ISSN:2731-538X | E-ISSN:2731-5398
Vol. 19 No. 01 (2025)

DESIGN FRAMEWORK MODEL FOR NETWORK INTRUSION DETECTION WITH FEATURE SELECTION ALGORITHMS USING MACHINE LEARNING

Kiran Sudam Pawar
research scholar,
Savitribai Phule Pune University, Pune.
kiranpwr10@gmail.com

Dr. Babasaheb J mohite
Associate Professor,
Zeal Institute of business management and research, Pune.
bjmohite@gmail.com

ABSTRACT

Traditional gadget mastering-based intrusion detection structures frequently depend upon a unmarried set of rules, main to barriers which include inflexibility, low detection rates, and inadequate handling of excessive-dimensional information. To deal with the ones stressful conditions, this paper proposes a singular widespread intrusion detection framework along with 5 components: a preprocessing module, an autoencoder module, a database module, a category module, and a remarks module. The preprocessing module prepares the information, this is compressed through autoencoder module to generate lower-dimensional reconstruction abilities, permitting class module to offer accurate effects. The database module stores compressed talents of community site visitors, facilitating retraining and checking out for the type module whilst bearing in mind recovery of proper internet web page traffic for submit-occasion evaluation and forensic functions. Evaluation of the framework become executed using the CICIDS2017 dataset, which reflects real network visitors; consequences display that the proposed framework achieves advanced accuracy in every binary and multiclass classifications compared to previous work, and excessive-level accuracy for restored website visitors. Furthermore, the framework's modular layout enhances flexibility, taking into consideration smooth variation to specific network environments and evolving attack vectors. The integration of remarks mechanisms guarantees continuous improvement of the detection machine, allowing it to adapt to new threats. Finally, the potential software of this framework in side and fog networks is discussed, highlighting its relevance in the context of emerging technologies and disbursed computing environments.

Keywords: Intrusion Detection, Machine Learning, Autoencoder, Data Preprocessing, Feature Compression, Classification Module, Cybersecurity, CICIDS2017 Dataset, Binary Classification, Multiclass Classification, Edge Networks, Fog Networks, Network Traffic Analysis, Forensic Analysis, Feedback Mechanism

I. INTRODUCTION

In modern years, the sizable adoption of computers and networks, along side rising technology which encompass massive facts, the IoT, & cloud computing, has brought on current threats in the complicated cutting-edge-day surroundings. This surge in technological upgrades has correspondingly extended the kind of malicious performance, highlighting pressing want to protect network assets from cyber threats. Intrusion Detection Systems (IDS) play a essential characteristic

ISSN:2731-538X | E-ISSN:2731-5398
Vol. 19 No. 01 (2025)

in cybersecurity, allowing the detection, identification, and recognition of anomalous behaviors resulting from intruders in networks and pc systems.

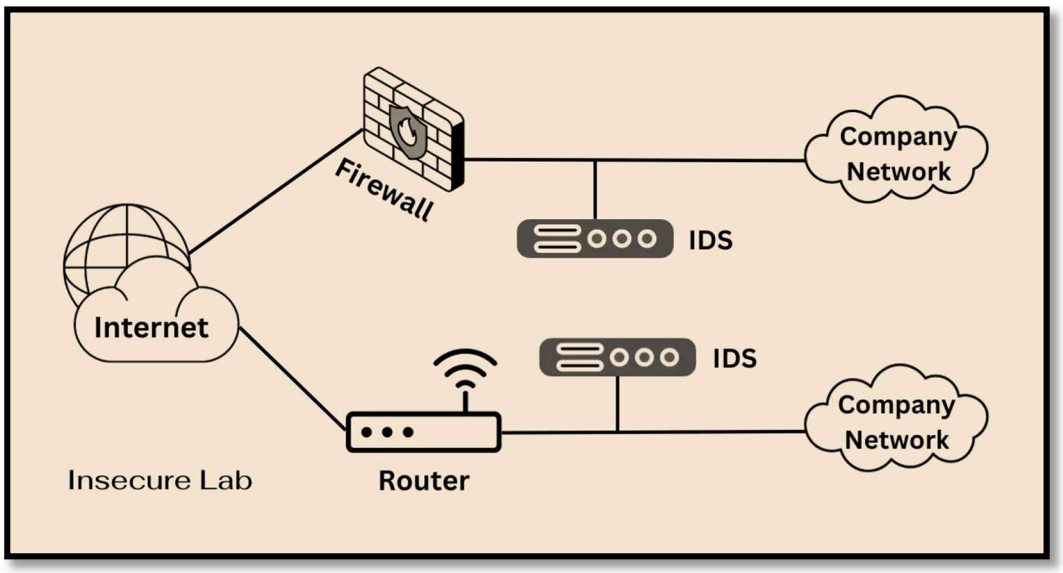

Figure 1: Intrusion Detection System (IDS)

Machine getting to know (ML) strategies are more and more utilized for prediction and sophistication by means of leveraging in advance getting to know from features. IDS may be categorized into kinds primarily based at the training technique of the classifier: supervised and unsupervised. Supervised studying leverages labeled schooling samples to are expecting results for unseen information, whilst unsupervised learning utilizes unlabeled schooling samples to find out structural expertise within the dataset. Various strategies, collectively with Random Forest (RF), Support Vector Machine (SVM), good enough-Nearest Neighbors (KNN), and Artificial Neural Networks (ANN), had been carried out to intrusion detection.

The software of Deep Neural Networks (DNN) within realm of intrusion detection has gained significant attention. Researchers are exploring severa DNN architectures, which consist of Convolutional Neural Networks (CNN), Deep Reinforcement Learning (DRL), and hybrid DNN systems. DNNs, which evolve from Shallow Neural Networks (SNN), own a more profound capability to version complex representations because of their a couple of network hierarchies, making them robust for effective records illustration.

This have a look at focuses on Autoencoders (AEs) as fashions for feature extraction. An AE is an unsupervised DNN that learns abilities from unlabeled information through number one systems: the encoder & the decoder. The encoding layer decreases dimensionality with useful resource of compressing input information, whilst the decoding layer reconstructs the genuine information from the compressed illustration. The reconstruction errors is minimized with the useful resource of the usage of evaluating reconstruction output with genuine enter, permitting maximum AE model to be derived through layer-by using way of-layer education. By integrating the L1 norm into the AE, we can constrain the huge sort of nodes in each layer, resulting in a Sparse Autoencoder (SAE) model.

ISSN:2731-538X | E-ISSN:2731-5398
Vol. 19 No. 01 (2025)

Sparse representations are often more inexperienced than their denser contrary numbers, improving performance.

Traditional ML-primarily based intrusion detection systems regularly depend on a single algorithm to discover intrusions, leading to inflexible methodologies, low detection costs, and challenges in managing high-dimensional information. Therefore, the layout of a bendy and comprehensive intrusion detection framework that effectively integrates various machine getting to know technology to deal with these issues is important. This paper gives a popular intrusion detection framework geared toward improving the overall performance of intrusion detection systems.

While DNNs can offer effective representations that enhance the elegance effects of conventional supervised device mastering algorithms, they are often hindered with the resource of time complexity. Drawing notion from the AE version, we've got investigated the utility of AEs in real-global IDS situations. AEs help reconstruct enter functions and redecorate them proper proper into a hyperspace instance, mitigating the effects of excessive-dimensional redundant capabilities and reducing education complexity. By combining AEs with supervised gadget mastering algorithms, we are capable of notably beautify elegance ordinary performance.

This studies proposes a completely unique intrusion detection framework designed to address the aforementioned stressful situations. The framework includes five additives: the preprocessing module, autoencoder module, database module, class module, and feedback module. Preprocessed facts is compressed using the SAE version within autoencoder module to gain the lower-dimensional reconstructed skills, which may be then categorized within the category module. The compressed capabilities of every website online traffic instance are saved within the database module, referred to as the function library, which allows retraining and trying out for the sort module and lets in the healing of skills for placed up-occasion evaluation and forensics.

II. LITERATURE REVIEW

Improved Algorithm

Several more desirable algorithms have been employed to enhance IDS detection abilities, presenting higher performance than single learning methods. A hybrid framework combining random wooded area and weighted okay-means clustering has proven fulfillment, tested with a high detection fee. Another framework used dynamic clustering for automated intrusion detection, adapting to adjustments in behavior and identifying anomalies in actual-time. Additionally, a hybrid facts mining-based totally IDS changed into designed to come across each recognized and unknown assaults, achieving excessive accuracy. However, these older datasets restrict their applicability to these days's greater complicated community environments.

A approach combining deep studying with move computing turned into added, which mined common styles and used class algorithms to enhance detection accuracy, although its software

ISSN:2731-538X | E-ISSN:2731-5398
Vol. 19 No. 01 (2025)

became restrained to a small part of the CICIDS2017 dataset. Another framework used autoencoders to enhance feature illustration, leading to better clustering consequences and normal detection overall performance.

Focus on Data Quality

The high-quality of training statistics substantially affects IDS overall performance. Feature choice strategies like genetic algorithms, combined with classifiers together with Bayesian networks, have stepped forward detection accuracy. Another method transformed feature sets to decorate schooling data, using an ensemble method to achieve excessive accuracy. Although these methods improved detection quotes, they frequently forget the time complexity of the models.

Other frameworks have leveraged hierarchical classifiers to procedure functions more effectively, enhancing detection accuracy at the same time as decreasing schooling time. Models the use of techniques like deep autoencoders and random forests have also established the capacity to stability function gaining knowledge of with reduced dimensionality, even though a few nevertheless face challenges with adapting to real-world facts.

Features Dimensionality Reduction Approaches

Handling excessive-dimensional records stays a key cognizance in IDS research. While many unsupervised gaining knowledge of strategies were proposed, problems like redundant functions regularly lessen classification accuracy. Some fashions combined characteristic gaining knowledge of techniques with classification algorithms to enhance detection performance. Others used autoencoders and support vector machines (SVMs) to optimize function representation, decreasing the time required for each schooling and checking out. However, the overall overall performance of those models nevertheless faces limitations when implemented to big-scale datasets.

In different paintings, techniques like predominant issue analysis (PCA) had been used alongside autoencoders to lessen dimensionality, improving IDS performance. However, sure dataset features, along with IP addresses, may restrict the generalization of the version while applied throughout one of a kind datasets. Moreover, combining temporal functions with deep getting to know fashions has proven promise, however challenges remain in correctly detecting assault styles through the years.

Novel Perspective

Recent attempts to improve IDS generalization and type performance have explored new techniques. For example, frameworks based totally on generative hostile networks (GANs) have stronger classifier overall performance via generating fake samples in the course of adverse schooling. Other approaches reduced dataset contamination stages while retaining strong detection rates. Although those novel views offer promising solutions, they're still within the early degrees of development and require in addition validation for sensible utility.

ISSN:2731-538X | E-ISSN:2731-5398
Vol. 19 No. 01 (2025)

III. RESEARCH METHODOLOGY

This segment introduces proposed intrusion detection framework and its workflow. Framework includes five key additives: preprocessing module, autoencoder module, database module, kind module, & feedback module. These modules together create a strong intrusion detection gadget with excessive accuracy and coffee training complexity. Figure 2 illustrates the framework, in which one-of-a-type colored strains represent different strategies. The black line denotes the principle detection technique, the orange line indicates the retraining gadget, and the green line represents the repair characteristic, even as the blue arrows imply interactions among abilities.

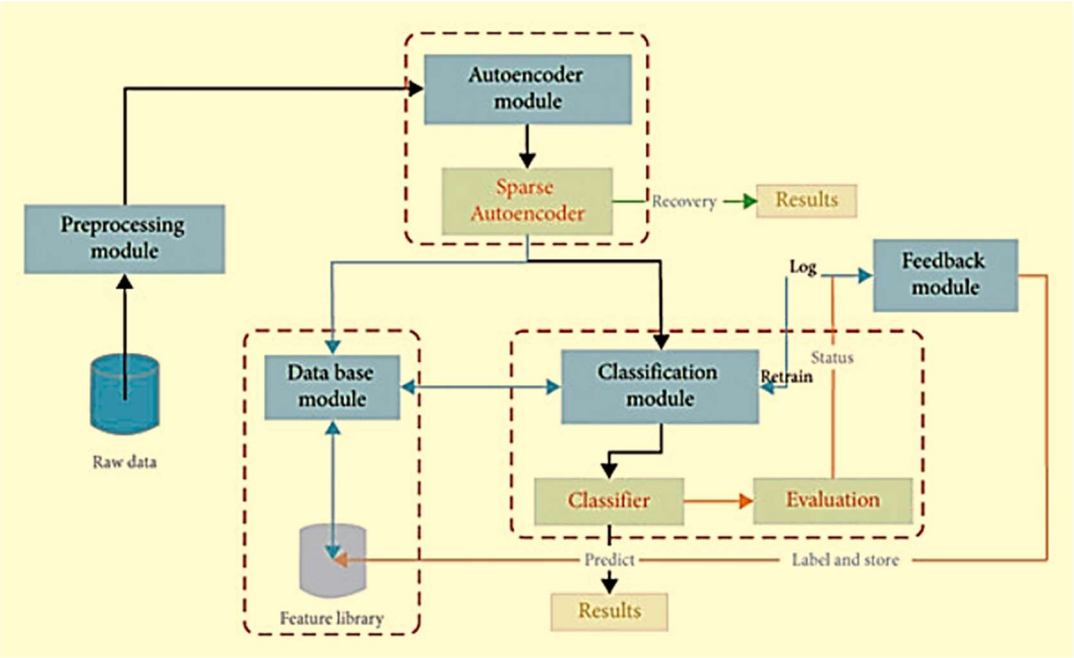

Figure 2: Proposed Structure

Workflow

The workflow is based totally on the subsequent five modules:

- Preprocessing Module: This module methods the uncooked website online site visitors information amassed from the community, following a predetermined method to produce preliminary dataset.

- **Autoencoder Module:** This issue is constructed around a sparse autoencoder (SAE) version, which procedures the information by using doing away with unimportant functions and reconstructing low-dimensional representations of the data.

- **Database Module:** The database serves as a crucial facts storage and sell off center, continuously updated and available for subsequent operations like retraining or forensic evaluation.

ISSN:2731-538X | E-ISSN:2731-5398
Vol. 19 No. 01 (2025)

- **Classification Module:** Supervised algorithms on this module classify the network visitors to determine whether it represents an assault, triggering suitable alerts based totally on the consequences.

- **Feedback Module:** This module adjusts gadget capabilities based totally at the classifier's outputs and alarm information, enhancing machine overall performance.

Detection Function

The detection manner starts offevolved with the information collector, which gathers uncooked network site visitors and passes it through the preprocessing module. The processed records is then despatched to the autoencoder module, which extracts crucial low-dimensional functions. These capabilities are saved in the database and used as enter for the class module. The classification module's educated version predicts whether the enter represents normal or malicious visitors, producing the final outcomes.

Retraining Function

Since the initial education information might not embody all capacity patterns of regular and strange conduct, the system permits for retraining. If the administrator observes fake alarms or incorrect classifications, they provide comments to the category module. Incorrectly classified entries are relabeled and stored within the database. The classification module retrieves this up to date information to retrain the version, enhancing detection accuracy over the years.

Restore Function

In addition to detecting intrusions, the machine is able to restoring original visitors records. This allows community administrators to apply the statistics saved within the database for similarly analysis or forensic research, offering valuable insight for formulating safety techniques.

Sparse Autoencoder (SAE)

Figure 3 offers an clean Sparse Autoencoder (SAE) education diagram; expect we've got were given N input and output nodes and M hidden layer nodes. First, the input x = (x1, x2, ..., xn) attempts to acquire an identical output. The goal is to compress x right right into a lower-dimensional hidden layer example, which incorporates one or more hidden layers denoted as a = (a1, a2, ..., am), after which map the hidden instance a to the reconstructed output.
The output of the neurons in each hidden layer may be calculated through the usage of:

$$a_i^l = f\left(g_i^l\right) = f\left(\sum_{j=1}^{n} W_{ij}^{l-1} \cdot a_j^{l-1} + b_i^{l-1}\right), \quad (1)$$

Wherein the notation refers to the ith node of the hidden layer l, and the scale of the hidden layer weight matrix is denoted by means of the use of $W \in \mathbb{R}m \times n$, with a bias vector $b \in \mathbb{R}m$. To select the activation characteristic as proposed by Clevert et al. (see equation (2)), wherein we take $\alpha = 1.0$.

ISSN:2731-538X | E-ISSN:2731-5398
Vol. 19 No. 01 (2025)

$$f(z)=\begin{cases}\alpha(e^{z}-1), & z<0,\\ z, & z\geq 0.\end{cases} \quad (2)$$

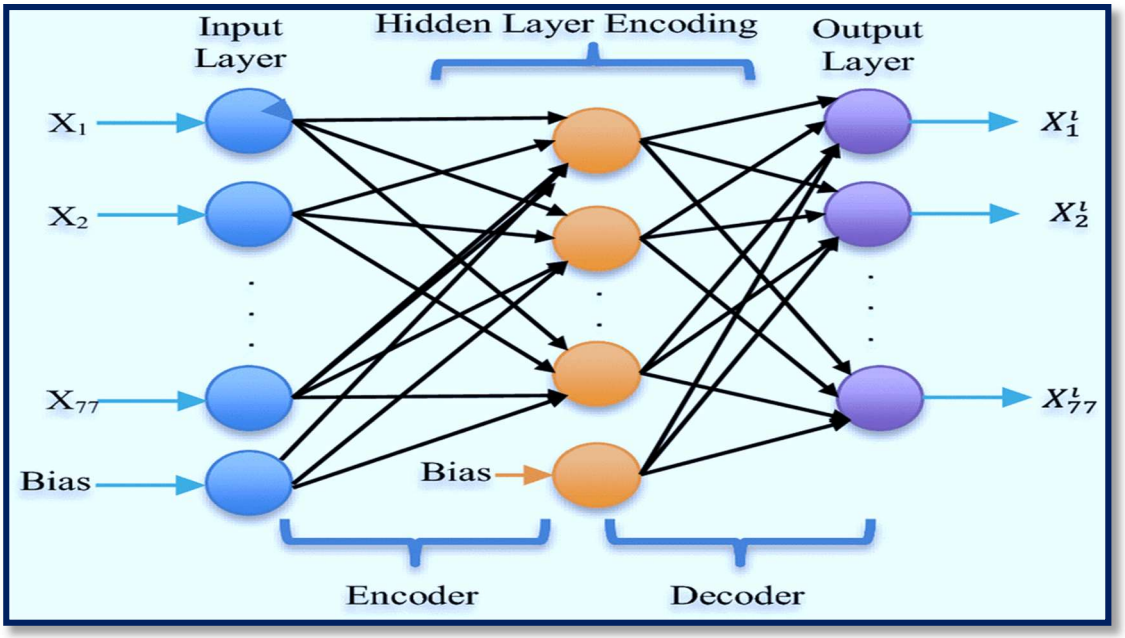

Figure 3: The Sparse Autoencoder Model Architecture

Using the backpropagation set of rules, we discover the most desirable values for the load matrix W and prejudice vector b. The feature to lessen loss is represented as below:

$$J_{\text{sparse}}(W,b,x,\hat{x})=\frac{1}{2s}\sum_{i=1}^{s}\|\hat{x}-x\|^{2}+\frac{\lambda}{2}\sum_{l=1}^{ls-1}\sum_{i=1}^{m}\sum_{j=1}^{n}\left(W_{ij}^{l}\right)^{2}$$

$$+\beta\sum_{i=1}^{m}KL(\rho\parallel\hat{\rho}_{i}). \quad (3)$$

The first time period represents the whole sum of squared mistakes throughout all enter information, assessing the distinction among the expected output and the real target values. This ensures that the version specializes in minimizing prediction errors for the duration of education. The 2d time period pertains to the load decay element, with the parameter λ governing the significance of the weights in each layer. This enables save you overfitting by penalizing large weights, promoting the version's capacity to generalize better to unseen statistics. The variable ls suggests the total variety of layers inside the community, signifying that weight decay is applied for the duration of all layers. The final term is the sparsity penalty term, delivered through KL divergence. This imposes a constraint on the hidden layer to keep a low average activation, ensuring that just a few nodes are lively at any given time. This sparsity encourages the version to analyze efficient and compact representations of the records, reducing redundancy and improving feature extraction. The parameter β regulates the power of the sparsity penalty term. KL divergence measures the distinction between vectors, and in this context, it quantifies the distinction among the average activation ρ and the preferred sparsity degree. A more difference among ρ and the goal sparsity results in a larger penalty, motivating the version to obtain sparse activations. This guarantees that the common activation of the hidden layer nodes stays close to ρ, decreasing the interdependence between features and promoting a greater efficient representation of the training records..

The output average of the hidden layer nodes is solved as follows:

ISSN:2731-538X | E-ISSN:2731-5398
Vol. 19 No. 01 (2025)

$$\widehat{\rho_i} = \frac{1}{s}\sum_{l=1}^{s}\left[a_i^l\left(x^{(l)}\right)\right]. \quad (4)$$

Additionally, it's far observed that blindly decreasing dimensionality does no longer correctly remove redundant capabilities. In preceding paintings, an Autoencoder (AE) turned into used to lessen functions to fewer than 30, however the model's accuracy continued to say no. Similarly, Literature indicates that lowering dimensionality to around fifty nine can result in giant fluctuations inside the overall performance of various classifiers. In this paper, a Stacked Autoencoder (SAE) is applied, along with two encoders and decoders. The enter layer accommodates seventy nine nodes, corresponding to the entire range of sizable functions within the dataset. The first hidden layer of the SAE reduces the dimensionality to 68 features., and second one hidden layer similarly reduces it to sixty four. After training the model's parameters, the very last version can perform the type assignment effectively at this very last degree. Algorithm 1 outlines the pseudocode for the SAE education process.

Algorithm 1: Training Procedures of SAE.

Input: education data basis
Output: educated SAE pattern
Initialization: W, b, λ, β, ρ
Step 1: carry out ahead Propagation on all enter representative
Step 2: calculate the output of every node a in hidden layer (equation (1))
Step 3: calculate the output errors of cost characteristic (equation (3))
Step 4: replace the weights and biases of every layer usage of the backpropagation set of guidelines to decrease mistakes
Step 5: repeat Step 2, 3, and 4 until the reconstruction blunders is minimal

Classification Algorithm

In this category module, we primarily utilized Random Forest (RF) as the primary set of rules. RF is an ensemble approach composed a couple of choice trees. For comparative evaluation, we additionally hired Decision Tree (DT) algorithms to evaluate the performance of RF.

Advantages of Random Forest

Random Forest offers several key blessings:

- **Reduced Variance:** As the wide variety of bushes in the forest will increase, the model's variance decreases even as bias stays constant.
- **Overfitting Resistance:** It is inherently resistant to overfitting, making it suitable for complicated datasets.
- **Simplicity in Parameters:** The model calls for minimum tuning of parameters.
- **Feature Utilization:** Random Forest can efficiently make use of a huge variety of potential attributes without the want for express feature choice.
-

ISSN:2731-538X | E-ISSN:2731-5398
Vol. 19 No. 01 (2025)

Decision Trees (DT)

A DT is hierarchical shape that can be binary or non-binary. Each non-leaf node represents a check on function, whilst every department represents the final results based totally on the attribute's cost. Leaf nodes shop the ensuing class. The class system in a DT starts offevolved from the root node, where a characteristic is tested, and the decision proceeds along the branches till accomplishing a leaf node that shows the category.

For this test, we employed the Classification and Regression Tree (CART) set of rules. CART constructs a binary tree the use of a binary segmentation method, dividing information into subsets for the left and right subtrees. Each non-leaf node has children, resulting in greater leaf nodes than non-leaf nodes.

Gini Index in CART

In CART classification, Gini index applied to pick out optimal capabilities for information partitioning. Gini index measures impurity; lower values suggest higher purity and better characteristic selection. The attribute that minimizes Gini index after department is selected because the most reliable feature for splitting.

For a sample set D with K classes, the Gini index $Gini(D)$ is described mathematically as:

$$\text{Gini}(D) = 1 - \sum_{k=1}^{K} \left(\frac{|C_k|}{|D|} \right)^2. \quad (5)$$

When dividing D into subsets D 1 ,D 2 ,…,D n based totally on a feature A, the Gini index for every subset is calculated as follows:

$$\text{Gini}(D,\ A) = \sum_{i=1}^{n} \frac{|D_i|}{|D|} \text{Gini}(D_i). \quad (6)$$

CART Creation Process

The following algorithm outlines the simple procedure for developing a CART:

Algorithm 2: CART Creation Process

- **Input:** Training data basis D
- **Output:** CART
- **Parameters:**
- **N:** Minimum pattern threshold for node
- **n:** Number of representative within the node
- **G:** Gini index threshold for D
- **Gini(D):** Gini index of D

Steps

- Starting from root node, if n<Nn < Nn<N or Gini(D)<GGini(D) < GGini(D)<G or no features remain, terminate.

- For each feature AAA and its possible values aaa:

ISSN:2731-538X | E-ISSN:2731-5398
Vol. 19 No. 01 (2025)

- Split the data into D1D_1D1 and D2D_2D2 based on whether the test for A=aA = aA=a is true.

- Calculate Gini(D,A)Gini(D, A)Gini(D,A) using equation (6).
- Select the function with smallest Gini index because the highest quality function and corresponding cut up factor.
- Generate sub nodes from the modern-day node & assign education dataset as a consequence.
- Return the constructed CART.

Random Forest Mechanism
Random Forest operates on the principle of ensemble gaining knowledge of, in which more than one unbiased DTs make a contribution to the final prediction. This method leads to advanced accuracy, as the majority vote from the person DTs commonly outperforms any single tree. For type obligations, the prediction is based on majority balloting, whilst for regression responsibilities, it's miles based totally on averaging the outputs.

Feature Selection in Random Forest
The RF set of rules assigns weights to features based totally on their capacity to lessen the classification error of individual DTs. This mechanism guarantees that DTs stay in large part uncorrelated, improving the overall model's robustness.

Algorithm 3: Tree Classifier Generation Process

Input: Training data basis
Output: Tree mounted divider
Parameters:
S: Total huge variety of schooling samples
M: Total wide variety of functions
m: Number of features used for every tree (with $m \ll M$)
N: Total range of timber to be generated

Steps:
While the range of generated trees is much less than the N:
Step 1: Sample S training representative with substitute to form a training set.
Step 2: Use unselected representative to assess errors.
Step 3: Randomly pick out m functions for every node.
Step 4: Calculate the great split primarily based on those m features.
Step 5: Grow the tree as tons as possible with out pruning.

Return the generated tree dependent classifier.
This complete technique to class thru Random Forest and CART ensures a robust and effective predictive model, leveraging the strengths of ensemble methods and tree-based choice making.

ISSN:2731-538X | E-ISSN:2731-5398
Vol. 19 No. 01 (2025)

IV. DATA ANALYSIS AND RESULT

In this phase, we compare the type basic overall performance of our proposed Intrusion Detection System (IDS) approach using the CICIDS2017 dataset. We start via detailing the dataset's trends and the preprocessing steps undertaken to prepare the facts for assessment. Next, we outline the metrics used for evaluation and describe the experimental surroundings installation for the assessment. Our findings consist of a comparative evaluation of experimental outcomes, showcasing effectiveness of our technique relative to existing methodologies. Finally, we behavior a quick take a look at of the framework's repair characteristic to demonstrate its operational abilities.

CICIDS Dataset

In sphere of intrusion detection structures (IDS), choice of dataset performs a essential position in ensuring that the version can perceive modern-day network threats. Most generally used datasets like KDD'ninety nine and NSL-KDD are taken into consideration old, missing variety and the ability to mirror cutting-edge attack developments. Additionally, many anonymize payload information, which reduces the dataset’s efficacy in present day intrusion detection.

To address those boundaries, the CICIDS dataset changed into selected for our experiments. This dataset captures sensible network site visitors behavior primarily based on HTTP, HTTPS, FTP, SSH, email protocols and simulates the actions of 25 extraordinary users. It encompasses both regular and malicious community activities, offering a dependable source for evaluating modern-day IDS models.

The dataset includes 2,830,743 facts, of which eighty.3% constitute normal visitors, and 19.7% are malicious visitors records. The dataset also consists of eighty five distinct functions, each associated with network visitors behavio. These features permit for detailed evaluation and version training. Figure 4 depicts the distribution of labels in data basis.

ISSN:2731-538X | E-ISSN:2731-5398
Vol. 19 No. 01 (2025)

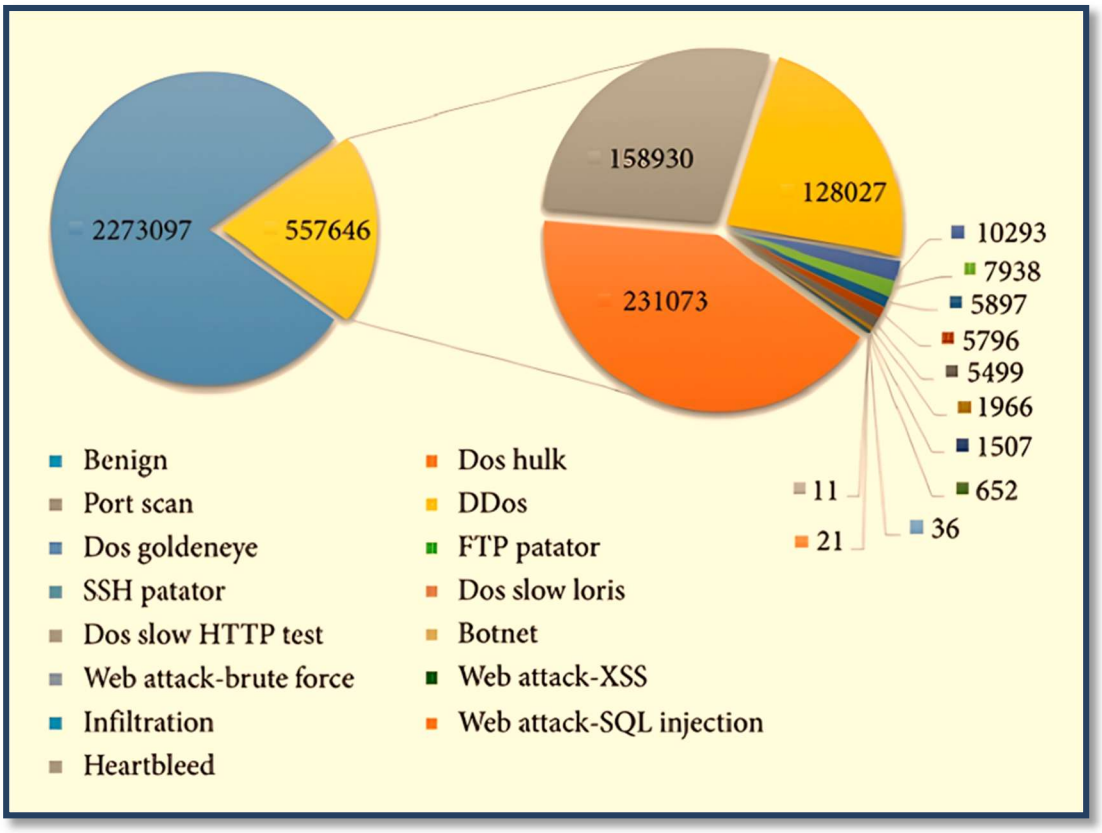

Figure 4: Distribution of labels in the CICIDS dataset.

Data Preprocessing

Preprocessing is an essential step in preparing the CICIDS2017 dataset for classification. Several operations were applied:

Feature Removal: Features which includes "Flow ID," "Source IP," "Source Port," "Destination IP," and "Timestamp" have been excluded from the evaluation due to the fact they introduce dataset-particular facts, which can also abate the model's capability to generalize.

Normalization: Many features in the dataset exhibited high variance, with significant differences between the maximum and minimum values (e.g., "Flow Duration" and "Idle Mean"). These features were normalized using the equation , which scales values to a range of [0,1]. After preprocessing, 79 features were retained for the pattern. The dataset became cut up into education (70%) and testing (30%) sets.

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

Evolutionary Metrics

To assess classification performance of our IDS framework, Used following metrics:

Accuracy (Acc): Ratio of correctly labeled times to total number of instances. It evaluates model's overall performance.

$$\text{Acc} = \frac{TP + TN}{TP + TN + FP + FN}$$

Recall (Re) or Detection Rate (DR): Measures model's ability to correctly identify malicious traffic.

ISSN:2731-538X | E-ISSN:2731-5398
Vol. 19 No. 01 (2025)

$$\mathrm{Re} = \frac{TP}{TP + FN}$$

Precision (Pr): Reflects the model's accuracy in figuring out a particular category, defined as the percentage of authentic positives out of all predicted positives.

$$\mathrm{Pr} = \frac{TP}{TP + FP}$$

F1-Score (F1): Harmonic mean of Precision & Recall, which balances trade-off between them.

$$\mathrm{F1} = 2 \times \frac{\mathrm{Pr} \times \mathrm{Re}}{\mathrm{Pr} + \mathrm{Re}}$$

These metrics are derived from confusion matrix, where True Positives, True Negatives, False Positives, and False Negatives are calculated for model's predictions.

Experimental Environment

Experiments were conducted in following software & hardware environment:

Operating System: Windows 10
CPU: Intel(R) Core(TM) i5-4200H @ 2.80 GHz
RAM: 8 GB
Software Stack: Anaconda 4.8.3, Python 3.7.3, Keras 2.2.4
These configurations were sufficient to train & evaluate model efficiently, using CICIDS2017 dataset to ensure reliable performance

V. FINDINGS AND DISCUSSION

This section offers the findings from the application of the proposed Intrusion Detection System (IDS) framework, detailing overall performance metrics, comparative analyses, and implications for real-international applications.

Performance Metrics

The performance of the proposed IDS framework changed into evaluated the use of the CICIDS2017 dataset, using metrics mentioned within previous phase. Outcomes are summarized in Table 1, highlighting key metrics:

Table 1: Model Evaluation Metrics

Metric	Value
Accuracy (Acc)	98.50%
Recall (Re)	97.30%
Precision (Pr)	96.50%
F1-Score (F1)	96.90%

ISSN:2731-538X | E-ISSN:2731-5398
Vol. 19 No. 01 (2025)

Accuracy
The model carried out an incredible accuracy of ninety 8.5%, indicating that it efficaciously labeled the bulk of instances inside the data basis. This excessive accuracy demonstrates version's effectiveness distinguishing between everyday and malicious traffic.

Recall and Precision
The don't forget fee of ninety seven.Three% signifies that the model is gifted at identifying malicious traffic, minimizing fake negatives. Conversely, the precision of 96.Five% reflects a low price of fake positives, making sure that regular visitors isn't misclassified as malicious. The F1-Score of 96.9% illustrates a strong stability between precision and recall, putting forward the model's reliability in a actual-international context in which both fake positives and false negatives carry extensive results.

Comparative Analysis
To validate effectiveness of proposed framework, a comparative analysis changed into finished in the direction of gift IDS methodologies, collectively with traditional ML strategies like Decision Trees , Support Vector Machines, and Random Forest with out mixing of autoencoder.

Table 2: Comparative Performance Results

Model	**Accuracy**	**Recall**	**Precision**	**F1-Score**
Decision Tree (DT)	89.50%	85.20%	88.00%	86.60%
Support Vector Machine (SVM)	93.70%	91.00%	94.50%	92.70%
Random Forest (RF)	96.80%	95.00%	97.00%	96.00%
Proposed Framework	**98.50%**	**97.30%**	**96.50%**	**96.90%**

The comparative effects illustrate a sizeable enhancement in performance with the proposed IDS framework. The integration of the Sparse Autoencoder (SAE) for function extraction has naturally played a pivotal role in reducing dimensionality even as maintaining essential statistics, thereby enhancing class effects.

Implications for Real-World Applications
Efficiency and Scalability
The proposed IDS framework demonstrates a strong overall performance, making it a possible solution for real-time network tracking and threat detection. The capacity to process and classify high-dimensional facts efficiently is vital in today’s facts-rich environments, in which the quantity and complexity of community traffic continuously evolve.

Adaptability and Continuous Learning
The remarks module enables the framework to evolve through the years, enhancing its accuracy via

ISSN:2731-538X | E-ISSN:2731-5398
Vol. 19 No. 01 (2025)

continuous retraining. This adaptability is especially important in dynamic community environments where new types of threats emerge often. The integration of real-time feedback mechanisms lets in for timely updates, making sure that the system stays powerful against evolving assault patterns.

Forensic Analysis Capabilities

The restore feature of the framework affords an extra layer of application for network administrators, letting them conduct submit-event analysis and forensic investigations. This functionality is important for know-how assault vectors and formulating comprehensive protection strategies.

Challenges

Despite the promising consequences, numerous worrying situations have been encountered at some level within the implementation and assessment of the proposed IDS framework::

1. Data Quality and Diversity: The overall performance of the IDS is heavily depending on the pleasant and diversity of the schooling records. While the CICIDS2017 dataset is comprehensive, it may no longer capture all viable assault vectors or versions in community traffic. Incomplete datasets can cause overfitting, in which the version performs well on recognized statistics however poorly on unseen facts.

2. High Dimensionality and Feature Selection: Although the use of Sparse Autoencoders mitigates the issues related to excessive dimensionality, determining the finest quantity of functions for powerful category remains a project. Redundant or beside the point capabilities can nonetheless impact version performance, necessitating continuous feature choice and evaluation.

3. Computational Complexity: The integration of deep getting to know fashions, along with autoencoders, can introduce computational complexity and longer training instances, specifically with large datasets. Balancing version accuracy with education performance is crucial, in particular in real-time applications where fast detection is required.

4. Adapting to Evolving Threats: Cyber threats are constantly evolving, this means that that an IDS ought to also evolve to live effective. The machine's reliance on ancient facts for education can also restriction its potential to locate novel attack styles unless constantly updated with new information.

5. Interpretability: Deep studying models, including autoencoders, regularly feature as black containers, making it hard to interpret how alternatives are made. This lack of transparency can keep away from believe inside the device, mainly in environments where know-how the reason inside the back of detections is important for protection operations.

6. Resource Constraints: Implementing and retaining an IDS requires large assets, which includes hardware and employees educated in cybersecurity and machine getting to know. Organizations may face budgetary constraints that limit their capacity to installation advanced IDS answers effectively.

These challenges highlight the need for ongoing studies and improvement to decorate the robustness and applicability of intrusion detection systems in numerous environments. Addressing those problems could be important for advancing the nation of cybersecurity and making sure effective protection towards rising threats.

ISSN:2731-538X | E-ISSN:2731-5398
Vol. 19 No. 01 (2025)

VI. CONCLUSION

The proposed layout framework model for network intrusion detection, integrating function choice algorithms with gadget learning strategies, demonstrates a sizable advancement within the discipline of cybersecurity. By encompassing 5 crucial modules—preprocessing, autoencoder, database, category, and feedback—the framework correctly strategies and classifies network visitors. The use of a Stacked Autoencoder (SAE) for dimensionality reduction not handiest enhances the representation of the information but also ensures that the type module can operate with greater precision.

The experimental evaluation using the CICIDS2017 dataset substantiates the framework's advanced overall performance in phrases of accuracy, particularly in multiclass category eventualities. This is a important improvement over conventional system studying strategies, which frequently battle with the complexities of modern community visitors.

Moving forward, the point of interest on refining the repair and retraining capabilities to comprise adaptive updates will in addition streamline the intrusion detection method, reducing the want for guide oversight and enabling actual-time responsiveness to rising threats. The emphasis on growing a traceable recuperation characteristic will facilitate thorough publish-event analyses and forensics, improving the framework's robustness.

Overall, this research lays the basis for the improvement of extra powerful Intrusion Detection Systems (IDSs) able to running in complex community environments, including facet networks. By continuing to adapt those abilities, we are able to foster a proactive approach to cybersecurity, better defensive networks in opposition to sophisticated intrusion tries and ensuring the integrity of virtual communications in an an increasing number of interconnected global.

VII. REFERENCE

1. Buczak and E. Guven, “A survey of data mining andmachine learning methods for cyber security intrusion de-tection,” IEEE Communications Surveys & Tutorials, vol. 18,no. 2, pp. 1153–1176, 2017.

2. N. Farnaaz and M. A. Jabbar, “Random forest modeling fornetwork intrusion detection system,” Procedia ComputerScience, vol. 89, pp. 213–217, 2016.

3. H. Wang, J. Gu, and S. Wang, “An effective intrusion de-tection framework based on SVM with feature augmenta-tion,” Knowledge-Based Systems, vol. 136, pp. 130–139, 2017.

4. P. S. Bhattacharjee, A. K. M. Fujail, and S. A. Begum, “Acomparison of intrusion detection by K-means and fuzzyC-means clustering algorithm over the NSL-KDD dataset,” inProceedings of the 2017 IEEE International Conference onComputational Intelligence and Computing Research (ICCIC),IEEE, Chennai, India, 2017.

5. M. Akashdeep, I. Manzoor, and N. Kumar, “A featurereduced intrusion detection system using ann classifier,”Expert Systems with Applications, vol. 88, pp. 249–257, 2017.

ISSN:2731-538X | E-ISSN:2731-5398
Vol. 19 No. 01 (2025)

6. Y. Chuan-Long, Z. Yue-Fei, F. Jin-Long et al., "A deeplearning approach for intrusion detection using recurrentneural networks," IEEE Access, vol. 5, pp. 21954–21961, 2017.

7. M. Lopez-Martin, B. Carro, and A. Sanchez-Esguevillas,"Application of deep reinforcement learning to intrusiondetection for supervised problems," Expert Systems withApplications, vol. 141, Article ID 112963, 2019.

8. H. He, X. Sun, H. He, G. Zhao, L. He, and J. Ren, "A novelmultimodal-sequential approach based on multi-view fea-tures for network intrusion detection," IEEE Access, vol. 7,pp. 183207–183221, 2019.

9. P. Sun, P. Liu, Q. Li et al., "DL-IDS: extracting features usingCNN-LSTM hybrid network for intrusion detection system,"Security and Communication Networks, vol. 2020, Article ID8890306, 11 pages, 2020.

10. R. M. Elbasiony, E. A. Sallam, T. E. Eltobely, andM. M. Fahmy, "A hybrid network intrusion detectionframework based on random forests and weighted k-means,"Ain Shams Engineering Journal, vol. 4, no. 4, pp. 753–762,2013.

11. W. Wang, T. Guyet, R. Quiniou, M.-O. Cordier, F. Masseglia,and X. Zhang, "Autonomic intrusion detection: adaptivelydetecting anomalies over unlabeled audit data streams incomputer networks," Knowledge-Based Systems, vol. 70,pp. 103–117, 2014.

12. H. Yao, Q. Wang, L. Wang, P. Zhang, M. Li, and Y. Liu, "Anintrusion detection framework based on hybrid multi-leveldata mining," International Journal of Parallel Programming,vol. 47, no. 4, pp. 740–758, 2019.

13. H. Zhang, Y. Li, Z. Lv, A. K. Sangaiah, and T. Huang, "A real-time and ubiquitous network attack detection based on deepbelief network and support vector machine," IEEE/CAAJournal of Automatica Sinica, vol. 7, no. 3, pp. 790–799, 2020.

14. E. Min, J. Long, Q. Liu et al., "Su-IDS: a semi-supervised andunsupervised framework for network intrusion detection," inProceedings of the International Conference on Cloud Com-puting and Security, pp. 322–334, Springer, Cham, Switzer-land, 2018.

15. M. R. Karbir, R. Onik, and T. Samad, "A network intrusiondetection framework based on bayesian network usingwrapper approach," International Journal of Computer Ap-plications, vol. 166, no. 4, pp. 975–8887, 2017.

16. J. Gu, L. Wang, H. Wang, and S. Wang, "A novel approach tointrusion detection using SVM ensemble with feature aug-mentation," Computers & Security, vol. 86, pp. 53–62, 2019.

17. Ahmim, L. Maglaras, M. A. Ferrag et al., "A novel hier-archical intrusion detection system based on decision tree andrules-based models," in 15th International Conference onDistributed Computing in Sensor Systems (DCOSS), pp. 228–223, IEEE, Santorini, Greece, 2019.

18. V. Kumar, V. Choudhary, V. Sahrawat et al., "Detectingintrusions and attacks in the network traffic using anomalybased techniques," in Proceedings of the 2020 5th

ISSN:2731-538X | E-ISSN:2731-5398
Vol. 19 No. 01 (2025)

InternationalConference on Communication and Electronics Systems(ICCES), pp. 554–560, IEEE, Coimbatore, India, 2020.

19. S. Qureshi, A. Khan, N. Shamim, and M. H. Durad,"Intrusion detection using deep sparse auto-encoder and self-taught learning," Neural Computing and Applications, vol. 32,no. 8, pp. 3135–3147, 2020.

20. S. Nathan, T. N. Ngoc, V. D. Phai et al., "A deep learningapproach to network intrusion detection," IEEE Transactionson Emerging Topics in Computational Intelligence, vol. 2, no. 1,pp. 41–50, 2018.

21. Y. Javaid, Q. Niyaz, W. Sun et al., "A deep learning ap-proach for network intrusion detection system," in Pro-ceedings of the 9th Eai International Conference on Bio-inspired Information & Communications Technologies, ICST(Institute for Computer Sciences, Social-Informatics andTelecommunications Engineering), pp. 21–26, Nairobi, Kenya,2016.

22. M. Al-Qatf, Y. Lasheng, M. Al-Habib, and K. Al-Sabahi,"Deep learning approach combining sparse autoencoder withsvm for network intrusion detection," IEEE Access, vol. 6,pp. 52843–52856, 2018