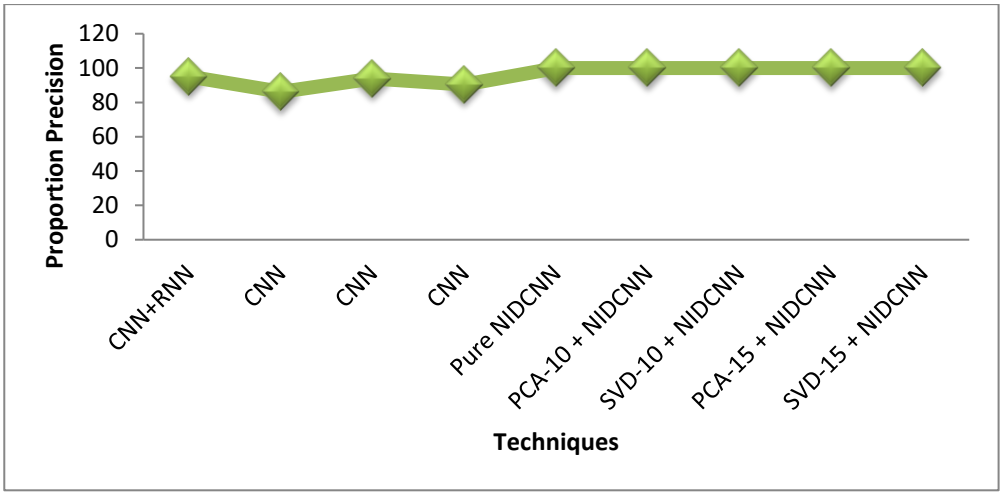
A thorough comparison of the architecture used with the UNSW-NB15 dataset can be seen in Table (7) figures(7,8,9, and 10) show the type of technique and performance evaluation to the same data.

Table (7).

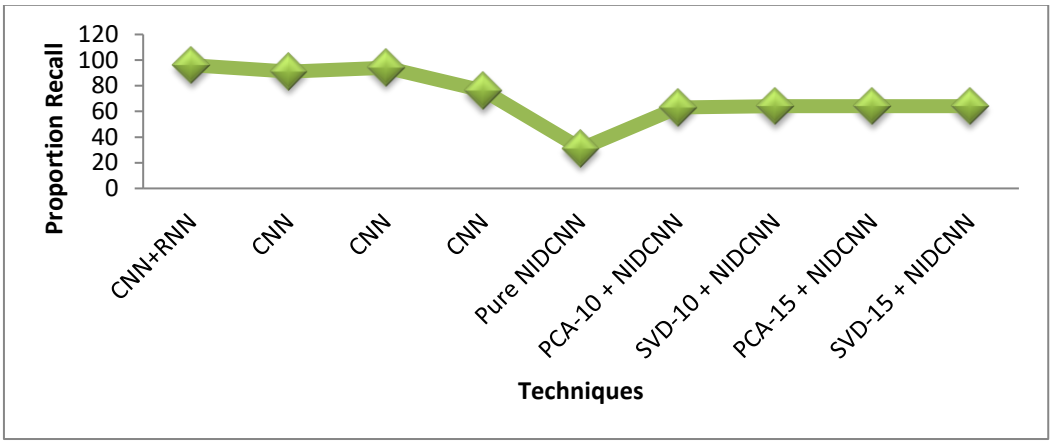
Ref. No.	Technique	Accuracy%	Precision %	Recall %	F-score %
[8]	CNN+RNN	94.98	95	96	98.5
[9]	CNN	89.93	86.15	91.15	90.43
[10]	CNN	95.6	94	94	94
[11]	CNN	76.3	90.4	76.1	78.2
Our Methods	**Pure NIDCNN**	**100**	**100**	**31**	**48**
	PCA-10 + NIDCNN	**100**	**100**	**63**	**78**
	SVD-10 + NIDCNN	**100**	**100**	**64**	**78**
	PCA-15 + NIDCNN	**100**	**100**	**64**	**78**
	SVD-15 + NIDCNN	**100**	**100**	**64**	**78**



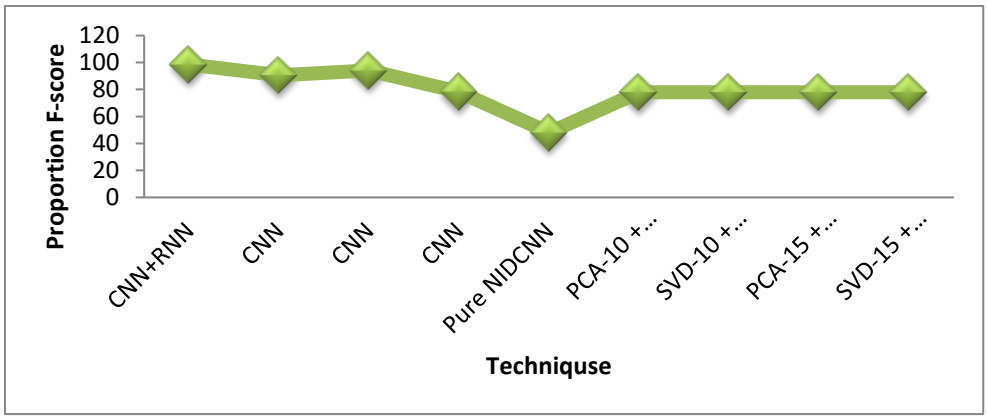
Fig(7). Relation between Accuracy and Techniques



Fig(8). Relation between Precision and Techniques



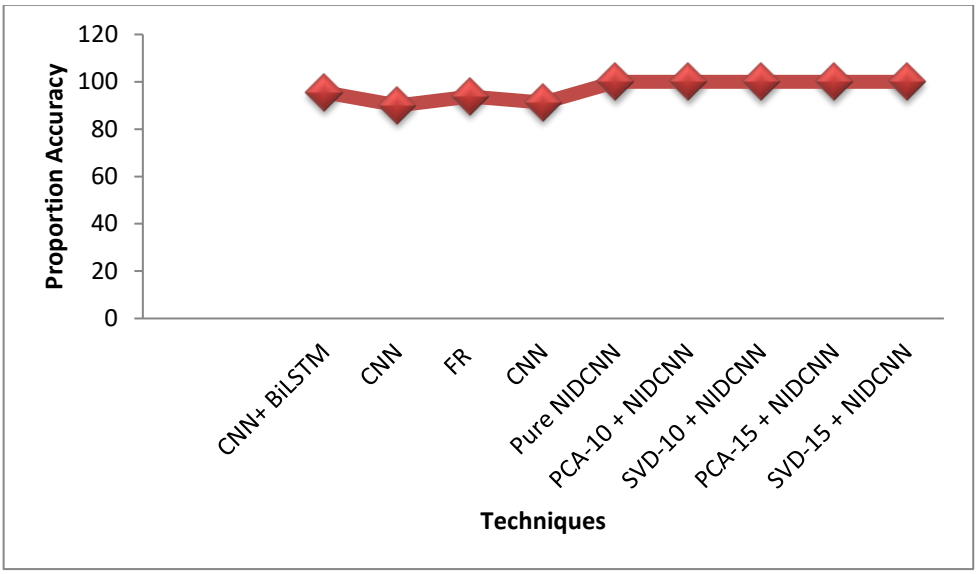
Fig(9). Relation between Recall and Techniques



Fig(10). Relation between F-Score and Techniques using other data sets can see in Table (8), Figures (11,12,13 and 14) show the type of technique and performance evaluation for it.

Table (8).

Ref. No.	Technique	Dataset %	Accuracy %	Precision %	Recall %	F-score%
[12]	CNN+ BiLSTM	KDD CUP 99	95.50	95.7	95.2	95.5
[13]	CNN	NSL-KDD	90.01	91	90	90
[14]	FR	KDD CUP	93.7	96	97	98
[15]	CNN	KDD99	91.23	93	95	96
Our Methods	**Pure NIDCNN**	**"UNSW-NB15"**	**100**	**100**	**31**	**48**
	PCA-10 + NIDCNN	**"UNSW-NB15"**	**100**	**100**	**63**	**78**
	SVD-10 + NIDCNN	**"UNSW-NB15"**	**100**	**100**	**64**	**78**
	PCA-15 + NIDCNN	**"UNSW-NB15"**	**100**	**100**	**64**	**78**
	SVD-15 + NIDCNN	**"UNSW-NB15"**	**100**	**100**	**64**	**78**



Fig(11). Relation between Accuracy and Techniques

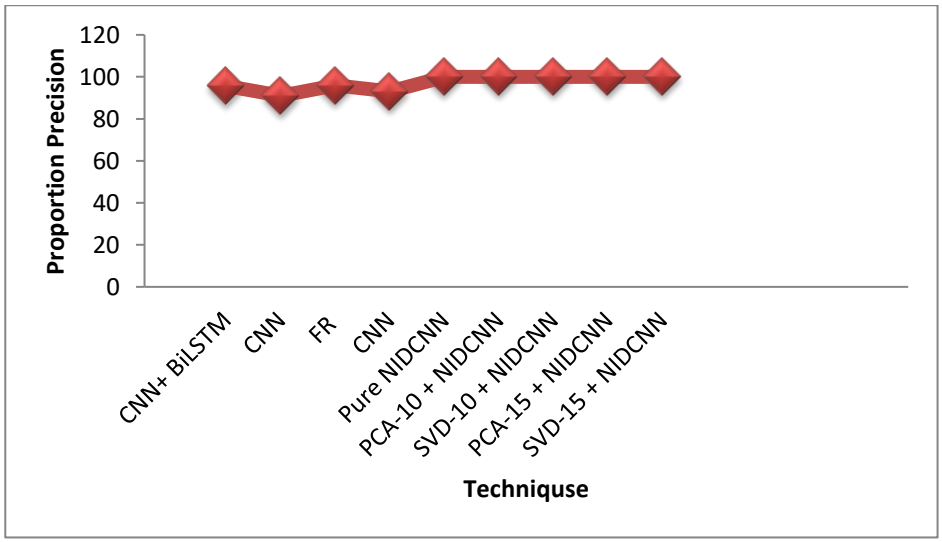
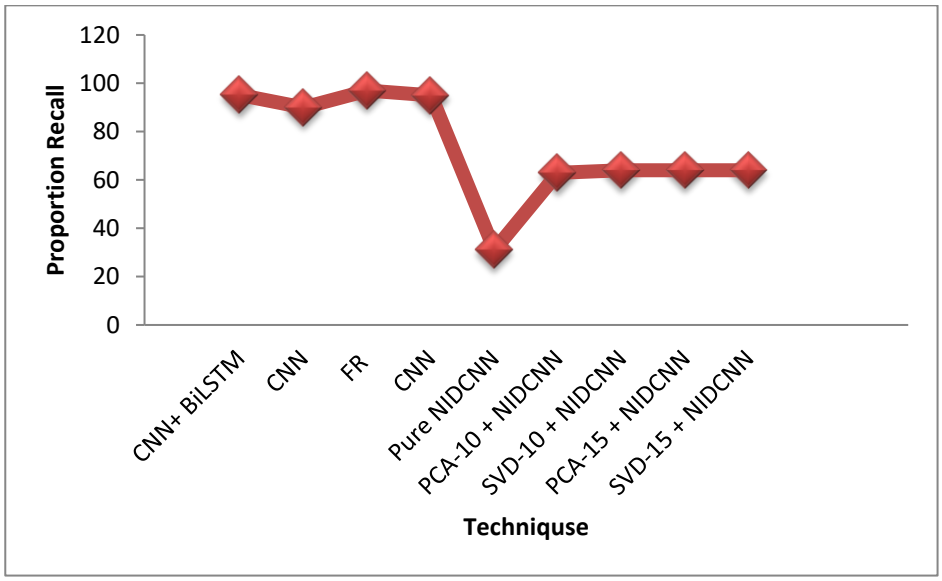
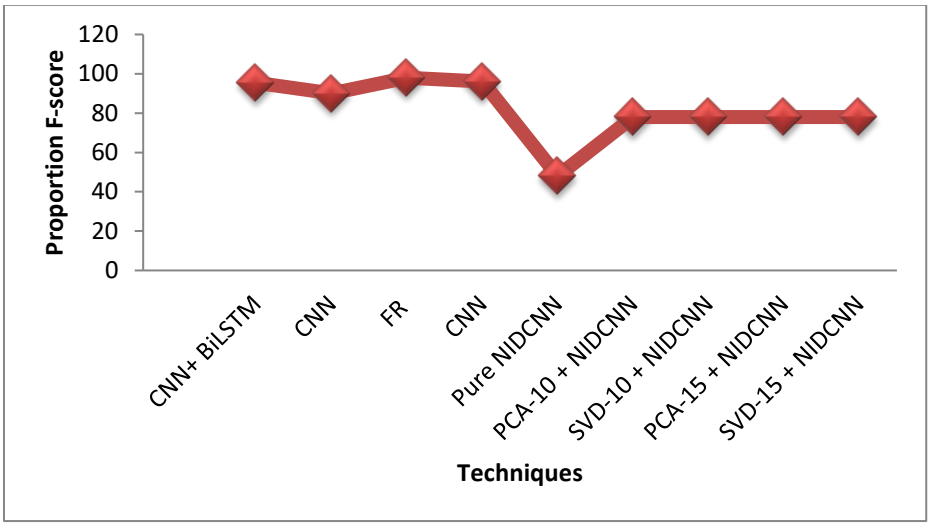


Fig (12). Relation between Precision and Techniques



Fig(13). Relation between Recall and Techniques



Fig(14). Relation between F-score and Techniques

6. Conclusions

proposition a 1D-CNN-based proposed NIDCNN model for identifying both normal and abnormal network packets. In order to execute multi-class classification detection, we applied the most recent DL algorithms and 1D-CNN. The model is simple and uses little processing power. achieved a total accuracy of 100% with this approach. Additionally, we used the PCA and SVD techniques to further analyze the smaller sets of input features, which significantly improved detection accuracy and decreased detection time. It is clear from the findings that the deep learning CNN with PCA and SVD method performs better than other classifiers given the parameters and data set under consideration. It is entirely accurate. The work can be expanded to include the crucial aspects of intrusion detection. The two technologies used provided an additional benefit. In addition to accuracy, we note that the time required to complete the work has become less, so there will be accuracy and speed. It is possible to use this technology to work in real-time.

References

[1] John Goodall et al.," The Work of Intrusion Detection: Rethinking the Role of Security Analysts" Proceedings of the Tenth Americas Conference on Information Systems, New York, New York, August 2004.

[2] P. Garcı´a-Teodoro et al.," Anomaly-based network intrusion detection: Techniques, systems and challenges" computers & security 28 (2009) 18–28.

[3] Salima Omar et al.," Machine Learning Techniques for Anomaly Detection: An Overview" International Journal of Computer Applications (0975 – 8887), Volume 79 – No.2, October 2013.

[4] M. Ferrag et al., "Deep Learning for Cyber Security Intrusion Detection: Approaches, Datasets, and Comparative Study," Journal of Information Security and Applications, vol. 50, 2020.

[5] Mohammad Sazzadul Hoque et al.," AN IMPLEMENTATION OF INTRUSION DETECTION SYSTEM USING GENETIC ALGORITHM" International Journal of Network Security & Its Applications (IJNSA), Vol.4, No.2, March 2012.

[6] P. Garcı´a-Teodoro et al.," Anomaly-based network intrusion detection: Techniques, systems and challenges" computers & security 28 (2009) 18–28.

[7] JAN J. GERBRANDS "ON THE RELATIONSHIPS BETWEEN SVD, KLT AND PCA" Department of Electrical Engineering, Delft University of Technology, Pattern Recoflnifion Vol. 14, Nos. I 6, pp. 375 381, 1981. Printed in Great Britain.

[8] P. Wu and H. Guo, "LuNet: A Deep Neural Network for Network Intrusion Detection," arXiv:1909.10031v2 [cs.AI] 6 Oct 2019.

[9] M. Azizjon et al., "1D CNN Based Network Intrusion Detection with Normalization on Imbalanced Data," International Conference on Artificial Intelligence in Information and Communication (ICAIIC), Fukuoka, Japan, 19-21 February 2020.

[10] L. Ashiku and C. Dagli, "Network Intrusion Detection System using Deep Learning," Complex Adaptive Systems Conference Theme: Big Data, IoT, and AI for a Smarter Future Malvern, Pennsylvania, vol. 185, pp. 239–247, June 16-18, 2021.

[11] M. Hooshmand and D. Hosahalli, "Network Anomaly Detection Using Deep Learning Techniques," CAAI Transactions on Intelligence Technology, vol. 7, pp. 228–243, 2022.

[12] J. Gao, "Network Intrusion Detection Method Combining CNN and Bilstm in Cloud Computing Environment," Hindawi, Computational Intelligence and Neuroscience, 2022.

[13] X. Yang et al., "An Enhanced Intrusion Detection System for IoT Networks Based on Deep Learning and Knowledge Graph," Hindawi, Security and Communication Networks, 2022.

[14] Selvakumar K, Sairamesh L, Kannan A. Wise intrusion detection system using fuzzy rough setbased feature extraction and classification algorithms. International Journal of Operational Research. 2019;35(1):87-107.

[15] R. Khan et al., "An Improved Convolutional Neural Network Model for Intrusion Detection in Networks," Cybersecurity and Cyber forensics Conference (CCC), 2019.

[16] A. Oprea, "The Use of Principal Component Analysis (PCA) in Building Yield Curve Scenarios and Identifying Relative-Value Trading Opportunities on the Romanian Government Bond Market," Journal of Risk and Financial Management, vol. 15, no. 247, 2022.

[17] F. Sultana et al., "Advancements in Image Classification Using Convolutional Neural Network," in Proceedings of the 2018 4th International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN), pp. 122–129, IEEE, Kolkata, India, November 2018.

[18] Iqbal Muhammad et al.," SUPERVISED MACHINE LEARNING APPROACHES: A SURVEY" , ICTACT JOURNAL ON SOFT COMPUTING, APRIL 2015, VOLUME: 05, ISSUE: 03.

[19] Sumit Das et al.," Applications of Artificial Intelligence in Machine Learning: Review and Prospect", International Journal of Computer Applications (0975 – 8887)Volume 115 – No. 9, April 2015.

[20] Shiliang Sun, Changshui Zhang and Guoqiang Yu, "A Bayesian Network Approach to Traffic Flow Forecasting",IEEE Transactions on Intelligent Transportation Systems, Vol. 7, No. 1, pp. 124-132, 2006.

[21] Hamid Parvin et al., "A Modification on K-Nearest Neighbor Classifier", Global Journal of Computer Science and Technology ,Vol. 10 Issue 14 (Ver. 1.0) November 2010 P a g e | 38.

[22] Aleksandar et al.," INTRUSION DETECTION: A SURVEY", Computer Science Department, University of Minnesota.

[23] S. Brunton and J. Kutz, "Singular Value Decomposition (SVD)," Data-Driven Science and Engineering (pp.3-46), 2019.

[24] K Rajasekaran and Dr. K Nirmala," Classification and Importance of Intrusion Detection System", (IJCSIS) International Journal of Computer Science and Information Security, Vol. 10, No. 8, August 2012.

[25] Qiang Guo et al." An Efficient SVD-Based Method for Image Denoising", IEEE transactions on circuits and systems for video technology, vol. 26, no. 5, may 2016.

[26] Sadiq Hussain et al.," Prediction Model on Student Performance based on Internal Assessment using Deep Learning", Paper—Prediction Model on Student Performance based on Internal Assessment using Deep Learning, iJET – Vol. 14, No. 8, 2019.

[27] Mathieu Lepot et al.," Interpolation in Time Series: An Introductive Overview of Existing Methods, Their Performance Criteria and Uncertainty Assessment",Water, 9, 796; doi:10.3390/w9100796 www.mdpi.com/journal/water, 2017.