**INTRUSION DETECTION SYSTEM USING MACHINE LEARNING**

**Name : Khushi Singh**

**Email :** [**khushi.singh2021b@vitstudent.ac.in**](mailto:khushi.singh2021b@vitstudent.ac.in)

**Name : Siddharth Upadhyay**

**Email :** [**siddharth.upadhyay2021@vitstudent.ac.in**](mailto:siddharth.upadhyay2021@vitstudent.ac.in)

**ABSTRACT**

Intrusion Detection Systems (IDS) have become crucial to protecting networks as well as confidential data due to the ever-rising sophistication of cyber attacks and the crucial role that information security plays in today's digital landscape. The capacity of conventional rule-based IDS solutions to recognise sophisticated and changing attack patterns is frequently constrained. In contrast, a potential strategy to improve the accuracy and effectiveness of IDS has evolved through the integration of machine learning and neural network approaches.

This study examines how neural networks and machine learning can be used to create robust and flexible IDS. IDS can distinguish between normal network operations and probable intrusions using machine learning, which makes use of large datasets and potent algorithms to discover unusual patterns. Furthermore, neural networks enable deep learning, enabling IDS to learn intricate features.

***Keyword:cyber-attack,neural;networks,security;breache;,logisticregression;artificial neural networks***

1. **INTRODUCTION**

**A**ssuring the security and integrity of digital assets has become a top priority in today's linked world, where information technology supports nearly every aspect of our lives. Organisations are engaged in a never-ending struggle to safeguard their networks, data, and intellectual property against unauthorised access and malevolent actions due to the exponential expansion of cyber threats and advanced hacking techniques[1]. A crucial line of defence, intrusion detection systems (IDS) act as watchful defenders who diligently track network activity and spot any security breaches[2].Network security has long been dependent on conventional IDS systems that use predetermined rules and signatures. Critical vulnerabilities are exposed as a result of the inherent limits of these rule-based techniques in identifying new and evasive assaults[3].

The purpose of this research study is to investigate how ML(machine learning) and NN(neural networks) can be used to create an advanced intrusion detection system (IDS) that is capable of quickly identifying and minimising intrusions. This research attempts to find the most appropriate ways for intrusion detection and provide insights into their strengths, limits, and prospective enhancements through a thorough analysis of several ML algorithms and NN architectures.

By shedding light on the efficiency of ML and NN approaches, the research's findings will advance the field of intrusion detection. The findings will guide the creation of more precise, adaptable, and reliable IDS systems that can successfully fend off changing cyberthreats.

More flexible and intelligent IDS technology are clearly needed as cyber attackers continue to change their strategies[4].

A paradigm shift in intrusion detection has been driven by the development of machine learning (ML) and neural networks (NN), enabling intrusion detection systems (IDS) to learn from data patterns, adapt to new attack routes, and increase detection accuracy[5]. This study explores the efficacy, difficulties, and prospective uses of Machine Learning and Neural Networks based Intrusion detection systems in contemporary cybersecurity environments.

1. **Background and Motivating Factors**

Traditional Intrsion detection systems were vulnerable to zero-day exploits and sophisticated attacks that eluded signature-based detection since they depended on static rules and signatures to recognise known attack patterns[6]. The threat landscape is rapidly changing, necessitating a more proactive, adaptable strategy to identify known and emerging threats.

The purpose of this research study is to investigate how ML and NN can be used to create an advanced intrusion detection system (IDS) that is capable of quickly identifying and minimising intrusions. This research attempts to find the most appropriate ways for intrusion detection and provide insights into their strengths, limits, and prospective enhancements through a thorough analysis of several ML algorithms and NN architectures.

The examination of probable future developments and trends in ML and NN-based IDS serves as the paper's conclusion, underscoring their continuous importance in the always changing cybersecurity scene.

In summary, the incorporation of ML and NN methods into IDS is a viable way to improve network security and counter new cyber threats[7]. A more robust and preemptive approach to intrusion detection is made possible by the power of data-driven decision-making, adaptive learning, and sophisticated pattern recognition[8]. This is made possible by ML and NN-based IDS. With the help of next-generation IDS systems that protect our digital assets with unmatched precision and vigilance, our research intends to further the corpus of knowledge on the revolutionary potential of ML and NN in cybersecurity[9].

1. **RELATED WORK**

Saranya, T., Sridevi, S., Deisy, C., Chung, T. D., & Khan, M. A. et al [1] analyzed the effectiveness of various ML techniques for IDS, including Modified K-Means, J.48, Support Vector Machine (SVM), decision table, PCA, Logistic Regression, decision tree, and Artificial Neural Network (ANN). Along with the aforementioned machine learning algorithms, their work implemented algorithms like Linear Discriminant Analysis (LDA), Classification and Regression Trees (CART), and Random Forest (RF) algorithms for categorising intrusion detection.

Sultana, N., Chilamkurti, N., Peng, W., & Alhadad, R et al [2] emphasised on the use of SDN as a platform for integrating NIDS with ML/DL techniques outside the scope of current evaluation works.

Venkatesan, S. al [5] analyed the effectiveness of various ML techniques for IDS, including Modified K-Means, Support Vector Machine (SVM), decision table, PCA, Logistic Regression, decision tree, and Artificial Neural Network (ANN). Along with the aforementioned machine learning algorithms His work implemented algorithms like Linear Discriminant Analysis (LDA), Classification and Regression Trees (CART), and Random Forest (RF) algorithms for categorising intrusion detection.

Sarhan, M., Layeghy, S., & Portmann, et al [6] presented thorough analysis of recent NIDSs based on machine learning methodologies, Their research came to the conclusion that nearly all of them fail to complete what they view as necessary steps for a trustworthy comparison and evaluation of NIDSs.

Mane, S., and Rao, D. [8] acuurately demonstrated use of deep neural networks.They provided an explainable AI framework to offer transparency across the machine learning pipeline. They accomplished this by utilising Explainable AI algorithms, which aim to reduce the black boxes that ML models are by explaining why a prediction was made.

Apruzzese, G., Pajola, L., & Conti, M. et al[9] .approach for the problem mainly emphasized on wider variety of plausible use-cases that may be evaluated via cross-evaluations, facilitating the finding of as-of-yet unidentified characteristics of cutting-edge ML-NIDS.Their paper suggested the first framework, XeNIDS. They demonstrated the reqirements, but also the dangers, of cross-evaluations of ML-NIDS using XeNIDS on six well-known datasets.

Thapa, N., Liu, Z., Kc, D. B., Gokaraju, B., & Roy, K. [14] presented a comparison of various ML and DL models on Coburg intrusion detection datasets (CIDDSs) is presented in this research. On the CIDDS dataset, they compare various ML- and DL-based models first. Then, they provide an ensemble model that integrates the top ML techniques.to obtain high-performance metrics, and DL models. Lastly, they compared our top models toutilising the CIC-IDS2017 dataset, and compared them to contemporary models.

Thapa, N., Liu, Z., Kc, D. B., Gokaraju, B., & Roy, K. [14] presented a comparison of various ML and DL models on Coburg intrusion detection datasets (CIDDSs) is presented in this research. On the CIDDS dataset, they compare various ML- and DL-based models first. Then, they provide an ensemble model that integrates the top ML techniques.to obtain high-performance metrics, and DL models. Lastly, they compared our top models toutilising the CIC-IDS2017 dataset, and compared them to contemporary models.

Magán-Carrión, R., Urda, D., Díaz-Cano, I., & Dorronsoro, B [13] put forward a thorough analysis of contemporary NIDSs based on machine learning methodologies, concluding that nearly all of them fall short of meeting the requirements that the authors of this paper view as essential for a valid comparison and evaluation of NIDSs. Second, a structured technique is suggested and evaluated using the UGR'16 dataset in order to see how well it may be used to solve network attack detection issues.

Chua, T. H., & Salam, I. [15] used a dataset that was produced after the training dataset to assess the ML-based IDS. As the testing dataset reflects the changes in the type of attack and the changes in network infrastructure over time, the suggested method may more accurately evaluate the long-term performance of an ML-based IDS. All models can perform well when the gap between the training and testing datasets is minor, according to our research using the LUFlow dataset.

Sarhan, M., Layeghy, S., Moustafa, N., & Portmann, M. [12] Using their publicly available packet capture files, this study offers NetFlow features from four benchmark NIDS datasets: UNSW-NB15, BoT-IoT, ToN-IoT, and CSE-CIC-IDS2018.For the purpose of resolving binary- and multiclass-based learning difficulties, the generated Netflow datasets have been tagged. According to preliminary findings, when compared to each dataset's original feature dataset, NetFlow features produce similar binary-class results and worse multi-class classification results.

Zamani, M., & Movahedi, M.[11]planned and evaluatedvarious techniques of machine learning can result in higher detection rates, lower false alarm rates and reasonable computation and communication costs, their effectiveness. We separate the schemes into techniques based on computational intelligence (CI) and techniques based on traditional artificial intelligence (AI). We describe how several CI technique traits can be applied to create effective IDS.

Dias, L. P., Cerqueira, J. D. J. F., Assis, K. D., & Almeida, R. C. [20] suggested an IDS system based on the KDDCUP'99 dataset and artificial neural network (ANN). When compared to conventional methods, experimental findings clearly demonstrate that the suggested system can achieve an overall accuracy of 99.9% regarding the classification of pre-defined classes of intrusion attacks.

Omer, K. A. A., & Awn, F. A. [19] employed artificial neural networks (ANNs), which they trained using feed-forward neural networks (FFNNs) and back-propagation algorithms (BPAs), to assess the performance of both IDS systems. In order to train and test both IDS systems, a subset of the KDD CUP'99 dataset was employed. The acquired experimental findings demonstrate that, in terms of classification rate, detection rate, and false negative error, the anomaly-based IDS system performs better than the misused-based IDS system.

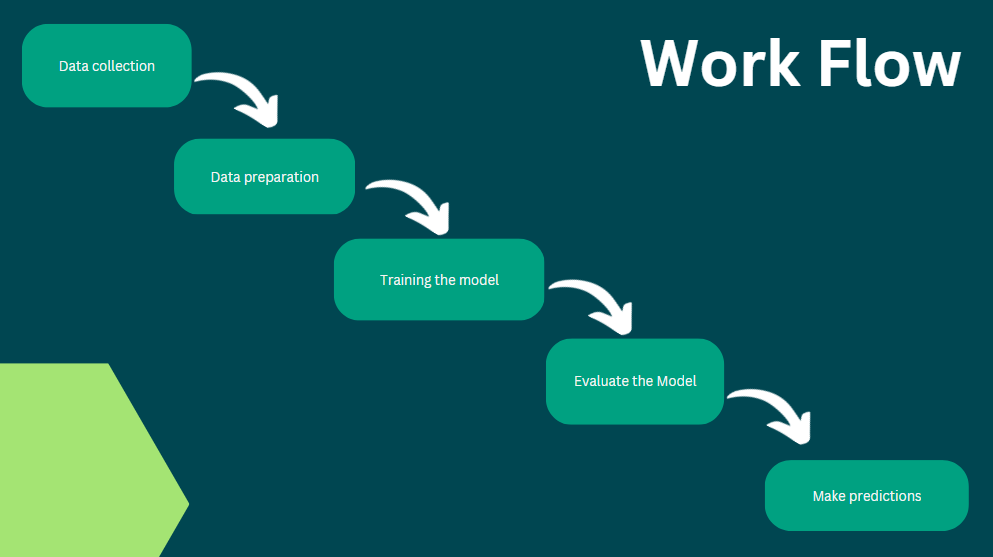
1. **METHODOLOGY**

Controlled trials are conducted in the experimental research design to examine how various ML algorithms and NN architectures affect the performance of the intrusion detection system. Here's an illustration:

"To assess the performance of various ML algorithms and NN architectures in our intrusion detection system, we utilised an experimental research approach. Datasets were prepared in two groups: one for modelling training and the other for performance evaluation.

We integrated Logistic Regression, K nearest neighbours (KNN) , Feedforward Neural Network, Convolutional Neural Network, and ML and DL techniques. On the training and testing datasets, each method and architecture was trained and evaluated separately.The primary objective of the experimental design was to identify the ML algorithms and NN architectures that yield the highest detection accuracy, lowest false positive rate, and robustness in detecting novel intrusion patterns."

1. **PROPOSED MODEL**

****

**Fig: Work flow diagram**

1. **DATA COLLECTION :**

The network intrusion detection dataset used in this research was obtained from Kaggle (named as Phase2.csv) it consists of a total of 25191 observations(0 to 25190) with each instance containing 42 features related to the network detection activities.

The dataset to be audited is publicly available and was provided which consists of a wide variety of intrusions simulated in a military network environment. It created an environment to acquire raw TCP/IP dump data for a network by simulating a typical US Air Force LAN. The LAN was focused like a real environment and blasted with multiple attacks. A connection is a sequence of TCP packets starting and ending at some time duration between which data flows to and from a source IP address to a target IP address under some well-defined protocol. Also, each connection is labelled as either normal or as an attack with exactly one specific attack type. Each connection record consists of about 100 bytes.

For each TCP/IP connection, 41 quantitative and qualitative features are obtained from normal and attack data (3 qualitative and 38 quantitative features) .The class variable has two categories:

• Normal

• Anomalous

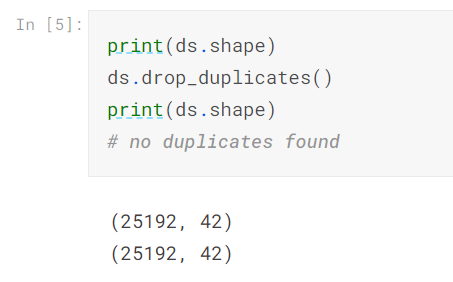
The dataset was developed to imitate network traffic in a military setting, complete with numerous intrusion attempts and standard network operations.

1. **DATA PREPROCESSING**

Before using the dataset for model training, we conducted data preprocessing to handle missing values and normalize the numerical features.

To ensure the integrity of the dataset we performed analysis on the missing values, duplicate values and outliers in the dataset but we found no missing or duplicate values in our dataset .

**Fig(1.1)**

****

**Fig(1.1)**

##### DATA TRANSFORMATION

##### We used the Standard scaling technique to rescale all numerical characteristics between 0 and 1 in order to handle all of the different scales of numerical features. By ensuring that each feature contributes equally to model training, this transformation minimises the dominance of features with higher values

##### Standard scaling, also known as z-score normalization or standardization, is a data preprocessing technique used to rescale numerical features in a dataset. It transforms the data such that the mean of the feature becomes 0, and the standard deviation becomes 1. This process makes the features comparable and helps certain machine learning algorithms converge faster and perform better.

Standard scaling process for a feature x is gien by the following formula:

Z = (x-μ)/σ

* Z is the standardized value of the feature
* X is the original value of the feature.
* μ is the mean(average of the feature across the dataset.
* σis the standard deviation of the feature across the dataset.

Here are the steps on how standard scaling works:

1. Calculate Mean and Standard Deviation: Compute the mean (μ) and standard deviation (σ) of the feature across the entire dataset.
2. Subtract Mean: For each data point in the feature, subtract the mean (μ) from the original value (x). This centers the data around zero.
3. Divide by Standard Deviation: Divide the centered values by the standard deviation (σ). This scales the data, making the standard deviation equal to 1.

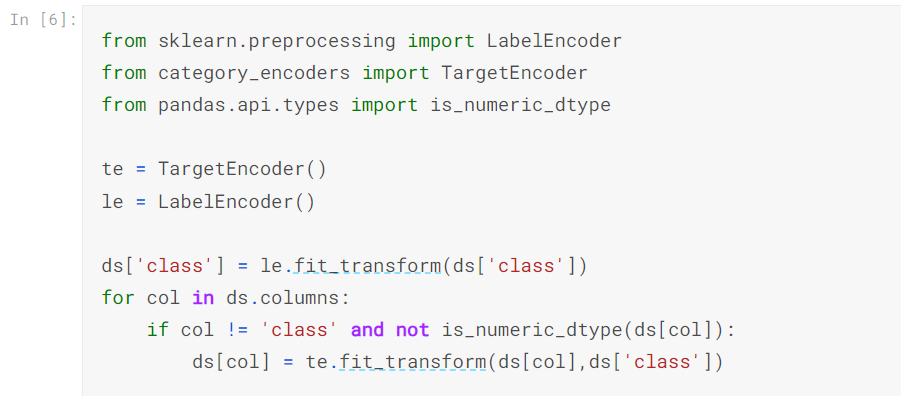
The standardised data that emerges will have a 0 mean and a 1 standard deviation. Standard scaling does not alter the distribution or structure of the data, but it does facilitate the processing of the features by some machine learning algorithms, particularly those that are sensitive to the scale of the input variables (such as gradient-based optimisation methods).

It's crucial to keep in mind that traditional scaling presume the feature has an approximately Gaussian (bell-shaped) distribution.

##### To handle categorical features we used target encoder and label encoder, which enabled the models to properly comprehend the categorical data.

Label encoding is a simple technique where each unique category in a categorical variable is assigned an integer value. The labels are typically encoded in ascending order, starting from 0. For example, if you have a categorical variable "Color" with categories ['Red', 'Blue', 'Green'], label encoding would map them to [0, 1, 2].

Target encoding is a more sophisticated technique that uses the target variable (the variable to be predicted) to encode categorical data. It is also known as mean encoding or probability encoding. The mean (or similar aggregate) of the target variable for each category is used to replace the categorical values.**Fig(1.2)**



**Fig(1.2)**

1. **FEATURE SELECTION AND DATA SPLITTING:**

We split the dataset into two mutually exclusive sets “the training set” and “the testing set”

in order to appropriately evaluate the generalisation performance of our machine learning models. We were able to train the models on one subset and assess their performance using previously unreleased data from the other group due to a technique known as "train-testsplit.”

The prominent Python method 'train\_test\_split' from the'sklearn.model\_selection' module was used to carry out the train-test split. The training set received 70% of the dataset, and the testing set received the remaining 30%. Based on accepted research practise, the 70-30 split was chosen to create a balance between having enough data for model training and making sure the test set was big enough.

It is crucial to remember that the stratified train-test split was carried out while taking the target variable's class distribution into account. This avoids class imbalance difficulties during model evaluation by ensuring that both the training and testing sets contain a representative percentage of cases from each class.

We made sure that the models weren't exposed to the testing data during training by employing a train-test split, maintaining the independence between the two sets. To generate unbiased estimates of the model's performance on unobserved data, this separation is essential. The testing set furthermore acts as an external validation set, enabling us to assess the model's capacity to generalise to fresh and undiscovered situations.

1. **TRAINING THE MODEL:**

The model training stage is a crucial step in creating an Intrusion Detection System (IDS) that is reliable and accurate. Specifically, we train the K-Nearest Neighbours (KNN), Logistic Regression (LR), Artificial Neural Network (ANN) models at this stage. Based on the attributes derived from the network traffic data, each model is intended to detect and categorise network activity as either normal or anomalous.

1. **K-NEAREST NEIGHBORS (KNN):**

It is a simple and intuitive machine learning algorithm used for both classification and regression tasks. It is a non-parametric algorithm, meaning it doesn't make any assumptions about the underlying data distribution.KNN identifies the K closest data points to a target instance in the feature space and assigns the target instance the class label that is most common among its K neighbors.

KNN is a flexible and simple method that works well with small to medium-sized datasets. However, because it is necessary to determine distances to every data points during prediction, its computational complexity rises with the amount of the dataset. For huge datasets, it might not be the most effective algorithm. Instance-based learning and neighborhood-based predictions in machine learning are concepts that can be understood by starting with KNN.

1. **LOGISTIC REGRESSION:**

A linear classification approach called logistic regression is used to model the probability of a binary outcome. For our intrusion detection task, where the binary result denotes the probability that a network activity is normal or abnormal, we used logistic regression. The logistic function, also referred to as the sigmoid function, converts a linear combination of input data into a probability score between 0 and 1, allowing the model to decide whether to classify anything as true or false in binary form.

sigmoid(z)= 1/(1+e -z)

1. **ARTIFICIAL NEURAL NETWORK (ANN):**

The adaptable Artificial Neural Network (ANN) design can recognise intricate patterns in input data. A feedforward ANN with several hidden layers was used in our intrusion detection system. The feature vectors generated from the network traffic are fed into the ANN's input layer, while the hidden layers change the input in non-linear ways. The final binary classification result is generated by the output layer. We used the sigmoid function for the output layer and the Rectified Linear Unit (ReLU) as the activation function for the hidden layers. To reduce the cross-entropy loss, the network is trained using the backpropagation algorithm. Backpropagation enables neural networks to gain knowledge from data and develop over time. The model continuously modifies its parameters to minimise the loss during iterations of the training process, effectively learning to improve predictions on fresh, unforeseen data.

1. **TESTING THE MODEL:**

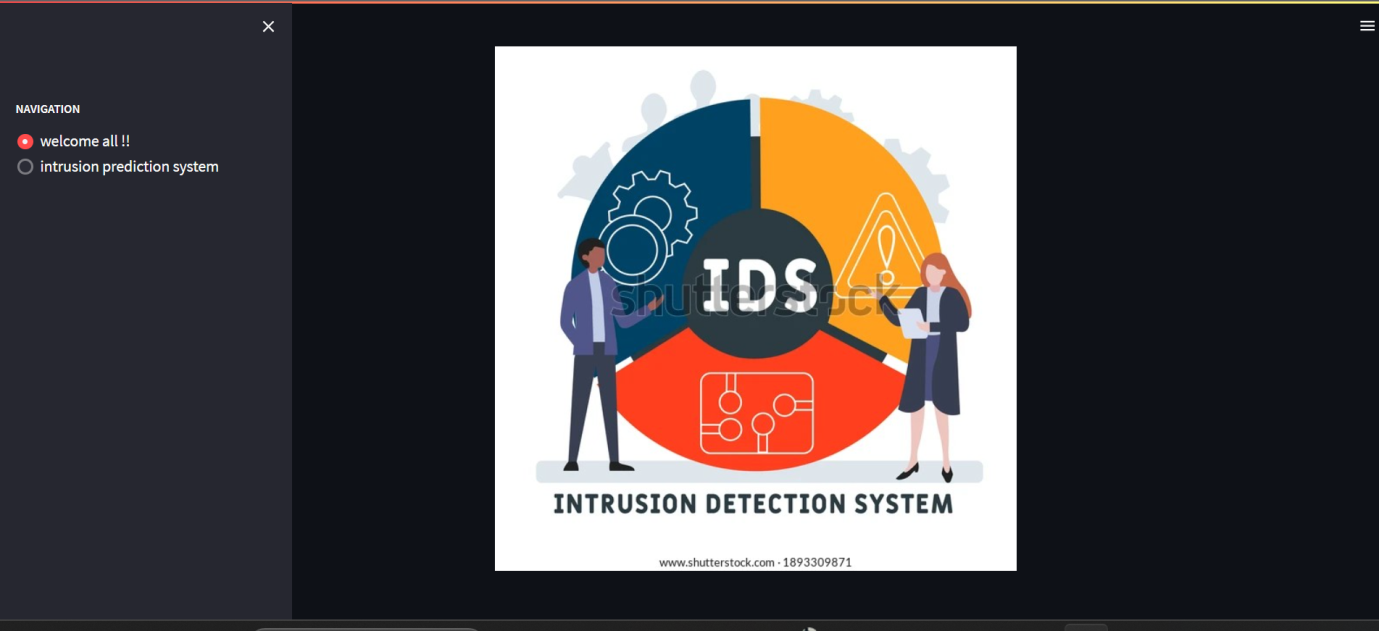
Using all the trained models with the help of various algorithms we predict a new set of values using our 30 percent testing data. These values are then compared with the actual values and from this we can estimate the accuracy of our trained model.

Using accuracy as a metric we can tell how well our model performs for random set of data.

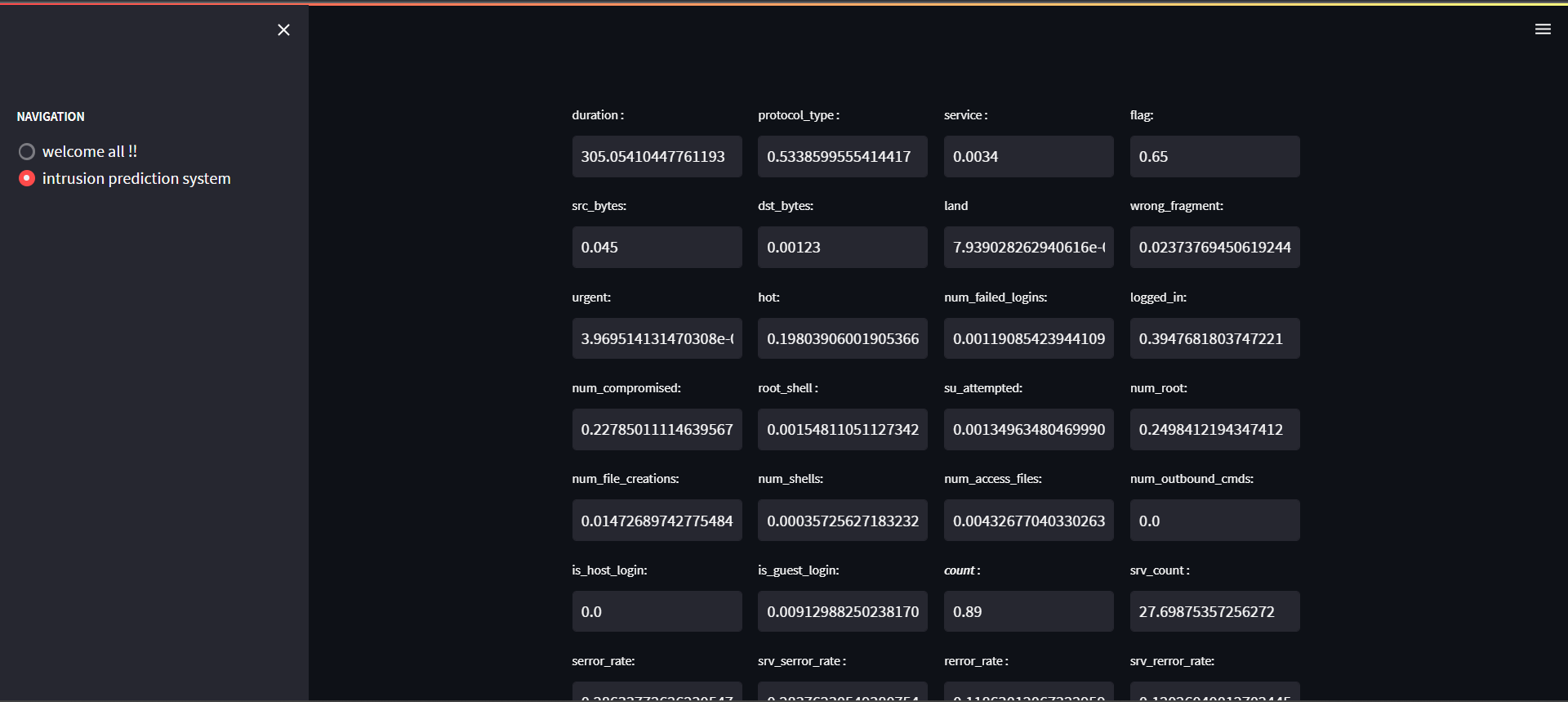
1. **HOSTING THE MODEL ( USING STREAMLIT ) :**

Streamlit is an open-source Python library used for creating interactive web applications for data science and machine learning projects. It allows data scientists and developers to turn data scripts into web applications quickly and easily, without requiring extensive web development knowledge.With Streamlit, you can transform your existing Python code, data visualizations, and machine learning models into interactive web apps that can be easily accessed and shared with others.

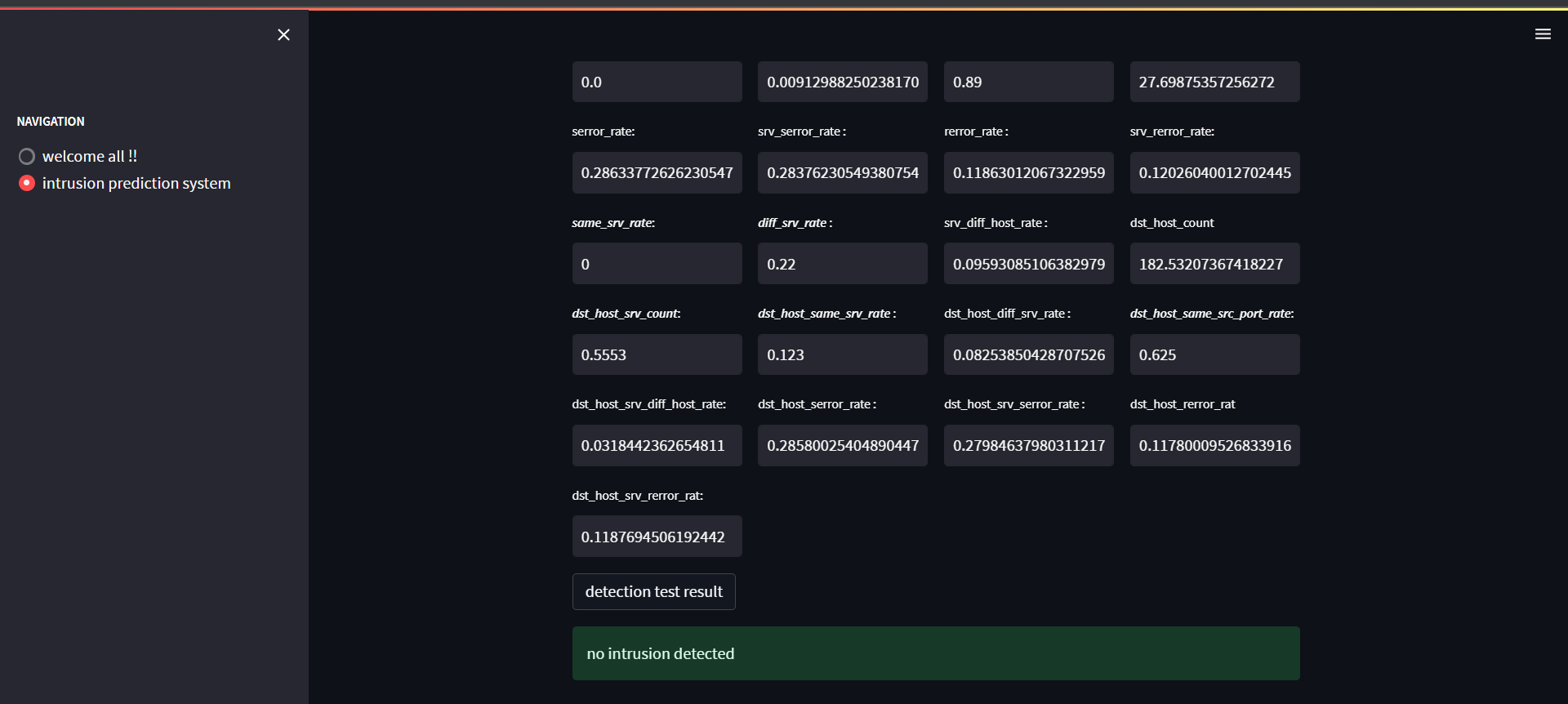
**Fig(2.1),Fig(2.2),Fig(2.3)**

****

**Fig(2.1) HOSTING ( STREAMLIT )**

****

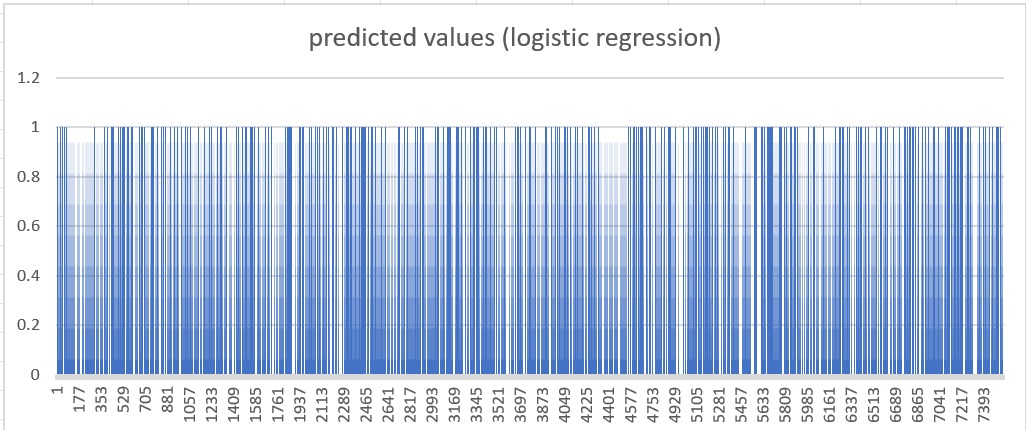
**Fig(2.2)**

****

**Fig(2.2)**

1. **EXPERIMENTAL RESULT/RESULT ANALYSIS**
2. **Logistic Regression**

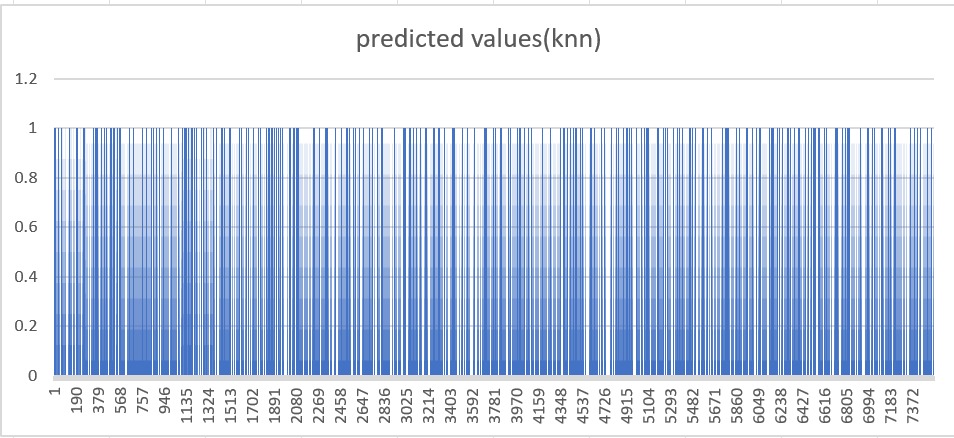
Logistic regression is a statistical method used to model the relationship between a dependent variable (often denoted as 'y') and one or more independent variables (often denoted as 'x').Here the regression model is drawn between predicted and true values of the target variable.As a result it shows that predicted and true values have less deviation between them

****

**Fig(3.1)**

1. **KNN**

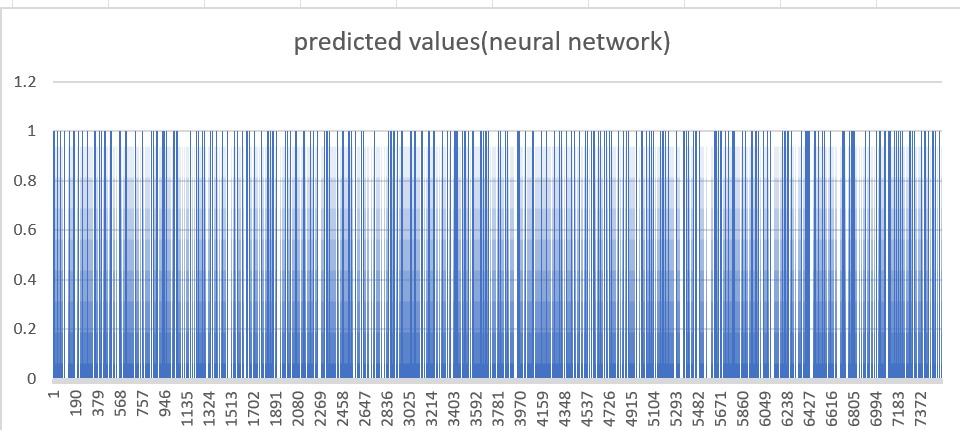
KNN is a non-parametric and lazy learning algorithm used for classification and regression tasks. Unlike linear regression, KNN does not create a predefined model but instead makes predictions based on the similarity of data points in the feature space. Here the KNN model shows analysis between predicted and true values of the target variable.As a result it shows that predicted and true values have less deviation between them

****

**Fig(3.2)**

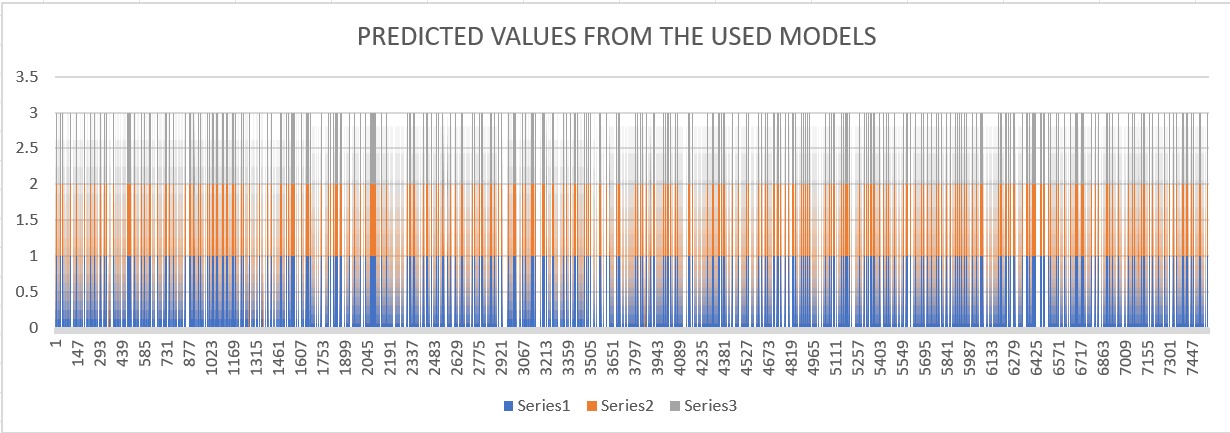
1. **ANN**

In an Artificial Neural Network (ANN) graph, the relationship between predicted and actual values can provide insights into the performance of the neural network on a specific task,. The below graphical result shows that predicted and true values are almost indepenedent.

****

**Fig(3.3)**

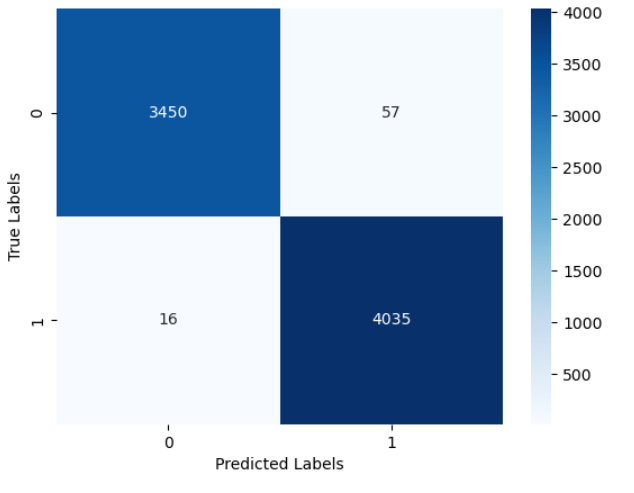
1. **ANALYSIS OF ALL THREE ALGORITHM**
2. **Area Graph for all the three models used**

****

**Fig(3.4)**

1. **CONFUSION MATRIX**

A confusion matrix is a table used to evaluate the performance of a classification model by comparing its predictions with the actual labels of a dataset. It provides a detailed breakdown of the correct and incorrect predictions made by the model for each class in the classification problem.For our approach, the confusion matrix turn out to be as follows(Fig).Since the anti diagonal elements(Not true values) are the least, our model perfectly fits with the dataset.

****

**Fig(3.4)**

1. **ACCURACY OF THE MODELS**

The below table shows analysis of all the three models based on precision, recall and F1 score.

* Precision measures the proportion of true positive predictions out of all positive predictions made by the model.
* Recall measures the proportion of true positive predictions out of all actual positive instances in the dataset.
* The F1-score is the harmonic mean of precision and recall. It provides a balance between precision and recall and is especially useful when there is an imbalance between the positive and negative classes. The F1-score considers both false positives and false negatives and penalizes the model for a large number of either type of error.

Our analysis , by training the model under different ML and DL algorithms turned out to be 97%,98% and 99% accurate on the dataset..

****

Table(1) Parameters for checking overall accuracy

1. **FUTURE WORK**

While our research has demonstrated promising results in the domain of intrusion detection using ML and DL, several avenues for future exploration and improvement remain open. We outline some potential directions for future work:

Incorporating real-time and dynamic updates: IDS models must be continuously monitored and adjusted as attack pathways and intrusion strategies change quickly. Future research can concentrate on creating methods that allow ML and DL models to be updated in real-time to successfully counter new threats.

Investigating hybrid approaches: Hybrid models that combine the advantages of ML and DL algorithms with rule-based techniques or expert systems should be looked at. Investigating the fusion of various methodologies might result in more reliable and complete intrusion detection systems.Real-time and dynamic updates are included: Rapid evolution of intrusion strategies and attack routes necessitates ongoing monitoring and IDS model change. Future research can concentrate on creating methods that allow ML and DL models to be updated in real-time to successfully counter new threats.

Enhancing explainability and interpretability: ML and DL models are frequently referred to as "black boxes," which makes it difficult to comprehend how they make decisions. Future research should focus on creating methods that allow intrusion detection systems to be explicable and interpretable, allowing security analysts to comprehend the reasoning behind reported incursions.

Adversarial attack detection: Attacker evasion or system manipulation may be tried as sophisticated intrusion techniques which are used to detect attacks. Designing ML and DL models that are resistant to adversarial attacks and capable of spotting and thwarting such attempts could be the main goal of future research.

Scalability and resource limitations: ML and DL models can be resource- and cost-intensive to compute. Future research should examine methods for improving intrusion detection systems' effectiveness and performance, especially in contexts with limited resources.

By addressing these areas of future research, we can further advance the field of intrusion detection using ML and DL, ultimately leading to more robust and effective security solutions in the face of evolving cyber threats.

1. **CONCLUSION**

This research paper has explored the application of machine learning (ML) and deep learning (DL) techniques, including logistic, K-nearest neighbors (KNN), and artificial neural networks (ANN), in the development of an intrusion detection system (IDS). Our findings demonstrate the effectiveness of these approaches in real-time intrusion detection.

Logistic regression models provide valuable insights into intrusion patterns, allowing us to predict the likelihood or severity of intrusions. KNN algorithms accurately classify network traffic as normal or malicious based on instance similarity. Additionally, deep learning models, specifically ANN architecture achieve superior detection rates and reduced false positives.

The results highlight the effectiveness of logistic regression, KNN, and ANN approaches in intrusion detection. Leveraging these ML and neural networks techniques contributes to accurate and efficient IDS solutions, actively protecting computer networks from evolving cyber threats. Further research can advance these approaches, improving performance and interpretability for more robust intrusion detection systems.

**16) REFERENCES**

[1] Saranya, T., Sridevi, S., Deisy, C., Chung, T. D., & Khan, M. A. (2020). Performance analysis of machine learning algorithms in intrusion detection system: A review. *Procedia Computer Science*, *171*, 1251-1260.

[2] Sultana, N., Chilamkurti, N., Peng, W., & Alhadad, R. (2019). Survey on SDN based network intrusion detection system using machine learning approaches. *Peer-to-Peer Networking and Applications*, *12*, 493-501.

[3] Musa, U. S., Chhabra, M., Ali, A., & Kaur, M. (2020, September). Intrusion detection system using machine learning techniques: A review. In *2020 international conference on smart electronics and communication (ICOSEC)* (pp. 149-155). IEEE.

[4] Abrar, I., Ayub, Z., Masoodi, F., & Bamhdi, A. M. (2020, September). A machine learning approach for intrusion detection system on NSL-KDD dataset. In *2020 international conference on smart electronics and communication (ICOSEC)* (pp. 919-924). IEEE.

[5] Venkatesan, S. (2023). Design an Intrusion Detection System based on Feature Selection Using ML Algorithms. *Mathematical Statistician and Engineering Applications*, *72*(1), 702-710.

[6] Sarhan, M., Layeghy, S., & Portmann, M. (2022). Towards a standard feature set for network intrusion detection system datasets. *Mobile networks and applications*, 1-14.

[7] Gad, A. R., Nashat, A. A., & Barkat, T. M. (2021). Intrusion detection system using machine learning for vehicular ad hoc networks based on ToN-IoT dataset. *IEEE Access*, *9*, 142206-142217.

[8] Mane, S., & Rao, D. (2021). Explaining network intrusion detection system using explainable AI framework. *arXiv preprint arXiv:2103.07110*.

[9] Apruzzese, G., Pajola, L., & Conti, M. (2022). The cross-evaluation of machine learning-based network intrusion detection systems. *IEEE Transactions on Network and Service Management*, *19*(4), 5152-5169.

[10] Mahfouz, A. M., Venugopal, D., & Shiva, S. G. (2020). Comparative analysis of ML classifiers for network intrusion detection. In *Fourth International Congress on Information and Communication Technology: ICICT 2019, London, Volume 2* (pp. 193-207). Springer Singapore.

[11] Zamani, M., & Movahedi, M. (2013). Machine learning techniques for intrusion detection. *arXiv preprint arXiv:1312.2177*.

[12] Sarhan, M., Layeghy, S., Moustafa, N., & Portmann, M. (2021). Netflow datasets for machine learning-based network intrusion detection systems. In *Big Data Technologies and Applications: 10th EAI International Conference, BDTA 2020, and 13th EAI International Conference on Wireless Internet, WiCON 2020, Virtual Event, December 11, 2020, Proceedings 10* (pp. 117-135). Springer International Publishing.

[13] Magán-Carrión, R., Urda, D., Díaz-Cano, I., & Dorronsoro, B. (2020). Towards a reliable comparison and evaluation of network intrusion detection systems based on machine learning approaches. *Applied Sciences*, *10*(5), 1775.

[14] Thapa, N., Liu, Z., Kc, D. B., Gokaraju, B., & Roy, K. (2020). Comparison of machine learning and deep learning models for network intrusion detection systems. *Future Internet*, *12*(10), 167.

[15] Chua, T. H., & Salam, I. (2022). Evaluation of machine learning algorithms in network-based intrusion detection system. *arXiv preprint arXiv:2203.05232*.

[16] Azizan, A. H., Mostafa, S. A., Mustapha, A., Foozy, C. F. M., Wahab, M. H. A., Mohammed, M. A., & Khalaf, B. A. (2021). A machine learning approach for improving the performance of network intrusion detection systems. *Annals of Emerging Technologies in Computing (AETiC)*, *5*(5), 201-208.

[17] Jasim, Y. A. (2018). Improving intrusion detection systems using artificial neural networks. *ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal*, *7*(1), 49-65.

[18] Mahmood, Z., Agrawal, C., Hasan, S. S., & Zenab, S. (2012). Intrusion detection in cloud computing environment using neural network. *International Journal of Research in Computer Engineering and Electronics*, *1*(1), 1-4.

[19] Omer, K. A. A., & Awn, F. A. (2015). Performance evaluation of intrusion detection systems using ANN. *Egyptian Computer Science Journal*, *39*(4), 32-42.

[20] Dias, L. P., Cerqueira, J. D. J. F., Assis, K. D., & Almeida, R. C. (2017, September). Using artificial neural network in intrusion detection systems to computer networks. In *2017 9th Computer Science and Electronic Engineering (CEEC)* (pp. 145-150). IEEE.