Mechanized network based cyber-attack detection and classification using DNN-generative adversarial model

Katikam Mahesh, Kunjam Nageswara Rao
Department of Computer Science and Systems Engineering, AUCE (A), Andhra University, Visakhapatnam, India

Article Info

Article history:

Received Dec 4, 2024
Revised Apr 4, 2025
Accepted Jul 2, 2025

Keywords:

Classification attacks
Cyberthreat detection
Deep neural network algorithm
Feature extraction
Intrusion detection system
Network security

ABSTRACT

These days almost everything is internet. Cyberattacks are the world's most pressing issues. Due to these attacks, Computer systems can be rendered inoperable, disrupted, destroyed or controlled via cyberattacks. Additionally, they can be used to steal, modify, erase, block, or alter data. Most organizations are facing this Issue and lose financially as well as in data security, there are numerous conventional intrusion detection systems (IDS) and firewalls are illustrations for network security tools which are not able to classify and detect different types of attacks in network. With machine learning approach using the Dataset KDD_CUP 99 as input, the synthetic minority oversampling technique (SMOTE) is one of the most often used oversampling methods for addressing imbalance issues. The proposed hybrid deep neural network (DNN), generative adversarial network (GAN), and exhaustive feature selection (EFS) can detect and classify several attack types including R2L, U2R, Probe, denial of service (DoS), and normal attacks types and inform to administrator to ring alarm sound to control and monitor network traffic in dynamically typed networks.

Corresponding Author:

Katikam Mahesh
Department of Computer Science and Systems Engineering, AUCE (A), Andhra University
Visakhapatnam-530003, India
Email: katikammahesh@gmail.com

1. INTRODUCTION

A cyberattack is an intentional attempt to hack into another person's or the information system of the organization [1]. Typically, the attacker gains an advantage by tampering in the victim's network [2]. An artificial neural network (ANN) with many generative adversarial network (GAN) and deep neural networks (DNNs) have the ability to model non-linear relationships just as shallow ANNs as shown in Figure 1.

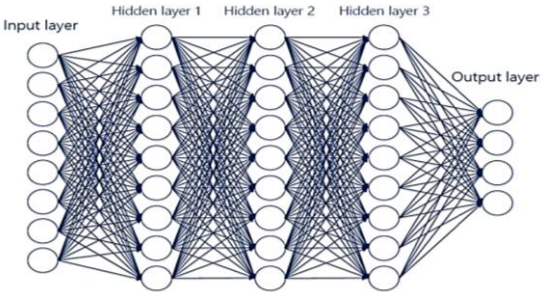


Figure 1. DNN for attack classification

Furthermore, current intrusion detection system (IDS) technologies are not able to manage network flow or detect, classify, and identify various types of cyberattacks on computer networks [3]. Different attack types, including denial of service (DoS), Probe, R2L, and U2R, can be identified and categorized by the proposed machine learning and deep learning technique with the KDD_CUP 99 dataset as input in a DNN [4]. A layer of input, an output layer, and at least a layer in the centre makes up a DNN. The network is deeper the fewer layers you are. Each of these levels uses a process called "feature selection" to carry out distinct classification and sorting tasks [5]. All we have to do is look at how the human brain functions to get a greater awareness of how a DNN works [6].

2. LITERATURE WORK

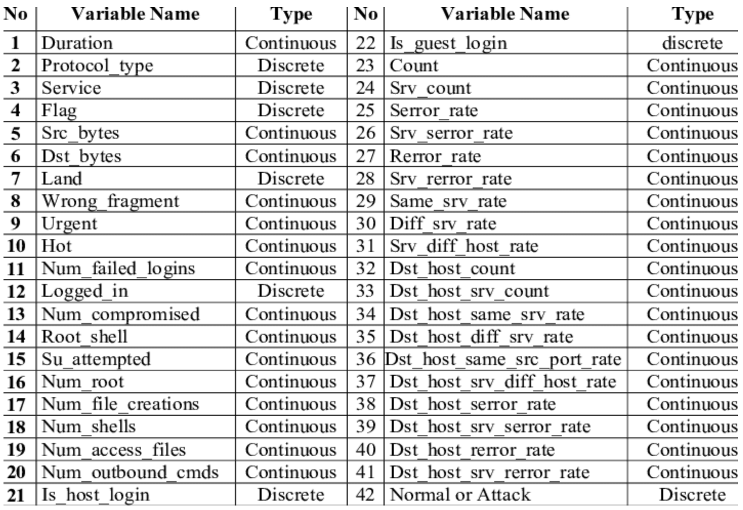
Several prior studies have highlighted using deep learning and machine learning approaches to improve the safety of systems for cyber security [7]. The background of security vulnerabilities systems: threat stack (IDS (accessed on 19 May 2022). But highly few frequently attacks are reported [8]. This work only classifies the normal attacks but no new types of attacks [9]. This work classifies the data but no clear about different attack detection. To eliminate all these drawbacks as well the proposed work can avoid pitfalls in past studies [10]. Furthermore, the many forms of cyberattacks on computer networks cannot be detected, classified, or identified by the IDS technologies currently in use [11]. With input from the KDD_CUP 99 dataset, the proposed DNN [12].

3. PROPOSED METHOD

The attack detection method's architecture is based on the security system's continuous data traffic monitoring of network systems to detect and categorize various security attack types. The attack detection strategy is created with the proposed DNN with GAN. GANs is a type of deep learning architecture [13]. It trains two neural networks to compete with one another to generate more real new data from a given training dataset [14]. In this case, you may create new photos from an existing image database or original music from a song library. With KDD_ CUP99 as input for identifying cyber-attacks that are introduced into the system by attackers and detected in system organizations which can identify and categorize different attack types such as normal, DoS, Probe, R2L, and U2R [15].

3.1. Data collection

KDD CUP 99 is an extremely demanding dataset as shown in Figure 2 with vast size, redundancy, an extensive number of variables (containing both numeric and categorical), and a skewed target variable [16]. It is a prominent dataset used for intrusion detection in academic literature. During a DARPA-sponsored event in 1999 at MIT's Lincoln Laboratory several attacks scenarios were simulated and features were collected, resulting in the creation of the dataset [17]. This dataset was a part of the 1999 KDD contest in intrusion detection. In the KDD training dataset, each of the nearly 4,900,000 unique connection vectors have resulted in 41 attributes and a label denoting as to is this a new or normal type of attacks [18].



No	Variable Name	Type	No	Variable Name	Type
1	Duration	Continuous	22	Is_guest_login	discrete
2	Protocol_type	Discrete	23	Count	Continuous
3	Service	Discrete	24	Srv_count	Continuous
4	Flag	Discrete	25	Serror_rate	Continuous
5	Src_bytes	Continuous	26	Srv_serror_rate	Continuous
6	Dst_bytes	Continuous	27	Rerror_rate	Continuous
7	Land	Discrete	28	Srv_rerror_rate	Continuous
8	Wrong_fragment	Continuous	29	Same_srv_rate	Continuous
9	Urgent	Continuous	30	Diff_srv_rate	Continuous
10	Hot	Continuous	31	Srv_diff_host_rate	Continuous
11	Num_failed_logins	Continuous	32	Dst_host_count	Continuous
12	Logged_in	Discrete	33	Dst_host_srv_count	Continuous
13	Num_compromised	Continuous	34	Dst_host_same_srv_rate	Continuous
14	Root_shell	Continuous	35	Dst_host_diff_srv_rate	Continuous
15	Su_attempted	Continuous	36	Dst_host_same_src_port_rate	Continuous
16	Num_root	Continuous	37	Dst_host_srv_diff_host_rate	Continuous
17	Num_file_creations	Continuous	38	Dst_host_serror_rate	Continuous
18	Num_shells	Continuous	39	Dst_host_srv_serror_rate	Continuous
19	Num_access_files	Continuous	40	Dst_host_rerror_rate	Continuous
20	Num_outbound_cmds	Continuous	41	Dst_host_srv_rerror_rate	Continuous
21	Is_host_login	Discrete	42	Normal or Attack	Discrete

Figure 2. Features of KDD_CUP99

3.2. Data preprocessing

Preparing information refers to the process used to collect data that was not processed that will be utilized by a machine learning model as shown in Figure 3. Considering only 50K data combined from all the types of protocols, verifying shape (number of rows and columns), looking up column names to identify features, verifying data integrity by checking for null values, and examining the distribution of the target column ('Label') to understand class distribution are all carried out as part of an exhaustive feature selection (EFS) [19]. An effort to enhance predictive model by trying to optimize features within a dataset made use of exhaustive feature selection [20]. This tests each potential feature combination will do 10-fold cross validation that utilized to limit the maximum number of features that may be selected from a dataset which comprised more than 80 features [21].

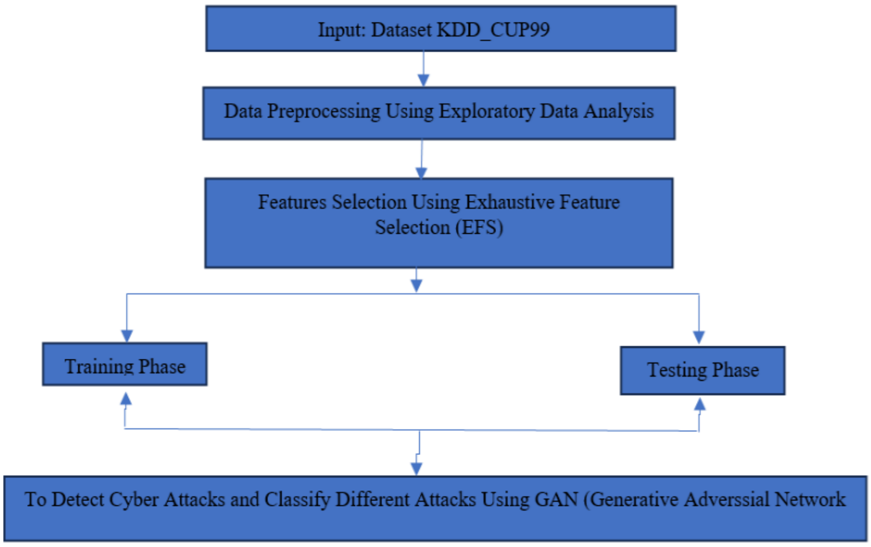

Input: Dataset KDD_CUP99
Data Preprocessing Using Exploratory Data Analysis
Features Selection Using Exhaustive Feature Selection (EFS)
Training Phase
Testing Phase
To Detect Cyber Attacks and Classify Different Attacks Using GAN (Generative Adverssial Network

Figure 3. Framework for attack classification

3.3. Feature selection

The parts deemed most significant and relevant are selected using a brute-force feature subset evaluation occurs in a feature selection process method [22]. Given any random regressor or classifier, the best subset is chosen by optimizing a given performance parameter. If the dataset has four features and the classifier is a random forest regressor, the algorithm will examine all fifteen feature combinations (assuming min_features=3 and max_features=20). fs = EFS (Random Forest Regressor ()), min_features=3, max_features=20, scoring='neg_mean_squared_error=10) Efs.fit (Y, X).

3.4. DNN-GAN model for attack detection

The network layers of the DNN and GAN models are combined in the hybrid DNN-GAN model. When viewed in Figure 4, the hybrid architecture combines the layers of the CNN with the GAN model, i.e., combining the output DNN with GAN to accurately identify the cyberattack [23]. First, the generator makes an image by providing any random integer. The discriminator accepts the image that was generated as input, and the real dataset is used to extract the actual images [24]. Both actual and phony photos are present in the discriminator, which now seeks to identify the real and fake images and predict the labels. Its output is the probability of a number between 0 and 1, where 1 denotes authenticity and 0 denotes a bogus forecast. The graphic below describes how a GAN operates. A DNN is made up of an input layer, an output layer, and at least one intervening layer [25]. The deeper the network, the more layers there are. In a method known as “feature hierarchy,” each of these tiers works out different kinds of specific sorting and classifying [26], [27].

3.4.1. Introduction to generative adversarial network (GAN)

A machine learning model called a generative adversarial network (GAN) is made to produce realistic data by identifying patterns in pre-existing training datasets. Through the use of deep learning techniques and an unsupervised learning framework, it functions in opposition to two neural networks, one of which generates data and the other determines whether the data is generated or real. The intricacy of calculations in generative models has made it more difficult to generate fresh data, including realistic images or text, even if deep learning has performed exceptionally well in tasks like image classification and speech recognition. When it pertains to managing complex network traffic, GANs are the main players.

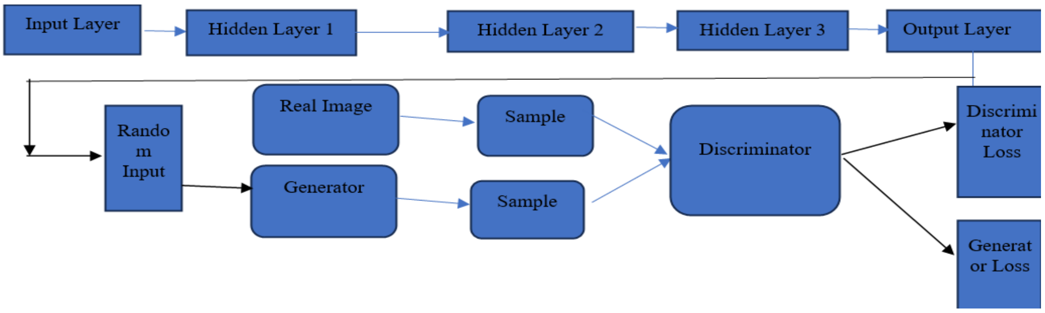


Figure 4. DNN-GAN model for attack detection

3.4.2. Generative adversarial network algorithm for detect cyber-threats and classify different attacks

Cybersecurity experts can improve their intrusion detection systems (IDS) by simulating the complex movements of possible attackers by creating hostile network traffic. Knowing the enemy's tactics before the combat starts is a proactive defines technique that this supports. By keeping one step ahead of any attacks, it serves as a digital sparring partner in this context, assisting firms in strengthening their cyber defences. Algorithm for detect cyber-threats and classify different attacks shown in Algorithm 1.

```
Algorithm 1: Detect cyber-threats
Input: Dataset KDD_CUP99
Step 1: The generator generates an image following receiving a random integer feed.
Step 2: A stream of images from the real, ground-truth dataset will be input into the discriminator combined with the newly created image.
Step 3: A stream of images from the real, ground-truth dataset is delivered into a discriminator along with the newly generated image.
Step 4: The discriminator really takes and pictures and returns likelihood that comprise of numbers between 0 and 1, when 0 indicates an image and 1 denotes an authentic image.
Output: To detect cyber threats and classify different attacks.
Stop
```

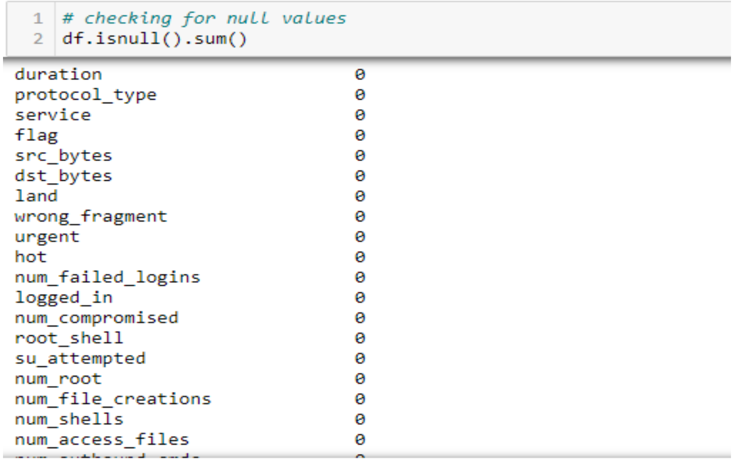
4. RESULTS AND DISCUSSION

4.1. Checking for null values

The results having no null dataset are recognized as missing values. Various symbols include blank cells, null values, or special characters like “NA” or “unknown,” are readily available to represent it. These missing data points save data analysis extremely hard and may result in biased or inaccurate findings. To obtaining the quality of dataset need to performing various data preprocessing techniques one of them are handling missing value and checking the null values. After checking null values, the model performs and takes correct decisions by replace with "NA" or Nan. Inadequate maintenance may cause past data to become corrupted. For a variety of reasons, some fields do not record observations. Human error might end up in a failure to record the values. The values were not purposefully served as by the user. This shows that the participant neglected to respond. The results as shown in Figure 5, there is no any null values found.

4.2. Visualization

Visualization of null values using values for the main variable of interest across two axis variables are represented as a grid of colored squares in a heat map as shown in Figure 6. A powerful tool for conveying numerical data is heatmap data visualization, which uses colours to represent numbers. It works particularly nicely for finding trends, patterns, and abnormalities in big datasets. The definition, types, benefits and best practices for heatmap data visualization will all be covered in this article. A heat map is a two-dimensional visualization of data where different values are presented as different colours. Users can rapidly understand the most important or important data points with the help of a basic heat map, which instantly offers a visual summary of data along two axes. The user can comprehend complex data sets with the help of more intricate heat maps. Data points in a data set can be viewed using a heat map. A common trait unites all heat maps: they convey the potential correlations between the variables depicted on the x-axis and y-axis by using various colors or shades of the same color to indicate various values.



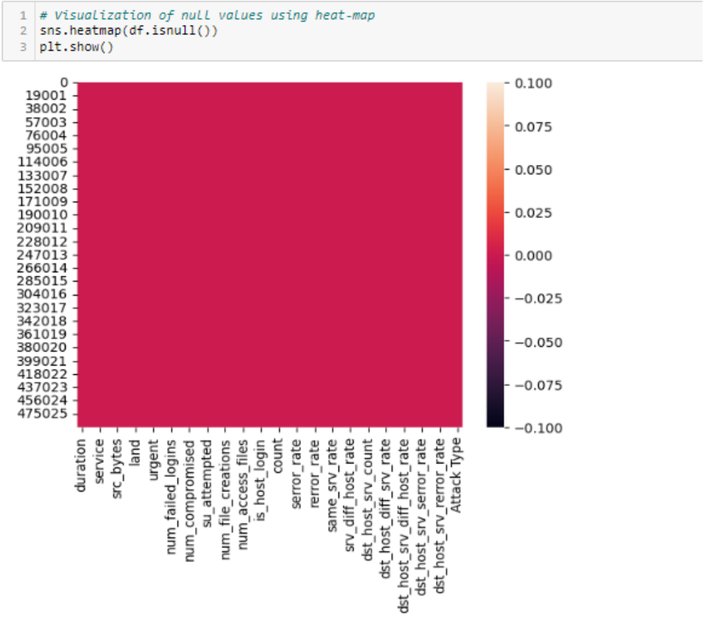
```
# checking for null values
df.isnull().sum()
```

```
duration                       0
protocol_type                  0
service                        0
flag                           0
src_bytes                      0
dst_bytes                      0
land                           0
wrong_fragment                 0
urgent                         0
hot                            0
num_failed_logins              0
logged_in                      0
num_compromised                0
root_shell                     0
su_attempted                   0
num_root                       0
num_file_creations             0
num_shells                     0
num_access_files               0
```

Figure 5. Checking null values for KDD_CUP99 dataset

4.3. Overall features getting only the categorical features

There is a limited range of possible values for categorical data as shown in Figure 7. For example, the various animal species found in a national park. The street names in a certain city whether an email is spam or not the exterior paint colours of houses. The working with numerical data section discusses binned numbers. Categorical data are those that can be grouped or defined. In statistics, this form of data is made up of variable categories or data that is grouped. It can be obtained from observations of quantitative data grouped within specified intervals or from observations of qualitative data that are distilled into counts. Qualitative data is another name for data that can be categorized. The simplest and primary categorical-column encoding approach is one-hot encoding. Each category ought to have its own multi-digit binary number. Therefore, the number of categories for the categorical characteristic to be encoded is equal to the number of digits. The term “one-hot” comes from the binary number's one digit being 1 and every other digit be zeros.

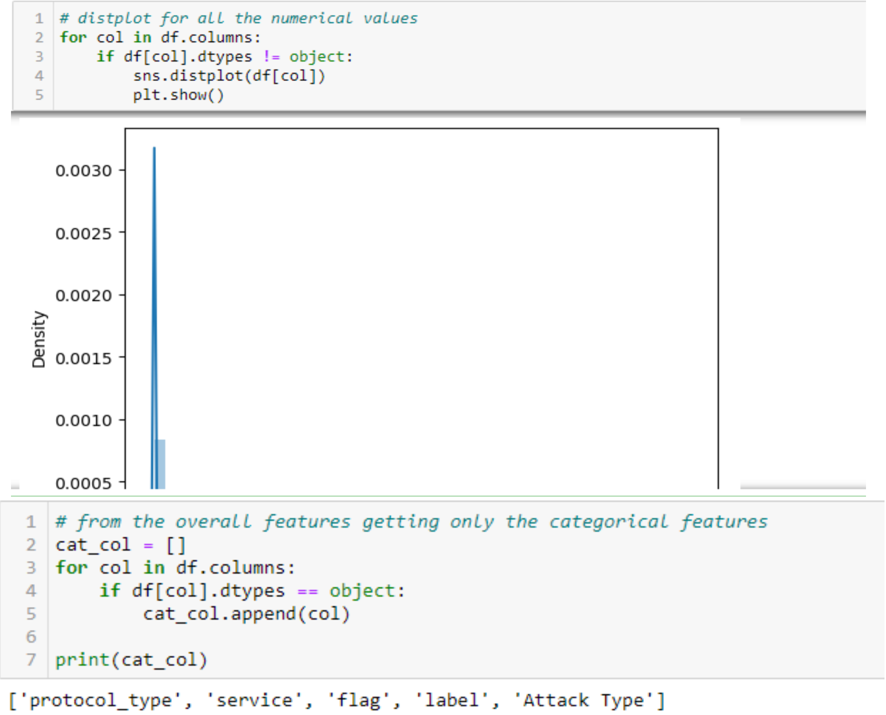


```
# Visualization of null values using heat-map
sns.heatmap(df.isnull())
plt.show()
```

Figure 6. Visualization of null values using heat map

4.4. Target attacks types

Targeted harm typically employs strategies that are comparable with those of regular online threats, like malware, exploits, corrupted or fake websites, and malicious emails. There are numerous ways in which targeted attacks are different from traditional internet threats. Usually, targeted attacks are carried out as campaigns. Advanced persistent threats (APTs) are not isolated incidents because they are often carried out as part of campaigns, which are a series of fruitless and successful attempts over time to access a target's network in greater and greater detail. The work targeted for detecting and classifying unique attacks types such as normal, dos, probe, r2l, u2r with a series that has the counts of each value is returned by the value. counts () method as shown in Figure 8. In other words, this method returns a number of unique items in any particular column from a data frame.



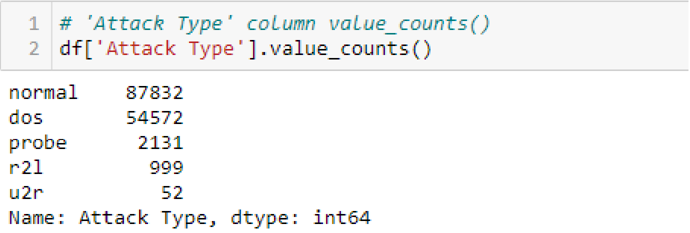
```
# distplot for all the numerical values
for col in df.columns:
    if df[col].dtypes != object:
        sns.distplot(df[col])
        plt.show()
```

```
# from the overall features getting only the categorical features
cat_col = []
for col in df.columns:
    if df[col].dtypes == object:
        cat_col.append(col)

print(cat_col)
```

```
['protocol_type', 'service', 'flag', 'label', 'Attack Type']
```

Figure 7. Categorical features



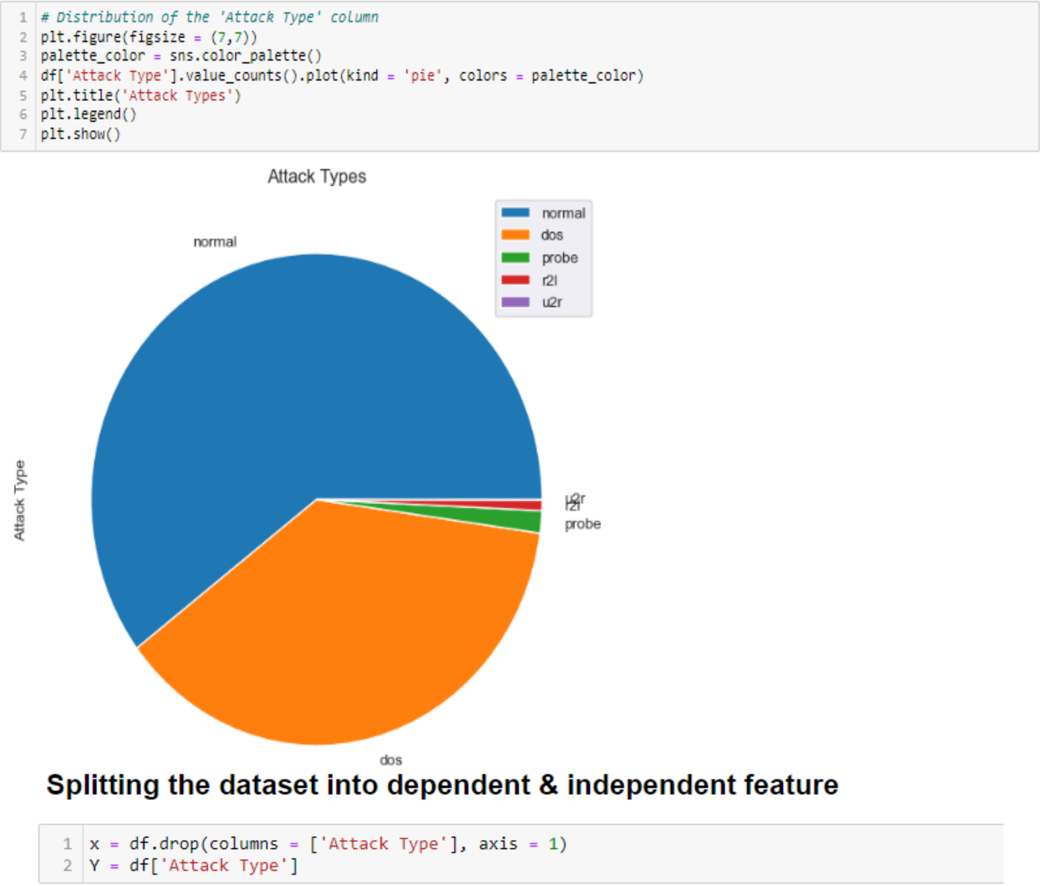
```
# 'Attack Type' column value_counts()
df['Attack Type'].value_counts()
```

```
normal    87832
dos       54572
probe      2131
r2l         999
u2r          52
Name: Attack Type, dtype: int64
```

Figure 8. Target attacks types

4.5. Distribution of attacks types

By giving the variables, a range that includes possible values, statistical distributions aid in recognizing issues and are highly useful in data science and machine learning. The attack distribution is shown in pie chart in Figure 9 with different attacks detection and classification. The detection and classification of attacks such as normal, DoS, probe, R2L, U2R can be shown in pie chart. The variable for which forecast or explanations are sought is known as the dependent variable. Predicting a target variable based on the values of a group of input variables is known as supervised learning. The price of a house could be the dependent variable in a regression problem, whereas the neighborhood, lot size, number of bedrooms, and other factors could be the independent variables. On the other hand, independent variables also called to as predictor variables are employed to explain or produce predictions for the variation in the dependent variable (the objective). Variables might have a continuous or categorical structure and can be either quantitative or qualitative. To improve its predictive value, data can be scaled or transformed. The result in Figure 9 shows the distribution of various attacks 'duration' column distribution splitting the dataset into dependent and independent features.



```
# Distribution of the 'Attack Type' column
plt.figure(figsize = (7,7))
palette_color = sns.color_palette()
df['Attack Type'].value_counts().plot(kind = 'pie', colors = palette_color)
plt.title('Attack Types')
plt.legend()
plt.show()
```

Splitting the dataset into dependent & independent feature

```
x = df.drop(columns = ['Attack Type'], axis = 1)
Y = df['Attack Type']
```

Figure 9. Distribution of attacks types

4.6. Normalization

Two important methods in data preprocessing are normalization and standardization. Adjusted by normalization, the data that falls between 0 and 1, but standardization involves rescaling the data to have the same standard deviation and mean. One of the widely used normalizing techniques, the min-max method, was used in the current study. Min-max scaling is commonly referred to as "normalization." Features are changed to a defined range, usually 0–1. The min-max scaling is shown in (1).

$$X - \frac{Xmin}{Xmax} - Xmin \ = \ Xnormalized \qquad (1)$$

The random feature value 'X' is the one that needs to be normalized. The dataset's minimal and maximum feature values are denoted by Xmin and Xmax, respectively. The normalized value is 0 when 'X' is the minimal value since the numerator is zero (Xmin - Xmin). The normalized value is 1 when 'X' is the maximum value since the numerator equals the denominator (Xmax - Xmin). The normalized value is in the range of 0 and 1 when X is neither minimum nor maximum. This method is known as the min-max scaling strategy. Z-score normalization (standardization) adjusts features to have a mean (μ) of 0 and a standard deviation (σ) of 1, using a Gaussian (bell curve) distribution of the data. The standardization is shown in (2).

$$`X' \ standardized \ = \ X - \frac{\mu}{\sigma} \qquad (2)$$

4.7. Handling the class imbalance using SMOTE Technique

One of the most common oversampling techniques for resolving difficulties with imbalance is the synthetic minority oversampling technique, or SMOTE technique. By replicating minority class situations at random, it aims to reach class distribution balance by utilizing old minority instances. SMOTE builds new minority instances that it generates from imblearn. Oversampling import SMOTE from collections import counter.

Before SMOTE, check the class distribution print ("Class distribution before SMOTE:", Counter(Y))
Setup to the SMOTE object: motes = SMOTE (random state = 42, sampling strategy = "auto")
Apply SMOTE to balance the dataset: smote.fit_resample (X, Y) = resampled, y_resampled
Verify the distribution of each class on SMOTE print ("Class distribution on SMOTE:", Counter(resampled).

Inferred from the experimental results, the GAN model achieves extraordinary accuracy, attack detecting and classification. Results validate the effectiveness of the optimized DNN-GAN model. With the old technique, using IDS just detecting and classifying only normal attacks but these are not able to monitoring the network traffic. So, with the achieved results of different attack types such as normal, DoS, Probe, R2L, U2R using proposed model: hybrid DNN-GAN which can monitor networks traffic efficiently. The comparative analysis is shown in Table 1.

Table 1. Comparative analysis

Aim	Attack type	
Cyber attack detecting and classification	Using IDS Normal attacks	Using proposed model: hybrid DNN-GAN Normal, DoS, Probe, R2L, U2R.

5. CONCLUSION

One of the biggest issues affecting the globe now is cyberattacks. Cyberattacks are unwanted attempts to enter computer systems unauthorized authorization having the intent to steal, show, change, disable, or destroy information. Due to the fact that millions of computers fall prey to this kind of activity every day, which costs organizations money by disclosing sensitive information to rival companies, data security has gained prominence and requires immediate attention. For the purpose of detecting network intrusion, there are numerous conventional network security tools and approaches available such as the ones listed below: antivirus software includes anti-malware, encryption and decryption, and access control protective firewalls. Furthermore, current IDS methodologies are unable to identify and classify many types of intrusions on computer networks. This work classifies the data but no clarity about different attacks detection. To eliminate all these drawbacks, the proposed work can avoid pitfalls in past studies. Furthermore, many forms of cyberattacks on computer networks cannot be detected, classified, or identified by the IDS technologies currently in use. Leveraging the KDD_CUP 99 dataset, the proposed kinds of assaults, among them DoS, Probe, R2L, and U2R can all be noticed and sorted via machine learning and methods of deep learning such as neural network algorithms (DNNs). 3 tiers: one for input, one as output, and a couple of intermediary elements make up a DNN. In future scope, detecting and classifying a greater number of different types of attacks will be made to reach accuracy.

FUNDING INFORMATION

Authors state no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Katikam Mahesh	✓	✓	✓	✓	✓	✓		✓	✓		✓			
Kunjam Nageswara Rao					✓					✓		✓	✓	

C :	**C**onceptualization	I :	**I**nvestigation	Vi :	**Vi**sualization
M :	**M**ethodology	R :	**R**esources	Su :	**Su**pervision
So :	**So**ftware	D :	**D**ata Curation	P :	**P**roject administration
Va :	**Va**lidation	O :	Writing - **O**riginal Draft	Fu :	**Fu**nding acquisition
Fo :	**Fo**rmal analysis	E :	Writing - Review & **E**diting		

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

Data availability does not apply to this paper as no new data were created or analyzed in this study.

REFERENCES

[1] Q. Abu Al-Haija and S. Zein-Sabatto, “An efficient deep-learning-based detection and classification system for cyber-attacks in IoT communication networks,” *Electronics*, vol. 9, no. 12, p. 2152, Dec. 2020, doi: 10.3390/electronics9122152.

[2] Y. Zhou, G. Cheng, S. Jiang, and M. Dai, “Building an efficient intrusion detection system based on feature selection and ensemble classifier,” *Computer Networks*, vol. 174, p. 107247, Jun. 2020, doi: 10.1016/j.comnet.2020.107247.

[3] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, “Deep learning approach for intelligent intrusion detection system,” *IEEE Access*, vol. 7, pp. 41525–41550, 2019, doi: 10.1109/access.2019.2895334.

[4] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, “A deep learning approach to network intrusion detection,” *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 1, pp. 41–50, Feb. 2018, doi: 10.1109/tetci.2017.2772792.

[5] M. Almehdhar *et al.*, “Deep learning in the fast lane: a survey on advanced intrusion detection systems for intelligent vehicle networks,” *IEEE Open Journal of Vehicular Technology*, vol. 5, pp. 869–906, 2024, doi: 10.1109/ojvt.2024.3422253.

[6] W. L. Al-Yaseen, Z. A. Othman, and M. Z. A. Nazri, “Multi-level hybrid support vector machine and extreme learning machine based on modified K-means for intrusion detection system,” *Expert Systems with Applications*, vol. 67, pp. 296–303, Jan. 2017, doi: 10.1016/j.eswa.2016.09.041.

[7] M. Belouch, S. El, and M. Idhammad, “A two-stage classifier approach using reptree algorithm for network intrusion detection,” *International Journal of Advanced Computer Science and Applications*, vol. 8, no. 6, 2017, doi: 10.14569/ijacsa.2017.080651.

[8] I. A. Mahar, W. Libing, G. A. Rahu, Z. A. Maher, and M. Y. Koondhar, “Feature based comparative analysis of traditional intrusion detection system and software-defined networking based intrusion detection system,” in *2023 IEEE 8th International Conference on Engineering Technologies and Applied Sciences (ICETAS)*, Oct. 2023, pp. 1–5, doi: 10.1109/icetas59148.2023.10346497.

[9] D. R. Dipta, J. Tan, and B. Gulmezoglu, “Systematical evasion from learning-based microarchitectural attack detection tools,” *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 14, no. 4, pp. 823–833, 2024, doi: 10.1109/jetcas.2024.3491497.

[10] M. Ahmed, A. Naser Mahmood, and J. Hu, “A survey of network anomaly detection techniques,” *Journal of Network and Computer Applications*, vol. 60, pp. 19–31, Jan. 2016, doi: 10.1016/j.jnca.2015.11.016.

[11] J. Chen, Y. Guo, K. Shi, and M. Yang, “Network intrusion detection method of power monitoring system based on data mining,” in *2022 2nd International Conference on Algorithms, High Performance Computing and Artificial Intelligence (AHPCAI)*, Oct. 2022, pp. 255–259, doi: 10.1109/ahpcai57455.2022.10087405.

[12] I. Ghafir and V. Prenosil, “Advanced persistent threat attack detection: an overview,” *International Journal of Advancements in Computer Networks and Its Security– IJCNS*, vol. 4, no. 4, pp. 50–54, Aug. 2014.

[13] A. S. Ashoor and S. Gore, “Difference between intrusion detection system (IDS) and intrusion prevention system (IPS),” in *International Conference on Network Security and Applications*, 2011, pp. 497–501, doi: 10.1007/978-3-642-22540-6_48.

[14] T. Vollmer, J. Alves-Foss, and M. Manic, “Autonomous rule creation for intrusion detection,” in *2011 IEEE Symposium on Computational Intelligence in Cyber Security (CICS)*, Apr. 2011, pp. 1–8, doi: 10.1109/cicybs.2011.5949394.

[15] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, “A detailed analysis of the KDD CUP 99 data set,” in *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, Jul. 2009, pp. 1–6, doi: 10.1109/cisda.2009.5356528.

[16] S. M. Gaffer, M. E. Yahia, and K. Ragab, “Genetic fuzzy system for intrusion detection: analysis of improving of multiclass classification accuracy using KDDCup-99 imbalance dataset,” in *2012 12th International Conference on Hybrid Intelligent Systems (HIS)*, Dec. 2012, pp. 318–323, doi: 10.1109/his.2012.6421354.

[17] Y. Chen and F. Yuan, “Dynamic detection of malicious intrusion in wireless network based on improved random forest algorithm,” in *2022 IEEE Asia-Pacific Conference on Image Processing, Electronics and Computers (IPEC)*, Apr. 2022, pp. 27–32, doi: 10.1109/ipec54454.2022.9777557.

[18] L. Chen, X. Kuang, A. Xu, S. Suo, and Y. Yang, “A novel network intrusion detection system based on CNN,” in *2020 Eighth International Conference on Advanced Cloud and Big Data (CBD)*, Dec. 2020, pp. 243–247, doi: 10.1109/cbd51900.2020.00051.

[19] I. Sharafaldin, A. Gharib, A. H. Lashkari, and A. A. Ghorbani, “Towards a reliable intrusion detection benchmark dataset,” *Software Networking*, vol. 2017, no. 1, pp. 177–200, 2017, doi: 10.13052/jsn2445-9739.2017.009.

[20] M. Bijone and J. Dangra, “A survey of signature based & statistical based intrusion detection techniques,” *International Journal for Scientific Research & Development*, vol. 4, no. 8, pp. 583–585, 2016.

[21] L. Mohan, S. Jain, P. Suyal, and A. Kumar, “Data mining classification techniques for intrusion detection system,” in *2020 12th International Conference on Computational Intelligence and Communication Networks (CICN)*, Sep. 2020, pp. 351–355, doi: 10.1109/cicn49253.2020.9242642.

[22] E. P. Nugroho, T. Djatna, I. S. Sitanggang, A. Buono, and I. Hermadi, “A review of intrusion detection system in IoT with smachine learning approach: current and future reearch,” in *2020 6th International Conference on Science in Information Technology (ICSITech)*, Oct. 2020, pp. 138–143, doi: 10.1109/icsitech49800.2020.9392075.

[23] F. Kuang, W. Xu, and S. Zhang, "A novel hybrid KPCA and SVM with GA model for intrusion detection," *Applied Soft Computing*, vol. 18, pp. 178–184, May 2014, doi: 10.1016/j.asoc.2014.01.028.

[24] I. Ahmad, M. Hussain, A. Alghamdi, and A. Alelaiwi, "Enhancing SVM performance in intrusion detection using optimal feature subset selection based on genetic principal components," *Neural Computing and Applications*, vol. 24, no. 7–8, pp. 1671–1682, Apr. 2013, doi: 10.1007/s00521-013-1370-6.

[25] F. N. M. Sabri, N. Md.Norwawi, and K. Seman, "Identifying false alarm rates for intrusion detection system with data mining," *International Journal of Computer Science and Network Security*, vol. 11, no. 4, pp. 95–99, 2011.

[26] L. Zhang, J. Zhang, Y. Chen, and S. Lao, "Hybrid intrusion detection based on data mining," in *2018 11th International Conference on Intelligent Computation Technology and Automation (ICICTA)*, Sep. 2018, pp. 299–301, doi: 10.1109/icicta.2018.00074.

[27] T. Soewu, Hemant, M. Rakhra, and D. Singh, "Analysis of data mining-based approach for intrusion detection system," in *2022 5th International Conference on Contemporary Computing and Informatics (IC3I)*, Dec. 2022, pp. 908–912, doi: 10.1109/ic3i56241.2022.10072828.

BIOGRAPHIES OF AUTHORS

Katikam Mahesh currently as a research scholar at Andhra University College of Engineering in Visakhapatnam India in the Department of Computer Science and Systems Engineering. He pursued his Master's in computer science and engineering from SISTAM College in Srikakulam. His area of interest includes networks security and machine learning. He has around 02 publications in the area of networks security and machine learning. He can be contacted at email: katikammahesh@gmail.com.

Dr. Kunjam Nageswara Rao working as Professor and Head in the Department of Computer Science and Systems Engineering, University College of Engineering, Andhra University, Visakhapatnam, Andhra Pradesh, India. He completed his B.Tech. in computer science and systems engineering from GITAM Engineering College, M.Tech. in computer science and technology from Andhra University and Ph.D. in computer science engineering from Andhra University. His current research includes bio-informatics, computer science information systems, health informatics, big data analytics, artificial intelligence and machine learning, algorithm and design analysis. He can be contacted at email: kunjamnag@gmail.com.