Machine Learning for Human Intelligence
2024, Vol. 2, Issue 2

Research Article

Machine Learning Enhanced Network Intrusion Detection System

Muhammad Yasir Iqbal[1,⋆]

[1]School of Mathematics Statistics and Mechanics, Beijing University of Technology, Beijing, China.; Email: yasirbasra190@gmail.com
⋆Corresponding author: Muhammad Yasir Iqbal (yasirbasra190@gmail.com)

Article History

Academic Editor:
Dr. Muhammad Nabeel Asghar
Submitted: June 21, 2024
Revised: August 19, 2024
Accepted: September 01, 2024

Keywords:
Network Intrusion Detection, artificial intelligence, 10-fold and JackKnife validation techniques, MLP classification algorithm

Abstract

Contemporary cybersecurity demands have elevated network protection to a fundamental requirement for any computational system. Safeguarding networks from unauthorized infiltration is essential for maintaining seamless operational continuity in advanced network infrastructures. Network protection has emerged as a dominant concern within the information technology domain. Cybercriminals and malicious actors execute countless successful penetration attempts against network systems. An Intrusion Detection System serves as a cornerstone in network defense, identifying and recognizing irregularities within network security frameworks. IDS effectiveness can be evaluated through its intelligence capacity, operational efficiency, and precise identification of both novel and familiar attack patterns. The maximum gain principle provides optimal anomaly detection capabilities. This research presents a machine learning architecture utilizing multilayer perceptron (MLP) classification achieving 99.98% accuracy. The methodology is validated using 10-fold and JackKnife cross-validation techniques. Critical performance indicators including Accuracy, Sensitivity, Specificity, and Matthew's Correlation Coefficient are analyzed to assess system performance. All evaluation metrics achieved peak performance ratios, demonstrating MLP's superiority as a classification approach. The proposed model's accuracy, sensitivity, specificity, and MCC values reached 99.99% when tested on the complete UNSW-NB15 dataset. These findings indicate significant accuracy improvements through various perceptron architectural configurations. Both K-fold and JackKnife methodologies successfully achieved 99.99% accuracy rates.

1 Introduction

In today's interconnected digital landscape, where every entity continuously generates data and maintains constant global connectivity with various devices, technological advancement has transformed modern life dramatically. The exponential growth in internet-based attacks presents significant security challenges, necessitating the development of adaptable, flexible, resilient, and security-focused solutions. The substantial transformation in modern network traffic characteristics [1, 2] creates vulnerabilities and security gaps that compromise existing protection mechanisms. Extensive research has been conducted on anomaly detection within Intrusion Detection Systems. A major challenge in network intrusion detection involves the massive data volumes generated and collected from network

users. As internet connectivity becomes ubiquitous across devices, users demonstrate increasing enthusiasm for connected technologies. User populations utilizing network services and resources expand continuously, generating enormous volumes of modern network data traffic [3, 4], alongside corresponding data growth. Protecting users and clients securely remains essential. Additionally, increased data volumes extend processing times, thereby impacting IDS model performance. Network intrusion can be characterized as a series of activities that compromise confidentiality, integrity, and availability [5, 6] as shown in Figure 1.

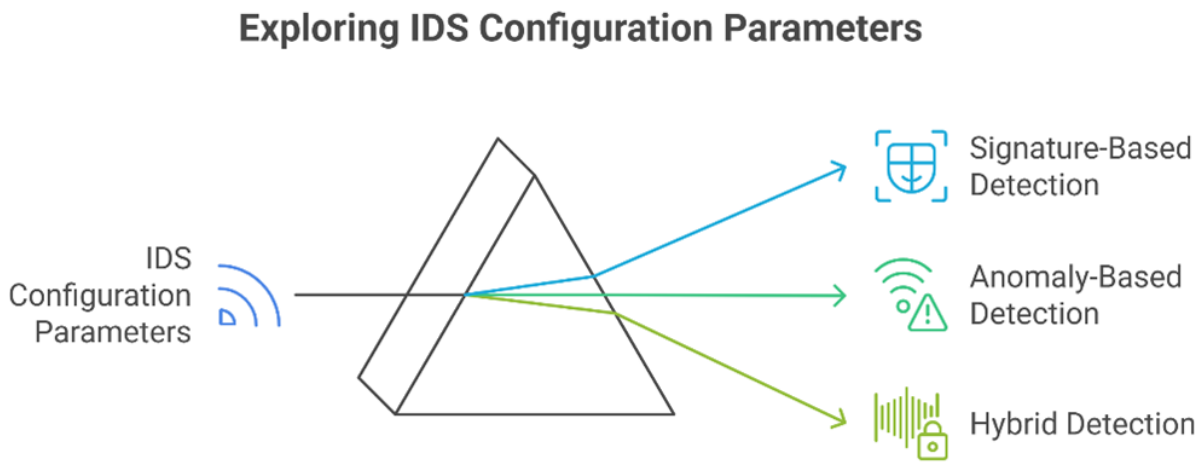

Figure 1: Core IDS Components .

Intrusion Detection encompasses monitoring and analyzing network or computer system events to identify security compromise indicators. Traditional intrusion prevention methods, including access control mechanisms, firewalls, encryption protocols, and password-based security systems have proven insufficient for complete network and system protection against evolving, sophisticated attacks and malicious software [7, 8]. Various datasets have been developed to address network anomalies over time. Each new dataset represents enhanced and redundancy-free improvements. The most recent IDS training and testing dataset is UNSW-NB15, introduced in 2015. It contains approximately 100GB of raw network [5] traffic data with more precise and targeted features and classifications to meet current and future IDS model development requirements. A fundamental objective in machine learning research involves systematically identifying complex data patterns and relationships to generate intelligent outcomes based on specific datasets and historical experience.

IDS systems are categorized into Host-based IDS (HIDS) and Network-based IDS (NIDS) [9, 10]. HIDS operates exclusively on individual host systems using signature databases. HIDS efficiency improves through signature database updates. Signatures represent databases containing comprehensive lists of known viruses and anomalies. When these predefined signatures encounter novel signatures, automatic updates do not occur, resulting in system protection failures. Passive IDS systems detect anomalies after system intrusion occurs. Conversely, Network Intrusion Detection Systems (NIDS) operate proactively, discovering and responding to real-time scenarios. Research focuses vary between HIDS and NIDS approaches. According to [11], HIDS closely resembles antivirus programs. Currently, researchers and scientists emphasize NIDS development due to its proactive nature compared to HIDS. Through Intrusion Detection capabilities, systems prevent compromise from network anomalies and attacks. Due to system and model complexity, legacy IDS solutions fail to meet contemporary requirements. Network system attacks occur continuously, numbering in millions daily [12]; therefore, encountering these attacks requires timely IDS system updates and redesigns. Primary IDS functionality involves classification methodologies and techniques implemented through various classification algorithms and approaches. IDS combines hardware components with software essentials. IDS repre-

sents a security component that prevents system compromise from suspicious activities and malware attacks. Anomalies can damage network data traffic and performance. Within intrusion detection frameworks, detection models learn from previously reported attack patterns and identify similar incoming traffic not encountered previously.

2 Related Work

Previous IDS research has employed numerous approaches from various knowledge domains to develop effective IDS solutions, though such approaches typically contain limitations. In [13], the author conducted a comprehensive survey defining Software Defined Network Technology, Deep Learning, and Machine Learning approaches, with deep learning practically implemented in Network Intrusion Detection Systems. They proposed combined ML/DL methodologies simultaneously. [14] introduced a Gradient Descent (GD) neural network-based algorithm. Before algorithm implementation, accuracy rates approached 50% with false positive and false negative rates at 50% respectively. F-measure mean values in GD-based neural networks reached approximately 39%, which improved to 60% using Fractal -based neural networks, demonstrating significance. Multilayer perceptron (MLP) [15] analysis was conducted on two datasets with different characteristics. The primary objective involved comparing flow- based IDS performance using MLP. MLP demonstrated superior performance compared to J48 decision trees [15]. Another proposed algorithm utilized SVM [16], randomly selecting three features that provided promising results with 98.76% accuracy, 0.09% false positive rate, and 1.15% false negative rate, all exceeding normal SVM values. One consideration is that this algorithm doesn't confirm which features contribute to optimal outputs.

A study [17] implemented a two-stage classifier approach: binary classification and multi-class classification for detection speed and accuracy prediction. UDP attack classification performance was suboptimal according to proposed algorithm results. The algorithm initially eliminates noisy or irrelevant features, reducing overall processing time. EM clustering, Logistic Regression, and Naïve Bayes represent ML approaches for distinguishing normal and abnormal behavior. Multiverse Optimization (MVO) algorithm [18] was proposed for ANN training. This model achieved 98.21% and 99.61% De-

Table 1: UNSW-NB15 Feature Categories

Attack Class	**Selected Attributes**
Normal Traffic	1,2,6,10,11,15,18,19,20,21,29,31,34,36,37,47
DoS Attacks	6,10,11,13,14,15,16,17,23,31,36,37,39,40,42,43,44,45,47
Fuzzing Attacks	2,4,6,10,11,14,15,16,28,29,31,36,37,39,40,41,42,45,46,47
Backdoor Attacks	6,10,11,14,15,16,37,41,42,44,45
Exploit Attacks	5,6,10,11,13,14,16,17,19,31,33,36,37,41,42,45,46
Analysis Attacks	6,10,11,12,13,14,15,16,34,35,37
Generic Attacks	6,9,10,11,12,13,15,16,17,18,19,20,23,28,31,33,34,46
Reconnaissance	9,10,14,16,17,19,20,27,28,30,31,34,37,41,42,43,44,45,46,47
Shellcode	4,6,9,10,12,13,14,15,16,17,18,23,37,44,45
Worm Attacks	5,9,10,11,13,14,17,23,37,41,46

tection Accuracy on NSL-KDD and UNSW-NB15 respectively as given in Table 1. [19] demonstrated that hybrid Genetic Algorithm and SVM approaches enable better training data selection, generating improved performance rates, true positive and false positive rates. This framework's performance on KDD datasets was exceptional compared to UNSW-NB15 because it encompasses all legacy features and groups in KDD 99 datasets. Initial data preprocessing employed the proposed Apriori algorithm with central point (CP) [20], demonstrating comparatively improved preprocessing times. The central point (CP) played crucial roles with Apriori algorithm assistance in reducing preprocessing time. Decreased preprocessing time positively affects overall system performance. Naïve Bayes and Logistic Regression techniques were implemented for normal and abnormal attack classification. Machine learning techniques analyzed features using KDD 99 and UNSW-NB15 datasets [12]. ML approaches

eliminated redundant and irrelevant features. After irrelevant feature elimination, processing time decreased while detection rates increased. In [21], the author implemented a two-level defense architecture model named NvCloud IDS (Network and Virtualization). This model analyzes incoming and outgoing network traffic behavior on cloud network servers (CNS) at the first defense level. At the second defense layer, VM traffic analysis separates normal and abnormal traffic as shown in Figure 2.

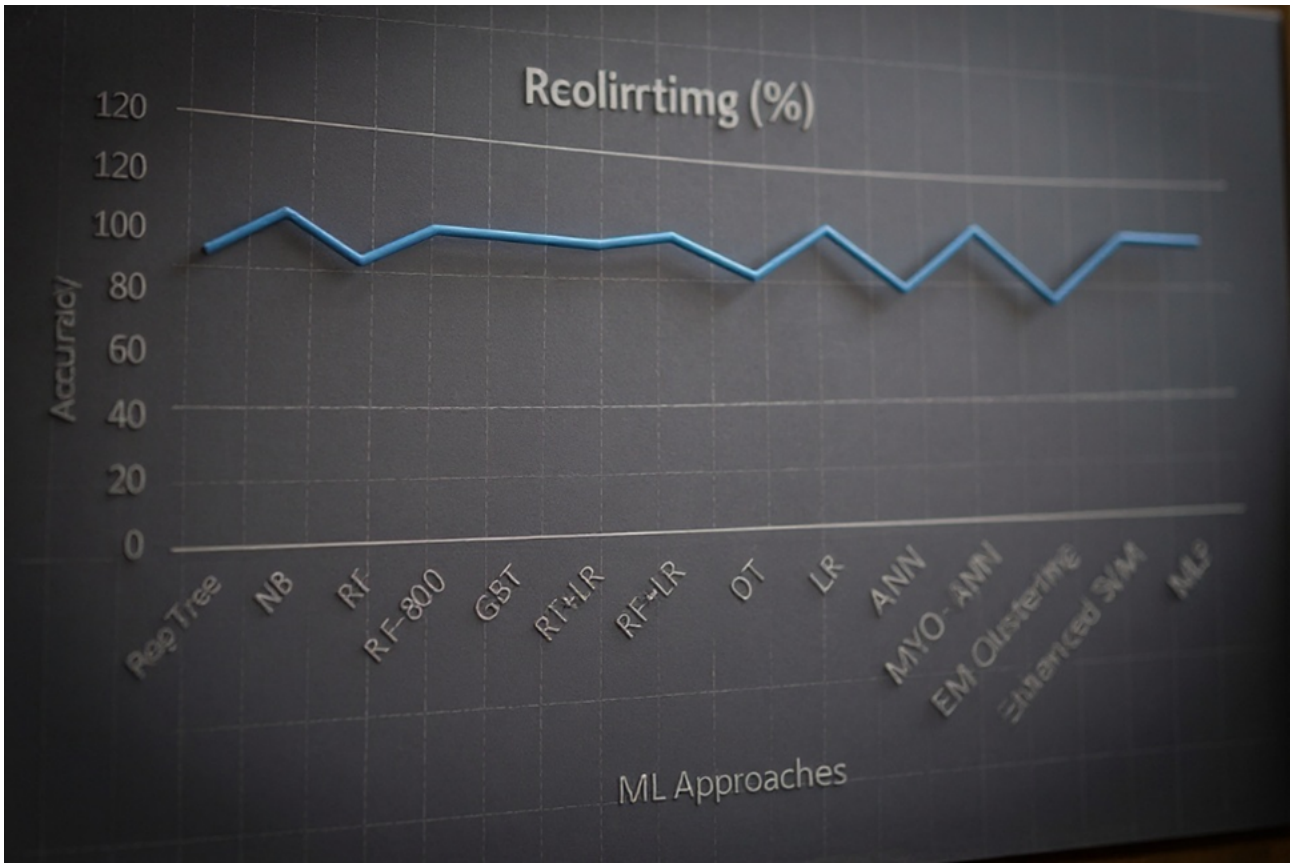

Figure 2: ML Approach Accuracy Comparison.

In [27], authors proposed Multi-SVM IDS development for smart grid models. All SVM models in grid networks operate on different network topologies and algorithms. Security checks are installed at network entrance points as IDS systems. One demonstrated point shows multi-SVM achieves higher detection ratios than Artificial Neural Networks. In [28], authors utilized MLP achieving 97.8% accuracy using Levenberg Marquardt Algorithm, which lacks precision due to insufficient proximity to 100

3 Materials and Methods

In [23], authors proposed Association Rule Mining (ARM) techniques applied to KDD 99 and UNSW-NB15 datasets for normal and abnormal parameter categorization. When numerous similar values exist in records, decision engine algorithms perform poorly. In [24], the dataset is explained, claiming UNSW- NB15 datasets are more complex than KDD 99 datasets due to similar behaviors in normal network traffic and contemporary attacks. K-means centroid-based machine learning classification techniques served as classifiers for KDD 99 and UNSW-NS15 datasets [25]. In [26], the new Earth Mover's Distance (EMD) approach achieved maximum 99.95% detection accuracy when implemented on KDD 99 datasets.

Selecting relevant and useful datasets for evaluating and assessing powerful IDS schemes is essential for dataset selection. Although various datasets are available for IDS system training and testing, some datasets have become outdated or expired over time due to new dataset additions or insufficient

information regarding novel network system attacks. Commonly used datasets in early network research included DARPA 98 and 99 [16], which were adequate for that era's network security, primarily designed for military communication networks. Over time, existing datasets failed to meet modern security parameters, leading to KDD CUP 1999 [16] introduction, basically an extension of existing DARPA 98 datasets with updates and new features. Considerable work has been accomplished using KDD datasets, though they include inappropriate raw data and redundancy. KDD 99 dataset updates and enhancements removed redundancy, creating the NSL-KDD dataset. However, over the past decade, NSL-KDD datasets have failed to address real-time problems due to dramatic real-world scenario changes.

In contemporary times, technological advancement peaks, resulting in enormous network data traffic volumes generated through various connected devices and nodes, including IoT devices, gadgets, sensors, handheld devices, computers, and networks, all contributing to data generation. As data volumes increase, network security concerns reach high-risk levels because legacy datasets lack information about new technologies and techniques, potentially compromising network security. To address these issues and abnormalities, a new dataset was introduced in 2015, namely UNSW-NS15, by [22, 23, 24]. It contains more features than NSL-KDD datasets, approximately 49 features and 9 categories or class types [25, 26, 27] excluding normal classes. UNSW-NS15 datasets consist of four primary data files. Files one, two, and three contain approximately 700,001 records each, while file four contains 440,044 records. The subsequent data preparation step involves combining all four files into a single file, creating a new file containing approximately 2.5 million records [28]. Managing such enormous files is challenging without machine learning techniques. This raw dataset contains numerous null values, leading to inaccurate research results. Approximately 87.3% of records are normal, while 12.7% represent abnormal records in UNSW-NB15 datasets.

3.1 Proposed Framework

The suggested concept layout is illustrated in Figure 3 below. This section proposes using LabelEncoder and standard scaler in data preprocessing phases for data normalization, reshaping data into appropriate formats. Subsequently, the dataset is split into training and testing sets. After dataset splitting, the proposed methodology is implemented and tested using 10 and 700 splits in k-fold cross-validation strategies for intelligent decision-making.

3.2 Data Preprocessing

Preprocessing involves various classification processes for reshaping or reformatting data into useful dimensions. This section preprocesses datasets by selecting machine learning methodologies and techniques. Data preprocessing importance is crucial for accurate results because data redundancy and integrity may influence generated outcomes. Different methods and statistical tools including WEKA, R- programming, Python, and SAS are available for performing these tasks. Data preprocessing's primary task involves transforming raw data into processed forms with more predictable patterns. At this stage, preprocessing is used for data regularization and standardization. Researchers in [29] noted that string values exist in most dataset columns; each string value is mapped to integer or numeric values because machine learning doesn't recognize string values. Dataset examination identified missing column headings for each column; therefore, all columns require labeling for easy identification. Second, NaN values are filled with zero values because nan values can impact research results. Third, data is converted from string to numeric data types because machine learning doesn't process string values, only numeric values with data scaling. Nine columns were identified containing object/string datatypes.

3.3 Data Normalization

Dataset normalization phases represent key model points. For this purpose, Python programming's LabelEncoder method is used to map or convert string/object data types into integer data types. Numerical representation was essential because machine learning techniques cannot process string

values. Standard Scaler is also used for data standardization. Data standardization is crucial because machine learning approaches perform better and faster when features are on similar scales or normally distributed. StandardScaler is computed using the following Equation 1:

$$z = (x - \mu)/s \tag{1}$$

Where μ represents mean and s represents standard deviation. Standard Scaler reformats data with

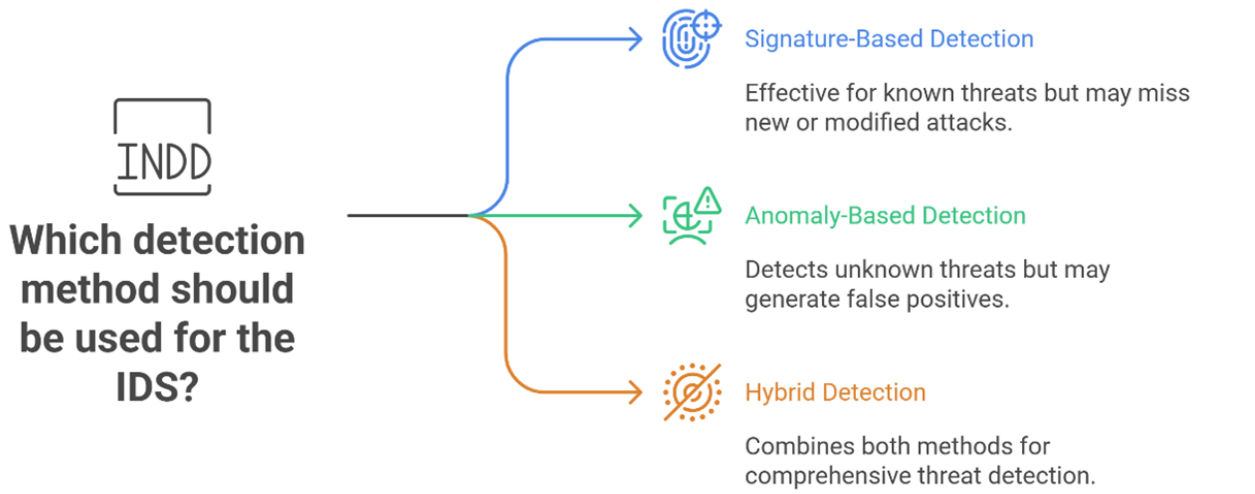

Figure 3: Proposed IDS Architecture.

0 mean and 1 standard deviation. Two main reasons for using Standard scaler are listed below [30]:

1. It is utilized when datasets contain negative values, helping arrange such negative data into normal distributions.
2. It is beneficial when classification is more important than regression.

The primary purpose for using standard scaler is that data contains negative values that don't demonstrate normal distribution.

3.4 Min-Max Scaling

According to [31], min-max normalization techniques are used for transforming outputs from one value group to new value groups. Generally, particular feature minimum values are converted to 0 and maximum values converted to 1, while other values between 0 and 1 transform into decimals between 0 and 1. Min-max scalability enables recent intrusions of particular classes to be added to datasets without requiring complete layer training, only disturbed layers require training [33]. Hyperbolic tangent functions are among the most significant activation functions, enclosed within -1 to 1 ranges [37], Equation 2.

$$\phi(x) = tanh(x) = (e^{x} - e^{-x})/(e^{x} + e^{-x}) \tag{2}$$

can be calculated using the following Equation 3:

$$x_n = \frac{(x_0 - min(x))}{(max(x) - min(x))} \tag{3}$$

where, xn = new variable x value $x0$ = current variable x value $min(x)$ = minimum dataset value $max(x)$ = $maximum$ dataset value Due to non-linear Multilayer Perceptron (MLP) popularity in

current research, MLP is briefly presented in this section. MLP represents one of the best ordinary function classifiers demonstrating capabilities and efficiency for handling various application fields [32]. MLP is also an optimal passive network model [33]. MLP is an artificial neural network branch consisting of at least three node layers with directed graphs [34]. It shares similar models with sigmoid functions with several sigmoid function benefits. These include derivatives utilized in neural network training, examined in future backpropagation algorithm sections. Where w denotes weight vectors, x represents input vectors, b represents bias, and phi represents non- linear activation functions. Neural networks are particularly competent for incident prediction when networks have large databases [28]. Each node is separate from input nodes. Testing and training phases can be implemented within short and long periods respectively. MLP requires considerable training time but has rapid testing times [32]. Total output numbers or anticipated classes and total hidden layer numbers represent significant MLP algorithm execution design considerations. Linear, Sigmoid, and Hyperbolic functions can be implemented in MLP algorithms [35].

3.5 Validation Techniques

This represents highly effective techniques for determining suggested model effectiveness. It further assists in preventing over-fitting while ensuring projected models are comprehensive for separate data by subdividing datasets into multiple folds using K-folds schemes [36]. Thus, all record entries in actual training datasets are used for both training and testing. Cross Validation is specifically employed in ML for evaluating machine model expertise on unseen data. The suggested model version splits datasets into 10 equal folds for K-Folds and 700 folds for JackKnife processing. In [38], K-Fold parameters divide data into k roughly equivalent chunks. Numerous justifications exist for applying Cross validation approaches including data inadequacy, avoiding overfitting issues, and classifying and fine-tuning suggested model parameters in Equation 4.

$$Sigmoidf(x) = \frac{1}{(1 + e^{-x})} \tag{4}$$

In this practice, datasets are arbitrarily divided into K components. Suggested models are consequently trained through k-1 MLP falls under supervised networks due to backpropagation usage [36]; therefore, required responses must be trained beforehand. MLP is typically used for pattern classification. MLP's advantage is that it not only detects attacks but also identifies attack types [39]. MLP also enhances portions and tested on remaining portions. This action is repeated K times, ensuring every K portion is exclusively applied once as test data. [40] explained that accuracy acquired in every single iteration is averaged to achieve optimal accuracy rates.

3.6 JackKnife Technique

Another cross-validation method is JackKnife, introduced by Quenouille [41]. Since data is used extensively, this technique's main purpose is overcoming over-fitting problems in data. JackKnife evaluation engages in computing statistical interest facts for all data combinations where one or more original data observations are removed, calculating estimates and averaging these calculations [42].

3.7 Results and Performance Evaluation

MLP classifier performance is based on accuracy, sensitivity, specificity [43], and Matthew's correlation coefficient. These three metrics are estimated through True Positive measures, False Positive measures, False Negative measures, and True Negative measures. These values are acquired from confusion matrices as shown in Table I. Confusion matrices compose instance numbers predicted accurately or inaccurately by classification models. Confusion matrices work as core source units containing all information about predicted and unpredicted attacks as shown in Table 2.

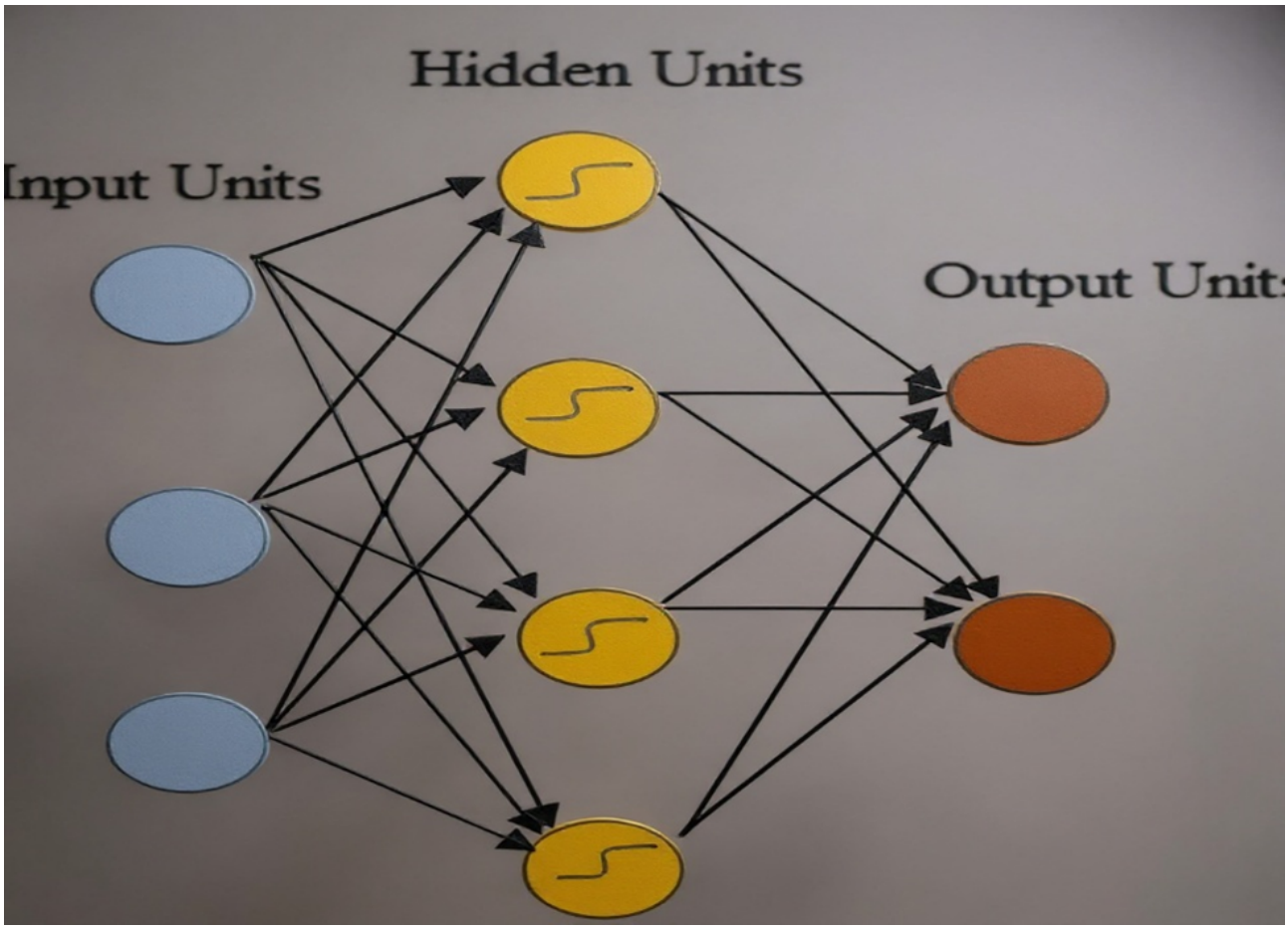

Figure 4: Multilayer Perceptron.

Table 2: Attack Evaluation Confusion Matrix

Attack Classification	**Connection Type**	
	Normal	Attack
Normal	True Positive 2218738	False Negative 27
Attack	False Positive 8	True Negative 321275

3.8 Training and Testing Data

This research's main theme is dataset in Equation 5

$$Sensitivity = (TP)/(TP + FN) \tag{5}$$

When discussing specificity, it represents probabilities of how efficiently proposed approaches can perfectly predict negative instances and is computed using Equation 6:

$$Specificity = (TN)/(TN + FP) \tag{6}$$

Earlier researchers have used selected datasets for generating optimal detection rates or accuracy. Our research uses all available data records in UNSW-15 datasets. UNSW-15 datasets contain four data files, each Matthew's Correlation Coefficient (MCC) is widely used as performance metrics for imbalanced and large data [44, 45]. It represents balanced measures usable even when classes have very different sizes. MCC represents significant correlation coefficients between undetected and detected dataset classifications. 10 iterations lead to better analysis and testing of complete dataset records. Independent tests using MLP classifiers measure 99.9887% accuracy. K-fold tests measure 99.999% accuracy using MLP classifiers. These results demonstrate method performance accuracy. 99.998% accuracy was achieved when executing JackKnife. Matthew's Correlation

3.9 Experimental Results

Coefficient also generated good results at 99.993% in both k-fold and JackKnife techniques. Specificity and sensitivity percentages are also very high at 99.998 and 99.997 respectively in both techniques. For evaluating each IDS model, ordinary benchmarks exist for determining whether efficiency is sufficient. Accuracy, True Positive, True Negative, and Precision represent major terms used in IDS model evaluation. Appropriate IDS would prefer higher Accuracy. Therefore, results causing higher accuracy percentages heavily depend on feature selection. Feature selection processing was not performed for data categorization. Default features were used as shown in Table 1; selected features differ according to category-wise classifications, with some features falling into multiple categories depending on selection criteria and algorithms. Table 2 shows different approaches, algorithms, and machine learning methods used in customized ways for producing higher accuracy ratios. Figure 2 shows method-wise accuracy percentages at 99.96% achieved using Naïve Bayes methods under certain conditions. Lowest accuracy percentages are 78.47% achieved using EM Clustering techniques as given in Table 3. All

Table 3: Classification Performance Metrics

Classification Metric	**Accuracy**	**Sensitivity**	**Specificity**	**MCC**
MLP	99.9987	99.9988	99.9191	99.9492
MLP+10-FOLD	99.9986	99.9987	99.9975	99.9937
MLP+JACKKNIFE	99.9986	99.9987	99.9975	99.9937

techniques and methods resulting in higher accuracy were obtained on selected dataset portions. These methods were not previously implemented on complete datasets; now machine learning methods are executed on complete datasets to observe impacts and results. For this purpose, k-fold and jackknife techniques falling under machine learning are used. In k-fold, datasets are processed in 10 iterations with each iteration producing accuracy results. At all iteration ends, all results are combined and average final results are calculated, which are better than single iteration results because in 10 iterations, datasets are divided into 4 training sets and 1 testing set. These sets change in all outputted nearly equal results, with result figures at 99.9937% for both K-fold and JackKnife methodologies. These results show that using K-fold techniques adds advantages because it only uses 10-folds while providing identical results instead of using 700-folds. It also affects execution time, which is also major parts for achieving optimal results.

4 Discussion

It is observed that proposed approaches showed ways to enhance intrusion detection overall performance through preprocessing, normalizing, and scaling datasets, splitting them into multiple equal folds. UNSW-NB15 datasets lack column headings, making data analysis difficult by columns; to overcome this, column headings were named. Sometimes data also contains missing values, represented as NaN values. To handle NaN values, different options are available, such as deleting rows with missing values, but this approach is not optimal because deleting entire rows affects results. In this research, NaN values were replaced with zero values.

Datasets generally comprise different data types including integers, strings, floats, etc., but machine learning techniques only understand numeric values, so to generate accurate results, non-numeric values must be converted to numeric values. After completing these steps, target columns are excluded from datasets, and remaining dataset portions are used for training and testing. Standard scalers are used for normalizing and reshaping data. After data normalization, MLP independent tests were conducted on normalized data, achieving 99.9887% accuracy figures. We wanted to optimize these results using K-fold techniques implemented on datasets. This technique splits data into equal portions; we figured k=10 in k- fold cross-validation, meaning it iterates data ten times with each portion used only once as testing portions.

This method ensures complete data is used as testing and training parts equally. If one part is used at any stage as testing, it will never be used in future as testing parts. This approach helps deeply an-

alyze and test complete data. Although this approach requires time for executing such huge datasets, this procedure is effective for learning intrusion detection systems in better ways. Machine learning approach Multilayer Perceptron (MLP) is applied as classifiers with different parameters such as 'hidden layer sizes'. Confusion matrices are calculated using MLP classifiers. Through these confusion matrices, 99.99% figures were achieved for Accuracy, Sensitivity, Specificity, and Matthew's Correlation Coefficient (MCC). We further tested datasets to obtain optimal results by dividing datasets more deeply. This was accomplished using JackKnife approaches, where k=700 folds were used, which is very lengthy process for executing large data amounts. But when analyzing JackKnife approach outputs, they behaved identically in results like 10-fold. Therefore, we conclude that MLP classifiers perform much better using 10-fold. Obviously, it requires considerable time for processing and executing 700 folds, then calculating values according to performance metrics. But when results are compiled as shown in Figure 7, identical results were shown as obtained from using 10-folds processes, meaning there is no need for processing lengthy and time-consuming processes like JackKnife.

5 Conclusion

In this information and digitalization era, either enhancing existing IDS model efficiency or devising such secure frameworks that address next-generation network security aspects is essential. A machine learning framework based on multilayer perceptron (MLP) classifiers with 99.98% accuracy is devised. This work is further validated through 10-fold and JackKnife cross-validation. Key metrics for observing accuracy impacts and other performance measurement metrics such as Sensitivity, Specificity, and Matthew's Correlation Coefficient are discussed. All metrics show good results, meaning MLP is the suitable classification technique. The suggested model's accuracy, sensitivity, specificity, and MCC rates computed as 99.99% using UNSW-NB15 datasets. K-fold and JackKnife are capable of earning 99.99% accuracy.

Supplementary Materials

All relevant data is within the manuscript and its supporting information files.

Funding

This research received no external funding.

Conflicts of Interest

The authors declare no conflict of interest.

References

[1] Qadeer, I., & Ehsan, M. K. (2021). Improved channel reciprocity for secure communication in next generation wireless systems. *Computers, Materials & Continua*, *67*(2), 2619-2630.

[2] Ehsan, M., Shah, A. A., Amirzada, M. R., Naz, N., et al. (2021). Characterization of sparse WLAN data traffic in indoor opportunistic environments as a prior for coexistence scenarios of modern wireless technologies. *Alexandria Engineering Journal*, *60*(1).

[3] Ehsan, M., & Dahlhaus, D. (2015, February). Statistical modeling of ISM data traffic in indoor environments for cognitive radio systems. In *IEEE Digital Information, Networking, and Wireless Communication (DINWC), 2015 Third International Conference* (pp. 88-93). Moscow, Russia.

[4] Mahmood, A., Hong, Y., Ehsan, M., & Mumtaz, S. (2021). Optimal resource allocation and task segmentation in IoT enabled mobile edge cloud. *IEEE Transactions on Vehicular Technology*, *70*(12).

[5] Tsai, C. F., Hsu, Y. F., Lin, C. Y., & Lin, W. Y. (2009). Intrusion detection by machine learning: A review. *Expert Systems with Applications*, *36*(10), 11994-12000.

[6] Garg, T., & Khurana, S. S. (2014). Comparison of classification techniques for intrusion detection dataset using WEKA. *International Conference on Recent Advances and Innovations in Engineering (ICRAIE-2014)*.

[7] Krishnan, R. B., & Raajan, N. R. (2016). An enhanced multilayer perceptron based approach for efficient intrusion detection system. *8*(4), 23139-23156.

[8] Biesecker, K., Foreman, E., Staples, B., & Jones, K. (2008). Intelligent transportation system (ITS) information security analysis.

[9] Yadav, M. R., & Kumbharkar, P. (2014). Intrusion detection system with FGA and MLP algorithm.

[10] Desai, A. S., & Gaikwad, D. P. (2016). Real time hybrid intrusion detection system using signature matching algorithm and fuzzy-GA. In *2016 IEEE International Conference on Advances in Electronics, Communication and Computer Technology (ICAECCT)*.

[11] Saxena, A. K., Sinha, S., & Shukla, P. (2017). General study of intrusion detection system and survey of agent based intrusion detection system. In *2017 International Conference on Computing, Communication and Automation (ICCCA)*.

[12] Janarthanan, T., & Zargari, S. (2017). Feature selection in UNSW-NB15 and KDDCUP99 datasets. In *2017 IEEE 26th International Symposium on Industrial Electronics (ISIE)*.

[13] Sultana, N., Chilamkurti, N., Peng, W., & Alhadad, R. (2018). Survey on SDN based network intrusion detection system using machine learning approaches. *Peer-to-Peer Networking and Applications*, *12*(2), 493-501.

[14] Siddiqui, S., Khan, M. S., Ferens, K., & Kinsner, W. (2017). Fractal based cognitive neural network to detect obfuscated and indistinguishable internet threats. In *2017 IEEE 16th International Conference on Cognitive Informatics & Cognitive Computing (ICCI*CC)*.

[15] Efferen, L. V., & Ali-Eldin, A. M. (2017). A multi-layer perceptron approach for flow-based anomaly detection. In *2017 International Symposium on Networks, Computers and Communications (ISNCC)*.

[16] Chowdhury, M. N. (2016). Network intrusion detection using machine learning. *Network Intrusion Detection using Machine Learning*.

[17] Belouch, M., El, S., & Idhammad, M. (2017). A two-stage classifier approach using RepTree algorithm for network intrusion detection. *International Journal of Advanced Computer Science and Applications*, *8*(6).

[18] Benmessahel, I., Xie, K., Chellal, M., & Semong, T. (2019). A new evolutionary neural networks based on intrusion detection systems using locust swarm optimization. *Evolutionary Intelligence*, *12*(2), 131-146.

[19] Gharaee, H., & Hosseinvand, H. (2016). A new feature selection IDS based on genetic algorithm and SVM. In *2016 8th International Symposium on Telecommunications (IST)*.

[20] Mogal, D. G., Ghungrad, S. R., & Bhusare, B. B. (2017). NIDS using machine learning classifiers on UNSW-NB15 and KDDCUP99 datasets. *IJARCCE*, *6*(4), 533-537.

[21] Mishra, P., Pilli, E. S., Varadharajant, V., & Tupakula, U. (2016). NvCloudIDS: A security architecture to detect intrusions at network and virtualization layer in cloud environment. In *2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*.

[22] Zhang, H., Wu, C. Q., Gao, S., Wang, Z., Xu, Y., & Liu, Y. (2018). An effective deep learning based scheme for network intrusion detection. In *2018 24th International Conference on Pattern Recognition (ICPR)*.

[23] Moustafa, N., & Slay, J. (2015). The significant features of the UNSW-NB15 and the KDD99 data sets for network intrusion detection systems. In *2015 4th International Workshop on Building Analysis Datasets and Gathering Experience Returns for Security (BADGERS)*.

[24] Moustafa, N., & Slay, J. (2016). The evaluation of network anomaly detection systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set. *Information Security Journal: A Global Perspective*, *25*(1-3), 18-31.

[25] Setiawan, B., Djanali, S., & Ahmad, T. (2017). A study on intrusion detection using centroid-based classification. *Procedia Computer Science*, *124*, 672-681.

[26] Tan, Z., Jamdagni, A., He, X., Nanda, P., Liu, R. P., & Hu, J. (2015). Detection of denial-of-service attacks based on computer vision techniques. *IEEE Transactions on Computers*, *64*(9), 2519-2533.

[27] Vijayanand, R., Devaraj, D., & Kannapiran, B. (2017). Support vector machine based intrusion detection system with reduced input features for advanced metering infrastructure of smart grid. In *2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS)*.

[28] Baharuddin, M. F. (2018). Malicious URL classification system using multi-layer perceptron technique. *Journal of Theoretical and Applied Information Technology*, *96*, 6454-6462.

[29] Divekar, A., Parekh, M., Savla, V., Mishra, R., & Shirole, M. (2018). Benchmarking datasets for anomaly-based network intrusion detection: KDD CUP 99 alternatives. In *2018 IEEE 3rd International Conference on Computing, Communication and Security (ICCCS)*.

[30] Rauch, S., & Panchal, S. (2019). When to use standard scaler and when normalizer? *Data Science Stack Exchange*. Retrieved from https://datascience.stackexchange.com/questions/45900/when-to-use-standard-scaler-and-when-normalizer

[31] Adeyemo, A., & Wimmer, H. (2018). Effects of normalization techniques on logistic regression on data science. In *2018 Proceedings of the Conference on Information Systems Applied Research Norfolk Virginia*, *11*(4813).

[32] Pal, S., & Mitra, S. (1992). Multilayer perceptron, fuzzy sets, and classification. *IEEE Transactions on Neural Networks*, *3*(5), 683-697.

[33] Ibrahim, H. E., Badr, S. M., & Shaheen, M. A. (2012). Adaptive layered approach using machine learning techniques with gain ratio for intrusion detection systems. *International Journal of Computer Applications*, *56*(7), 10-16.

[34] Britton, E. G., Tavs, J., & Bournas, R. (1995). TCP/IP: The next generation. *IBM Systems Journal*, *34*(3), 452-471.

[35] Almesidin, M., Alzubi, M., Kovacs, S., & Alkasassbeh, M. Evaluation of machine learning algorithms for intrusion detection system.

[36] Mohamed, H., Hefny, H., & Alsawy, A. (2018). Intrusion detection system using machine learning approaches. *Egyptian Computer Science Journal*, *42*(3).

[37] Shah, D. (2017). Activation functions. *Medium*. Retrieved from https://towardsdatascience.com/activation-functions-in-neural-networks-58115cda9c96

[38] Tobi, A., & Duncan. (2019). Improving intrusion detection model prediction by threshold adaptation. *Information*, *10*(5), 159.

[39] Yadav, S., & Shukla, S. (2016). Analysis of k-fold cross-validation over hold-out validation on colossal datasets for quality classification. In *2016 IEEE 6th International Conference on Advanced Computing (IACC)*.

[40] Chou, J. S., Chiu, C. K., Farfoura, M., & Al-Taharwa, I. (2011). Optimizing the prediction accuracy of concrete compressive strength based on a comparison of data-mining techniques. *Journal of Computing in Civil Engineering*, *25*(3), 242-253.

[41] Kokkinidis, K., Mastoras, T., Tsagaris, A., & Fotaris, P. (2018). An empirical comparison of machine learning techniques for chant classification. In *2018 7th International Conference on Modern Circuits and Systems Technologies (MOCAST)*.

[42] Abdi, H., & Williams, L. J. Jackknife. Retrieved from https://utdallas.edu/ herve/abdi-Jackknife2010-pretty.pdf

[43] Chauhan, H., Kumar, V., Pundir, S., & Pilli, E. S. (2013). A comparative study of classification techniques for intrusion detection. In *2013 International Symposium on Computational and Business Intelligence*.

[44] Krishnan, R. B., & Raajan, N. R. (2016). An enhanced multilayer perceptron based approach for efficient intrusion detection system. *International Journal of Pharmacy & Technology*, *8*(4), 23139-23156.

[45] Boughorbel, S., Jarray, F., & El-Anbari, M. (2017). Optimal classifier for imbalanced data using Matthews correlation coefficient metric. *PLoS One*, *12*(6).