Multistage Optimized Fuzzy Based Intrusion Detection protocol for NIDS in MANET

S.Venkatasubramanian
Associate Professor, Department of Computer Science and Engineering, Saranathan college of Engineering, Trichy, India

Abstract - **Malicious activities that harm the operation of a network can be sensed using intrusion detection systems (IDS), which are critical in this regard. There are wireless networks called mobile ad hoc networks (MANETs) that don't require infrastructure to operate and can transfer data wirelessly. They are difficult to secure because of their decentralized character and the scarcity of resources. As a result, Network Intrusion Detection System (NIDS) is an excellent network security tool for detecting unknown threats in network traffic. Traditional machine learning models, such as Support Vector Machine (SVM), and others, are now the basis for most network anomaly detection systems. These methods can indeed provide some impressive results, but accuracy suffers, as they rely primarily on the manual creation of traffic features, which is no longer necessary in the age of big data This research provides a unique multi-stage optimized Fuzzy based NIDS framework to address the issues of low accuracy and feature engineering in IDS while also retaining detection performance. With the help of oversampling techniques, this research examines how small a training sample should be to get the best results. In addition, it compares and investigates the impact of detection performance and time complexity on two feature selection systems as information gain and correlation based. It is also being explored how to improve the NIDS's performance using hyper parameter (HP) optimization methods like Random Search (RS) and Genetic Algorithm (GA). The NSL-KDD dataset is used to assess the presence of the framework. Additionally, hyper-parameter adjustment with detection accuracies improves model performance.**

Index Terms - **Fuzzy based Framework; Mobile ad hoc networks; Hyper Parameter Optimization Techniques; NSL-KDD dataset;**

I.INTRODUCTION

A MANET is a sophisticated wireless network made up of a large number of mobile nodes that spontaneously establish a network without any physical infrastructure, allowing individuals, groups of people, and organizations to collaborate and communicate without the need for a solid infrastructure [1]. Messages are relayed from the source to the destination by other nodes in a MANET if the nodes are within the expanse or designated boundary. The most important security objectives must be met to keep an ad hoc network safe and secure. They include non-repudiation, authentication, and integrity as well as secrecy and availability. One of the most important parts of a security strategy is the use of NIDS for malicious node detection[2]. In the security of the network, there are two corporate detection methods to Network Intrusion Detection Systems (NIDS): signature-based detection and anomaly-based detection. Moreover, NIDS rely on the concept of "traffic identification", that is, extracting the useful features from the captured traffic flow, and then classifying the traffic record to either normal or attack by using one of the previously trained machine learning algorithms [3]. Since network viruses, eavesdropping, and malicious attacks have increased, network security has become a top priority for the government and society. Intrusion detection, on the other hand, can effectively deal with these issues. Network information security relies heavily on intrusion detection. Due to Internet business' accelerated growth, the types of traffic that pass via networks are expanding daily, and network behavior characteristics are getting more complicated, posing new problems for intrusion detection. Isolating and detecting malicious network traffic is a critical issue that can't be avoided.

Network traffic can indeed be split into two types such as normal and malicious traffics). Aside from that, there are five other types of network traffic: normal; dos; root to local; and user to root; as well as a probe. As a result, detecting intrusions might be viewed as a classification issue. The accuracy of IDS can be greatly enhanced by enhancing the efficiency of classifiers in efficiently identifying hostile traffic. In intrusion detection, machine learning approaches [6]–[9] are commonly used to identify malicious traffic. These methods, on the other hand, belong to the shallow learning category and frequently emphasize feature selection and engineering. With low recognition accuracy and a high false alarm rate due to large intrusion data sets, they are unable to successfully tackle the big intrusion data categorization problem.

Deep learning-based approaches for IDS have been proposed one after the other in recent years. The authors of [10] present a method for classifying malware traffic using a convolutional neural network and a picture of traffic. As an alternative to using handcrafted features, this approach uses the original traffic as the classifier's input data. According to [11], Recurrent Neural Networks (RNN) can detect network traffic behavior by representing it as a series of changing states over time. Classifying incursion traffic is made easier with the use of an LSTM network, which the authors demonstrate in [12]. Agreeing to the findings of the experiments, the LSTM algorithm is capable of learning all the attack classes that are hidden in the training data. Regardless of which approach you to choose, all network traffic is treated as if it were made up of nothing more than a series of traffic bytes. They don't employ network traffic field knowledge to its maximum potential. It is analogous to handling traffic as if it were an unrelated entity, like CNN does, and ignores the internal relations of the network traffic. In the first place, network traffic is organized into levels. A network traffic unit is a collection of data packets traveling over a network. There are numerous bytes in a data packet, and it is a traffic unit. Second, traffic characteristics within and between packets differ greatly. It is necessary to extract sequential information from successive packets independently.

The main contributions can be summarized as follows:

- To reduce computing complexity and improve detection accuracy, propose an innovative, multi-stage Fuzzy-based NIDS architecture.
- Fuzzy logic is used to detect intrusions that might otherwise cause the network to refuse or extract confidential information from an active connection.
- Investigate the influence of sampling approaches and establish the smallest training sample size necessary for efficient IDS.
- Find out the effect of NIDS detection performance and train and test the complexity that is affected by different feature selection strategies.
- HP optimization approaches and their impact on NIDS detection performance should be proposed and investigated.
- Comparison of performance with previous research shows improved detection accuracy and a reduction in the size of the training sample and feature set for the proposed framework.

II. LITERATURE REVIEW

In the study [13], an IDS has been proposed as a malicious detection in computer networks. After performing a recursive feature removal via the random forest, the suggested IDS is tested against the CICIDS2017 dataset. An advanced multilayer perception (DMLP) algorithm is then used on the selected features in [14], resulting in an accuracy rate of 91% in the final analysis. There are two steps to this model: the first is sparse Auto Encoder (AE) for unsupervised feature learning and another is a softmax classifier trained on the resulting training data. They used their model on the NSL-KDD dataset and were able to attain an accuracy of above 90%.

NIDS model using convolutional neural networks is proposed by the authors in [15]. Not only does the CNN model minimize false alarm rates (FAR), it also increases class accuracy when dealing with tiny quantities. As an example, to reduce dimensionality, Keerthi et al. [16] employed one strategy. Random forest and C4.5 classifier techniques were used in their PCA tests with the KDD CUP and UNB ISCX datasets as input. They evaluated the classification accuracy of the C4.5 classifier with that of the 10 main components.

This study by Natesan et al. [17] used a parallel computing model and nature-inspired feature selection method to get the best detection rate possible. In addition, the Map-Reduce programming ideal is utilized for the selection of the optimal subset with the least amount of computational effort. Rough Set Theory (RST) and SVM are both used in IDS.The Wei et al [18] DL-based DBN model has been suggested to be optimized through the combination of particle swarm, and GA. The model has been evaluated using the dataset NSL-KDD. The findings indicated significant improvements in the U2R and R2L class detection rates. The principal disadvantage of the model proposed is that the model's complicated structure increases its training time.

By merging CNN with long-range bidirectional short-term memory, Jiang et al.[19] suggested an efficient IDS system in a deeper hierarchy. A Synthetic Minority Over-sampling Technique (SMOTE)is used to increase marginal samples, which helps the ideal learn the features properly. The class imbalance issue is handled. The CNN has been utilized to extract spatial properties while temporal functions were utilized by BiLTSM. Use NSL-KDD datasets to experiment. The methodology provided achieves greater accuracy and detection rate performance. Minority data class detection rates have slightly increased but comparing other attack classes are still quite poor. The training time is higher due to the intricate structure. Zhang et al. [20] suggested a complicated CNN and gcForest multi-layer IDS model. In addition, they introduced a new P-Zigzag technique for transforming raw data into two-dimensional gray characteristics. For initial detection, they utilized a better CNN model in an initial coarse grit layer. gcForest (caXGBoost) is then employed in the finely grained layer for further classification of the anomalous classes into N-1 classes. They have joined UNSW-NB15 with CIC-IDS2017 datasets using a dataset. The findings from the experiments indicate that the projected model improves greatly in comparison with single algorithms the accuracy and detection rate while minimizing the FAR.

A model IDS based on the new idea of DL few-shot learning (FSL) was proposed by Yu and others [21]. The goal is to train on balanced data from the dataset using a modest quantity. DNN and CNN are embedded in the model for the extraction and scaling of the critical feature. The model efficiency of reasonable rates of detection for minority classes was demonstrated in the experimental findings obtained using NSL-KDD datasets. To obtain such exceptional performance for the considered data set, only less than 2 percent of data were used for training. Efficient CNN-based IDS was proposed by Xiao et al [22]. The key idea is initially to use Principle Component Analysis and AE to do feature extraction. The one-dimensional (feature set) vector is transformed into a 2-D matrix and entered into the Neural network. KDD Cup'99 dataset experiments were conducted. Experiments demonstrate its usefulness in terms of algorithmic time consumed during training and testing. The key problem in comparison to other attack classes is poorer detection rates for the R2L classes.

However many earlier studies have been done on the intrusion detection issue, the models that have been offered have several drawbacks. It's common knowledge that class imbalance occurs in intrusion detection datasets, although many of these studies ignore it. Also, rather than following a systematic approach, the size of the training sample is often chosen at random. Only a few studies have taken hyper-parameter optimization into account by combining several strategies instead of sticking with one.

III. PROPOSED METHODOLOGY

As a result of this paper's optimizations, computational complexity can be reduced while still achieving high detection rates. This is implemented in stages, each with a new set of techniques. See Fig. 1 for a visual representation of the suggested methodology's process flow.

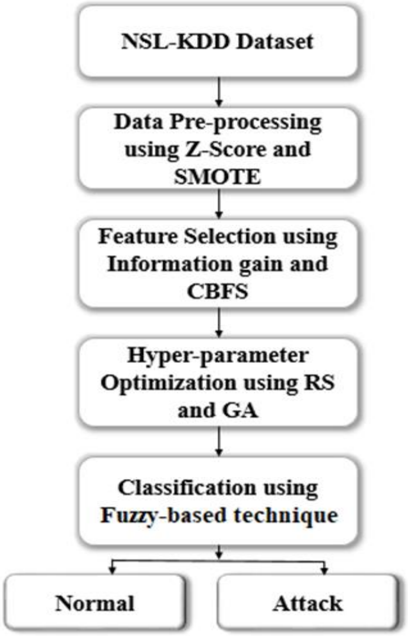
NSL-KDD Dataset
Data Pre-processing using Z-Score and SMOTE
Feature Selection using Information gain and CBFS
Hyper-parameter Optimization using RS and GA
Classification using Fuzzy-based technique
Normal
Attack

Fig 1: Working Flow of Proposed Methodology

The following is a rundown of the many methods that were employed.

A . *Data Pre-processing*:

In the data pre-processing stage, the Z-score method is used to normalize your data and the SMOTE [23] algorithm to oversample the minority class.The training and testing instances, which are described in Table 1 are given as input to normalization and SMOTE processes during pre-processing.

A.1. Z-Score Normalization:

Z-score data normalization is the initial step in the data pre-processing stage. To convert any categorical features to numerical ones, the data must first be encoded with a label encoder before being used. Normalization is carried out by determining for each data point in the dataset the normalized value x_i norm, as shown below:

$$x_{norm} = \frac{x_i - \mu}{\sigma} \quad (1)$$

Where, σ is signified as the standard deviation, and is the feature's mean vector. Because Fuzzy-based techniques perform better with normalized datasets, it's important to point out that the Z-score data is normalized. [24].

A.2 SMOTE Technique

Second, the SMOTE technique is used to perform minority class oversampling. Using this technique, more instances of the underrepresented minority class will be synthetically created, which will help to balance the DL classification model's performance [25]. The training model's performance can be improved by execution minority class oversampling, especially for datasets including network traffic that are prone to this problem [26].

The SMOTE algorithm creates new cases of the minority class based on the analysis of the original instances. As a result, the technique creates a single set X minority that contains all instances of the minority class. A new synthetic instance X new is generated as follows for each of the instances X_{inst}within $X_{minority}$. [27]:

$$X_{new} = X_{inst} + rand(0,1) * \left(X_j - X_{inst}\right), \quad j = 1, 2, \dots, k \quad (2)$$

A random number is generated in the range [0,1], and a random sample is taken from the set of the k nearest neighbors of $X_{inst}\left(X_{1,}X_2, X_3, \dots . X_k\right)$. Unlike other oversampling techniques, SMOTE creates new, high-quality instances that statistically match samples from the minority class. After SMOTE and the normalization process, the data of NSL-KDD is normalized and has balanced data of NSL-KDD for the next process called feature selection to select the optimal features.

B. Feature Selection

The research examines two feature selection strategies, information gain-based, and correlation-based, to better understand model detection enactment and temporal complexity. This is especially crucial when developing proposed models for large-scale systems thatmake high-dimensional data[28].

C.1: *Information Gain-based Feature Selection(IGBFS):*

The IGBFS algorithm is the first to be studied. It utilizes information theory ideas like entropy and mutual information to pick out the best features of NSL-KDD data. When creating the feature subset for the ML model, the IGBFS evaluates each feature depending on how much information it provides about the target class (measured in bits). The features with the most data are selected as part of the feature subset. The function for evaluating features is, therefore, [29-31]:

$$I(S;C) = H(S) - H(S|C) = \sum_{s_i \in S} \sum_{c_j \in C} P\left(s_i, c_j\right) log \frac{P(s_i, c_j)}{P(s_i) \times P(c_j)} \quad (3)$$

where $P\left(s_i, c_j\right)$ is signified as the feature having a value $s_i$and class being c_j, $P(s_i)$is signified as the probability of feature having a value s_i, and $P(c_j)$ is signified as the probability of class being c_j. $I(S;C)$issignified as the mutual information between feature subset S and class C, $H(S)$ is signified as the entropy of discrete feature subset S.Finally, the $I(S;C)$ has the mutual information of features, which is passed through CBFS for the further selection process.

C.2. Correlation-based Feature Selection (CBFS)

The CBFS algorithm is the other feature selection technique under consideration. With its ease of use and ranking attributes according to their correlation with the target class, it is frequently utilized [32-34]. If a characteristic is thought to be relevant by CBFS, it is included in the subset. When employing CBFS, the feature subset evaluation is performed using Pearson's correlation coefficient. As a result, the function of evaluation is:

$$Merit_S = \frac{k\times\overline{r_{cf}}}{\sqrt{k+k\times(k-1)\times\overline{r_{ff}}}} \quad (4)$$

where$Merit_S$ as the merit of the feature subset S, k issignified as the number of features in feature subset S,$\overline{r_{ff}}$ is signified as the average feature-feature Pearson correlation$\overline{r_{cf}}$ is signified as the average class-feature Pearson correlation. Even though, the optimal features are selected by CBFS, HP optimizations are important because they directly control the behavior of the training algorithm and have a significant impact on the performance of the model is being trained.

D. Hyper-parameter Optimization

This work explores two different HP optimization approaches, specificallyRS, and GA meta-heuristic algorithms [35], [36].

D.1. Random Search:

The RS method is the first HP optimization technique. A heuristic optimization model is what this technique belongs to [37]. RS investigates different combinations of the optimization parameters like the grid search algorithm [38, 39]. For the sake of simplicity, let's assume the following model:

$$\underset{parm}{max} f(parm) \quad (5)$$

The objective function f should be maximized (usually the model's accuracy) and the collection of tuning parameters is called parm. While grid search searches through all potential possibilities, the RS method merely randomly selects a sample of features from CBFS to test. This is in contrast to grid search. This means that when there are only a few hyper-parameters to consider, RS outperforms grid search. Additionally, this technique enables parallel optimization, which reduces the computational complexity even further. Therefore, the computational complexity of the proposed model is reduced, while using RS.

D.2. Meta-heuristic Optimization Algorithms

Using these techniques, we hope to find or generate an effective solution to the optimization difficult at hand. They are good candidates for HP optimization because they solve combinatorial optimization issues with decreased computing complexity. This study examines well-known HP optimization called GA.

Meta-heuristic algorithms inspired by evolution and natural selection are common [40] There are several applications for it in combinatorial optimization where biologically inspired procedures like mutation, crossover, and selection are employed to find high-quality features. The search space can be efficiently searched by GA algorithms using these operators. The GA algorithm operates as follows when used with ML hyper-parameter optimization:

a. Create a new population of chromosomes by generating a random set of solutions. Hyper-parameter value combinations can be found on every chromosome.
b. Use a fitness function to assess the overall fitness of each chromosome. If you use the chromosomal vectors, the function is usually the ML model's accuracy level.
c. Descend the list of chromosomes in order of relative fitness.
d. Crossover and mutation processes can be used to produce new chromosomes to replace those that are no longer needed.
e. If the performance doesn't improve after repeating steps b)-d), then stop.

When the effective optimal features are obtained using these RS and GA techniques, these inputs are given to the Fuzzy classifier for the final process.

E. Fuzzy based Classifier

To find, whether the optimal features of NSL-KDD are normal or attack by using Fuzzy rules (i.e. if-then rules). Using formal mathematics and logic, the fuzzy notion can make proper judgments and create qualitative data or projected data. Fuzzy ideas can deal with challenges of a humanistic nature. True (Attack) or false (normal) are the two meanings of the word hazy. Any action or occurrence that alters the current condition, whether temporarily or permanently, is neither true nor untrue. The fuzzy idea is critical in new applications because of this. The result is either 0 or 1 in Boolean logic. Values in a fuzzy system are defined as being between 0 and 1, or yes/no. On the other hand, uses a scale of 1.0 to denote "absolute or extreme truth," with 0.0 denoting ultimate falsehood. The truth value depicts the fuzzy system's dynamic range.

Fuzzy logic uses fuzzy ideas to handle the reasoning capability. Fuzzy logic is a way of describing fuzziness that isn't logic. Uncertainty can only be represented by the fuzzy set. Consider a situation where network users are unsure whether a trustworthy node is malicious or innocent. Due to the current state of uncertainty, it is extremely difficult to distinguish between malicious and benign nodes. Assign a value to each potential crisp set to reflect this type of uncertainty. In other words, it's a measure of how confident the nodes in a network are. A fuzzy measure is another name for uncertainty. After examining all available data, fuzzy measures resolve the issue and then make the most appropriate choice for the provided input.

1. Rules: For improved decision-making, it contains a set of rules and regulations that govern. The fuzzy rule has been reduced as a result of the more effective decision-making mechanism.
2. Inference:This fuzzy input set matches the rules, and if it doesn't match an existing rule, a new rule based on fuzzy input will be established. The new rules are then merged to arrive at the most appropriate option.
3. Fuzzifier:The crisp input must be converted to fuzzy input set values before the inference process can begin. Fuzzifier is used to carry out the conversion process described above.
4. Defuzzifier:To help end-users, it transforms the fuzzy result of an inference process into a crisp output. Figure 4 shows the working flow of the proposed Fuzzy model.

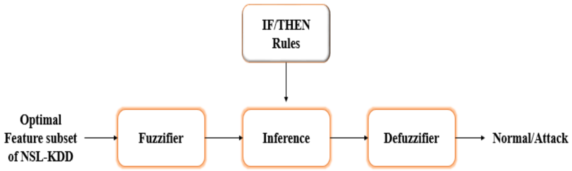

Fig 2: Working Procedure of Proposed Fuzzy Model

The benefits of Fuzzy logic systems are:

- The system has a high level of reliability.
- System performance by adding or removing components as needed.
- Fuzzy logic systems are simple to implement and understand.
- In the rising trends, it delivers a very efficient solution to difficult problems. It addresses engineering uncertainty.

F. Security Considerations

The proposed multi-stage Fuzzy-based NIDS architecture is based on signatures and is an NIDS [41]. As a result, the framework picks up on trends from previously reported attacks. A distinction should be made between the framework and an anomaly-based NIDS because the framework is trained using a binary classification model to identify any unusual behavior as an assault. It is possible to use this framework as a single module as part of a larger security policy framework. Other techniques such as firewalls, deep packet inspection, and user authentication procedures can be included in this security framework/policy [42-43]. This would provide a safe architecture with multiple layers that can protect user data and information while also maintaining privacy and security.

IV. RESULTS AND DISCUSSION

All tests were conducted on an Ubuntu 14.0.4 LTS with Python. Use Scikit-learn to implement all traditional machine learning algorithms. Using GPU-enabled TensorFlow4, three DNNs were developed with a higher Keras5 framework backend. The GPU was NVidia GK110BGL Tesla K40 and the CPU was configured to run on 1 Gbps Ethernet network (32 GB RAM, 2 TB hard disk. The following test cases were selected to assess the performance of the proposed and different classical deep learning classifiers on NSL-KDD dataset.

A. Dataset Description

We considered the widely available and widely used leak detection data sets in earlier work: the NSL-KDD data set. The dataset has normal data and four different types of attacks include Probe, U2R, R2L, and DoS. There are 42-dimensional features are presented in each intrusion record and it is categorized into a 3-dimensional symbol feature, a traffic type label, and a 38-dimensional digital feature. Table 1 shows the description of the data set.

Table 1: Dataset Description

Category	Train	Test
DoS	11656	7458
Probe	45927	2421
R2L	995	2754
Normal	67343	9711
U2R	52	200
Total	125973	22544

B. Performance metrics

The basis truth value is necessary for the evaluation of the various statistical measures. In the instance of binary classification, the foundation truth consisted of several connection registers that were normal or attack. Let L and A be the sum of usual and attack logs in the test dataset and use the subsequent terms to determine the excellence of the classification model:

- True Positive (TP) - the sum of connection records properly categorised to the Usual class.
- False Negative (FN) - the sum of Attack connection records incorrectly categorised to the Usual connection record.
- True Negative (TN) - the sum of connection records properly categorised to the Attack class.
- False Positive (FP) - the sum of Normal linking records wrongly categorised to the Attack linking record.

The following evaluation metrics are examined based on the above given terms.

$$Accuracy = \frac{TN+TP}{TP+TN+FN+FP} \times 100 \qquad (6)$$

$$F - measure = \frac{2TP}{(2TP+FP+FN)} \times 100 \qquad (7)$$

$$\text{Precision}=\frac{\text{TP}}{(\text{FP+TP})}\times 100 \qquad (8)$$

$$\text{Recall}=\frac{\text{TP}}{(\text{FN+TP})}\times 100 \qquad (9)$$

C. Performance Evaluation of Proposed Model

The proposed evaluation has been segregated into major parts such as binary classification and Multi-class classification. The binary classification has detecting the attack or normal communication. Multi-class classification has to detect the various types of attack, which is presented in the dataset.

C.1. Multi-class classification

The detailed results for the classification of the proposed system for multi-class are reported in this section. Table 2 and Figure 3 show the performance analysis of the proposed model on multi-class data classification.

Table.2.Comparative analysis of multi-class on Proposed Fuzzy-based Classifier

Category	Accuracy (%)	Precision (%)	Recall (%)	F-score (%)
Normal	79.08	87.27	94.60	91.47
DoS	82.75	93.16	87.47	83.42
Probe	83.43	75.81	78.62	87.28
U2R	81.33	71.04	76.47	92.94
R2L	87.19	72.32	69.04	81.15

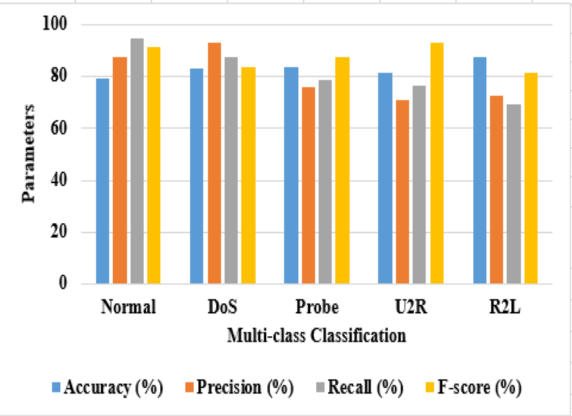

Figure 3: Graphical Representation of Proposed Fuzzy-based Classifier for different categories on NSL-KDD Dataset.

While in the normal category, the proposed method achieved 79.08% of accuracy, 87.27% of precision, 94.60% of recall, and 91.47% of F1-measure. While comparing with other categories on recall experiments, the proposed Fuzzy technique achieved high performance on normal category only. As like, the proposed method achieved high precision (i.e. 93.16%) on DoS category and high F1-measure (i.e.92.94%) only on U2R category. In other categories like Probe, U2R, R2L, the proposed method achieved nearly 71% to 75% of precision, 69% to 78% of recall, and 81% to 87% of accuracy, where Fuzzy technique achieved less recall value (i.e.69.04%) on R2L category only.

C.2 Binary classification

The detailed results for Binary classification of several classical ML and DL classifiers and proposed systems are reported in this section. Figure 4 shows the graphical analysis of the proposed classifier.

Table.3.Comparative analysis of binary class on Proposed with various existing algorithms.

Algorithm	Accuracy (%)	Precision (%)	Recall (%)	F-score (%)
Random Forest (RF)	88.70	90.41	89.21	90.02
Support Vector Machine (SVM)	91.50	91.82	90.81	92.27
Convolutional Neural Network (CNN)	91.90	91.78	92.52	92.41
Bi-directional Long Short-Term Memory (Bi-LSTM)	93.50	92.63	91.38	93.18
Recurrent Neural Network (RNN)	92.70	93.90	92.93	94.32

LSTM	94.27	94.91	93.47	94.63
Proposed Fuzzy-based technique	95.32	96.95	94.24	95.02

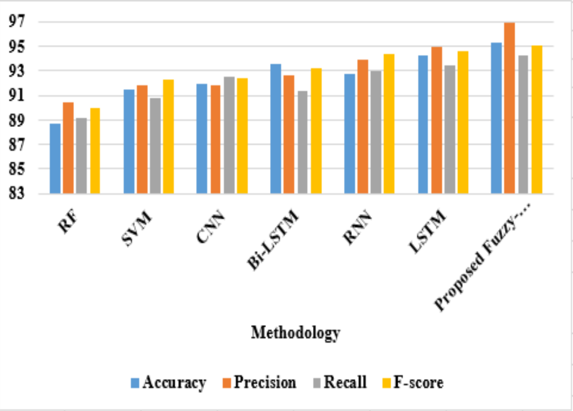

Figure 4: Graphical Representation of Proposed Fuzzy-based technique with existing techniques for binary data classification

From the above table 3, it is proved that the proposed Fuzzy-based technique achieved better accuracy (95.32%), precision (96.95%), recall (94.24%), and F-score (95.02%) than existing ML and DL techniques. The existing techniques namely SVM and CNN achieved nearly 91% to 92% of accuracy, precision, recall, and F-score, where the other techniques such as Bi-LSTM, RNN, and LSTM achieved nearly 92% to 94% of accuracy, precision, recall, and F-score on binary data classification. While comparing with all techniques, Random Forest (RF) provides low results in all parameters, i.e. 88.70% of accuracy, 90.41% of precision, 89.21% of recall, and 90.02% of F-score.

From the existing techniques [13-22] used in Section 2, limited techniques such as DMLP-AE [14], DL-based DBN model [18], SMOTE-CNN [19], and FSL [21] uses the NSL-KDD dataset for validation, whereas other techniques use KDDCup99', UNSW-NB15 with CICIDS2017 and UNB-ISCX datasets for validation. The existing DMLP-AE achieved 91% of accuracy, where the proposed method achieved 95.32% of accuracy. The DL-based DBN method has high training time, where the proposed method uses the multi-stage optimization techniques for each step and minimized the training time of the Fuzzy-based technique. The minority data class rate is high in SMOTE-CNN [19], which is overcome by developing two different stages for balancing the dataset using Z-score and SMOTE in the proposed method. For the training process, FSL [21] uses only less amount of data for the training process that cause low accuracy, where the proposed model uses 80% of training data and achieved more than 92% of accuracy on NSL-KDD dataset.

C.3. Minimal feature analysis

Optimizing functionality is an essential step to detect intrusion. This is a key step towards identifying more correctly the different sorts of attacks. Without the optimization of features, a misclassification of assaults may be possible and the development of a model would take a long time. The methods for selecting functions reduced the training and testing time greatly, as well as an enhanced rate for detecting intrusions. Two experiment trials are performed on limited feature sets on the NSL-KDD to assess the performance of the proposed method and static machine learning classifications. Table 4 provides detailed results. In comparison to tests in 4 feature sets, the experiments with 11 and 8 feature sets were good. In addition, experiments with 11 sets of functionalities were successful compared to the 8 sets. The performance difference of 11 to 8 minimum set of features is minor.

Table.4. Comparative analysis of test results using minimal feature sets.

	Accuracy (%)		
Algorithm	11 features	8 features	4 features
RF	88.27	89.67	87.18
SVM	91.82	92.43	88.79
CNN	92.04	92.94	91.41
Bi-LSTM	93.13	93.06	92.07
RNN	94.43	94.16	93.66
LSTM	94.89	94.87	94.09
Proposed Fuzzy-based Technique	96.90	95.08	94.39

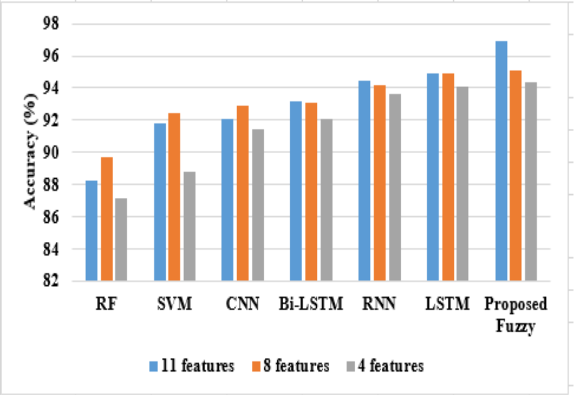

Figure 5: Graphical Representation of proposed Fuzzy-based technique with existing classifiers in terms of accuracy while reducing the features set.

The above table consists of validated results of existing techniques with proposed methods for different attacks namely Normal, DoS, Probe, U2R, and R2L.When the number of features is minimized, the accuracy of the proposed Fuzzy-based technique is also minimized. For instance, its accuracy is 96.90%, when the 11 features are reduced and its accuracy is 95.08%, when 8 features are reduced, finally, it reaches 94.39% of accuracy when only 4 features are reduced. The RF technique achieved low accuracy, i.e. nearly 87% to 89% for all features reduction while comparing with existing techniques. When the feature set is 8, the existing techniques such as SVM, CNN, Bi-LSTM, RNN, and LSTM achieved 92.43%, 92.94%, 93.06%, 94.16%, and 94.87% of accuracy, but the same techniques achieved 88.79%, 91.41%, 92.07%, 93.66%, and 94.09% of accuracy only, while the feature set is 4.

V. CONCLUSION

Due to the growing reliance of individuals and businesses on the Internet and their concerns about the security and privacy of their activities, the field of cyber-security has attracted considerable attention from both industry and academia. Modern Internet-based networks are better guarded against intrusions and aberrant activity because to the increased deployment and allocation of resources. Thus, various NIDS kinds have been put up in the literature. There is still room for improvement in NIDS performance despite the advancements that have been made. High volumes of network traffic data, constantly changing settings, and a variety of attributes acquired as part of training datasets (high dimensional datasets) can all be used to get further insights. Fuzzy-based detection models' performance can be improved by selecting the best subset of features and fine-tuning their parameters. Due to the reduced computational complexity, while still maintaining high detection performance, this paper developed an improved Fuzzy-based NIDS framework with many stages of optimization. As a preliminary step, this research looked at the effects of oversampling approaches on the size of the training sample for the models and came up with a minimal training size that was effective in detecting intrusions. Researchers found that utilizing SMOTE oversampling to minimize the number of training datasets had positive benefits IGBFS and CBFS feature selection strategies have been used in this study, and their effects on feature set size, training sample size, as well as model detection performance have all been studied. The results of the experiments revealed that the feature selection approaches could lower the size of the feature set in question. For future work, other models, such as deep-learning classifiers with learning rate optimization techniques, can be investigated because they excel on non-linear and large datasets.

REFERENCES

[1] E. M. Shakshuki, N. Kang and T. R. Sheltami, EAACK - a secure intrusion detection system for MANETs,IEEE Trans. Ind. Electron.2013, 1089-1098.

[2] T.Kavitha, K.Geetha, R. Muthaiah, Intruder node detection and isolation action in mobile ad hoc networks using feature optimization and classification approach, Journal of Medical Systems, 2019.

[3] B. Dong, X. Wang, Comparison deep learning method to traditional methods using for network intrusion detection, in: 2016 8th IEEE International Conference on Communication Software and Networks (ICCSN), IEEE, 2016, pp. 581–585.

[4] B. B. Zarpelo, R. S Miani, C. T. Kawakani, and S. C. de Alvarenga, ‘‘A survey of intrusion detection in Internet of Things,’’ J. Netw. Comput. Appl., vol. 84, pp. 25–37, Apr. 2017.

[5] B. Mukherjee, L. T. Heberlein, and K. N. Levitt, ‘‘Network intrusion detection,’’ IEEE Netw., vol. 8, no. 3, pp. 26–41, May 1994.

[6] S. Kishorwagh, V. K. Pachghare, and S. R. Kolhe, ‘‘Survey on intrusion detection system using machine learning techniques,’’ Int. J. Control Automat., vol. 78, no. 16, pp. 30–37, Sep. 2013.

[7] N. Sultana, N. Chilamkurti, W. Peng, and R. Alhadad, ‘‘Survey on SDN based network intrusion detection system using machine learning approaches,’’ Peer-to-Peer Netw. Appl., vol. 12, no. 2, pp. 493–501, Mar. 2019.

[8] S. Garg and S. Batra, ‘‘A novel ensembled technique for anomaly detection,’’ Int. J. Commun. Syst., vol. 30, no. 11, p. e3248, Jul. 2017.

[9] F. Kuang, W. Xu, and S. Zhang, ‘‘A novel hybrid KPCA and SVM with GA model for intrusion detection,’’ Appl. Soft Comput., vol. 18, pp. 178–184, May 2014.

[10] W. Wang, M. Zhu, X. Zeng, X. Ye, and Y. Sheng, ‘‘Malware traffic classification using convolutional neural network for representation learning,’’ in Proc. Int. Conf. Inf. Netw. (ICOIN), 2017, pp. 712–717.

[11] P. Torres, C. Catania, S. Garcia, and C. G. Garino, ‘‘An analysis of Recurrent Neural Networks for Botnet detection behavior,’’ in Proc. IEEE Biennial Congr. Argentina (ARGENCON), Jun. 2016, pp. 1–6.

[12] R. C. Staudemeyer and C. W. Omlin, "ACM press the south African institute for computer scientists and information technologists conference - east London, South Africa (2013.10.07-2013.10.09) proceedings of the south African institute for computer scientists and information technologists co,’’ in Proc. South African Inst. Comput. Scientists Inf. Technol. Conf., 2013, pp. 252–261.

[13] S. Ustebay, Z. Turgut, M. A. Aydin, Intrusion detection system with recursive feature elimination by using random forest and deep learning classifier, in: 2018 International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism (IBIGDELFT), IEEE, 2018, pp. 71–76.

[14] N. Shone, T. N. Ngoc, V. D. Phai, Q. Shi, A deep learning approach to network intrusion detection, IEEE Transactions on Emerging Topics in Computational Intelligence 2 (1) (2018) 41–50.

[15] K. Wu, Z. Chen, and W. Li, ‘‘A novel intrusion detection model for a massive network using convolutional neural networks,’’ IEEE Access, vol. 6, pp. 50850–50859, 2018.

[16] K. KeerthiVasan and B.Surendiran, “Dimensionality reduction using Principal Component analysis for network intrusion detection,” Elsevier, 2016.

[17] Natesan P., Rajalaxmi R.R., and Gowrison G., “Hadoop based parallel Binary Bat Algorithm for Network Intrusion Detection”, Springer,Int J Parallel Prog, PP. 1-20,2016.

[18] Wei P, Li Y, Zhang Z, Hu T, Li Z, Liu D. An optimization method for intrusion detection classification model based on deep belief network. IEEE Access. 2019;7:87593-87605.

[19] Jiang K, Wang W, Wang A, Network Intrusion WH. Detection combined hybrid sampling with deep hierarchical network. IEEE Access. 2020;8:32464-32476.

[20] Zhang X, Chen J, Zhou Y, Han L, Lin J. A multiple-layer representation learning model for network-based attack detection. IEEE Access. 2019;7:91992-92008.

[21] Yu Y, Bian N. An intrusion detection method using few-shot learning. IEEE Access. 2020;8:49730-49740.

[22] Xiao Y, Xing C, Zhang T, Zhao Z. An intrusion detection model based on feature reduction and convolutional neural networks. IEEE Access. 2019;7:42210-42219.

[23] Gonzalez-Cuautle, D., Hernandez-Suarez, A., Sanchez-Perez, G., Toscano-Medina, L.K., Portillo-Portillo, J., Olivares-Mercado, J., Perez-Meana, H.M. and Sandoval-Orozco, A.L., 2020. Synthetic minority oversampling technique for optimizing classification tasks in botnet and intrusion-detection-system datasets. Applied Sciences, 10(3), p.794.

[24] K. M. Ali Alheeti and K. McDonald-Maier, “Intelligent intrusion detection in external communication systems for autonomous vehicles,” Systems Science and Control Engineering, vol. 6, no. 1, pp. 48–56, 2018.

[25] Z. Chen, Q. Yan, H. Han, S. Wang, L. Peng, L. Wang, and B. Yang, "Machine learning-based mobile malware detection using highly imbalanced network traffic," Information Sciences, vol. 433, pp. 346–364, 2018.

[26] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “Smote: synthetic minority over-sampling technique,” Journal of artificial intelligence research, vol. 16, pp. 321–357, 2002.

[27] X. Tan, S. Su, Z. Huang, X. Guo, Z. Zuo, X. Sun, and L. Li, “Wireless sensor networks intrusion detection based on smote and the random forest algorithm,” Sensors, vol. 19, no. 1, p. 203, 2019.

[28] M. B. ÃGatalkaya, O. KalÄ ˘ spsÄ ´ sz, M. S. AktaÅ§, and U. O. Turgut, “Data feature selection methods on distributed big data processing platforms,” in 2018 3rd International Conference on Computer Science and Engineering (UBMK), Sep. 2018, pp. 133–138.

[29] R. S. B. Krishna and M. Aramudhan, “Feature Selection Based on Information Theory for

Pattern Classification," in 2014 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT), Jul. 2014, pp. 1233–1236.

[30] B. Bonev, "Feature selection based on information theory," Ph.D. dissertation, University of Alicante, Jun. 2010.

[31] J. Li, K. Cheng, S. Wang, F. Morstatter, R. P. Trevino, J. Tang, and H. Liu, "Feature selection: A data perspective," ACM Computing Surveys (CSUR), vol. 50, no. 6, p. 94, 2018.

[32] M. A. Hall, "Correlation-based feature selection for machine learning," Ph.D. dissertation, University of Waikato Hamilton, 1999.

[33] A. Moubayed, M. Injadat, A. Shami, and H. Lutfiyya, "Relationship between student engagement and performance in e learning environment using association rules," in 2018 IEEE World Engineering Education Conference (EDUNINE), Mar. 2018, pp. 1–6.

[34] P. Koch, B. Wujek, O. Golovidov, and S. Gardner, "Automated hyperparameter tuning for effective machine learning," in Proceedings of the SAS Global Forum 2017 Conference, 2017, pp. 1–23.

[35] L. Yang and A. Shami, "On hyperparameter optimization of machine learning algorithms: Theory and practice," Neurocomputing, 2020.

[36] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," Journal of machine learning research, vol. 13, no. Feb, pp. 281–305, 2012.

[37] M. Injadat, A. Moubayed, A. B. Nassif, and A. Shami, "Systematic ensemble model selection approach for educational data mining," Knowledge-Based Systems, vol. 200, p. 105992, 2020.

[38] Injadat, M., Moubayed, A., Nassif, A.B. and Shami, A., 2020. Multi-split optimized bagging ensemble model selection for multi-class educational data mining. Applied Intelligence, 50(12), pp.4506-4528.

[39] L. Bianchi, M. Dorigo, L. M. Gambardella, and W. J. Gutjahr, "A survey on metaheuristics for stochastic combinatorial optimization," Natural Computing, vol. 8, no. 2, pp. 239–287, 2009.

[40] G. Cohen, M. Hilario, and A. Geissbuhler, "Model selection for support vector classifiers via genetic algorithms. an application to medical decision support," in International Symposium on Biological and Medical Data Analysis. Springer, 2004, pp. 200–211.

[41] F. Salo, M. Injadat, A. B. Nassif, A. Shami, and A. Essex, "Data mining techniques in intrusion detection systems: A systematic literature review," IEEE Access, vol. 6, pp. 56 046–56 058, 2018.

[42] A. Moubayed, A. Refaey, and A. Shami, "Software-defined perimeter (sdp): State of the art secure solution for modern networks," IEEE Network, vol. 33, no. 5, pp. 226–233, 2019.

[43] P. Kumar, A. Moubayed, A. Refaey, A. Shami, and J. Koilpillai, "Performance analysis of sdp for secure internal enterprises," in 2019 IEEE Wireless Communications and Networking Conference (WCNC), 2019, pp. 1–6.