Webology (ISSN: 1735-188X)
Volume 18, Number 2, 2021

Gumbel-Softmax: A Cyber-Attack Classification Mechanism Based On Variational Autoencoder

Ms. Bhawana Parihar[1] , Ms. Varsha Sharma[2] , Dr. Atul Kumar Rai[3]

[1]Asst. Prof. CSE Department, B T Kumaon Institute of Technology, Dwarahat (Almora) Uttrakhand, INDIA.

[2]Asst. Prof. CSE Department, Kothiwal Institute of Technology and Professional Studies, Moradabad, UP, INDIA.

[3]Asst. Prof. CSE Department, Kothiwal Institute of Technology and Professional Studies, Moradabad, UP, INDIA.

Abstract: The utilization of computer networks has significantly risen, accompanied by a proliferation of applications operating on these networks. Consequently, the significance of network security has escalated, given that every system possesses security vulnerabilities that could potentially result in heightened cyberattacks adversely affecting the economy. It is now imperative to accurately identify system and network weaknesses in real-time. In order to serve as an overview for the subsequent growth of intrusion detection systems, this article compares the efficacy of artificial intelligence techniques used in IDS, such as train a model using variational autoencoder Gumbel SoftMax. The NSL-KDD dataset is utilized to train the model. The outcomes show that our suggested method performs better than existing ML algorithms. The accuracy, precision, recall, and f1score of the proposed model are higher than those of existing machine learning algorithms, including SVM linear, SVMrbf, Adaboost, GaussianNB, and Logistic Regression. In our research, we discovered that the gaussianNB approach we proposed—which combines one hot encoding technique with normalization—performs better than previous machine learning models. We achieved 92.9% accuracy, 98.8% pressure, and 92.3 recall.

Keywords: Artificial Intelligence; Anomaly-based Method; Host-Based IDS; Network-Based IDS; NSL-KDD; Signature-based Method.

Introduction: In today's society, Intrusion Detection Systems (IDS) play a crucial role in protecting digital environments. Due to the complexity of cyber threats and the increased dependence on technology, IDS is essential in proactively identifying and mitigating possible security breaches [1]. IDS employs anomaly detection in addition to signature-based approaches to detect established attack patterns and identify emergent threats and odd activity, offering a layered protection against a variety of assaults. IDS acts as a vigilant guardian, assisting to maintain the integrity, confidentiality, and availability of digital assets, making it an essential component of contemporary cybersecurity strategies in a landscape where data breaches, ransomware, and other malicious activities pose significant risks to both

organizations and individuals[2]. IDS is a security system designed to detect and alert administrators of unauthorized access or malicious activity on a computer network. These systems monitor network traffic and use algorithms and signatures to identify potential threats, for example, if a hacker attempts to access a network using a known malicious IP address. The IDs will detect the attempt and send an alert to the network administrator, allowing them to take appropriate action to prevent the attack. The IDs provides real time monitoring and alerts, allowing organizations to quickly respond to potential threats and minimize the risk of a data breach[3].

Securing both corporations and consumers is crucial, given the significant trust placed in them. While automated security systems have been developed, none have proven as effective as Intrusion Detection Systems (IDS), also known as ideas platforms. An IDS is an application or device that continually monitors network traffic, analyzing patterns and alerting administrators to unusual behavior [4]. If malicious content is detected, it notifies the security team for investigation and remediation. To avoid impacting performance, IDS solutions often use techniques like switched port analyzers or access codes to analyze a copy of the data traffic. Unlike Intrusion Prevention Systems (IPS) that block threats, IDS identifies attack patterns with network packets, monitors user behavior, and ensures compliance with security policies, acting proactively to detect anomalies before hackers achieve their objectives. However, an IDS must be tailored to an organization's specific needs and context, necessitating the expertise of a trained analyst [5]. There are two main approaches to IDS: passive and reactive. In a passive system, IDS detects potential threats, logs information, and alerts administrators [6]. In a reactive system, it responds to suspicious activity by taking actions like logging off users or reconfiguring firewalls to block malicious sources. There are various types of intruders that IDS must be aware of, including external hackers attempting to breach systems, unauthorized individuals exploiting user privacy (MASQUERADE), and insiders aiming to weaken security defenses or aid others. IDS platforms primarily use two methods for intrusion detection: signature-based and anomaly-based. Signature-based IDS compares network traffic and log data to existing attack patterns, while anomaly-based IDS identifies deviations from normal behavior. IDS deployment tactics include network-based IDS, host-based IDS, and cloud-based IDS. Network-based IDS uses sensors at strategic points within the network, host-based IDS deploys agents on servers and endpoints, and cloud-based IDS is optimized for cloud environments.

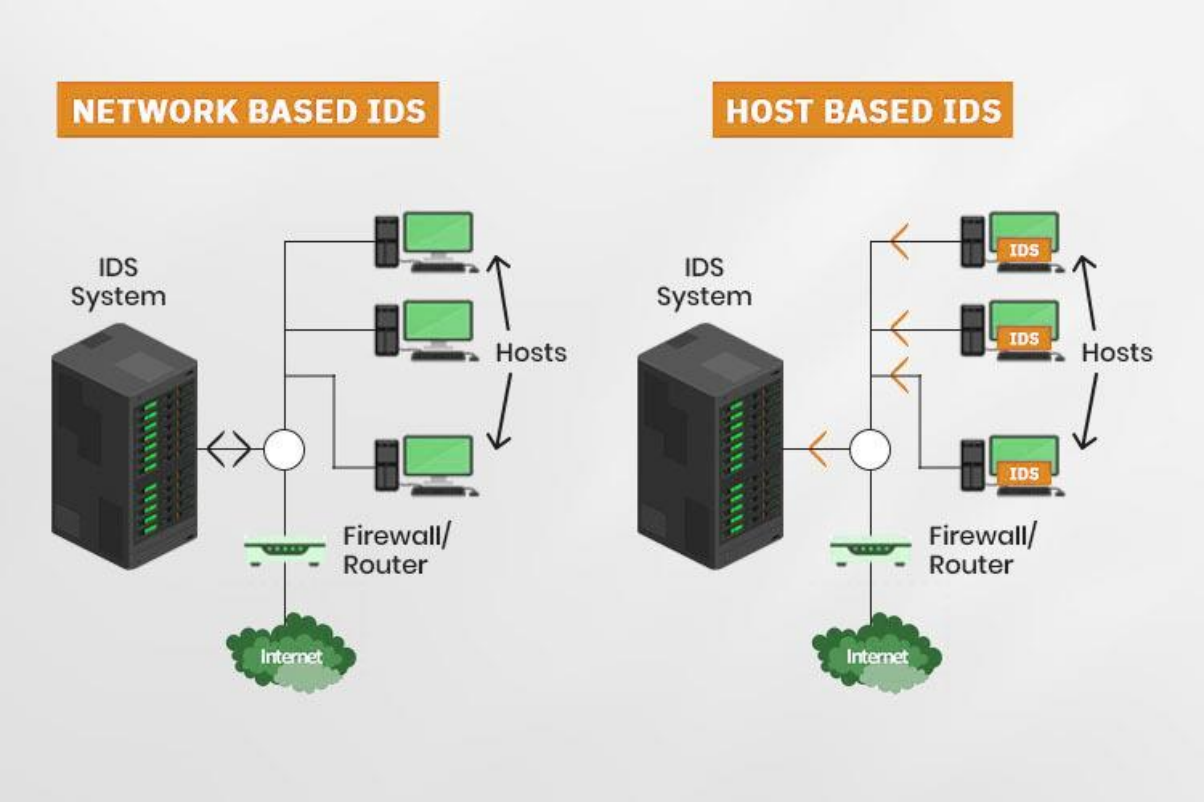


Figure 1: work flow of different types of IDS

Related Work: In [7], authors have put their efforts on offering detailed instructions on methodology selection and use for the training of machine and deep learning models. Two data sets are used as input, and five MDL models are assessed. main objectives are to reduce the percentage of undetected attacks, the number of false alarms, and the duration of the testing process as a whole. Based on this configuration, the proposed system is able to forecast both known and unknown computer network threats in close to real time. In [8] this study, authors have suggested a unique 5-layer autoencoder (AE)-based model that is better suitable for applications involving network anomaly detection and based on the findings of a thorough and meticulous examination of a number of performance metrics utilized in an AE model. In order to mitigate model bias brought on by data imbalance across various data types in the feature set, we employ a novel data pre-processing mechanism in our suggested model that transforms and eliminates the most detrimental outliers from the input samples. In order for the model to determine if a sample of network traffic is normal or abnormal. In [9] this study, authors have suggested BLSTM (Bidirectional Long Short-Term Memory) and attention mechanisms are combined in the BAT model. The network flow vector, which is made up of packet vectors produced by the BLSTM model and may extract the essential properties for classifying network traffic, is screened using an attention mechanism. We also use many convolutional layers to capture the regional characteristics of the traffic data. Due to the usage of numerous convolutional layers while processing input samples. In [10] this study, machine learning (ML) classification algorithms such as support vector machines (SVM), K-nearest neighbors (KNN), logistic regression (LR), Naive Bayes (NB), multi-layer perceptrons (MLP), random forests (RF), extra-tree classifiers (ETC), and decision trees (DT) were used to categorize data as

normal or intrusive. Four feature subsets taken from the NSLKDD dataset were used to assess the model performance. In [11] this study, authors have suggested deep learning-based solution for intrusion detection that can be used to address the issue in part. The suggested approach made use of autoencoder, a well-known deep learning tool. Deep autoencoder's encoder was used to compress the less crucial characteristics and extract the crucial information without the need for a decoder. In [12] this study, authors have suggested NSL-KDD features that have the greatest impact on the detection outcome is one of the goals of this effort. We will thus ingest the unsettling component of the dataset. We used the Condensed Nearest Neighbors (CNN) algorithm as our initial strategy to create our Network IDS (NIDS). Given that it takes sample distribution into account, this approach is particularly good for classification and regression.

Proposed Model: We have primary focus was on Gumbel-Softmax estimator technique using Variational AutoEncoder for classification of NSL-KDD, This study specifically the classification algorithms can identify irregularities in network traffic patterns. The researchers prioritize preprocessing and normalization due to its significance, especially when dealing with the challenges posed by processing datasets.

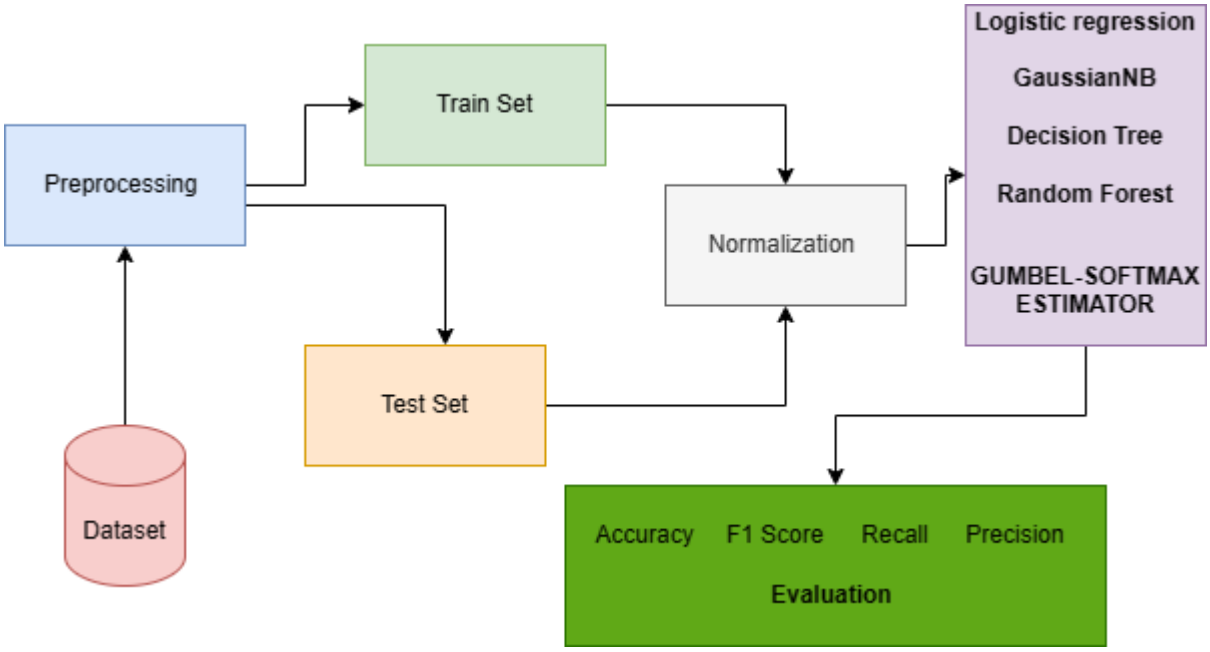


Figure 2: Work flow of proposed model.

Dataset: In the field of intrusion detection, New Selected Learning-Knowledge Discovery in Databases Data set (NSL-KDD stands out as the fundamental dataset. It was developed using the KDD99 dataset and has been improved to accommodate several levels of difficulty in its test set. Due to several restrictions, it might not perfectly reflect real-world network circumstances, but it is nevertheless a useful benchmark dataset for academics trying to compare different intrusion detection strategies. Table 1 and figure 2 provides the detail distribution of NSL-KDD dataset.

Table 1: Details description Network Traffic of NSL-KDD dataset

Type	Train	Test	Imbalance ratio	Total
DoS	45927	7458	1.44	53385
Probe	11656	2421	5.47	14077
R2L	955	2421	19.85	3882
U2R	52	67	647.51	199
Normal	67343	9711	-	77054

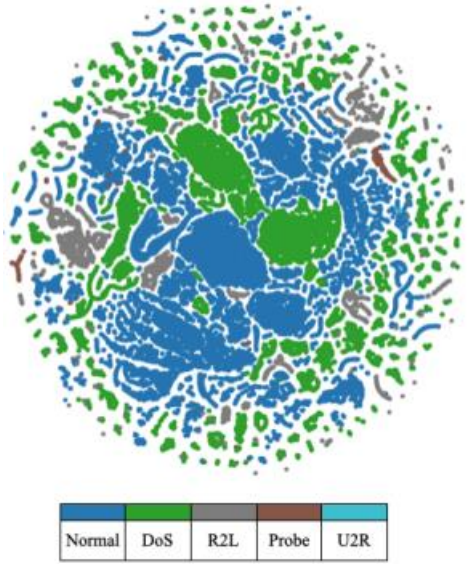


Figure 2: Details distribution Imbalanced Network Traffic of NSL-KDD dataset [13]

THE GUMBEL-SOFTMAX DISTRIBUTION

In order to start, let's define the Gumbel-Softmax distribution, which is a continuous distribution over the simplex that may roughly approximation samples from a categorical distribution. A categorical variable with class probabilities of 1, 2,..., k is called z. In the remaining sections of this study, we assume that categorical samples are represented as k-dimensional one-hot vectors positioned at the corners of the (k 1)-dimensional simplex, k1. The element-wise mean Ep[z] = [1,..., k] of these vectors may be defined as a result. Using the Gumbel-Max technique (Gumbel, 1954; Maddison et al., 2014), it is easy and quick to choose samples z from a categorical distribution with class probabilities:

$$z = \text{one\_hot}(\underbrace{\arg \_\max}_{i} [gi + \log\pi_i) \qquad (1)$$

where the samples g1–gk are samples taken from Gumbel(0, 1)1. In order to approximate arg max continuously and differently, we utilize the softmax function to produce k-dimensional sample vectors y k 1, where

$$y_i = \frac{\exp((\log(\pi_i)+g_i/\tau}{\sum_{j=1}^{k} \exp((\log(\pi_i)+g_i/\tau} \qquad \text{for } i = 1,2, \dots \dots k \qquad (2)$$

The Gumbel-Softmax distribution has the following density:

$$p_\pi(y_1, y_{2\ldots\ldots}, y_k) = \Gamma(k)\tau^{k-1}\left(\sum_{i=1}^{k} \pi_i / y_i^{\tau}\right)^{-k} \prod_{i=1}^{k} \pi_i / y_i^{\tau+1} \quad (3)$$

The concrete distribution is the name given to this distribution, which was independently found by Maddison et al. (2016). The Gumbel-Softmax distribution and the categorical distribution p(z) become identical when the softmax temperature approaches 0 and samples from the distribution become one-hot.

GUMBEL-SOFTMAX ESTIMATOR

Due to the smoothness of the Gumbel-Softmax distribution for > 0, it has a well-defined gradient y/ with respect to the parameters. Backpropagation may therefore be used to generate gradients by swapping categorical samples for Gumbel-Softmax samples. The Gumbel-Softmax estimator is the term used to describe the process of substituting non-differentiable categorical data with a differentiable approximation during training. Although Gumbel-Softmax ones may be distinguished from instances from the associated categorical distribution at non-zero, they are not the same. There is a trade-off for learning, where samples are smooth but the gradient variation is small, and small value, where samples are close to high but the gradient variance is huge. According to our tests, the softmax value may be annealed in a number of ways while still performing effectively. This method can be understood as entropy regularization if is a learned parameter (rather than annealed according to a fixed schedule). In this case, the Gumbel-Softmax distribution can adaptively change the "confidence" of suggested samples throughout the training process. Simple feedforward neural networks, several structures that autoencoders may adopt. Variants include stacked autoencoders, denoising autoencoders, and variational autoencoders (VAEs) bring further methods and capabilities to enhance performance or produce new data samples. There are several fields where unsupervised learning and data representation are essential, including image and audio processing, recommendation systems, and many more.

Logistic regression: The Classification problems, which entail predicting one of two potential outcomes based on input data, are commonly handled by the statistical modeling approach known as logistic regression. Logistic regression is intended to predict a binary result, often expressed as 0 or 1, true or false, yes or no, etc. as opposed to linear regression, which predicts a continuous numerical value. The procedure used in logistic regression models the connection between a dependent variable (the binary result) and one or more independent variables (features or predictors). In order to convert a linear combination of the input characteristics into a probability score that lies between 0 and 1, it employs the logistic function, commonly known as the sigmoid function. The possibility that the binary event will occur is represented by this probability score (1).

GaussianNB: When working with continuous data, like as measurements, sensor readings, or any other data that may be presumed to follow a Gaussian distribution, GaussianNB is very helpful. However, it might not work effectively when dealing with high-dimensional data or datasets where the premise of feature independence is violated. It's important to keep in mind that while the GaussianNB approach is quick and easy to use, it might not always yield the most precise results when compared to deeper neural networks, support vector machines, or

random forests. The unique attributes of the dataset and the objectives of the machine learning assignment should be taken into consideration while choosing a model.

Decision Tree: A decision tree is made up of nodes, which stand in for decisions or options, and edges, which represent potential outcomes. Internal nodes and leaf nodes grow out from the root node at the beginning. Leaf nodes contain the ultimate forecasts or values, whereas internal nodes include conditions or queries. Decision trees provide a number of benefits, including their simplicity, readability, and capacity for both category and numerical characteristics. They can, however, be prone to overfitting, particularly if the tree is deep and collects data noise.

Random Forest: A potent ensemble machine learning technique called Random Forest is employed for both classification and regression problems. Because of its excellent predicted accuracy and resistance to overfitting, it is a decision tree algorithm extension that is well-known. However, the number of trees in the ensemble (a hyperparameter) needs to be carefully controlled to attain the optimal performance because Random Forests might not be as interpretable as a single decision tree. Despite being a reliable and strong technique, Random Forests may not always be the ideal option for very big datasets or when computing efficiency is a top priority.

Result ANALYSIS:

This study's objective is to assess the gumbel-softmax estimator's performance in relation to the NSL-KDD dataset. Identification and detection of attack-related behaviors. Given that IDS require a high detection rate and a low false alarm rate, we assess accuracy, precision, recall, and f1score and present the comparative findings for a range of attacks.

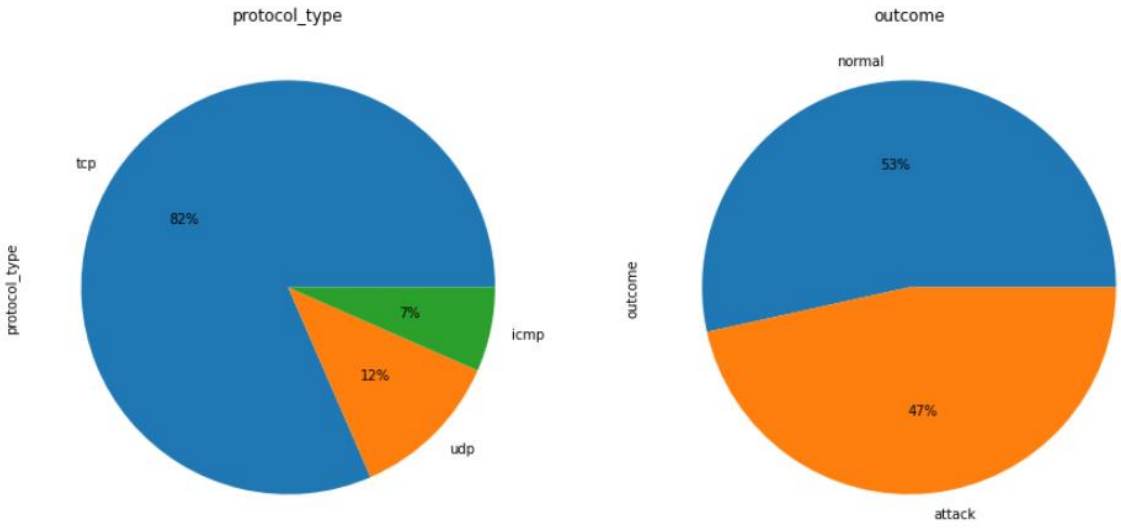


Figure 3: Details distribution Imbalanced Network Traffic of NSL-KDD dataset [13]

Accuracy: The Accuracy [14] measures how accurately a model predicts or categorizes data items. Although accuracy is a simple and obvious statistic, it may not always be the ideal option, especially when working with datasets that are unbalanced. It is calculated using equation (4) as follows:

$$\mathrm{ACC} = \frac{\mathrm{TP+TN}}{\mathrm{TP+TN+FP+FN}} \quad (4)$$

F1 Score (F1): When dealing with unbalanced datasets or when both precision and recall are crucial, the F1 Score is a statistic frequently used in machine learning, notably in classification tasks, to evaluate a model's performance. It balances these two measurements and is the harmonic mean of precision and recall. It is calculated using equation (5) as follows:

$$\mathrm{F1} = \frac{2\mathrm{TP}}{2\mathrm{TP+FP+FN}} \quad (5)$$

Recall: The recall Recall, sometimes referred to as sensitivity or the true positive rate, indicates how well the model finds all positive situations while not missing too many. It is calculated as the ratio of true positives to all real positives. It is calculated using equation (6) as follows:

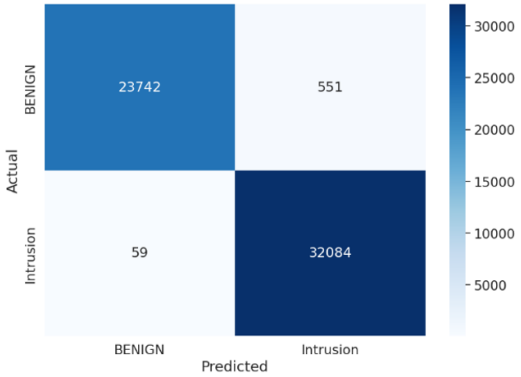
$$\mathrm{TPR} = \mathrm{TP}/\mathrm{TP} + \mathrm{FN} \quad (6)$$

Precision: Precision evaluates how effectively the model properly detects positive situations without producing too many false positive predictions. It is calculated as the ratio of true positives to the total number of anticipated positives. It is calculated using equation (6) as follows:

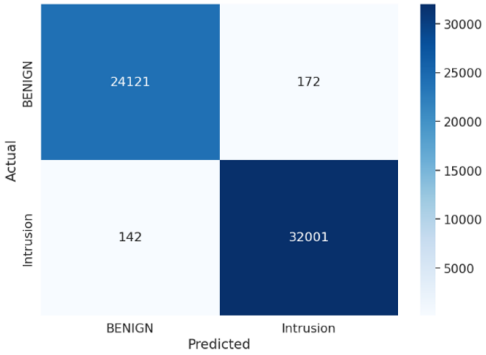
- $\mathrm{PPV} = {}^{\mathrm{TP}}/_{\mathrm{TP} + \mathrm{FP}}$ (7)

Here, TP signifies true positives, TN signifies true negatives, FP signifies false positives, and FN signifies false negatives.

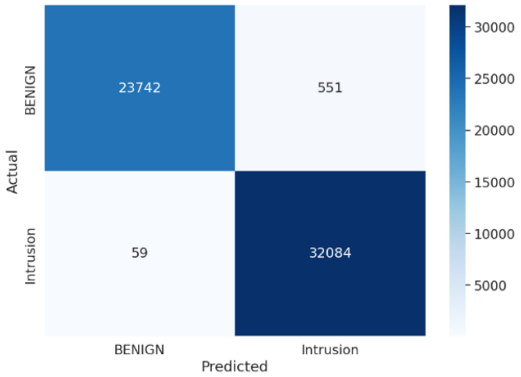
The suggested model outperforms the tested machine learning models in terms of Sensitivity (0.9961), Precision (0.9949), Accuracy (0.9961), and F1 Score (0.9955), demonstrating outstanding performance across key criteria. Decision Tree, Logistic Regression, and Gaussian Naive Bayes models all perform admirably, with Sensitivities of 0.9975, 0.9941, and 0.9975, respectively. Random Forest fared well in terms of Sensitivity (0.9986) and Accuracy (0.9931) while achieving the greatest Precision (0.9853). In light of the trade-offs between sensitivity, precision, and overall model correctness, these results indicate that the suggested model is the best appropriate for the job at hand. The confusion matrix for several machine learning models is displayed in Figure 4. The outcome analysis for several models is presented in Table 2. The study of the results from various models is shown in Figure 5.



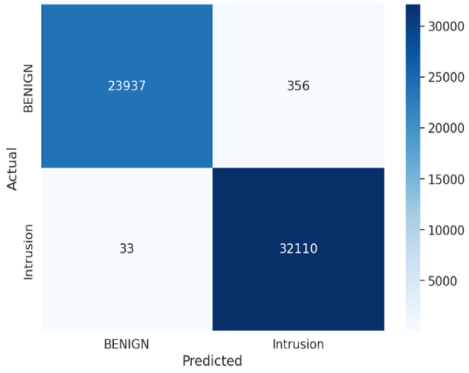
Logistic regression



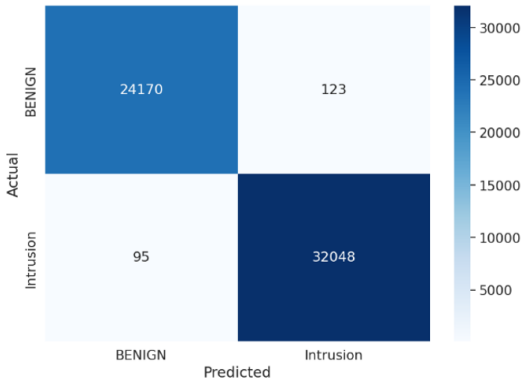
Gaussian NB



Decision Tree



Random Forest



Proposed Model

Figure 4: confusion matrix of different machine learning models

Measure	Logistic regression	GaussianNB	Decision Tree	Random Forest	Proposed Model	[13]
Sensitivity	0.9975	0.9941	0.9975	0.9986	0.9961	0.9697

Measure	Logistic regression	GaussianNB	Decision Tree	Random Forest	Proposed Model	[13]
Precision	0.9773	0.9929	0.9773	0.9853	0.9949	0.9746
Accuracy	0.9892	0.9944	0.9892	0.9931	0.9961	0.9699
F1 Score	0.9873	0.9935	0.9873	0.9919	0.9955	0.9704

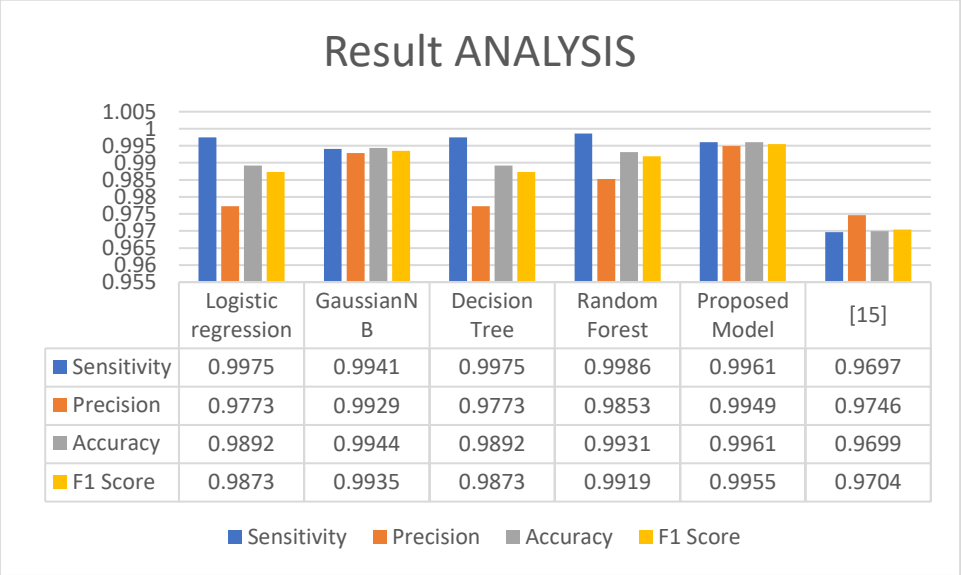


Figure 5: The study of the results from various models

Conclusion: This research suggests a unique method for intrusion detection in network security that combines statistical analysis and deep learning approaches. The model shows notable development in identifying intrusions in business and industrial networks. Utilizing traditional measuring tools, the efficacy of the suggested IDS was assessed. Highly linked characteristics were extracted using statistical analysis and put into deep learning models like AE and LSTM as well as conventional machine learning methods. The tests took into account binary situations and were performed on datasets: NSL KDD, with 99% accuracy achieved 98% accuracy reached on the NSL KDD dataset with the LSTM classifier, the findings demonstrated excellent accuracy.

References:

[1] H. Chae, B. Jo, S.-H. Choi, and T. Park, "Feature selection for intrusion detection using NSL-KDD," Recent advances in computer science, vol. 20132, pp. 184–187, 2013.

[2] B. Ingre and A. Yadav, “Performance analysis of NSL-KDD dataset using ANN,” in 2015 international conference on signal processing and communication engineering systems, 2015, pp. 92–96.

[3] L. M. Ibrahim, D. T. Basheer, and M. S. Mahmod, “A comparison study for intrusion database (Kdd99, Nsl-Kdd) based on self organization map (SOM) artificial neural network,” Journal of Engineering Science and Technology, vol. 8, no. 1, pp. 107–119, 2013.

[4] L. Dhanabal and S. P. Shantharajah, “A study on NSL-KDD dataset for intrusion detection system based on classification algorithms,” International journal of advanced research in computer and communication engineering, vol. 4, no. 6, pp. 446–452, 2015.

[5] S. Revathi and A. Malathi, “A detailed analysis on NSL-KDD dataset using various machine learning techniques for intrusion detection,” International Journal of Engineering Research \& Technology (IJERT), vol. 2, no. 12, pp. 1848–1853, 2013.

[6] S. Lakhina, S. Joseph, and B. Verma, “Feature reduction using principal component analysis for effective anomaly--based intrusion detection on NSL-KDD,” 2010.

[7] B. M. Serinelli, A. Collen, and N. A. Nijdam, “Training Guidance with KDD Cup 1999 and NSL-KDD Data Sets of ANIDINR: Anomaly-Based Network Intrusion Detection System,” Procedia Comput Sci, vol. 175, pp. 560–565, 2020, doi: https://doi.org/10.1016/j.procs.2020.07.080.

[8] A. R. Wani, Q. P. Rana, U. Saxena, and N. Pandey, “Analysis and detection of DDoS attacks on cloud computing environment using machine learning techniques,” in 2019 Amity International conference on artificial intelligence (AICAI), 2019, pp. 870–875.

[9] T. Su, H. Sun, J. Zhu, S. Wang, and Y. Li, “BAT: Deep Learning Methods on Network Intrusion Detection Using NSL-KDD Dataset,” IEEE Access, vol. 8, pp. 29575–29585, 2020, doi: 10.1109/ACCESS.2020.2972627.

[10] I. Abrar, Z. Ayub, F. Masoodi, and A. M. Bamhdi, “A Machine Learning Approach for Intrusion Detection System on NSL-KDD Dataset,” in 2020 International Conference on Smart Electronics and Communication (ICOSEC), 2020, pp. 919–924. doi: 10.1109/ICOSEC49089.2020.9215232.

[11] C. Zhang, F. Ruan, L. Yin, X. Chen, L. Zhai, and F. Liu, “A Deep Learning Approach for Network Intrusion Detection Based on NSL-KDD Dataset,” in 2019 IEEE 13th International Conference on Anti-counterfeiting, Security, and Identification (ASID), 2019, pp. 41–45. doi: 10.1109/ICASID.2019.8925239.

[12] F. Z. Belgrana, N. Benamrane, M. A. Hamaida, A. M. Chaabani, and A. Taleb-Ahmed, “Network Intrusion Detection System Using Neural Network and Condensed Nearest Neighbors with Selection of NSL-KDD Influencing Features,” in 2020 IEEE International Conference on Internet of Things and Intelligence System (IoTaIS), 2021, pp. 23–29. doi: 10.1109/IoTaIS50849.2021.9359689.

[13] L. Liu, P. Wang, J. Lin, and L. Liu, “Intrusion Detection of Imbalanced Network Traffic Based on Machine Learning and Deep Learning,” IEEE Access, vol. 9, pp. 7550–7563, 2021, doi: 10.1109/ACCESS.2020.3048198.

[14] P. K. Mall, P. K. Singh, and D. Yadav, “GLCM based feature extraction and medical X-RAY image classification using machine learning techniques,” in 2019 IEEE Conference on Information and Communication Technology, 2019, pp. 1–6.