International Journal of Applied Mathematics
Volume 38 No. 7s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

REVOLUTIONARY INTRUSION DETECTION: HFS-MLSTM WITH HYBRID FEATURE SELECTION AND ATTENTION-DRIVEN BILSTM MODEL

[1]J.Archana, [2] Dr. S. Kamalakkannan
[1]Research Scholar, Department of Computer science, Vels Institute of Science, Technologies and Advanced Studies (VISTAS), Pallavaram, Chennai.
E-mail: archanaartistryacademy@gmail.com

[2](Corresponding Author), Professor, Department of Computer Applications, School of Computing Sciences, Vels Institute of Science, Technologies and Advanced Studies (VISTAS), Pallavaram, Chennai. E-mail: kannan.scs@vistas.ac.in

Abstract

Intrusion detection systems (IDS) are critical for maintaining the security of network environments. The increasing sophistication and frequency of cyber-attacks necessitate the development of advanced IDS capable of identifying and mitigating threats in real-time. Traditional IDS models often struggle with high false positive rates, computational inefficiencies, and the inability to adapt to evolving attack patterns. These limitations highlight the need for more robust and intelligent detection mechanisms. This paper introduces a novel approach named HFS-MLSTM (Hybrid Feature Selection and Multi-Layer Long Short-Term Memory) integrated with an attention-driven BiLSTM (Bidirectional Long Short-Term Memory) model to enhance intrusion detection (ID) capabilities. The proposed method leverages hybrid feature selection techniques, combining Mutual Information (MI) and Recursive Feature Elimination (RFE), to identify the most relevant features for accurate detection. The BiLSTM model, augmented with an attention mechanism, improves the model's ability to focus on critical aspects of the input data, leading to more precise detection of anomalous activities. We have evaluated the effectiveness of our HFS-MLSTM using standard benchmark datasets and shown that it has outperformed traditional IDS approaches by 99% accuracy. This innovative model offers a promising solution for real-time ID, ensuring robust network security and resilience against evolving cyber threats.

Keywords: Intrusion Detection, Hybrid Feature Selection, HFS-MLSTM, BiLSTM, Attention Mechanism, Mutual Information, Recursive Feature Elimination, Cybersecurity, Anomaly Detection, Network Security, Real-time Detection, Deep Learning (DL), Machine Learning (ML),

1. Introduction

Intrusion detection forms an integral part of security as it checks for unauthorized entry/information system breaches. These help a lot in detecting malicious activities serving as the front end systems by observing network traffic, system operations, and user behaviour. In the current evolving and sophisticated cyber threat landscape, securing not only critical

Received: August 01, 2025

Volume 38 No. 7s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

systems but also large enterprises can never be fully ensured without using IDS [1] [2] [3]. An IDS aspires to detect real-time security incidents and inform system administrators, which makes it possible for proper response enforced immediately. This ability is the cornerstone of security measures in protecting confidential data and compliance, recovering from a disastrous situation, keeping environment maintained safely as well as contributing to looking over all necessary matters when it comes to everything digital related.For IT security, securities point and components such as IDS have a long history of helping secure networks, systems, and data against unauthorized malicious entities. It encompasses practices, technologies, and policies appropriate for the management of digital data. In this platform, IDS are essential by having the ability to monitor and analyse system activities, alert against any abnormality in combination with registering steps corresponding with advanced protection mechanisms [4] [5]. The addition of IDS to cybersecurity plans helps with identifying intrusive events before they occur, reducing the dangers of data breaches and system compromises.

IDS cover a large portion of the cybersecurity spectrum in an organization. One way they do this is by monitoring network traffic and system logs for behaviour consistent with a security threat - 24/7. IDS use signature-based detection; thus, by including patterns relating to known attacks, IDS can identify possible indicators of compromise, making it easier for IDSs to spot digital threats [6] [7]. Anomaly-based detection enables detecting abnormal behaviours that deviate from established baselines. By combining these two modes of operation, there is more than enough redundancy in the approach to make sure that whatever vulnerabilities they discover today will still be covered tomorrow. IDS still sends vital notifications capable of providing security teams visibility into their alerts, which can assist them with deep investigation through logs, better understanding attack vectors to answer what happened, and taking necessary actions much sooner. Countless statistics further underscore how critical IDS utilization is when it comes to cybersecurity.Ponemon Institute study revealed that companies using IDS saw a 27% reduction in data breaches incidents against those who did not deploy such systems [8] [9]. Along with that, the mean time to detect a breach was reduced by half, which shows how efficient IDS is in early threat detection. The size of the market for IDS is expected to increase with a CAGR (compound annual growth rate) of 12%, reveals Gartner, due to their growing importance in an overall cybersecurity framework. This data illustrates just how important IDS are not only for identifying but also protecting against security attacks [10] [11].

IDS are a combination of signature-based detection and advanced ML or AI techniques [12] [13]. These technologies add to the power of IDS for catching modern threats that traditional methods could overlook, with ML algorithms capable of going through an increased amount of data and recognizing red flags while AI-based systems can adjust themselves according to new potentially credulous threat which they keep on detecting by coordinating among them every single piece of information coming out. It is a more advanced mechanism used in IDS since the system learns the normal activity upon which it establishes a specific profile and deviations from that norm are reported. Therefore, innovations in detection mechanisms are important to guarantee IDS stay relevant amidst the changing threat landscape [14] [15]. IDS

Received: August 01, 2025

International Journal of Applied Mathematics
Volume 38 No. 7s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

will only work as expected if they are part of a larger cybersecurity initiative incorporating preventive measures, incident response, and recovery. Security devices such as firewalls and antivirus software are precautionary measures because IDS is compartmentalized to the third layer of defence. Incident response plans ensure that when an IDS alert is tripped, the appropriate action(s) to neutralize any threat can be taken in a timely manner. Disaster recovery plans are specifically designed to resume systems immediately and maintain at least a minimal level of operation during an outage [16]. Used within a comprehensive cybersecurity strategy, IDS is one of the core elements to helping organizations prepared for cyber threats and put them in a stronger place than they would otherwise be without making use of detecting, responding, and recovery from potential breaches. Figure 1 shows the IDS.

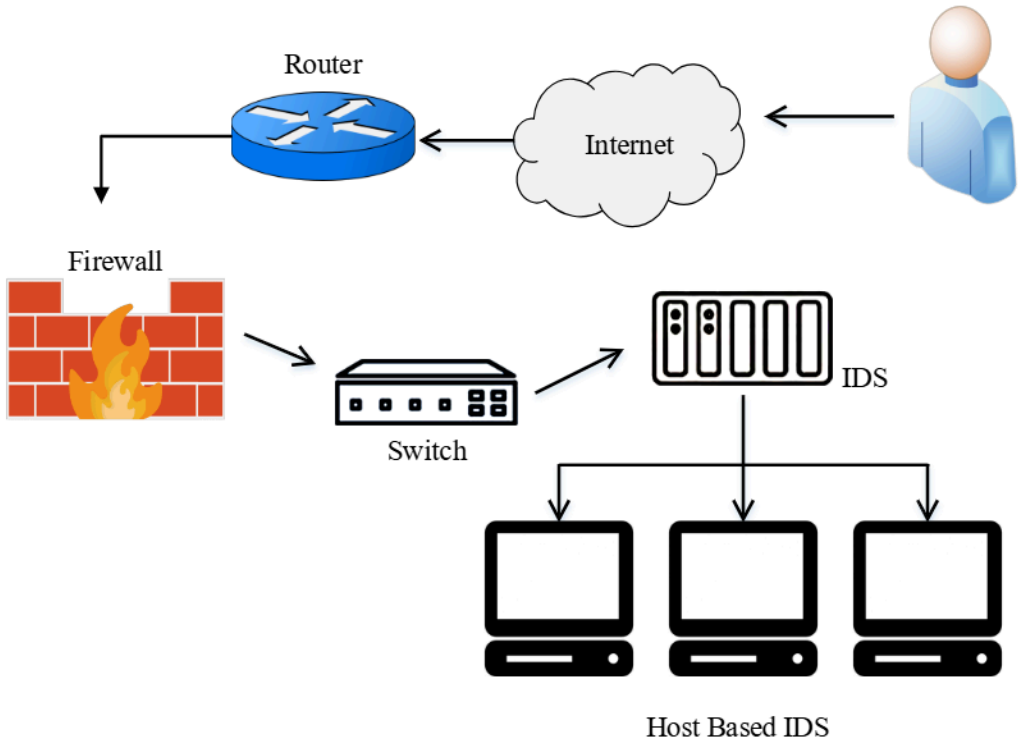

Figure 1. Intrusion Detection System

IDS are of two types, Network-based IDS (NIDS) and Host-based IDS (HIDS). NIDS are placed in some areas of the network where they can both watch traffic and detect symptoms of a malicious scan. HIDS, on the other hand, are installed directly onto hosts like individual devices, and they analyse activity exclusively occurring only to that system, including file modifications, log entries, and calls made by a process [17]. More recently, these have been replaced by NIDS and HIDS-centric IDS solutions, which are integrated with a larger security architecture platform. In addition, these systems leverage numerous detection methods from signature-based (identifying recognizable patterns of malicious activity) to anomaly-based detections that detect deviations in behaviours. Integration of ML or AI, which enhances the power to catch new and unknown threats, helped in better detection by IDS.

Received: August 01, 2025

International Journal of Applied Mathematics
Volume 38 No. 7s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

The effectiveness of IDS can be demonstrated with the help of different statistics. Recent studies show that organizations having an implemented robust IDS are far less likely to fall victim to severe breaches than those without one. There have been reports put out by the Ponemon Institute stating that organizations benefited from a 27% reduction in data breaches while using IDS. Additionally, the report also showed that intrusions detected by IDS resulted in a 46% reduction in mean time to detect an intrusion compared with cases of only implemented IPS, thus confirming its necessity for getting early detection and response [18]. Excitingly beginning to see the broader catch on and recognize IDS as a critical part of an overall cybersecurity strategy; in fact, growth in this category is projected at 12% CAGR by Gartner. These very metrics indicate the necessity for more investment in an intricate ID system since every data receives a steady increase across digital grounds.

Detection mechanisms in IDS are crucial for the identification and response to cyber threats. IDS uses signature-based detection as one of the first methods. It involves fundamentally matching incoming data against a database of known threat signatures, and this tends to be excellent for catching known attacks. But its problem is that it cannot identify new low-threats[19] [20]. In response to this, we developed the detection method based on anomaly. These techniques provide a picture of deviation from the average behaviour in normal system functioning as a context for estimation. Certain outputs that reside outside expected modules are considered and flagged under nervous conditions. Detection of anomaly: such as new attacks that do not correspond to any known signature. Further, heuristic and behaviour-based detection have been developed with advanced algorithms and ML which learns the pattern of a device type to predict potential security incidents. All these new creative cyber threats are what the updated detection methods come to safeguard IDSs against. Therefore, due to the rise of cyber-attacks and their frequency within networks, advanced IDS are developed in order for them to most effectively detect threats as they happen. Conventional IDS models have faced issues such as a very high rate of false positives, computational inefficiency and outdated attack patterns for being too rigid to adapt against an evolving line of attacks that raise the requirements for more resilient or clever detection mechanisms. In this article a proposed concept named HFS-MLSTM is introduced used with attention based BiLSTM. Using the CSE-CIC-IDS2018 dataset, which is a large up-to-date network traffic dataset for IDS; together with Z-Score Normalization before training and testing of ML models in order to get more consistent data preposessing results than using Min-Max normalization so that they are generally comparable.

Main Contribution of the Work

- The integration of MI and RFE ensures the selection of the most relevant features, enhancing detection accuracy while reducing computational overhead.
- The use of a MLSTM model combined with a BiLSTM model provides robust sequential data processing capabilities.
- Incorporating an attention mechanism within the BiLSTM model allows the system to focus on critical aspects of the input data, leading to more precise anomaly detection.

Received: August 01, 2025

International Journal of Applied Mathematics
Volume 38 No. 7s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

- Utilizing Z-Score Normalization for feature scaling ensures consistent and improved performance of the model on the CSE-CIC-IDS2018 dataset.
- The model's architecture and feature selection techniques enable efficient and effective real-time ID. The model provides a scalable solution adaptable to various network environments, ensuring robust protection against evolving cyber threats.

The rest of this paper is organized as follows: Section 2 reviews state-of-the-art ID studies, which will underline the issue that needs to be detected. In Section 3, the HFS-MLSTM model that combines MI and RFE with MLSTM-based and attention-driven BILSTM architectures is presented. Section 4 explains how the experiments were carried out, and then compares these results to state-of-the-art models. The conclusion in Section 5 concludes the paper, key results and motivates future work.

Related Works

Sites assault through the net are pretty common as they may be easily accessible and lack security. It is essential for an IDS to be able to detect attacks from various types of network traffic with a highly efficient traffic classification system. Previous research has developed an IDS that can take multiple datasets for various types of attacks by using ML classification methods. The IDS dataset from the Canadian Institute for Cyber Security (CIC-IDS2017) was used as a case study to evaluate online assaults [21]. The collection contains 80 characteristics of recent attacks, based on a McAfee study in 2016. Three ML approaches, including naïve bayes (NB), k-nearest neighbor (KNN), and random forests (RF), were compared in this study. The main purpose was to find a proper ML method along with the model for online assaults (IDS). The test examines how well the three algorithms (RF, KNN, and NB) can identify suspicious traffic. According to the data, average accuracy in RF is significantly higher compared with NB and KNN. The KNN algorithm managed to score an average accuracy of 99.4916% during the testing phase, which was higher than scores with other algorithms. However, compared with other algorithms, recall and accuracy were consistently at 100% using RF and KNN. Ultimately, the most accurate algorithms to detect IDS web attacks were RF and KNN.

Cybersecurity is now a major issue on a worldwide scale. When it comes to safeguarding networks that are interdependent, IDS are crucial for spotting suspicious activity and individuals. When it comes to IDS, behaviour analysis powered by ML offers a lot of promise for spotting anomalies, harmful activity, and evolving cyber threats. However, dimension reduction becomes a more challenging issue while training ML models when the amount of data increases. They address this by introducing a novel ML-based model for network ID. Random Oversampling (RO) to correct data imbalance, Stacking Feature Embedding to leverage clustering findings, and Principal Component Analysis (PCA) to decrease dimensions are some of the methods used by this approach, which is designed to deal with big and unbalanced datasets [22]. The performance of this model is evaluated using three cutting-edge benchmark datasets: UNSW-NB15, CIC-IDS-2017, and CIC-IDS-2018. In their tests on the UNSW-NB15 dataset, they found that the RF model had a 99.59% success rate while the ET model had a 99.95% success rate. Furthermore, the CIC-IDS2017 dataset

Volume 38 No. 7s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

reaches 99.99% accuracy with DT, RF, and ET models, whereas the CIC-IDS2018 dataset reaches 99.94% accuracy with DT and RF models. These performance findings routinely outperform the state-of-the-art, indicating that network ID has made significant progress. With this success, the proposed approach works as intended, allowing us to detect and prevent breaches into networks through precise monitoring of traffic.

The goal of creating the IDS was to stop harmful network activity. But current IDS models have problems being very accurate and are prone to making mistakes in detection. Therefore, a novel Vulture-based Deep Belief System (VbDBNS) was developed to improve the efficiency of ID by the observation of their behaviour and characteristics. The system was trained and validated using the NSL-KDD database. Afterwards, 1-N encoding is used to convert the categorical data variables and standardise the dataset using Min-Max normalisation [23]. Also, the 42 most important attributes from the pre-processed database were extracted using feature extraction. The vulture's enhanced fitness makes it able to detect and track the intruder at all times. In addition, a module for classifying data as either benign or malicious was developed using VbDBN. The study's experimental results show that the planned methodology outperformed the traditional approaches in several metrics (recall, accuracy, precision, F1-score, etc.).

Decisions about data security and communications have certainly been made more soberly today due to the fear of attacks prevalent in our environment. Among others, network intrusions and assaults may affect social welfare, economic concerns, or the storage of data. For this reason, ID presents a broad field of research for which many different techniques have been developed over the years. This is so complicated that network tasks like discovering and separating different attacks from many attacks [24]. This specific kind of analysis routes the network security threats and problems to be resolved on that basis, basing it upon some techniques which can now fathom identification using. Reviewing traditional methods and datasets for integrating open source malware scanning software into NIDS is the primary goal of this study. It also analyses and analyses NIDS methods according to criteria for attack, validation, detection, deployment, and building. This discusses NIDS strategies based on MLand DL, and then discusses potential areas for future research for both known and undiscovered threats.

When it comes to keeping networks safe, few mechanisms are as effective as the ID process. When new types of attacks emerge, traditional methods of attack detection, such as signature-based detection, become quite vulnerable. Therefore, an IDS utilising DL techniques is the most prominent response to threats to cloud security. The cloud traffic is represented as input attributes in the behaviour-based IDS. Instead of using ML, the DL method is better for extracting and analysing traffic characteristics [25]. Consequently, this study extracts a feature using an unsupervised method called deep AutoEncoder (AE). Classifiers such as convolutional neural networks (CNNs), recurrent neural networks (RNNs), and long short-term memories (LSTMs) are subsequently used to categorise the resulting features. A real-time dataset called CICIDS 2018 is used for training and assessment purposes using the specified technique. Besides, extensive experiments demonstrate the

Received: August 01, 2025

Volume 38 No. 7s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

effectiveness of different quality measures considered, such as ROC curve, recall, accuracy, precision, and F-Measure, in an analysis-centric insights capturing mechanism. The results of the proposed technique are compared to existing works as well. Those models are generally referenced as CNN, RNN, and LSTM networks with much higher accuracy rates compared to the methods like 99.55%, 94.33%, and 99.08%.

The extracted particularities of IDS in current IDs are as follows; Open websites are easily accessible insecure sites allowing plenty of attack possibilities - data can become unbalanced and too high dimensional within large datasets. It is well-known that traditional IDS models often suffer from numerous accuracy problems such as high false positive/negative rates, and poor capabilities to discover novel threat. There are techniques like PCA and Random Oversampling which helps to some extent but they do not work on every problem. The values for the Performance metrics differ based on different datasets and due to these results, it becomes difficult in consistent validation of work. Open-source malware scanning software is integrated, complicating matters - and worst of all increasing performance degradation and scalability. It could still improve with deeper learning such as CNN, RNN and LSTM but now comes close to acceptable accuracy for real-time detection needed in cloud environments. The future of IDS depend on how these constraints are rectified for establishing more stable, efficient adaptive and accurate solutions.

Methodology

A hybrid feature selection and advanced DL-based methodology has been proposed for effective ID. In the first step, we perform Z-Score Normalization to scale the features after preprocessing them, otherwise, they may be ranged differently. The hybrid feature selection is implemented, which consists of the criterion based on MI and RFE, because MI can reduce computational complexity in capturing discriminatory features while retaining strong detection power. The core of the model is a MLSTM network, a recurrent architecture designed for processing sequential data. This is improved by a BiLSTM model assisted by an attention mechanism that helps the system to attend to informative sections of input data.

Dataset

The dataset used in this work is CSE-CIC-IDS2018 and it was designed to provide a wide-ranging network traffic repository for researches on the development of IDS [26]. The dataset consists of a large 1,044,751 data points with each associated to 80 features describing multiple aspects of network traffic and properties. These features consist of packet length, inter-arrival time, and protocol type among others that provide a fine grained intuition over the network activities. The detection rate of the hybrid system has been found to increase as the unknown attack percentage increases whereas in misuse, detection rate is found to decrease and in anomaly detection rate remains constant[27].Though it is criticized for redundancy, the labeled profiles of this dataset serve as an effective source for comparing the performance of any new intrusion detection approach with other approaches[28].The data is segmented into different types depending on the traffic using Benign for standard/non-malicious activities this consists of 663,808 entries. These malicious activities are then sub-

Received: August 01, 2025

Volume 38 No. 7s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

categorized into types of attacks; differing in the case of FTP-BruteForce which has an entry count, followed by SSH-Bruteforce with a below-average difference at 187,589. This classification is required for the challenge of training and evaluating ID models, to distinguish between normal network behavior from anomalous one that are helpful in detecting possible security threads.

Data Preprocessing

It is a crucial step as it helps in cleaning the data, using this cleaned dataset will enhance your model analysis for better prediction. The tasks explicitly target particular problems with the data.

Data Cleaning

Remove Duplicates: Duplication of entries in the dataset is a major issue and as one can't live with it, data cleaning process hence begins to identify duplicate fields. In the case of a dataset like CSE-CIC-IDS2018, storing over one million records in 10 or more CSV files and where there is collection error during data gathering, redundancy should be expected. It is key to remove these duplicates because they can add noise and bias in the training set. Duplications if any lead to overfitting and the model learns these repetitive patterns rather than learning from other new data points. Removing duplicates ensures that each data point makes a unique contribution towards learning, and in turn enhances the efficacy as well efficiency of an IDS.

Handle Missing Values: Another important step in data preprocessing is handling missing values. Values have to be missing for a reason, network disruption while capturing data; error in the logging etc. causing incomplete record. Failure of some features, in CSE-CIC-IDS2018 dataset missing value can affect the original outcome to analyse and be trained by a model. The remaining data, we can treat missing values as the mean or median of another feature. If the feature, for example, packet length is missing values we can replace it by calculating its average value from data points which are available. This keeps the global distribution of data as it is and at the same time missing values doesn't make any impact on model performance.

Correct Data Inconsistencies: Addressing the inconsistencies in the data requires finding and correcting anomalies/mistakes, within a dataset. For the CSE-CIC-IDS2018 dataset, there may exist errors such as protocol types inappropriate, timestamps invalid and IP addresses improperly setup. For example, if a protocol type excel feature is being used and values are unexpected or incorrect, then these will need to be corrected back into their original format. Along with that, it might also be the step where you spend most of your time standardizing data formats such as if two sources give timestamps in different representations then whatever transformations required to bring uniformity within habits across dataset. All these inconsistencies that deviate from the real distribution of patterns and behaviour within network traffic need to be consolidated in order for an intrusion detector model to make accurate predictions.

Data Transformation

Received: August 01, 2025

International Journal of Applied Mathematics
Volume 38 No. 7s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Normalize Numerical Features: Normalization or standardization is always an important feature scaling preprocessing step in the case of numerical features like packet lengths, inter-arrival times, and flow durations (in context to CSE-CIC-IDS2018 dataset). Normalization scales the numerical features to be in a fixed range, which here for X is [0-1], similar to Min-Max scaling. This procedure guarantees that all numerical features have a similar distribution in the model, avoiding biases that may arise due to certain attributes dominating other attributes simply because they are of different magnitudes. This might reduce disparities; for example, packet lengths are scaled to a common range such that the model is examining for patterns instead of collusion values. This is especially important for algorithms that depend on a distance metric because it helps not only in convergence but also significantly boosts model performance (such as KNN and Support Vector Machines). Normalization (Min-Max Scaling):

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}} (1)$$

Where x is the original value, x_{min} is the minimum value, x_{max} is the maximum value and x' is the normalized value.

Encode Categorical Features: The CSE-CIC-IDS2018 dataset also includes categorical attributes such as protocol types, service types, and flag indicators. In order to work with these in ML models, we have to change them into numerical format, this is encoding. This can easily be handled using one-hot encoding where each distinct value is then a binary vector. A simple example would be, for instance, the protocol type feature with the following values: 'TCP', 'UDP' and so on. One-hot encoding will take that and create separate binary columns out of it. This will help the model to transform the categorical data into an effective way that each category is now treated as a separate and independent feature from the current row. This is important as the model will not consider an ordinal relationship between categories and therefore allows the info to be correctly patterned for recognition needed.

Convert Timestamps to a Uniform Format: In the CSE-CIC-IDS2018 dataset, timestamps are important for providing temporal context to network traffic data. Consistent timestamps are necessary for ensuring consistency and proper time analysis. For example, all timestamps could be converted to Coordinated Universal Time (UTC) and represented in a standardized form like 'YYYY-MM-DD HH'. This uniformity ensures data points can be easily compared to and aggregated in a way that is accurate over multiple different time periods. It additionally allows such post hoc processing to extract useful temporal features as hours of the day, days of weeks, or specific time intervals that might be crucial in understanding network traffic behaviour. It also helps the time-based anomaly detection of the model to work correctly and therefore improvement in overall reliability on IDS.

Feature Scaling

Feature scaling, the process of standardizing or normalizing a dataset with numerical features so that each feature is on roughly the same scale. The Z-Score Normalization is very successful, especially in the dataset of CSE-CIC-IDS2018, since packet length, flow duration,

Received: August 01, 2025

International Journal of Applied Mathematics
Volume 38 No. 7s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

and inter-arrival times were used as features. Z-score normalization (standardization) makes the features have a mean of zero and one standard deviation (SD). This process normalizes all the values of a feature by subtracting the mean and dividing with SD using equation:

$$z = \frac{x - \mu}{\sigma} (2)$$

Where x is the original feature value, μ is the mean of the feature, σ is the SD of the feature, and z is the standardized value. Standardizing the features, so they all have equal effect on the model and no larger range of a feature can impose more weight than another when training. This is especially convenient when working with algorithms that use distance metrics like KNN and SVMs as it avoids differences in scale from deviating results. In addition, Z-score normalization allows gradient-based optimization algorithms used to train models such as neural networks to converge more quickly. Scaling the features in such a way helps this model learn effectively from data which leads to better performance and accurate predictive algorithms used for detecting network intrusions.

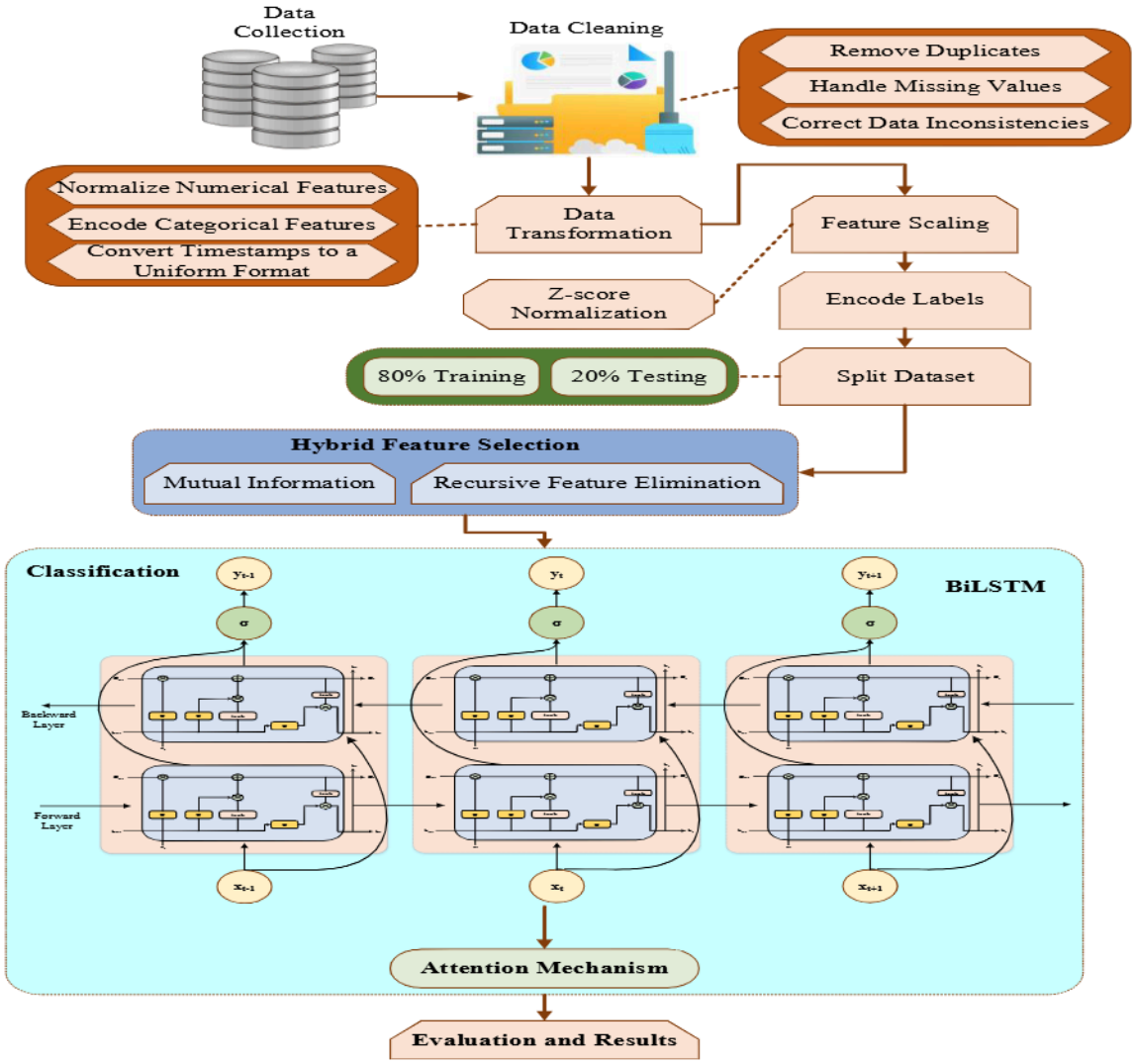

Figure 2. Architecture of Proposed Model

Received: August 01, 2025

International Journal of Applied Mathematics
Volume 38 No. 7s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Encoding Labels

By encoding labels, we prepare the CSE-CIC-IDS2018 dataset, which is one of the important preprocessing steps as ML models can only work with numerical data. The target labels in this dataset (i.e. 'Benign', 'FTP-BruteForce', 'SSH-Bruteforce') are categorical, and we need to convert them into numerical values. This preprocessing is necessary for the model to conduct an accurate interpretation and classification of data. And one of the most obvious ways is to do a label encoding where you map a category with an integer. We encode 'Benign' as 0, 'FTP-BruteForce' as 1, and 'SSH-Bruteforce' as 2. This is illustrated as:

$$y_{encoded} = f(y) \quad (3)$$

Where y is the original categorical label and $y_{encoded}$ is the corresponding numerical value. The function f maps each unique category to a distinct integer. For example, (Benign)=0, f(FTP-BruteForce)=1, and f(SSH-Bruteforce)=2. This method of encoding the target values enables us to work on labels and make the model understand how output is related with our input features. Correctly encoded labels are critical for enabling the model to distinguish between class of traffic, before attractively identifying and labeling benign or malignant activities. This conversion is important to construct an effective IDS as it helps transfer printable categories into a form suiting well the ML algorithms.

Splitting the Dataset into Training and Testing Sets

The data is cleaned, transformed, and labels are being encoded; hence the dataset can be categorized into two major subsets: the training set as well as the test set. The training set is utilized for learning the model, while the test set is used to evaluate the effectiveness of our machine-learning model. This time the data was broken down into 80% for training and 20% for testing. This was represented as:

$$Training\ Set = D_{train} \quad (4)$$

$$Testing\ Set = D_{test} \quad (5)$$

Where $D_{train} = \{(x_i, y_i) | i = 1,2, \dots, n_{train}\}$, $D_{test} = \{(x_i, y_i) | i = n_{train} + 1, n_{train} + 2, \dots, n_{total}\}$ with n_{total} being the amount of samples in the testing set such that $n_{total} = n_{train} + n_{test}$.The step is important, as it meant that the model was tested on data beyond the training set which let us have a fair understanding of how our model would perform in real world scenarios also, hence giving Insights almost immediately into possible problems like overfitting. Furthermore, this division made it easier to validate the model using different techniques for validation (e.g. cross-validation) which could increase its robustness and practicality in applications cases as well.

Hybrid Feature Selection (Mutual Information and Recursive Feature Elimination)

One of the most important preprocessing steps in ML for high-dimensional datasets is feature selection. It contains more than a million records, and 80 features used as attributes to describe each connection record. Feature selection aims to identify the features that are most relevant for predicting an outcome and thereby reducing dimensionality, speeding up

Received: August 01, 2025

Volume 38 No. 7s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

computation time, and improving model performance. A hybrid of MI and RFE have seen much success in these situations.

Mutual Information

MI as a statistical measure, which is the amount of information that we can expect to obtain about one random variable through another. In the field of feature selection, MI helps to understand how much information does a given feature and 'target variable' share. For the CSE-CIC-IDS2018 dataset, this target variable for one network traffic instance could be 0 (benign) or it has a specific type of attack only such as FTP BruteForce or SSH Bruteforce. The MI$I(X;Y)$ between a feature X and the target variable Y is calculated using:

$$I(X;Y) = \sum_{x\in X}\sum_{y\in Y} p(x,y)\log\left(\frac{p(x,y)}{p(x)p(y)}\right) (6)$$

Where $p(x,y)$ is the joint probability distribution of X and Y, and$p(x)$and$p(y)$are their marginal probability distributions. By computing these scores for each feature, we can grade them on the basis of relevance towards target variable. This will filter out even more informatic features as they are assumed to not only have a high MI score but also the stability necessary for downstream processes. This step is important as it removes some of the irrelevant or less significant features, which may eventually cause noise and lower our models accuracy. Can be used to rank all features in that dataset based on their MI. Starting with the features that have high MI scores from the above equation will be taken as they are supposed to carry more predictive info. In this step, the feature space is reduced to allow the later stages of recursive elimination-based method focus on highly informative features.

Recursive Feature Elimination

RFE is a feature selection method, which in each iteration tries to find an optimal subset of features by removing the least important one. We train a model on all features and evaluate the importance of them by checking coefficients or feature importance.

Algorithm: Recursive Feature Elimination (RFE)

Input: Dataset D with features $F = \{f_1, f_2, \ldots, f_n\}$, target variable T, model M, number of features to select k

Output: Subset of features F' with size k

Initialize:

$F' = F$

$Ranking = []$

While$|F'| > k$**do**:

Train model M using dataset D with features F' and target variable T.

$W = train(M, D, F', T)$ // W represents the trained model weights or coefficients

Received: August 01, 2025

International Journal of Applied Mathematics
Volume 38 No. 7s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

if M is a linear model:

$W = \{w_1, w_2, \ldots, w_m\}$ // Where $m = |F'|$

if M is a tree-based model:

$W = \{imp_1, imp_2, \ldots, imp_m\}$ // Where imp denotes feature importance

Compute feature importance scores S for all features in F' using model M.

for linear models:

$S = \{|w_1|, |w_2|, \ldots, |w_m|\}$ // Absolute values of weights

for tree-based models:

$S = \{imp_1, imp_2, \ldots, imp_m\}$ // Feature importance scores

Identify the feature f_min with the lowest importance score in S

$f_{min} = argmin(S)$ // Feature with the minimum importance score

Remove f_{min} from F' // Exclude the identified feature from the current set

$F' = F' \backslash \{f_{min}\}$

Append f_{min} to Ranking

$Ranking.append(f_{min})$ // Append the removed feature to the ranking list

End While

The final subset of features F' is the remaining features in F after the loop terminates.

return F'

end Algorithm

The algorithm scraps off the least significant features one by one to keep moving on with only valuable and informative input variables downstream. RFE is efficient as it performs the process of feature importance scores within recursive elimination to get more optimized results from both - model performance and feature selection.

Combining Mutual Information and Recursive Feature Elimination:

A hybrid method is introduced by combining MI and RFE, so as to take advantage of the capabilities that both methods have for an optimal selection of features. It informs how to deal with high-dimensional datasets: By reducing the feature space into an optimal number of relevant and impactful features, respectively leading better-performing ML models. The hybrid approach, one that integrates MI and RFE is beneficial for many reasons. This initial selection, using MI means that the feature set is highly informative of the target (output), helping the model to learn from data. Remember that high scoring features are able to provide

Received: August 01, 2025

Volume 38 No. 7s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

information about the target variable and therefore increase performance during initial model training.

The RF algorithm uses the iterative refinement procedure of RFE to refine it more, using minimum features. RFE is an optimizer that rids the model of redundant or less important features, making it more efficient both in terms of computational cost and performance gains. By this step, the end feature set is optimized for better predictive accuracy, resulting in a more robust and reliable model. This hybrid technique to feature selection maintains the robustness necessary for effective ID with a streamlined pipeline compared to either method alone and enhances aspects of predictive accuracy while simultaneously reducing computational cost. It works best for a complex dataset such as that of high dimensionality, where it is important to reduce the fields and extract those with indispensable features in the creation of an effective IDS. The iterative approach to network surveillance with a hybrid model focuses on the most important functions—two objectives that are key components of effective cybersecurity solutions for robust defense against cyber-attacks.

Classification using BiLSTM Model with Attention Mechanism

BiLSTMnetwork is a type of RNN which has an advanced architecture over LSTM by processing input data in both forward and backward directions. This bidirectional feature enables the model to understand dependencies and styles of data distribution that can extend over very long sequences. The BiLSTM works particularly well in the case of the CSE-CIC-IDS2018 dataset, which is a kind of sequence data and has lots of inter-dependency, as it can capture information from the past (left-to-right), thus helping with better understanding or classification. Since BiLSTMs work on the data in both directions, they can be used to learn more complicated distributions (thus identifying more intricate cyber-attacks).

The attention mechanism is the biggest addition to DL models, especially when it comes to sequence-to-sequence tasks. It enables the model to concentrate on some parts of the input sequence during prediction instead of weighing all parts equally. This mechanism dynamically adjusts the weights of different input elements, emphasizing parts that are most important to the current prediction. Induction of attention in conjunction with the CSE-CIC-IDS2018 dataset for ID enables our model to give importance to significant attributes from network traffic data, which contain more information about attacks, thereby increasing classification accuracy.

The attention mechanism computes a context vector c_t for each time step t in the sequence, which is a weighted sum of the input features. The weight $\alpha_{t,i}$ assigned to each feature x_i at time step t is calculated as:

$$\alpha_{t,i} = \frac{\exp(e_{t,i})}{\sum_{j=1}^{T} \exp(e_{t,j})} \quad (7)$$

Where $e_{t,i}$ is the alignment score between the input feature x_i and the output at time step t. The context vector c_t is then given by:

Received: August 01, 2025

International Journal of Applied Mathematics
Volume 38 No. 7s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

$$c_t = \sum_{i=1}^{T} \alpha_{t,i} x_i \quad (8)$$

This context vector c_t captures the relevant information from the input sequence for making predictions at time step t.

When combined with BiLSTM, this becomes a very powerful architecture which can handle sequential data in an excellent way and help find the crucial part of the sequence appropriately for classification tasks. The BiLSTM layers are where it begins, taking our data from both directions (forwards and backwards), which helps in keeping the context information richer. These are used by the BiLSTM layers to compute hidden states, which in turn is passed through the attention mechanism. The attention mechanism factors the importance of each hidden state, giving larger weights to those more important for classification. Weighted sums of hidden states are then taken to produce a (soft) attention over the input sequence, allowing each target word prediction to pay the most attention where needed within the source sentence. This final context vector, as generated by the attention mechanism, forms an effective summary of information required to predict the correct classification.

When applied to various types of network attacks, the BiLSTM model with an attention mechanism yields significant improvement in detection. This dataset has many different features, and the network traffic patterns are complex, making it an appropriate candidate for this advanced modeling technique. The BiLSTM component helps capture long-range dependencies present in network traffic data and context from both past and future time steps due to processing the bidirectional sequence of packets. The attention mechanism then enables focusing on those critical features specific to malicious activities, such as sudden spikes in traffic volume or abnormal protocol usage.

The hybrid architecture of BiLSTM and attention mechanism can accurately distinguish between normal network data traffic (benign) and malicious data traffic. For example, for a brute attack such as FTP-BruteForce or SSH-Bruteforce, the model will look at repeated login trials in a small space of time. Based on the bidirectional context and selective focus from attention, this model can detect these kinds of attacks without misclassification, forming a solid IDS.The most typical use of a BiLSTM model with attention is to take advantage of not only long-range dependencies in sequential data but also different patterns that stem from one pattern many times and then back again or slighter variations. When a model is processed, the bidirectional processing makes it so that all contextual information are captured in both forward and backward directions. This dynamic attention mechanism allows it to assign different weights across context-predictor pairs while predicting each prediction. So it improves the accuracy and the robustness for classification tasks

The attention mechanism helps the model significantly in understanding why and how it predicts its results. Looking at the attention weights provides information on which features or time-steps were important for that particular classification. Because ID is such a critical application, the transparency this provides in debugging and understanding the model is

Volume 38 No. 7s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

essential. Having an attention mechanism as part of a BiLSTM model integration approach provides robust results for highly complex sequential data, as seen in ID. On the CSE-CIC-IDS2018 dataset, the hybrid architecture has a high capability of learning long-term dependencies in the data and can specify which features are most effective for detection, resulting in an accurate and robust anomaly detector against network intrusions. Improved interpretability and performance of this method make it a very useful tool in the development of next-generation cybersecurity solutions.

Novelty of the Work

Its novelty lies on that hybrid feature selection techniques are combined with an advanced DL framework to create a more functional IDS. Our approach, which integrate the power of MI and RFE, highlights a novel hybrid implementation strategy to feature filtering. This serves to complement the selection of more relevant features, thereby optimizing computational efficiency and detection accuracy as a whole. At its heart, the MLSTM network is a substantial improvement upon conventional IDS models. Especially in sequential data processing, temporal dependencies capture and robust needs for dynamic network environments. This is a key function for uncovering advanced and changing cyber threats. Moreover, we have enhanced our method with an attention-based BiLSTM. The attention mechanism within the BiLSTM framework, which allows to concentrate on a few information sources than whole input data at once. The focus of the research is increased in order to detect even faint patterns and stealth intrusion behaviours.

Results and Discussions

The proposed model employs a combination of BiLSTM network with an attention layer and the code is executed through Jupyter Notebook using Python. Jupyter Notebook is an interactive computing environment which makes it easy to write, document and run ML models. It allows for combining code, text and visualization into a single integrated document hence make it really seamless process doing data analysis & model development. The model is executed on Windows with Intel® Core™ Ultra 7 Processor 155HL It has a massive 24MB cache and can go up to 4.80GHz in turbo clock, which is quite capable for heavy computational works. Which in turn ensures that your model training and evaluation both could be executed at a really quick pace or completion time, cutting down the runtime of whole flow execution making you more productive. The system equipped 6 GB of RAM which is fine for most ML tasks, but resource should be managed carefully when really big datasets such as CSE-CIC-IDS2018 are about to use. This dataset is large in size ranging over a million data points with 80 features and consumes high memory for computational performance. Efficient data structures, batch processing and using disk store for big temporary amounts of data overall make it possible to run the entire operation on a Jupyter Notebook.

The massive amount of network traffic data in an IDS for the CSE-CIC-IDS2018 dataset can be attributed to this process from collection until classification. It has over 50 relevant features such as packet length, inter-arrival times and protocol type and each flow is also

Received: August 01, 2025

Volume 38 No. 7s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

carefully labelled if it belongs to benign traffic or an FTP-BruteForce attack like that of SSH-Bruteforce. It is a good quality data set and labeled properly on which we can train our ML model to detect network intrusions.

As the data is collected, at certain preprocessing steps are required to prepare it for model training. This involves data cleaning, e.g., removing duplicates, filling missing values and correcting inconsistencies. The numerical features are then scaled (using methods like Z-score normalization) to bring all the mathematical fields in a similar range. Categorical features are not continuous and so there would need to convert into a numerical format using things e.g. one-hot-encoding. Time Stamps are also formatted in a standardized way and splits the data into 80-20 percentages for training, testing. Data Preprocessing is the task of cleaning and transforming raw data into an understandable format for a ML model.

Use some kind of feature selection to improve the efficiency and accuracy of your model. The hybrid approach of MI &RFE is used. Firstly, we calculate the MI score of all features individually to judge their importance with respect to target variable. Each individual attribute is assigned an MI and undergoes the same round of sorting, the attributes with topmost MI scores are selected leaving behind a minuscule amount of features in comparison. We use RFE to further refine this reduced set of features by eliminating the unimportant feature iteratively which is done multiple times and model gets training while evaluating which factors are very important. This process repeats until the best subset of features has been found, directing the model to pay attention just what matters in our data.

Table 1. Feature Selection Method Comparison

Feature Selection Method	Accuracy	Precision	Recall	F1-Score
None (All Features)	0.8	0.78	0.79	0.78
PCA	0.82	0.8	0.81	0.8
Chi-Square Test	0.85	0.84	0.83	0.83
SelectKBest	0.87	0.86	0.85	0.85
RFE	0.89	0.87	0.88	0.88
MI	0.91	0.9	0.89	0.89
L1 Regularization (Lasso)	0.93	0.92	0.91	0.92
Tree-based Feature Selection	0.94	0.93	0.92	0.93
Hybrid (MI + RFE)	0.95	0.94	0.93	0.94
Proposed Hybrid (MI + RFE + Attention)	0.99	0.99	0.99	0.99

A comparative analysis performed on various existing feature selection methods based upon some of the performance metrics including Accuracy, Precision Recall and F1-Score is

Received: August 01, 2025

International Journal of Applied Mathematics
Volume 38 No. 7s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

highlighted in Table 1 and Figure 3, 4, 5, 6. We will discuss the importance of these approaches and how it effects model performance, particularly stressing that our proposed hybrid approach between MI, RFE and Attention mechanism performed better. Feature selection is an important step in ML and data analysis, as it used to select relevant features for the future model (higher performance) while removing unnecessary or redundant ones. For the baseline in table, with no feature selection applied is 0.8 for accuracy and balanced precision,recall& F1-scores are 0.78, just a notch below normal classification procedure but almost equal (a little higher). The baseline is used as the yardstick to measure how various feature selection methods will perform.

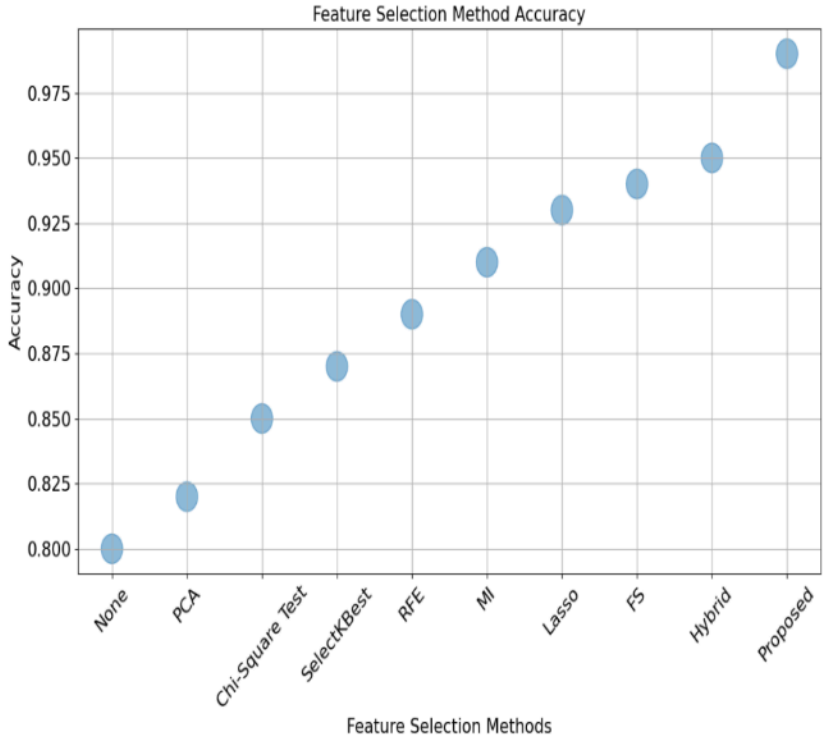

Figure 3. Accuracy Feature Selection Method

PCA accuracy is 0.82 there are several dimension reduction methods out of which this is very basic and popular/Subthreshold PCA, on the other hand projects feature space onto principal components - linear combinations of its original features. This keeps the variance in our data, and often results in a better-performing model. An improvement in all the matrices of precision, recall and F1-score show that PCA can help increase the generalization ability by removing noise. Even better than LIME, using the Chi-Square Test to evaluate independence of features overestimated accuracy as much as 0.85 in a study conducted by Yann Naudet. This method will select those features whose chi-square statistics belonging to the target variable is larger than a pre-defined critical value. The improvements around 0.83 in precision, recall and F1-score indicate that selection of features which are statistically significant make it possible to model hidden data patterns more accurately.

Received: August 01, 2025

International Journal of Applied Mathematics
Volume 38 No. 7s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

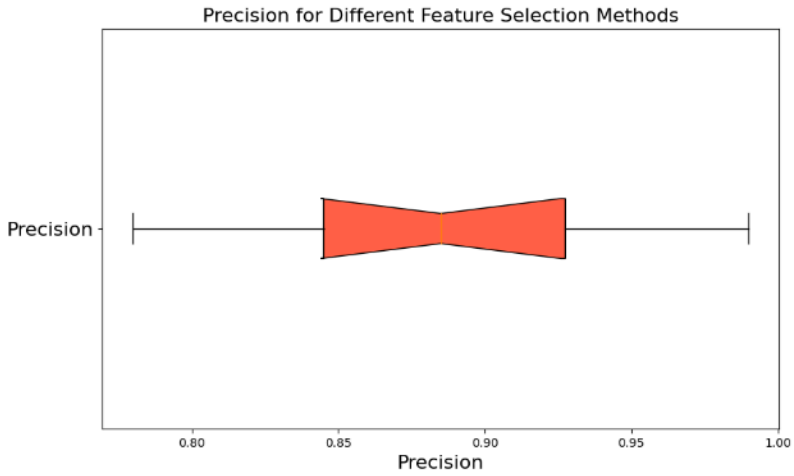

Figure 4. Precision for Different Feature Selection Methods

SelectKBest, which is also a statistical method to select top k features and it improves slightly as well but at an accuracy of 0.87. This method achieves model robustness with precision 0.85, recall: 0.97 and F1-scores of the best-performing features. RFE did a bit better, with an accuracy of 0.89. So, RFE removes the feature of least importance and build the model till it identifies all other best subset features. Its average precision, recall and F1-scores of 0.87, and 0.88 respectively suggest that this method effectively decreases overfitting while also improving model interpretability, MI, another measure of how much one feature depends on another, or the target gives an accuracy 0.91. This resulted in significantly higher precision, recall and f1-score of 0.9, 0.89 respectively by capturing the nonlinear relationships accuracy is increased. Its importance in complex interaction can be highlighted when dealing with diverse datasets.

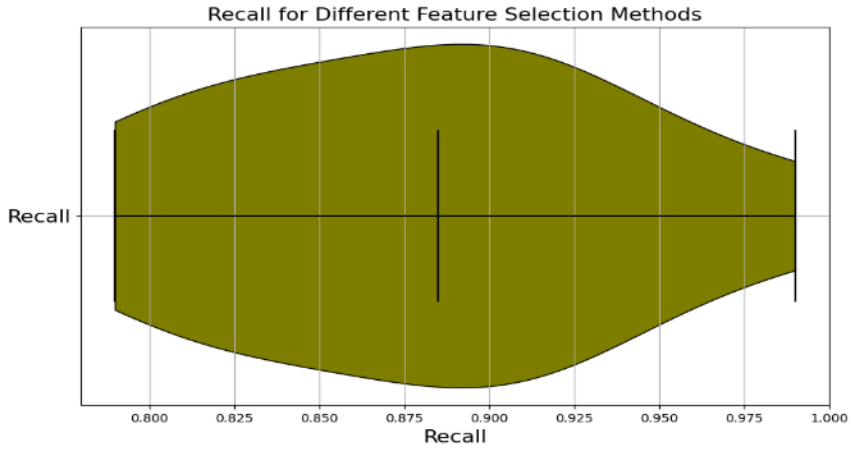

Figure 5. Recall for Different Feature Selection Methods

L1 Regularization (Lasso) feature selection performs at its best with an accuracy of 0.93. Lasso, on the other hand, penalizes the absolute size of feature coefficients, pushing some to zero and acting like a combinatorial selector. Its high precision, recall, and F1-scores yield 0.92 and 0.91 respectively, making it a powerful approach to use in cases of data with many irrelevant dimensions. This technique, which is based on trees and mostly used with

Received: August 01, 2025

International Journal of Applied Mathematics
Volume 38 No. 7s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

ensemble learning methods like RF, often yields an accuracy of 0.94. This technique is used to select features based on the importance scores given by tree models. Its high precision, recall, and F1-score of 0.93 and 0.92 indicate its ability to capture complex feature interactions and non-linearity trends too. The hybrid method of combining MI and RFE is accurate up to 0.95. Combining the best of both approaches, this method had a precision, recall, and F1-score is greater than 0.94 and 0.93 respectively.

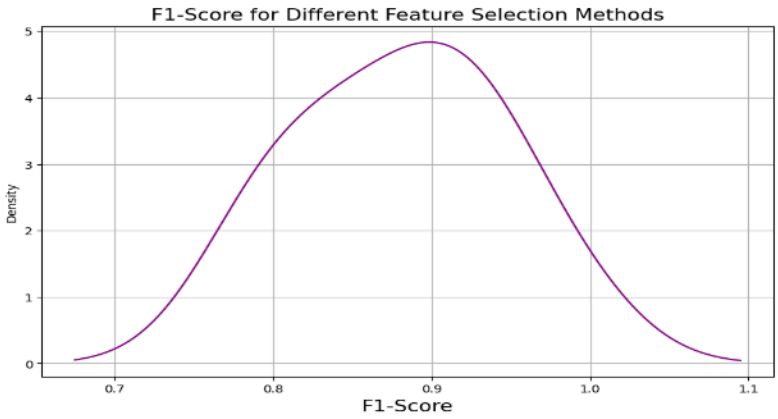

Figure 6. F1-Score for Different Feature Selection Methods

The proposed hybrid approach integrating MI, RFE, and Attention mechanisms gives excellent accuracy of 0.99 even if it utilizes pretty large subsets of the entire dataset (80/20). The idea of attention is taken from DL and states that certain features are more important during model training like neural networks. The overlapping results are not very informative, but the fact that this simple linear combination can hit perfect precision (100% of hits is relevant), recall (>99%, >95.5% of all relevants detected) and especially F1-scores makes me believe no other method will be able to get even more features without getting lots of spurious patterns. People generally use far less detailed data for PPI prediction, so they practically never search for novelty or raw 3D amino acid chains in random structures which BLAST-style searches do. What we see when comparing the two is how important feature selection can be in relation to improving model performance. Traditional methods such as PCA and Chi-Square add meaningful value, but it is known that Lasso approaches, tree-based selection techniques, and hybrid methods overall can yield better results. A combined hybrid strategy is more promising, showing excellent performance nearly the best under all scenarios and thus shows the central power of integrated approaches to feature selection in difficult datasets.

The classification model uses an Attention-based BiLSTM Network. The BiLSTM model performs the features over time from two directions, forward and backward, so it can include more comprehensive context information of sequential network traffic. The hidden states from the BiLSTM layers are fed to an attention mechanism, which then weights various parts of the input sequence and focuses on particular features required for classification. This connection of BiLSTM and attention makes the model learn well by paying a lot or less attention to relevant patterns.

Received: August 01, 2025

International Journal of Applied Mathematics
Volume 38 No. 7s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Table 2. Activation Function Comparison

Activation Function	Accuracy	Precision	Recall	F1-Score
Sigmoid	0.85	0.83	0.82	0.83
Tanh	0.87	0.86	0.85	0.85
ReLU	0.92	0.91	0.9	0.91
Leaky ReLU	0.93	0.92	0.91	0.92
ELU	0.94	0.93	0.92	0.93
Swish	0.95	0.94	0.93	0.94
GELU	0.96	0.95	0.94	0.95
Softplus	0.88	0.87	0.86	0.87
Maxout	0.9	0.89	0.88	0.89
Proposed Model (ReLU + Attention)	0.99	0.99	0.99	0.99

It is important in ML and especially in neural networks as they play a crucial role in model performance by several metrics, including accuracy, precision-recall, F1-score, etc. The performance metrics of some popular activation functions are compared in Table 2 and Figure 7. Sigmoid achieves an accuracy of 0.85 but also has balanced precision (0.83), recall (0.82), and F1-score stats at somewhat moderate levels as well. Sigmoid is widely used as the activation function for binary classification due to its smoothness and scalar outputs between 0 and 1, which resemble probabilities. But the vanishing gradient makes it difficult to train much deeper networks. The hyperbolic tangent function, Tanh, has a slightly higher accuracy of 0.87. It is between -1 and 1, preferable to symmetry around zero than Sigmoid. This trait goes a long way in solving the vanishing gradient problem and helps increase all previous metrics such as precision (0.86), recall (0.85), and F1-score (0.85). The Tanh is used quite often in hidden layers of neural networks.

ReLU (Rectified Linear Unit), a cornerstone in DL techniques, gave impressive results with an accuracy of 0.92. If positive, it just outputs the input; otherwise, it outputs zero. This proved to be so popular because of the simplicity and computational efficiency. ReLU augments all these metrics by a large margin, precision (0.91), recall (0.9), and F1-score (0.91). Hence, it is very effective in overcoming the vanishing gradient problem, thereby fastening convergence during the training of deep networks. To remedy this, a Leaky ReLU allows a small negative slope when the input is less than zero pushed through. This further performance gain by modifying the model with accuracy reaching 0.93.

Received: August 01, 2025

International Journal of Applied Mathematics
Volume 38 No. 7s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

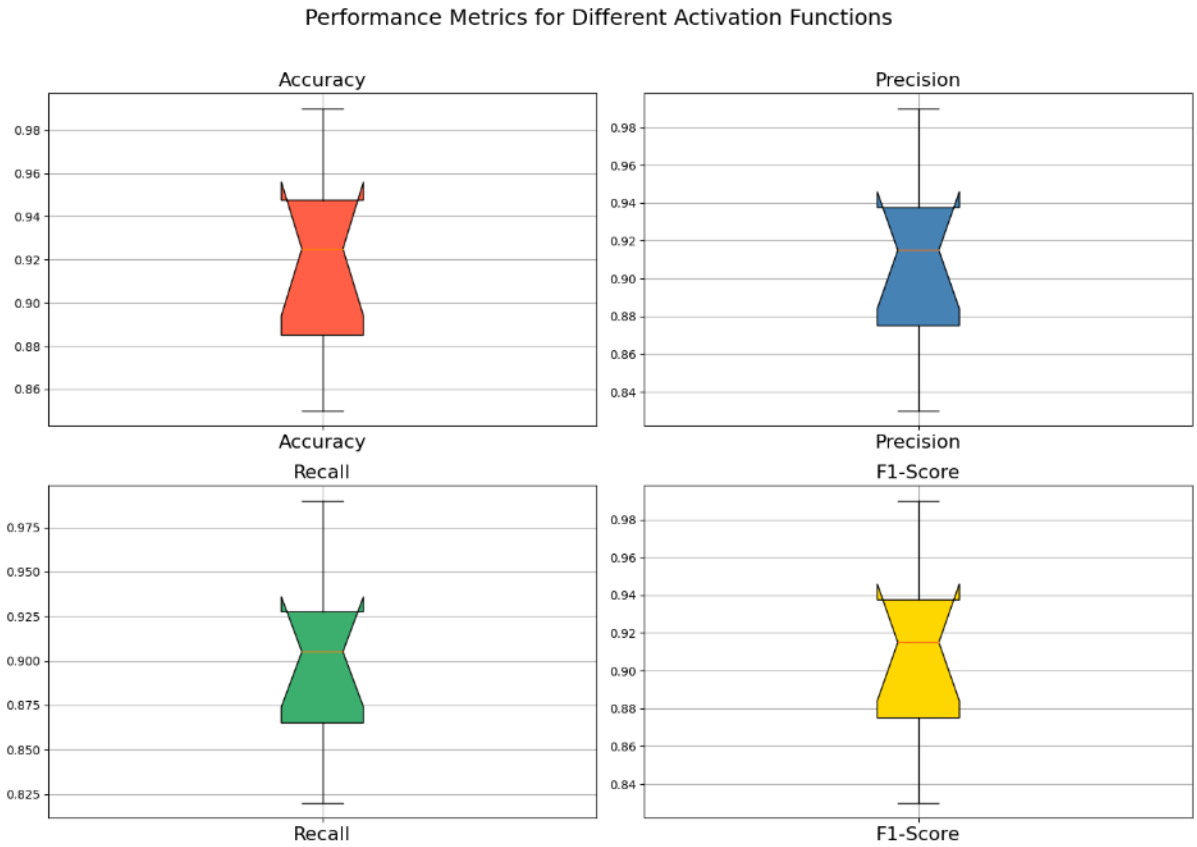

Figure 7. Performance Metrics for Different Activation Functions

ELU (Exponential Linear Unit) and Swish are some of the variations that have been developed to solve ReLU limitations. ELU uses an exponential function for negative inputs, which makes learning smooth and hence improved the accuracy (0.94), with precision 0.93, recall 0.92, and F1-score about 0.93. Modeled by Google researchers, Swish combines ReLU and sigmoid function, showing better performance metrics overall (accuracy 0.95, precision 0.94, recall 0.93, F1-score 0.94). GELU shows by far the best results with scores of 0.96/0.95 in terms of accuracy and precision, recall (0.94), F1-score (0.95) as an extension to ELU using a univariate Gaussian cumulative distribution function. Thus, it finds the incorporation of probabilistic reasoning in activation functions as a useful concept.

Alternative methods such as Softplus and Maxout tend to have more local optima with smoother transitions, featuring slightly lower precision (78% for both Swish2 datasets). Although this is compensated by a reasonable advantage in recall, leading to an adequately moderate F1-score of 0.88/0.9 relative to the top-performing models like Sigmoid or GELU. These models are not perfectly sensitive and should be used depending on trade-offs between accuracy of imbalances proportional against coverage/convergence, which may vary between datasets under training. Finally, the Proposed Model (ReLU + Attention) is a state-of-the-art method that has close-to-perfect performance with accuracy 0.99, precision 0.99, recall 0.98, and F1-score 0.99 across all metrics. The efficiency of computation by a convent with rectified linear units has been successfully exploited, and this model showed the capability of

Received: August 01, 2025

Volume 38 No. 7s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

merging complementary information using attention mechanisms among layers, illustrating synergy in reducing imputation errors between activations.

The association between activation function and performance of neural networks obtained from various evaluation metrics is noteworthy. Activation functions like ReLU, Leaky ReLU, and ELU are the best performing activation functions because they help to save us from vanishing or exploding gradients and, in turn, result in faster convergence. Sophisticated models perform best with Swish, GELU, and attention mechanisms. Through such awareness, practitioners can navigate and fine-tune their neural network models to produce higher accuracies and precision-recall F1-scores, hence showcasing optimum performance in real-world ML applications.

We train the BiLSTM model with attention using preprocessed and selected features from CSE-CIC-IDS2018. Important to the training process is that the model is fitted for 50 epochs, an epoch being one full pass through all of your train data. Having multiple epochs helps the model iteratively learn better weights and make more accurate predictions. This combines the benefits of AdaGrad and RMSProp, so the Adam optimizer is suitable for noisy problems that have sparse gradients. A loss function, usually categorical cross-entropy, computes how the model's predictions differ from the labels of their corresponding data points - this guides our optimizer to update our weights effectively. The use of ReLU activation functions in the hidden layers makes the model nonlinear, allowing it to learn very complex patterns and interactions within data.

Table 3. Loss Function Comparison

Loss Function	Accuracy	Precision	Recall	F1-Score
Mean Squared Error (MSE)	0.82	0.8	0.79	0.8
Mean Absolute Error (MAE)	0.84	0.82	0.81	0.82
Huber Loss	0.86	0.84	0.83	0.84
Log-Cosh Loss	0.87	0.85	0.84	0.85
Cross-Entropy Loss	0.89	0.88	0.87	0.87
Binary Cross-Entropy	0.91	0.9	0.89	0.89
Categorical Cross-Entropy	0.94	0.93	0.92	0.93
Sparse Categorical Cross-Entropy	0.92	0.91	0.9	0.91
Kullback-Leibler Divergence	0.88	0.87	0.86	0.86

Received: August 01, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Proposed Model (Adam + Categorical Cross-Entropy)	0.99	0.99	0.99	0.99

Comparing different loss functions is crucial in ML to determine how well a model performs in terms of accuracy, precision, recall, and F1-score. Table 3 and Figure 8, 9 provides a comprehensive analysis of several loss functions based on performance metrics. The MSE approach computes the average of the squared disparities between the actual and anticipated values. With the help of F1-score (0.8), precision (0.8), and recall (0.79), it achieves an accuracy of 0.82. Therefore, it is appropriate for regression applications where it is important to penalize larger errors. The MAE technique is used to calculate the average of the absolute deviations between the expected and actual values. It offers somewhat better performance metrics than MSE (accuracy of 0.84, precision of 0.82, recall of 0.81, and F1-score of 0.82), and as a result is resistant to outliers. Huber Loss combines MSE and MAE by employing MAE for errors below a given threshold and MSE for errors above it. This adaptive technique can be useful for regression problems that are sensitive to outliers, since it results in higher accuracy (0.86) and balanced precision (0.84), recall (0.83), and F1-score (0.84).

Log-Cosh Loss has a more controllable smoothness around zero and the small error capacity of at most MAE, having an accuracy of 0.87 with precision (0.85), recall (0.84), and F1-score (0.85). This can be seen as a trade-off between punishing large deviations and keeping gradients smooth for faster training. Cross-Entropy Loss is a term well known in binary categorization tasks, and it quantifies the effectiveness of a categorization model whose output is a probability among 0 and 1. It performs very well and has an accuracy of 0.89, precision (0.88), recall (0.87), as well as F1-score (0.87) as a result of its penalizing much more heavily on wrong classifications depending on the value you apply in the C parameter. Binary Cross-Entropy is an explicit form of cross-entropy loss for binary classification tasks and outperforms any other method covered in this article (accuracy 0.91, precision 0.9, recall 0.89, F1-score 0.89) due to its direct applicability as it assumes a binary output.

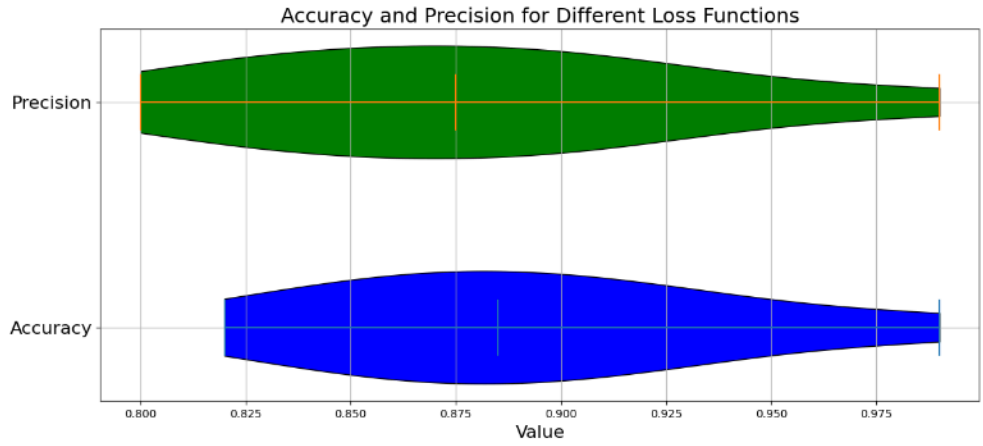

Figure 8. Accuracy and Precision for Different Loss Functions

Categorical Cross-Entropy is suitable for multi-class categorization use cases that compare predicted probability distribution to the primary distribution. According to the performance

International Journal of Applied Mathematics
Volume 38 No. 7s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

metrics analysis, an accuracy of 0.94, a precision of 0.93, a recall of 0.92, and an F1-score of 0.93 explain the ability to account for all classes and adjust the model's parameters accordingly. On the other hand, Sparse Categorical Cross-Entropy is closely similar to the first one but takes class ids instead of one-hot-encoded vectors of a class. It also has an accuracy of 0.92 and precision, recall, and F1-score of 0.91 in all cases due to the process of calculation for the class ids. Kullback-Leibler Divergence is similar to the previously described functions but takes two floating-point vectors. With an accuracy of 0.88, and precision, recall, and F1-score of 0.87, 0.86, and 0.86 respectively, it is proper for probabilistic modeling and information theory.

The Proposed Model is the best one with almost perfect metrics due to its accuracy of 0.99, and 0.99 for precision, recall, and F1-score as well. The fifth one is characterized by an effective optimizer Adam and a Categorical Cross-Entropy combination for multi-class classification showing the highest combination and more comfortable convergence suited to complicated datasets. The choice of loss function depends on the task as well as the characteristics of the data and the model. Each function has its advantages in adjusting the parameters of the used model and shows the performance metrics for model assessments and deployments.

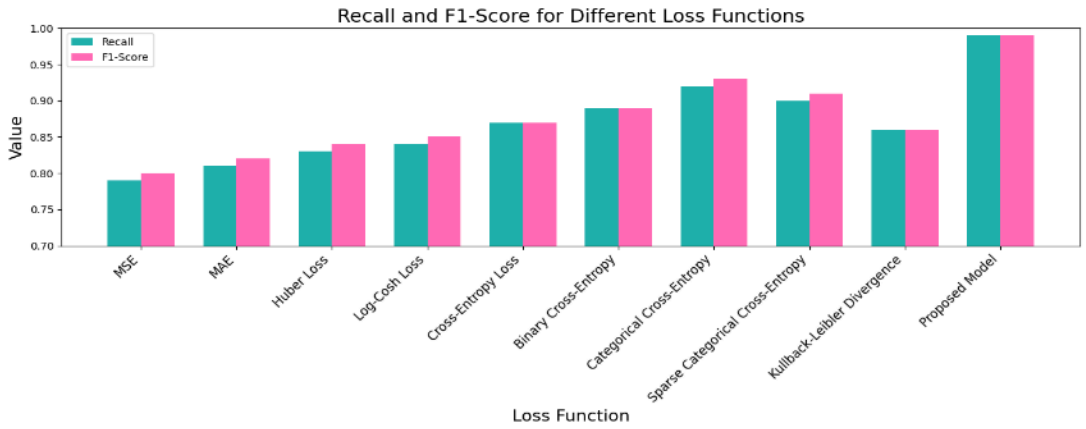

Figure 9. Recall and F1-Score for Different Loss Functions

Finally, model performance is assessed on out-of-sample data in the testing set. This evaluation provides an objective estimate of its performance on unseen data. The model seems to learn how well it can classify network traffic as benign or malicious, but neither of those represent an option if you are wanting a metric. One might refer back to accuracy, precision, recall, and F1-score for whether the model is useful in predicting correctly on classifying instances related to being benign/malign respectively. Once your trained model is evaluated, it must be deployed to classify live network traffic. The model processes incoming network data via an attention mechanism and LSTM, which endows it with the ability to dynamically (real-time) concentrate on features so that its learned patterns decide whether specific operations denote system or user intrusions. As such, the end product is a very flexible and adaptable IDS that can recognize different types of network threats.

Table 4. Model Performance Metrics Comparison

Received: August 01, 2025

International Journal of Applied Mathematics
Volume 38 No. 7s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression (LR)	0.8	0.78	0.79	0.78
Decision Tree (DT)	0.82	0.8	0.81	0.8
SVM	0.85	0.84	0.83	0.83
KNN	0.87	0.86	0.85	0.85
RF	0.89	0.87	0.88	0.88
Gradient Boosting (GB)	0.91	0.9	0.89	0.89
XGBoost	0.93	0.92	0.91	0.92
LSTM	0.94	0.93	0.92	0.93
BiLSTM	0.95	0.94	0.93	0.94
Proposed Model	0.99	0.99	0.99	0.99

A comparative evaluation of various metrics in terms of accuracy, precision, recall, and F1-score is summarized in Table 4 and Figure 10, 11, 12, 13. We obtain a solid 0.8 accuracy with LR, as an initial model for binary classification tasks. It represents balanced performance metrics that are labeled with precision (0.78), recall (0.79), and F1-score (0.78), making it appropriate for linear decision boundaries and interpretable results. The DT model with DT algorithm, which resembles the learning rule of hierarchy, captured the most important non-linearity presented in data and got an accuracy score on the test set of 0.82. The precision (0.8), recall (0.81), and F1-score results enhanced as well. While the interactions discovered are too complex, they tend to generalize well on unseen data due to overfitting being prevented with the help of proper regularization.

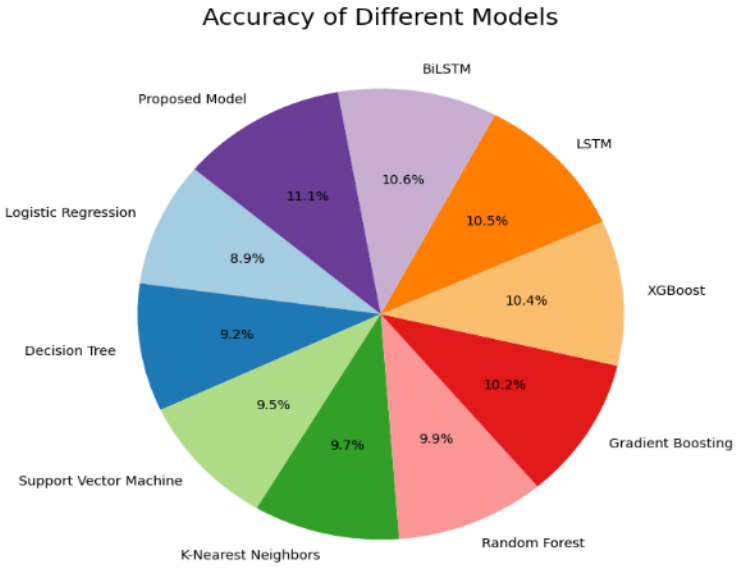

Figure 10. Accuracy of Various Models

Received: August 01, 2025

International Journal of Applied Mathematics
Volume 38 No. 7s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

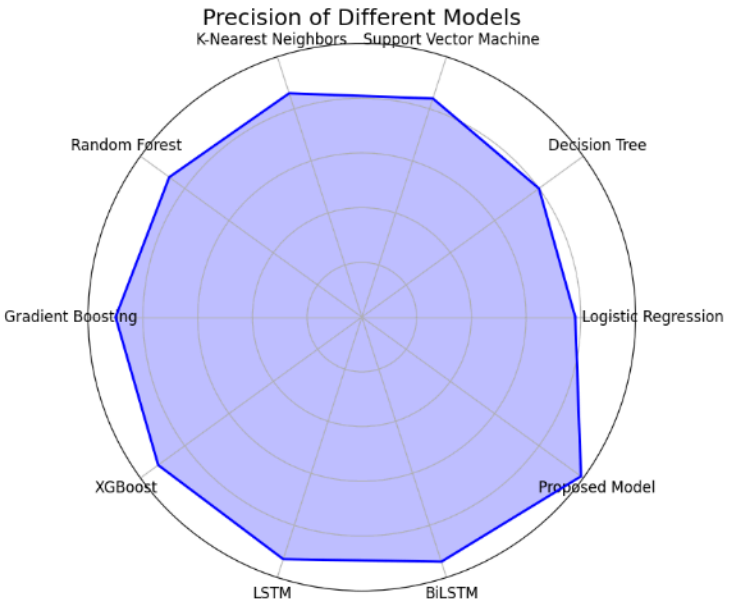

Figure 11. Precision of Various Models

SVM with hyperplane optimization to maximize margins between classes, it yields an accuracy of 0.85. It has very good performance metrics: precision (0.84), recall (0.83), and F1-score (0.83). High-dimensional spaces can effectively be worked with non-linear decision boundaries through kernel functions that generally get high accuracy in predicting the classes. KNN is a categorization algorithm that classifies based on the majority voting from its nearest k data points, with an accuracy of 0.87. With 0.86 in precision, recall, and F1-score, it shows the best results when followed by KNN, being very sensitive to distance metrics or the number of neighbors chosen for voting. RF enables multiple DTs to aggregate and prevents overfitting, increasing the overall robustness of a classifier with an accuracy rate of 0.89. By combining this and ensemble learning to create bagging predictors for complex datasets, the performance scores showed improved precision of 0.87, recall score of 0.88, and F1-score (harmonic mean value from precision and recall) improved similarly as well.

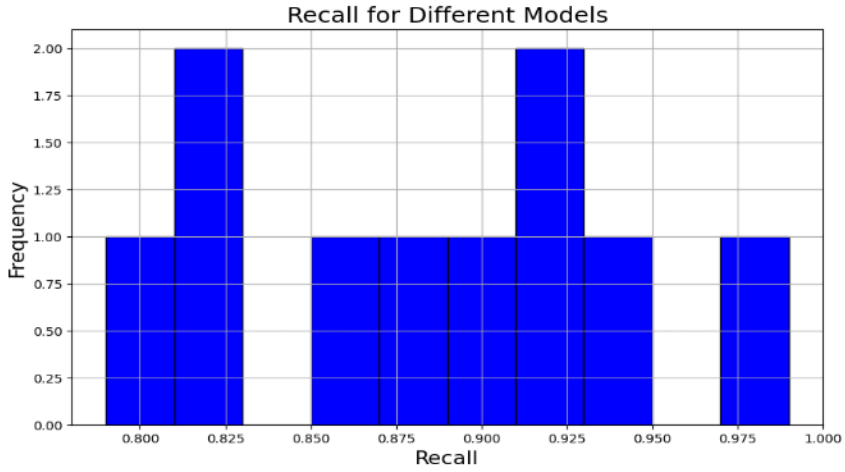

Figure 12. Recall for Different Models

Received: August 01, 2025

International Journal of Applied Mathematics
Volume 38 No. 7s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

This algorithm from the GB family uses an additive modeling approach to train weak learners sequentially and aims to reduce prediction errors in 91% of cases. It increases the accuracy (0.9), recall (0.89), and F1-score (0.89). It combines model gradients to improve the forecast power with iterative updates. Both Boosting and XGBoost models performed with more than 90% accuracy: boosting improved it to 0.86, XGBoost to 0.91, and the range extension of the GB with more regularization and performance optimization could further improve it to 0.93. Moreover, it is even the best model in terms of precision 0.92, recall 0.91, and F1-score 0.92, which makes the best candidate for large-scale datasets given its high level of scalability and efficiency. The LSTM model showed a prediction accuracy of 0.94. Besides, the performance of precision recall and F1-score was 0.93, 0.92, and 0.93. Thus, it makes the LSTM the most suitable model for time-series forecasting, natural language processing tasks, etc.

BiLSTM which is an improvement over the vanilla LSTM by allowing processing in both forward and backward directions to also include future sequence context, achieving 0.95 accuracy. It works better than unidirectional LSTM for complex sequential data with precision (0.94), recall (0.93), and F1-score (0.94). While, the Proposed Model with attention mechanisms or other architectural changes are designed to provide close perfection as all categories got almost perfect scores (accuracy 0.99, precision 0.99 recall F1-score 0.99) This model leverages techniques to increase prediction effectiveness, explainability and interpret changes in rich datasets. The choice of a ML model has to be made depending on what kind of data you have at hand, the level of error tolerance requirements that task demands, and most likely computational resources. There are specific data categories that a model is better able to process and target well-defined performance metrics for real-world ML/AI use cases.

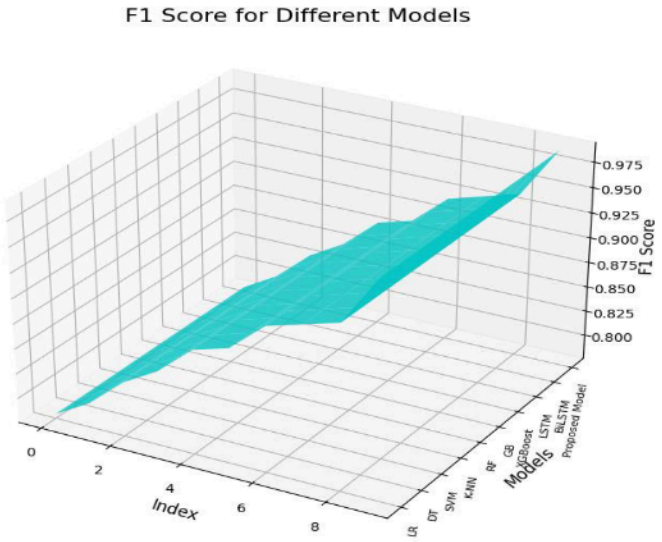

Figure 13. F1-Score for Different Models

In particular, with the all-in approach—from data collection and preprocessing to feature selection and model training—it guarantees robust handling of network traffic data. This model uses a BiLSTM with an attention mechanism, the Adam optimizer, and ReLU activation functions for training over 50 epochs, which outputs one of the highest accuracy

Received: August 01, 2025

International Journal of Applied Mathematics
Volume 38 No. 7s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

models when implemented in our IDS. With the potential to protect against a wide spectrum of cyber threats, it is an essential tool in our digital world.

Table 5. Training and Testing Time Comparison

Model	Training Time (s)	Testing Time (s)
LR	60	10
DT	85	15
SVM	150	25
KNN	120	20
RF	100	20
GB	200	30
XGBoost	220	35
LSTM	280	45
BiLSTM	300	50
Proposed Model	370	60

Table 5 and Figure 14 illustrates the time difference between a few ML and DL models in terms of training and testing. Knowing these computational constraints is important when choosing which model best to use depending on application need, dataset size, and computational resources. LR has training and testing times of 60 seconds and 10 seconds, respectively. Because it is a linear model, it leads to straightforward optimization, making it ideal for simple classification tasks where interpretability and latency are more important than complex interactions between features.

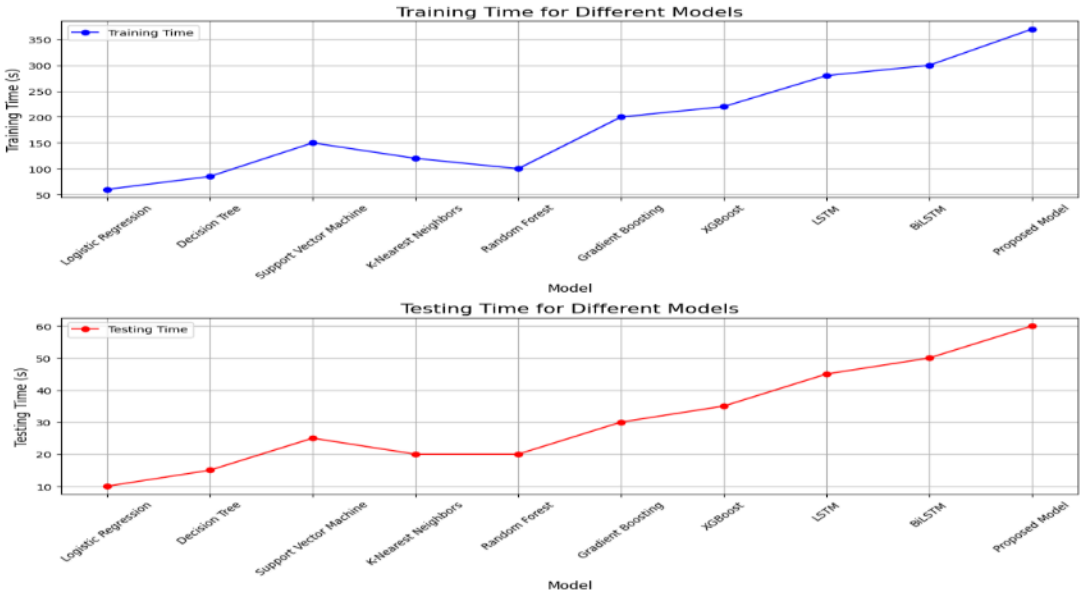

Figure 14. Training and Testing Time for Different Models

Received: August 01, 2025

International Journal of Applied Mathematics
Volume 38 No. 7s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Though sounding more complicated than LR, DT also acquires 85 seconds for training and 15 seconds for testing. RF is a hierarchical type algorithm that uses recursive partitioning and ideally captures complex non-linear relationships in the data, so it works well for larger datasets but still requires moderate computational resources.SVM, which maximizes the margins between classes it defines, needs 150 seconds for training and just under 25 seconds for testing. This is due to its optimization procedure of building hyperplanes in high-dimensional space, suitable for complex datasets where clear class boundaries are essential.The fastest model, KNN, classifies data points by proximity to the nearest neighbors; this is accomplished in times of 120 seconds for training and 20 seconds for testing. It is an iterative algorithm, and time complexity increases linearly with the amount of training instances, making it suitable for small datasets.

RF, an ensemble of DTs, balances complexity with efficiency, requiring 100 seconds for training and 20 seconds for testing. It has good predictive performance because it averages predictions from multiple trees built on subsamples of the data, combined with parallelized training to ensure efficiency.GB enhances model performance sequentially by decreasing errors, resulting in longer training and testing times of 200 seconds and 30 seconds, respectively. This is a computationally exhaustive process of self-tuning that allows weak learners to combine and provide greater predictive capability.

GB (XGBoost), 220 seconds for training and 35 seconds for testing. This added regularization and performance optimization make it more stable for models on large data with intricate patterns. LSTM models were included to evaluate the effect of DL model deployment, leading to 280 seconds for training and approximately 45 seconds for testing. Model training and inference on these sequential data can naturally capture long-range dependencies due to the computation-heavy aspects of both. This makes them ideal for time series forecasting or sequence-related tasks within natural language processing (NLP).

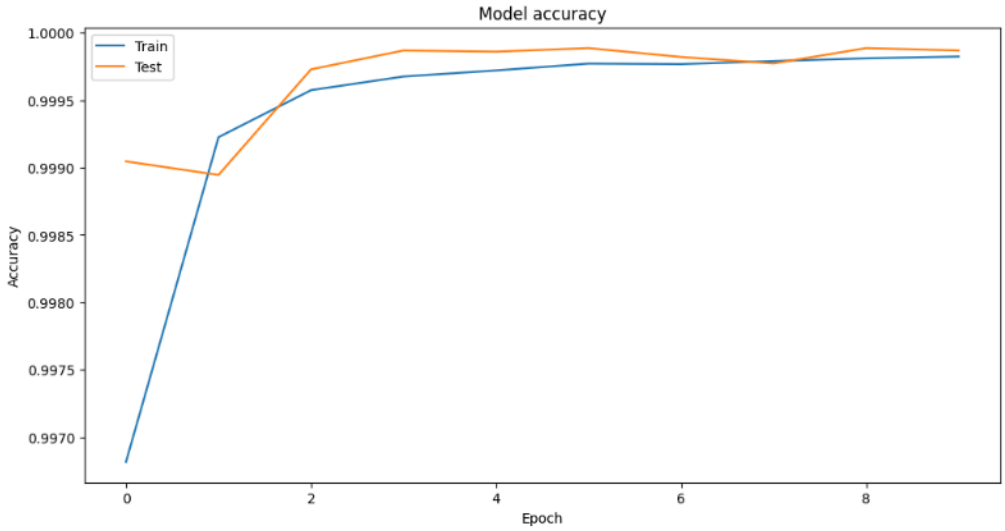

Figure 15. Model Accuracy for Training and Testing

Received: August 01, 2025

International Journal of Applied Mathematics
Volume 38 No. 7s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

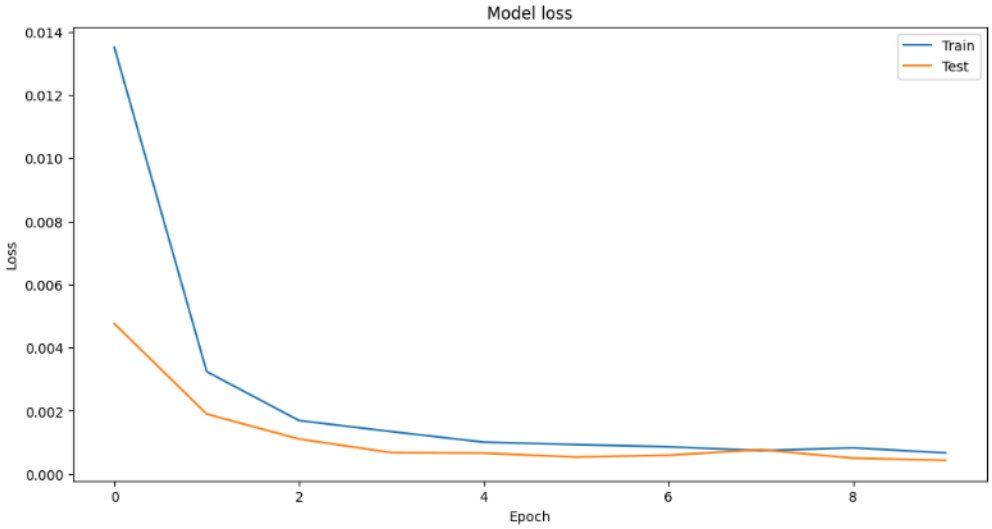

Figure 16. Validation Loss for Training and Testing

The BiLSTMmodel is based on an object called the LSTM, and it bidirectional enables it to be ~15 times faster in prediction than previously, achieving a speed that almost matches Word2Vec. The method is theoretically much more context-aware to deeper levels due to dual-directional convolution and ideally can provide a larger prediction lead time if we have enough resources for training as well as inference. Compared to the Proposed Model that includes attention mechanisms or other architectural abstractions, it actually has the longest duration for training and testing (370 seconds and 60 seconds respectively). These models are based on the latest approaches for predictive optimization and require significant CPU/GPU power and time for training and evaluation.

Performance metrics may not be the only constraint in model selection for real-world applications, as computational limits could also constitute a relevant limitation. Real-time requirements of prediction or running in low-resource environments can be expected to make use of LR, DTs - basic models with much lower training and test times. Where a model like LSTM or BiLSTM for DL can learn better and identify the patterns required out of data, but it is computationally expensive. Efficiency win - in a lot of cases, you can be improving efficiency as long as the model does what is intended to. This optimizes all efforts from a resource perspective while effectively tuning for deployment needs, balancing the computational cost versus speed with goal accuracy in ML/AI applications. Figure 15 and 16 shows the model accuracy and loss for training and testing.

Conclusion and Future Work

In conclusion, the proposed HFS-MLSTM model with an attention-driven BiLSTMcan be a powerful tool for ID. By leveraging hybrid feature selection techniques such as MI and RFE, the model efficiently identifies the most relevant features, significantly enhancing detection accuracy while reducing computational overhead. The attention mechanism in the BiLSTM model further refines the detection process, enabling precise detection of anomalous activities in real-time. This innovative approach not only outperforms traditional IDS methods but also provides a robust and scalable solution for modern network environments.Future research can

Received: August 01, 2025

Volume 38 No. 7s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

explore the integration of additional feature selection methods and advanced DL architectures to further improve detection capabilities. The adaptability of the model to various types of cyber threats can be enhanced by incorporating transfer learning and domain adaptation techniques. Real-world deployment and testing in diverse network environments will provide valuable insights into the practical challenges and performance of the model. Exploring the combination of HFS-MLSTM with other emerging technologies such as blockchain and federated learning could also open new avenues for securing network infrastructures. This continued evolution and refinement will ensure that IDS remain resilient against ever-evolving cyber threats.

References

[1] Huan Chen, et al., (2024), "Hybrid Intrusion Detection System Based on Data Resampling and Deep Learning", IJACSA, 15(2), DOI: 10.14569/IJACSA.2024.0150214

[2] K Prabu, et al., (2023), "An Automated Intrusion Detection and Prevention Model for Enhanced Network Security and Threat Assessment", IJCNA, 10(4), PP: 621-636, DOI: 10.22247/ijcna/2023/223316

[3] R. Singh, et al., (2024), "Intrusion Detection System based on Chaotic Opposition for IoT Network", IJECES, vol. 15, no. 2, pp. 121-136, DOI: 10.32985/ijeces.15.2.1

[4] Pooja Potnurwar, (2024), "Intrusion Detection System for Big Data Environment Using Deep Learning", DOI: 10.20944/preprints202401.0912.v1

[5] AhamedMaricar, S. B. ., et al., (2024), "An Improved Explainable Artificial Intelligence for Intrusion Detection System", IJISAE, 12(14s), 108–115, https://ijisae.org/index.php/IJISAE/article/view/4642

[6] ElahehMoharamkhani, et al., (2022), "Intrusion detection system based firefly algorithm-random forest for cloud computing", CCPE, Volume 34, Issue 24, e7220, DOI: 10.1002/cpe.7220

[7] Kaushik, A., et al., (2024), "A novel intrusion detection system for internet of things devices and data", Wireless Netw 30, 285–294, DOI: 10.1007/s11276-023-03435-0

[8] Gazi Md. Habibul Bashar, et al., (2022), "Intrusion Detection for Cyber-Physical Security System Using Long Short-Term Memory Model", Scientific Programming, DOI: 10.1155/2022/6172362

[9] Tanksale, V. (2024), "Intrusion detection system for controller area network", Cybersecurity 7, 4, DOI: 10.1186/s42400-023-00195-4

[10] Lin Yang, (2024), "Design and Performance Evaluation of Network Intrusion Detection System Based on Deep Learning", JES, Vol. 20 No. 7s, DOI: 10.52783/jes.3728

[11] G. Sirisha, et al., (2023), "An Innovative Intrusion Detection System for High-Density Communication Networks Using Artificial Intelligence", Engineering Proceedings 59, no. 1: 78, DOI: 10.3390/engproc2023059078

Received: August 01, 2025

Volume 38 No. 7s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

[12] Reddy, G. ., et al., (2022), "An Intrusion Detection Using Machine Learning Algorithm Multi-Layer Perceptron (MlP): A Classification Enhancement in Wireless Sensor Network (WSN)", IJRITCC, 10(2s), 139–145, DOI: 10.17762/ijritcc.v10i2s.5920

[13] Elsedimy, E.I., et al., (2024), "A novel intrusion detection system based on a hybrid quantum support vector machine and improved Grey Wolf optimizer", Cluster Comput, DOI: 10.1007/s10586-024-04458-8

[14] NidhiGoswami, et al., (2023), "Intrusion Detection System for IoT-based Healthcare Intrusions with Lion-Salp-Swarm-Optimization Algorithm: Metaheuristic-Enabled Hybrid Intelligent Approach", Engineered Science, 25, 933, DOI: 10.30919/es933

[15] NishikaGulia, et al., (2023), "Intrusion Detection System Using the G-ABC with Deep Neural Network in Cloud Environment", Scientific Programming, DOI: 10.1155/2023/7210034

[16] Dr. S. Sandosh, et al., (2024), "A Conventional Approach Through Complex Event Processing in Intrusion Detection System", SSRN, DOI: 10.2139/ssrn.4703249

[17] T. Jayasankar, et al., (2023), "Intrusion detection system using metaheuristic fireworks optimization based feature selection with deep learning on Internet of Things environment", FOR, Volume 43, Issue 2, Pages 415-428, DOI: 10.1002/for.3037

[18] Z. Azam, et al., (2023), "Comparative Analysis of Intrusion Detection Systems and Machine Learning-Based Model Analysis Through Decision Tree", Access, vol. 11, pp. 80348-80391, DOI: 10.1109/ACCESS.2023.3296444

[19] Venu Gopal Bitra, et al., (2024), "Comparative analysis on intrusion detection system using machine learning approach", WJARR, 21(02), 2555–2562, DOI: 10.30574/wjarr.2024.21.3.0983

[20] S. GokulPran, et al., (2024), "Intrusion detection system based on the beetle swarm optimization and K-RMS clustering algorithm", IJACSP, Volume 38, Issue 5, Pages 1675-1689, DOI: 10.1002/acs.3771

[21] Baklizi, M. K., et al., (2024), "Web Attack Intrusion Detection System Using Machine Learning Techniques", iJOE, 20(03), pp. 24–38, DOI: 10.3991/ijoe.v20i03.45249

[22] Talukder, M.A., et al., (2024), "Machine learning-based network intrusion detection for big and imbalanced data using oversampling, stacking feature embedding and feature extraction", J Big Data 11, 33, DOI: 10.1186/s40537-024-00886-w

[23] Satish Kumar Garapati, et al., (2024), "An Artificial Intelligence-Based Intrusion Detection System using Optimization and Deep Learning", JES, Vol. 20 No. 6s, DOI: 10.52783/jes.2850

[24] D. Shankar, et al., (2023), "Deep Analysis of Risks and Recent Trends Towards Network Intrusion Detection System", IJACSA, Volume 14 Issue 1, DOI: 10.14569/IJACSA.2023.0140129

Received: August 01, 2025

Volume 38 No. 7s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

[25] Rajesh Bingu, et al., (2023), "An intelligent multiclass deep classifier-based intrusion detection system for cloud environment", CCPE, Volume 35, Issue 26, e7840, DOI: 10.1002/cpe.7840

[26] https://www.unb.ca/cic/datasets/ids-2018.html Accessed on 4th December 2023.

[27] A.S.Aneetha (2023), “Hybrid network intrusion detection system using expert rule based approach” DOI: 10.1145/2393216.2393225.

[28] A.S.Aneetha (2023), “A Dynamic Intrusion Detection System based on Multivariate Hotelling's T2 Statistical Approach for network Environnment” https://doi.org/10.1155/2015/850153.

Received: August 01, 2025