Intrusion Watchdog: Enhancing Intrusion Detection System with Machine Learning using 3-way Feature Selection Technique

Nitesh Bharot[1], Priyanka Verma[1], Nisha Ghangare[1], John G. Breslin[1], and Sanjeev Kumar Gupta[2]

[1] Data Science Institute, University of Galway, Ireland
{firstname.lastname}@universityofgalway.ie
[2] Rabindranath Tagore University, Bhopal, India
sanjeevgupta73@yahoo.com

Abstract. With the escalating risk of cyberattacks and breaches in today's interconnected industries, the need for robust cybersecurity measures has become paramount. Effective intrusion detection is a critical aspect of cybersecurity, requiring timely identification and mitigation of malicious activities. This paper explores the application of Machine Learning (ML) techniques to enhance intrusion detection capabilities. ML revolutionizes Intrusion Detection Systems (IDS) by automating the recognition of attacks and patterns within network data, leading to timely identification and mitigation of cyber threats. ML-powered IDS adapt to emerging attack techniques, reducing false positives, handling class imbalance, and utilizing ensemble techniques like Random Forest for improved accuracy. By leveraging these advanced ML algorithms, such as Random Forest and Edited Nearest Neighbor, this research addresses challenges posed by class imbalance and noisy data. A novel ensemble feature selection technique is introduced, boosting the accuracy and efficacy of IDS by retaining relevant features. The comprehensive evaluation shows that the proposed approach outperforms Deep Learning techniques, contributing to improved network security against dynamic cyber threats.

Keywords: Intrusion Detection · Ensemble Techniques · Feature Selection · Machine Learning

1 Introduction

The escalating risk of cyber attacks and breaches has become a paramount concern with the widespread integration of computer networks across diverse industries, including finance, healthcare, and transportation. Ensuring the security of sensitive information, maintaining system reliability, and safeguarding user privacy are all contingent upon robust cybersecurity measures. A critical facet of cybersecurity lies in effective intrusion detection, aimed at the timely identification and mitigation of malicious activities, unauthorized access, and anomalous behavior within network environments [12].

Intrusion Detection Systems (IDS) play a pivotal role in safeguarding network environments, operating through two fundamental architectures: Host-Based IDS (HIDS) and Network-Based IDS (NIDS). These architectures offer distinct advantages and drawbacks, each catering to specific requirements within the realm of cybersecurity [8]. HIDS operates by monitoring and analyzing activities occurring on individual machines. It finds favor among network administrators due to its capacity to access encrypted information while traversing a network. Despite this advantage, managing HIDS proves challenging, necessitating configuration and information management for every host. In contrast, NIDS are specialized intrusion detection devices strategically deployed throughout networks. These devices employ software or hardware components and passively observe traffic flowing across network nodes. NIDS boasts a dual interface, enabling them to both listen to network conversations and manage reporting and control functions.

Conventional approaches to intrusion detection have traditionally leaned on rule-based or signature-based methodologies, which exhibit inherent limitations in discerning novel and intricate threats. The intricate nature and evolving landscape of cyber attacks often result in substantial rates of false positives and false negatives within these techniques.

In response to these challenges, the utilization of Machine Learning (ML) has garnered substantial attention as a promising avenue in the realm of intrusion detection [3]. ML's capacity to autonomously learn patterns and identify anomalies within extensive and intricate datasets has positioned it as a compelling solution to address the inadequacies of traditional methods. Further complexity arises from the uneven distribution of network traffic data. Intrusions represent a rare occurrence compared to the prevalence of regular traffic, posing a significant class imbalance issue that can hinder the accurate identification of intrusions.

Despite these formidable challenges, the potential of ML approaches in addressing intrusion detection within cybersecurity remains substantial [16]. The utilization of advanced ML algorithms such as ensemble methodologies like Random Forest (RF), and data refinement techniques like Edited Nearest Neighbour (ENN), holds the promise of refining the precision and efficacy of IDS. These advanced techniques offer the prospect of mitigating the impact of noisy data and tackling the class imbalance problem, thereby contributing to more accurate and robust intrusion detection capabilities. By leveraging the capabilities of ML, IDS has moved beyond static signatures and embraced dynamic patterns, enabling them to effectively recognize attack patterns [17].

This paper delves into the application of ML techniques for bolstering intrusion detection capabilities, exploring its potential to enhance accuracy, adaptability, and overall network security in the face of dynamic and sophisticated cyber threats. Through comprehensive analysis and evaluation, this research contributes to the advancement of intrusion detection systems, ultimately fostering heightened cyber resilience across critical industries. The main contribution of this paper are :

- This study harnesses the power of RF to bolster the effectiveness of IDS. By employing RF, the research aims to capitalize on its capability to handle complex and diverse datasets while mitigating overfitting.
- It introduces an ensemble 3-way feature selection technique that amalgamates the correlation coefficient method, ANOVA feature selection method, and information gain method. This combined approach aims to identify and retain the most relevant and informative features, thereby augmenting the quality of the data utilized for intrusion detection, and ultimately advancing the accuracy and efficacy of the IDS.
- It makes a significant contribution by incorporating the Edited Nearest Neighbor (ENN) sampling methodology. By integrating ENN into the framework, the study addresses the challenge of imbalanced datasets prevalent in IDS.
- The outcomes of this study underscore its efficacy, revealing that the proposed ensemble feature selection technique, coupled with the ENN sampling methodology and harnessed by the RF algorithm, outperforms Deep Learning (DL) techniques.

2 Related Work

In Wagh et al. [18], the author surveyed the integral role of IDS in identifying system attacks and discerning between normal and attack activities. With the application of ML techniques, IDS has seen substantial advancements in intrusion detection capabilities. It provides a comprehensive survey of various ML approaches within the context of IDS. Additionally, they introduced a system design aimed at enhancing intrusion detection accuracy while minimizing false alarms.

Sudar et al. [14] compared ML techniques like Naive Bayes (NB), Decision Trees (DT), RF, and Multilayer Perceptron (MLP) for IDS within Software-defined networking (SDN) considering accuracy and error rates. Chandre et al. [4] offered a performance analysis of diverse ML and DL techniques for intrusion detection and prevention within wireless sensor networks. Experiments on the WSN-DS dataset reveal superior intrusion detection results from DL, specifically the Convolutional Neural Network (CNN) classifier.

Muna et al. [11] introduce a DL-based anomaly detection technique for IICSs, utilizing TCP/IP packet data and testing on NSL-KDD and UNSWNB15 datasets. Sequential training using deep auto-encoders and neural networks yields enhanced detection rates and fewer false positives compared to recent techniques, making it suitable for real-world IICS environments.

Alessa et al. [2] consolidates the UNSW-NB15 dataset and extends its labels to encompass attack families for multi-classification. DL models' performance is assessed across binary and multi-class categories, with results surpassing related works at 99.59% multi-class and 99.26% binary accuracy.

Studies in [6] employ NB, RF, J48, and ZeroR algorithms on the UNSW-NB15 dataset for cyberattack classification. Additionally, K-MEANS and Expectation Maximization (EM) clustering group data based on attack or normal

attributes. Correlation-based Feature Selection (CFS) optimizes feature subsets. RF and J48 yield the best results (97.59% and 93.78%), offering an effective approach for studying intrusion detection in extensive networks.

Moualla et al. [10] introduces a dynamic, multiclass ML-based network IDS, countering current cyberattacks using the UNSW-NB15 dataset. The approach employs SMOTE for class imbalance, Extra Trees Classifier for feature selection, pre-trained ELM models for attack detection, and a connected layer for combined learning, showcasing superior accuracy, false alarm rates, ROC, and PRCs.

Kanimozhi et al. [9] worked on NIDS, monitors and identifies intrusions within networking. Its study on UNSWNB15 employs feature selection to retain key attributes for enhanced attack detection accuracy and lowered false alarm rate. The approach combines RF with DT using Anaconda3 and Conda. The model effectively identifies normal and attack instances with improved accuracy through DL techniques.

Fuat et al. [5] leveraged DL and ML techniques for successful attack detection and classification on UNSW-NB15 and NSL-KDD datasets. Impressive accuracies, such as 98.6% for UNSW-NB15 two-class and 97.8% for NSL-KDD, underscore ML's pivotal role in IDS. Ahmad et al. [1] proposed feature clusters (Flow, MQTT, TCP) from the UNSW-NB15 dataset, addressing issues like overfitting and imbalance. Employing supervised ML algorithms (RF, SVM, ANN), achieving 97.37% accuracy in multi-class classification. Cluster techniques yield 96.96%, 91.4%, and 97.54% accuracy in Flow & MQTT, TCP, and top feature clusters with RF.

While enhancing efficiency and resource utilization, IoT's expansion has also led to a surge in network attacks, particularly concerning botnet intrusions. Iqbal et al. [7] strives to employ advanced ML techniques, specifically leveraging the PyCaret library, to detect botnet attacks in IoT networks more effectively.

Priya et al. [13] optimized intrusion detection efficiency, by focusing on selecting optimal features and classification methods. Utilizing modern datasets like UNSW-NB-15 and CICDDoS2019, it focused on multi-class attack classification for anomaly detection via ML. Feature selection using the Ranker method and attribute evaluators like Information Gain, Gain Ratio, One Rule, and Correlation were applied. The study also comprehensively analyzes the performance of various ML algorithms.

Verma et al. [15] proposed a Federated Learning-enabled Deep Intrusion Detection (FLDID) framework. It employs a hybrid DL approach comprising CNN, Long Short Term Memory (LSTM), and MLP to tackle IDS. Comprehensive experiments on a publicly available dataset demonstrate the framework's superiority over existing methods for safeguarding smart industries against cyber threats.

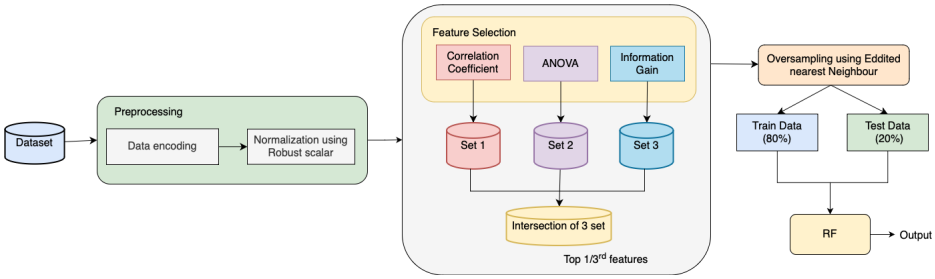

Fig. 1. Architecture of proposed framework

3 Proposed Work

This section deals with the description of the proposed framework. It begins with a description of the proposed framework flow and later describes the various components of the proposed framework.

Intrusion detection is a critical component of modern cybersecurity, aimed at safeguarding computer systems and networks from unauthorized access, malicious activities, and potential threats. With the increasing complexity and diversity of cyberattacks, traditional rule-based methods have proven insufficient to adequately identify and respond to these threats in real-time. ML techniques have developed as revolutionizing the way to defend against evolving cyber threats. ML techniques play a pivotal role in enhancing the accuracy and efficiency of IDS. Therefore we propose Intrusion Watchdog, an effective technique for IDS using the ML technique.

In the initial phase, the data is gathered and preprocessed to be effectively utilized in subsequent calculations and computations. The process involves a series of steps aimed at refining the raw data and ensuring it aligns with the requirements of the designed model. The UNSW-NB15 dataset, chosen for this study, undergoes this preprocessing to eliminate noise, handle categorical data, and normalize values, ultimately enhancing the efficiency of the model. Next, we applied 3 way-feature selection process where we utilized the power of 3 feature selection techniques to refine the dataset. These feature selection techniques, Correlation Coefficient, ANOVA, and Information Gain feature selection, were utilized to effectively analyze the dataset and extract significant features. After amalgamating the results from feature selection we employed ENN as a class balancing technique to resolve class imbalance issues. Finally, these techniques were unified and the system was classified using RF as classifier. Results from RF indicates that our 3-way feature selection mechanism in combination with ENN and RF outperforms various ML and DL classifier showing the efficacy of our proposed approach.

3.1 Preprocessing

The first step involves eliminating unnecessary columns from the UNSWNB15 dataset that might not contribute significantly to the analysis, such as 'dur' and 'state'. With a large number of features in the dataset, streamlining the data helps in reducing complexity and potential noise. Subsequently, given the presence of categorical features in the dataset, a crucial task is to convert them into numeric values. This transformation is achieved through a technique known as feature mapping. This process assigns a unique numeric value to each categorical label, making the data compatible with various ML algorithms. To mitigate the issue of missing values, they were identified and replaced with mean values. This step ensures that the dataset remains comprehensive and minimizes the impact of missing information. Next, for normalization purposes, the RobustScaler technique was employed. RobustScaler facilitates the scaling of features in the dataset to a standardized range, aiding ML algorithms in processing the data more efficiently. This technique is rooted in the computation of the median and Inter-Quartile Range (IQR) of each feature.

$$IQR = Q_3 - Q_1 \tag{1}$$

where Q_1 is the 25^{th} percentile and Q_3 is the 75^{th} percentile of feature values.

$$y = \frac{x - x_{median}}{x_{IQR,75\%} - x_{IQR,25\%}} \tag{2}$$

The median serves as a measure of central tendency, while the IQR quantifies the spread of data, being less sensitive to outliers compared to traditional measures like range or standard deviation. The scaling function in RobustScaler normalizes the data around the median and scales it by the IQR, resulting in data centered at a median of 0 and a range of -1 to 1. This approach offers robustness to the presence of outliers, enhancing the reliability of ML models trained on the processed data. By aligning the data around the median and adjusting for the spread using the IQR, RobustScaler ensures that the scaled data remains more resilient to anomalies and extreme values.

3.2 Feature selection

Furthermore, the incorporation of ensemble-based feature selection techniques is done, namely Correlation Coefficient, Analysis of Variance (ANOVA), and Information Gain. These techniques individually select the top 33% of features from the data. The intersection of the selected features from these three methods determines the final set of features to be used. Each of these techniques plays a unique role in streamlining the dataset and enhancing the relevance of features used in subsequent analyses.

We utilized the Pearson Correlation Coefficient, which quantifies the linear correlation between two continuous variables. A correlation threshold of 10 percent was employed to select relevant features. The correlation coefficient ranges

between -1 and 1 where -1 signifies a perfect negative correlation, 1 represents a perfect positive correlation, and 0 denotes no correlation. This statistical measure assists in identifying relationships between variables and is fundamental in eliminating redundancies or irrelevant features from the dataset.

$$Pearson\,Coefficient = \frac{\sum(x_i - x_{mean})(y_i - y_{mean})}{\sqrt{\sum(x_i - x_{mean})^2 \sum(y_i - y_{mean})^2}} \tag{3}$$

where $x_i\,and\,x_{mean}$ are sample and sample mean.

Also the ANOVA technique was employed to identify the top one-third of features from the dataset. The parameters were set for 33 % of the number of columns in the oversampled class. ANOVA is a statistical methodology that aims to compare means among two or more groups of data. The fundamental principle of ANOVA involves partitioning the overall variation in a dataset into two distinct types of variation: variation observed between groups and variation existing within groups. Furthermore, the application of ANOVA helps in distinguishing the variations that are attributable to differences between groups from those attributed to inherent fluctuations within individual groups. This division of variation is crucial for analyzing the significance of group differences and making informed conclusions based on statistical evidence.

$$Variance\,ratio = \frac{MeanSquare_1}{MeanSquare_2} \tag{4}$$

where $Mean\ Square_1$ = due to groups (between groups) and $MeanSquare_2$ =due to error (within groups, residual mean square)

$$Mean\ Square\,due\,to\,groups = \frac{\sum_{j=1}^{p}(\frac{gT_j^2}{t_i}) - \frac{GT^2}{t}}{p-1} \tag{5}$$

$$Mean\ Square\,due\,to\,error = \frac{\sum_{j=1}^{p}\sum_{i=1}^{t_i} o_{ji}^2 - \sum_{j=1}^{p}\frac{gT_i^2}{t_i}}{t-p} \tag{6}$$

where p = 33% of columns, gT = group total, GT = grand total and t = total observations.

Also to facilitate the process of feature selection, mutual information scores (information gain) were retrieved for all features using the scores attribute of the feature selector. These scores were then sorted in ascending order, and the indices corresponding to the top-k features were extracted. By reversing the order of these indices, the indices of the top-k features in descending order of their mutual information scores were obtained. Information gain is a well-established technique rooted in information theory. It quantifies the amount of information that a specific feature contributes to the understanding of the target variable.

$$Information\,Gain = -\sum p(a)log(p(a)) \tag{7}$$

To conclude the process of final feature selection, the top 33 % of features with the highest mutual information scores were chosen. These features were

considered for further computational analysis. This systematic methodology ensures that the features with the most substantial contributions in explaining variations in the target variable are prioritized for subsequent analyses.

3.3 Class Balancing

Class balancing is crucial to prevent biased ML models. It ensures equal representation of all classes, enhancing predictive accuracy and preventing the dominance of majority classes, leading to more effective and fair decision-making in various applications. In our methodology, we have incorporated the Edited Nearest Neighbors (ENN) algorithm, which is a modified version of the K-Nearest Neighbors (KNN) algorithm, specifically designed for the purpose of oversampling. The ENN technique, originally developed by Wilson in 1972, aims to refine the dataset by identifying potentially misclassified or noisy instances through a localized analysis of their neighbors.

Through the application of ENN, the study aims to alleviate the class imbalance issue by selectively removing redundant and noisy instances, thereby fostering a more balanced and representative dataset. This approach not only enhances the overall performance of the IDS but also contributes to the robustness and reliability of the results.

The ENN algorithm operates as follows: for each data point in the dataset, it identifies its KNN based on a given distance metric. Subsequently, it examines whether the majority class of the K-nearest neighbors aligns with the class of the central data point. If there is a discrepancy between the central data point's class and the majority class of its neighbors, the algorithm removes both the central data point and its KNN from the dataset. The mathematical steps for ENN involve computing distances between instances (using a distance metric like Euclidean distance) and evaluating class labels to determine which instances to remove. The actual implementation involve computing the distance matrix, identifying nearest neighbors, and comparing class labels.

3.4 Classification using RF

In our proposed approach, we utilized RF as the classifier algorithm for IDS. RF algorithm helps address the imbalanced nature of intrusion detection datasets and handles complex relationships between features and classes. By leveraging the ensemble nature of RF, we aimed to achieve higher accuracy and robustness compared to individual models. RF has ability to reduce overfitting through ensemble averaging, provide insights into feature importance, handle non-linearity, and effectively manage imbalanced datasets by aggregating predictions from multiple decision trees. Its capacity to handle complex relationships, reduce bias, and offer stable performance, along with its ease of tuning and parallelization, makes it a robust choice. Moreover, RF's resilience to outliers and consistent performance across diverse datasets further elevate its appeal as a versatile and reliable algorithm for classification tasks.

Through its ensemble nature and inherent feature selection mechanisms, RF contributes to enhanced accuracy in detecting both known and novel attack patterns, thereby fortifying the IDS against evolving cyber threats. This integration of RF underscores its potential as a valuable tool in the arsenal of intrusion detection, promising improved performance and heightened cybersecurity measures.

4 Results & Discussion

This section analyzes the results of the proposed framework tested on the UNSWNB15 dataset. The widely used UNSW-NB15 dataset serves as a valuable resource for assessing the efficacy of ML techniques in intrusion detection. Developed by the University of New South Wales (UNSW), this Australian dataset emulates real network traffic scenarios in a controlled environment. Comprising 2.5 million instances, it encompasses diverse network traffic sources, capturing both typical activities and various cyberattack types. The dataset is divided into 49% regular traffic and 51% diverse attacks, including denial-of-service (DoS), IDS probes, remote-to-local (R2L), and user-to-root (U2R) attacks. As a benchmark, it aids in evaluating the performance of intrusion detection methodologies and with table 1 giving the description of classes, their labels, and counts in the training and testing set.

Table 1. Dataset Description

Class Name	Class Label	Training Set	Testing Set
Analysis	0	2000	677
Backdoor	1	1746	583
DoS	2	12264	4089
Exploits	3	33393	11132
Fuzzer	4	18184	6062
Generic	5	40000	18871
Normal	6	56000	37000
Reconnaissance	7	10491	3496
Shellcode	8	1133	378
Worms	9	130	44
Total	10	175341	82332

4.1 Comparison with ML techniques

Table 2 offers an extensive evaluation of various intrusion detection techniques, shedding light on their effectiveness in different scenarios. The evaluation encompasses multiple metrics, including precision, recall, and F1-score, across a range of classes (0 to 9). These metrics are crucial in assessing the performance

of these techniques, as they provide insights into the techniques' ability to correctly identify instances of both normal and intrusive behavior.

Among the techniques considered, the proposed framework demonstrates a high precision of 0.99, indicating its capability to accurately identify true positive instances. Additionally, its recall of 1 for class 0, class 4, and class 6, among others, signifies that it successfully captures a significant portion of actual positive instances. The f1-scores further highlight the balance between precision and recall, with values above 0.9 for most classes. The weighted average is 0.99, indicating the overall proficiency of the proposed framework across the dataset. This table could be used to compare techniques and select the most appropriate one based on the specific requirements of their systems. Ultimately, this evaluation contributes to enhancing the security and robustness of systems against potential cyber threats by identifying the techniques that excel in detecting and mitigating anomalies and attacks.

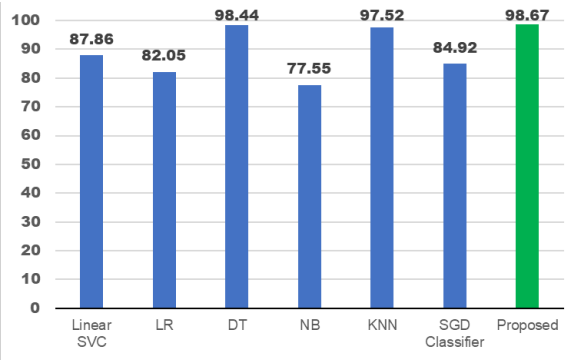

Fig. 2. Comparison of proposed work with ML techniques

Fig. 2 showcases a comprehensive evaluation of different machine learning classifiers in terms of their accuracy. Each classifier's performance is crucial for assessing its suitability in various applications. Among the classifiers assessed, the Linear SVC classifier achieved an accuracy of 87.86%, indicating a decent performance. LR follows with an accuracy of 82.05%, demonstrating its applicability in certain scenarios. Notably, the DT classifier exhibited a high accuracy rate of 98.44%, making it particularly promising for accurate predictions. In contrast, the NB classifier achieved a relatively lower accuracy of 77.55%, implying its limitations in certain complex contexts. It clearly indicated that the proposed framework outperforms other methods by achieving an accuracy of 98.67%. This occurs as the proposed framework employs effective data preprocessing and feature-selection techniques and works by creating an appropriate amount of balanced data.

Table 2. Multi-class comparison with ML classifiers

Technique	Perameters	Class 0	1	2	3	4	5	6	7	8	9	Weighted Avg
Lineear Regression	Precision	0	0	0	0.05	0.88	0.8	0	0	0	0	0.7
	Recall	0	0	0	0	1	0.96	0	0	0	0	0.82
	F1-score	0	0	0	0.01	0.94	0.88	0	0	0	0	0.76
Linear SVC	Precision	0	0	0.82	0.75	0.89	0.91	0.41	0	0	0	0.86
	Recall	0	0	0.57	0.24	1	0.94	0.32	0	0	0	0.88
	F1-score	0	0	0.67	0.36	0.94	0.93	0.36	0	0	0	0.86
Decision Tree	Precision	1	0.73	0.96	0.883	1	0.99	1	0.89	0.69	0.68	0.98
	Recall	0.5	0.82	0.95	0.81	1	0.99	0.99	0.94	0.6	0.94	0.98
	F1-score	0.67	0.77	0.96	0.82	1	0.99	0.99	0.92	0.64	0.79	0.98
Naïve Bayes	Precision	0	0	0.35	0.2	0.96	0.87	0.1	0.02	0	0.33	0.8
	Recall	0	0	0.21	0.13	1	0.81	0.11	1	0	0.06	0.78
	F1-score	0	0	0.27	0.16	0.98	0.84	0.1	0.04	0	0.11	0.78
KNN	Precision	0.67	0.81	0.9	0.77	1	0.98	0.99	0.79	0.64	0.48	0.97
	Recall	1	0.35	0.95	0.77	1	0.98	0.99	0.66	0.3	0.81	0.98
	F1-score	0.8	0.49	0.93	0.77	1	0.98	0.99	0.72	0.41	0.6	0.97
SGD Classifier	Precision	0	0	0	0.07	0.96	0.8	0.63	0	0	0	0.76
	Recall	0	0	0	0	1	0.96	0.67	0	0	0	0.85
	F1-score	0	0	0	0.01	0.98	0.87	0.65	0	0	0	0.8
Proposed	Precision	1	0.75	0.97	0.87	1	0.99	1	0.91	0.83	0.88	**0.99**
	Recall	1	0.82	0.97	0.81	1	0.99	0.99	0.86	0.5	0.94	**0.99**
	F1-score	1	0.78	0.97	0.84	1	0.99	1	0.88	0.62	0.91	**0.99**

The KNN classifier, known for its ability to capture local patterns, demonstrated an impressive accuracy of 97.52%. This reinforces its efficacy in scenarios where proximity-based learning is advantageous. The Stochastic Gradient Descent (SGD) Classifier, often employed in large-scale machine learning tasks, achieved an accuracy of 84.92%, which may indicate its ability to handle large datasets efficiently.

The standout performance is seen in the proposed framework, which achieved an accuracy of 98.67% suggesting that it offers a robust solution that outperforms the other classifiers in this context.

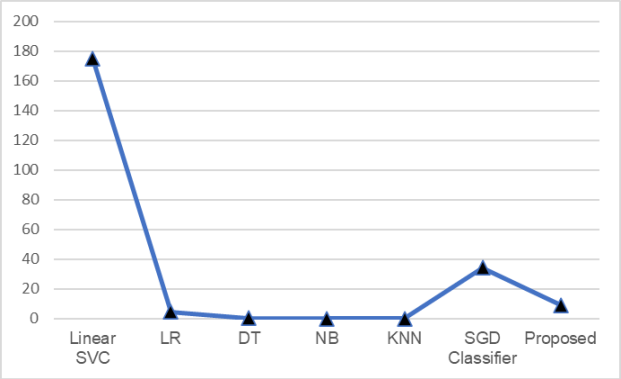

Fig. 3. Time comparison of proposed work with ML techniques (in sec)

Fig. 3 presents the training times of various ML classifiers. Training time is a crucial factor in model development, influencing efficiency and resource allocation. Among the classifiers, Linear SVC demonstrated the highest training time of 174.7097 seconds, possibly due to its computational complexity. LR and the proposed framework show relatively shorter training times of 4.7402 seconds and 9.1421 seconds, respectively, indicating their efficiency in model building. DT classifier stands out with an incredibly short training time of 0.2696 seconds, showcasing its speed in constructing DT but Table 2 indicates that it lags to perform well in case of multi-class classification. Proposed framework on other hand though takes more time than DT, LR, NB and KNN but it shows higher performance in terms of accuracy and multi-class classification.

Table 3. Multi-class comparison with DL techniques

Technique	Perameters	Class 0	1	2	3	4	5	6	7	8	9	Weighted Avg
Time Distributed (Hybrid)	Precision	0	0	0.84	0.66	1	0.95	0.89	0	0.83	0	0.94
	Recall	0	0	0.82	0.45	1	0.98	0.99	0	0.17	0	0.95
	F1-score	0	0	0.83	0.54	1	0.96	0.94	0	0.28	0	0.94
CNN+LSTM	Precision	0	0	0.83	0.9	1	0.95	0.88	0	0	0	0.94
	Recall	0	0	0.89	0.31	1	0.98	0.99	0	0	0	0.95
	F1-score	0	0	0.86	0.47	1	0.96	0.93	0	0	0	0.94
CNN+LSTM+MLP	Precision	0	0	0.8	0.83	1	0.95	0.9	0	0.5	0	0.94
	Recall	0	0	0.87	0.38	1	0.97	0.98	0	0.03	0	0.95
	F1-score	0	0	0.83	0.52	1	0.96	0.94	0	0.06	0	0.94
CNN	Precision	0	0.45	0.86	0.62	1	0.93	0.9	0	0	0	0.94
	Recall	0	0.73	0.76	0.29	1	0.97	0.96	0	0	0	0.94
	F1-score	0	0.56	0.81	0.4	1	0.95	0.93	0	0	0	0.94
LSTM	Precision	0	0.33	0.78	0.53	1	0.92	0.62	0	0	0	0.91
	Recall	0	0	0.75	0.3	1	0.97	0.96	0	0	0	0.92
	F1-score	0	0.01	0.77	0.39	1	0.94	0.62	0	0	0	0.91
MLP	Precision	0	0	0.72	0.11	0.95	0.89	0.64	0	0	0	0.86
	Recall	0	0	0.56	0	1	0.94	0.78	0	0	0	0.89
	F1-score	0	0	0.63	0	0.98	0.92	0.7	0	0	0	0.87
Proposed	Precision	1	0.75	0.97	0.87	1	0.99	1	0.91	0.83	0.88	**0.99**
	Recall	1	0.82	0.97	0.81	1	0.99	0.99	0.86	0.5	0.94	**0.99**
	F1-score	1	0.78	0.97	0.84	1	0.99	1	0.88	0.62	0.91	**0.99**

4.2 Comparison with DL techniques

The table 3 delves into precision, recall, and F1-score for each DL technique, providing a comprehensive overview of their effectiveness in identifying normal and anomalous network activities within the UNSW-NB15 dataset.

The proposed framework displays remarkable performance across multiple classes. It achieves high precision, recall, and f1-scores for several classes, indicating its strong ability to identify both normal and intrusive activities. These high scores are consistently observed throughout the table, reflecting the technique's strong performance overall.

This evaluation serves as a valuable resource for intrusion detection practitioners and researchers, offering insights into the strengths and weaknesses of different techniques when applied to the UNSW-NB15 dataset. The provided metrics empower decision-makers to select techniques that align with their specific requirements and security needs. Ultimately, this analysis contributes to enhancing network security by aiding in the selection of effective intrusion detection strategies.

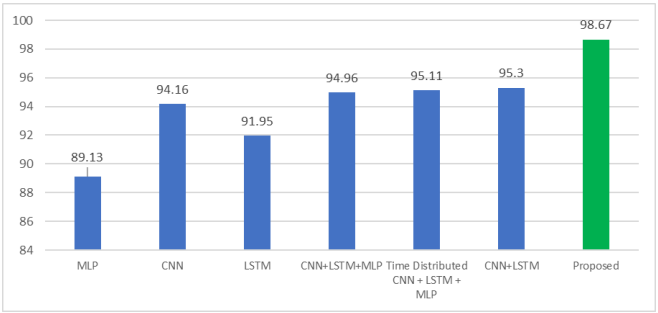

Fig. 4. Comparison of proposed work with DL techniques

Fig. 4 showcases accuracy values of different DL models, reflecting their effectiveness in making accurate predictions. Notably, the proposed framework stands out with an accuracy of 98.67%, surpassing other DL models. It indicates its potential to perform better than DL models as it achieved around 3% increase in accuracy values than DL models.

Fig 5 presents training time data for various DL models, shedding light on their computational efficiency during the model training process. MLP required a training time of 32.1376 seconds, indicating its relatively fast convergence whereas CNN followed with a training time of 45.3643 seconds, demonstrating its efficiency in handling image-based data. LSTM exhibited a longer training time of 245.137 seconds, attributed to its sequential data processing nature. The combined model of CNN, LSTM, and MLP (CNN+LSTM+MLP) took 157.258 seconds to train, highlighting the additional computational requirements when integrating multiple architectures. Notably, the proposed framework exhibited a

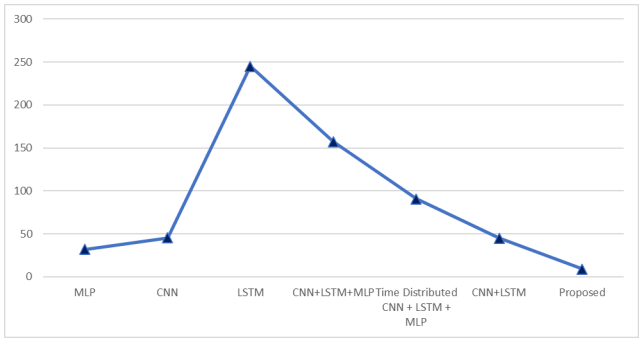

Fig. 5. Time comparison of proposed work with DL techniques

significantly shorter training time of 9.1421 seconds, indicating its efficiency in convergence during the training phase. Its quick training time further supports its suitability for practical applications in IDS.

4.3 Comparison with existing solutions

Table 4. Comparison with state of art techniques

Methods	Predicted Multi-class classification	Accuracy	Precision	Recall	F1-Score	Training Time
[2]	Yes	99.59	-	-	-	-
[6]	No	97.6	97.6	97.6	97.6	-
[10]	Yes	98.191	-	-	-	-
[9]	No	96	97	96	97	-
[5]	Yes	98.6	98	98	98	-
[1]	Yes	97.37	-	-	-	-
Proposed	Yes	98.67	98.66	98.67	98.66	9.1421

The comparative analyses of different methods used for multi-class classification are described in Table 4. The methods are evaluated based on various performance metrics. Methods that showcased the concept of multi-class classification are marked as “Yes” while rest are marked as “No”.

The proposed method, as indicated, is capable of multi-class classification. It achieves an accuracy of 98.67% and demonstrates high precision, recall, and F1-Score values of 98.66%, 98.67%, and 98.67% respectively with a training time of 9.1421 seconds. Overall, the table provides a comprehensive overview of the performance of various methods and highlights the effectiveness of the proposed approach in terms of accuracy and predictive capabilities.

5 Conclusion

Proposed framework presents a comprehensive approach for enhancing intrusion detection accuracy in datasets using RF as the classifier. The proposed methodology combines the power of ENN for oversampling, a 3-way feature selection technique involving Correlation Coefficient, ANOVA, and Information Gain, and the robustness of Random Forest. This ensemble approach demonstrates superior performance in identifying intrusions, effectively addressing class imbalance, and optimizing feature representation for improved model generalization, achieving an accuracy of 98.67%. It showcases a robust framework for enhancing intrusion detection systems, contributing to the advancement of cybersecurity applications.

Acknowledgment

The research leading to these results has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No. 101100680 (GN5-1), and also by grants from Science Foundation Ireland under Grant Numbers 16/RC/3918 and 12/RC/2289_P2. For the purpose of Open Access, the author has applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission.

References

1. Ahmad, M., Riaz, Q., Zeeshan, M., Tahir, H., Haider, S.A., Khan, M.S.: Intrusion detection in internet of things using supervised machine learning based on application and transport layer features using unsw-nb15 data-set. EURASIP Journal on Wireless Communications and Networking **2021**(1), 1–23 (2021)
2. Aleesa, A., Younis, M., Mohammed, A.A., Sahar, N.: Deep-intrusion detection system with enhanced unsw-nb15 dataset based on deep learning techniques. Journal of Engineering Science and Technology **16**(1), 711–727 (2021)
3. Bharot, N., Verma, P., Sharma, S., Suraparaju, V.: Distributed denial-of-service attack detection and mitigation using feature selection and intensive care request processing unit. Arabian Journal for Science and Engineering **43**, 959–967 (2018)
4. Chandre, P.R., Mahalle, P.N., Shinde, G.R.: Deep learning and machine learning techniques for intrusion detection and prevention in wireless sensor networks: comparative study and performance analysis. Design frameworks for wireless networks pp. 95–120 (2020)
5. Fuat, T.: Analysis of intrusion detection systems in unsw-nb15 and nsl-kdd datasets with machine learning algorithms. Bitlis Eren Üniversitesi Fen Bilimleri Dergisi **12**(2), 465–477 (2023)
6. Hammad, M., El-medany, W., Ismail, Y.: Intrusion detection system using feature selection with clustering and classification machine learning algorithms on the unsw-nb15 dataset. In: 2020 International Conference on Innovation and Intelligence for Informatics, Computing and Technologies (3ICT). pp. 1–6 (2020). https://doi.org/10.1109/3ICT51146.2020.9312002

7. Iqbal, F.B., Biswas, S., Urba, R., et al.: Performance analysis of intrusion detection systems using the PyCaret machine learning library on the UNSW-NB15 dataset. Ph.D. thesis, Brac University (2021)
8. Jain, J.K., Waoo, A.A.: An artificial neural network technique for prediction of cyber-attack using intrusion detection system. Journal of Artificial Intelligence, Machine Learning and Neural Network (JAIMLNN) ISSN: 2799-1172 **3**(02), 33–42 (2023)
9. Kanimozhi, V., Jacob, P.: Unsw-nb15 dataset feature selection and network intrusion detection using deep learning. International Journal of Recent Technology and Engineering **7**(5) (2019)
10. Moualla, S., Khorzom, K., Jafar, A.: Improving the performance of machine learning-based network intrusion detection systems on the unsw-nb15 dataset. Computational Intelligence and Neuroscience **2021**, 1–13 (2021)
11. Muna, A.H., Moustafa, N., Sitnikova, E.: Identification of malicious activities in industrial internet of things based on deep learning models. Journal of information security and applications **41**, 1–11 (2018)
12. Pareriya, R., Verma, P., Suhana, P.: An ensemble xgboost approach for the detection of cyber-attacks in the industrial iot domain. Big Data Analytics in Fog-Enabled IoT Networks: Towards a Privacy and Security Perspective p. 125 (2023)
13. Priya Devi, A., Johnson Singh, K.: A machine learning approach to intrusion detection system using unsw-nb-15 and cicddos2019 datasets. In: Smart Computing Techniques and Applications: Proceedings of the Fourth International Conference on Smart Computing and Informatics, Volume 1. pp. 195–205. Springer (2021)
14. Sudar, K.M., Deepalakshmi, P.: Comparative study on ids using machine learning approaches for software defined networks. International Journal of Intelligent Enterprise **7**(1-3), 15–27 (2020)
15. Verma, P., Breslin, J.G., O'Shea, D.: Fldid: Federated learning enabled deep intrusion detection in smart manufacturing industries. Sensors **22**(22), 8974 (2022)
16. Verma, P., Breslin, J.G., O'Shea, D., Mehta, N., Bharot, N., Vidyarthi, A.: Leveraging gametic heredity in oversampling techniques to handle class imbalance for efficient cyberthreat detection in iiot. IEEE Transactions on Consumer Electronics (2023). https://doi.org/10.1109/TCE.2023.3319439
17. Verma, P., Tapaswi, S., Godfrey, W.W.: A request aware module using cs-idr to reduce vm level collateral damages caused by ddos attack in cloud environment. Cluster Computing pp. 1–17 (2021)
18. Wagh, S.K., Pachghare, V.K., Kolhe, S.R.: Survey on intrusion detection system using machine learning techniques. International Journal of Computer Applications **78**(16), 30–37 (2013)