RESEARCH PAPER

Neural-Shield A Dual-Stage Hybrid Machine Learning Framework for Real-Time Network Intrusion Detection

Dr. Kavitha A[1], Dr. Janarthanan P[2]*, Dr. Prabakaran R[3], Suriya N[4], Prisha S[5], Sabarish Kumaresan[6]

[1]PG Dept. of CSE, Dwaraka Doss Govardhan Doss Vaishnav College,Chennai-600 106, India
[2,4,5,6]Dept. of CSE, Sri Venkateswara College of Engineering, Sriperumbudur, Pennalur – 602117.
[3]Dept. of CSE , Kingston Engineering college, Vellore – 632059.
Email: kavithaa@dgvaishnavcollege.edu.in[1], janap@svce.ac.in[2], hodaids@kingston.ac.in[3], suriyan@svce.ac.in[4]*
pri003sha@gmail.com[5], sabarishkv2005@gmail.com[6]
Corresponding author
Dr. P. Janarthanan
Dept. of CSE, Sri Venkateswara College of Engineering, Sriperumbudur, Pennalur – 602117.
Email: janap@svce.ac.in

ABSTRACT
with cyber attacks becoming more and more sophisticated, upgrading the current NIDS based on signatures to be able to detect innovative and enciphered intrusions is a pressing need. Current approaches face severe deficiencies in achieving low latency and high accuracy when dealing with imbalanced classes of traffic. NEURAL-SHIELD is a brand new hybrid machine learning network intrusion detection system intended for real-time intrusion identification. The main layer is represented by Random Forest, while gradient-boosting ensemble is used in the secondary layer to resolve cases that require discriminative precision. This extraction procedure relies on a sliding window approach in order to extract patterns from the traffic stream without performing deep packet inspection. Class imbalance problem is solved through the implementation of SMOTE technique meaning Synthetic Minority Over-sampling Technique. Performance of NEURAL-SHIELD was tested by NSL-KDD data set which is known as a good benchmark. Results of the experiment show that NEURAL-SHIELD has the following indicators: 99.65% accuracy, 99.6% precision and 99.5% F1 measure. It outperforms other systems by detecting U2R and R2L attacks better than any other solution.

Keywords - Network Intrusion Detection, Hybrid Machine Learning, Cyber security, Zero-Day Detection, NSL-KDD..
How to cite this article Moumita Kundu , Pintu Kumar De ,Tapan Kumar ChatterjeeSeparation of the Active Ingredient Andrographolide from the Plant Andrographis paniculata and Evaluation of Its Immunomodulatory EffectsInt J Drug Deliv Technol. 2026;16(8): 146-155; DOI: 10.25258/ijddt.16.8.13
Source of support: Nil. ,
Conflict of interest: None

1.INTRODUCTION

With fast development of internet-infrastructure-related infrastructures, the area of potential attacks has increased drastically. It has become necessary to secure today's networks which must support large amounts of data with constantly changing behaviour characteristics. Traditional NIDSs using rule based, signature matching, and DPI techniques are intrinsically based on knowledge about previous attacks and their signatures as well as direct access to packet payloads. Although these approaches can work reasonably well in cases of known threats, their capabilities in detecting new forms of attacks such as zero-day and polymorphic attacks are quite limited [1], [2].

With the advent of transport layer encryption protocols like TLS and HTTPS, packet-based methods have become relatively less useful owing to the fact that the packets are usually encrypted, thereby hiding their content from inspection. Behavioral analysis, based on metadata information gathered through monitoring, has thus become a fundamental approach for detecting intrusions [3]. Incorporation of various types of machine learning in intrusion detection poses an obvious dilemma between accuracy and efficiency, where accurate algorithms demand significant computation costs, making them impractical in a real-time environment. On the other hand, while lightweight algorithms are computationally efficient, they are discriminatory in nature [4], [5].

One major issue facing intrusion detection systems arises from the highly skewed distribution of data sets found both in the real world and within commonly used benchmarks. Specifically, key attack classes like U2R and R2L represent less than 1% of the samples in common benchmark data sets, while normal traffic dominates the vast majority of network flows [6]. This skewing results in a bias in classifier design, which achieves very high overall accuracy while significantly reducing recall on the important attack types. Previous studies have looked at using a multi-stage approach or an ensemble classifier approach to overcome this issue, whereby the attack detection process is broken down into sub-processes, yet

**Author for Correspondence:janap@svce.ac.in*

current approaches suffer from lacking an effective confidence-based routing system, relying heavily on deep learning algorithms, or failing to generalize across multiple attacks [7], [8].

The emergence of distributed computing approaches such as cloud computing, Internet of Things (IoT), and Software-Defined Networking (SDN) adds to difficulty in intrusion detection due to their creation of heterogeneous and voluminous traffic that cannot be efficiently analyzed using simple single-step classification approaches [9], [10].

2.LITERATURE REVIEW

NIDS has evolved in terms of shifting focus away from traditional rule-based techniques of intrusion detection towards the adoption of data-driven techniques that can detect attacks patterns that had gone unnoticed before. The evolution of the first NIDS focused on the use of signature detection that detected known attacks successfully

The Haider et al. study evaluated the use of various machine learning and deep learning models for network intrusion detection on the dateset. The results revealed that hybrid learning methods yielded better detection results than the methods with a single learning model; but the suggested method showed high computational complexity, limiting its usability in delay-sensitive scenarios [11]. The Shenbagalakshmi et al. study introduced a new IDS scheme, combining feature selection and algorithm optimization to enhance the speed of classifications. Even though some improvements were obtained in terms of the efficiency of the classification process, certain weaknesses emerged in detecting the less common attacks, such as U2R and R2L attacks due to poor discriminative power of the features selected based on few training samples [12].

In their study, Christy et al. came up with a multiple-stage intrusion detection technique aimed at improving the detection ability in the cloud-aided vehicular ad hoc network. This was achieved through selecting specific featutes and using ensemble learning techniques. This model showed improved results in the process of classification in the distributed network setting [13]. Alashjaee et al. proposed a deep learning structure that utilizes the methods of convolutional features learning, sequential features, and weighted attention models. This helped improve multi-class intrusion classification performance [14].

An ensemble architecture that combined the use of attention mechanisms with dilated convolution was suggested by Ashraf et al. to enhance intrusion detection capability in face of evasion attacks. This architecture exhibited good intrusion detection ability due to its capability to learn from traffic features, but the problem with its great complexity in computation making the architecture unsuitable for implementation in real-time [15]. Goldschmidt et al. conducted an analysis of the existing benchmark datasets employed for intrusion detection evaluation and highlighted some structural weaknesses such as time redundancy, high class imbalance, and lack of coverage of new attacks types [16].

Naveen et al. investigated the use of machine learning algorithms in optimizing intrusion detection systems in IoT networks. Their investigation revealed that ensemble machine learning methods and techniques for optimization of machine learning algorithms greatly enhanced the efficiency of the detection process in diverse network setups [17]. Patel et al. conducted an extensive comparative evaluation of machine learning algorithms used for intrusion detection in network systems and found that the efficacy of various algorithms depended greatly on their environment [18].

Ahmed et al. studied machine learning-based intrusion detection techniques for IoT devices having limited computing capabilities. A number of lightweight algorithms were considered in their study and found to be effective in producing good results under the limited resource environment [19]. Alomari et al. introduced a lightweight machine learning-based intrusion detection system particularly suitable for IoTs, where it was proved that well-designed lightweight models performed similarly to heavyweight counterparts despite stringent resource limitations [20].

Kumar et al. have studied combinations of multi-classification algorithms in order to enhance intrusion detection precision, proving that algorithm complementarily helped obtain improvements in detection precision as opposed to when using individual algorithms [21]. NIDS systems, which were evaluated on multiple datasets, were analyzed by Alzahrani et al. It turned out that generalization capabilities of NIDS could be better assessed when evaluated on multiple data sets rather than just one [22]. An ensemble learning model for detecting intrusions in networks was proposed by Mazini et al., who proved that more diverse individual algorithms within ensembles led to better results and reduced false positives [23].

Synthetic Minority Over-sampling Technique (SMOTE), which is an approach used to synthesize minority class instances via interpolation of existing minority class instances, was introduced by Chawla et al. as an attempt to solve class imbalance problem that occurs in various machine learning classification problems [24]. The theory of Random Forest algorithms was introduced by Breiman et al., who provided proofs on how randomization of data samples and features leads to development of highly robust classifiers with low variance [25].

3.PROPOSED METHODOLOGY

The NEURAL-SHIELD architecture adopts a two-stage hybrid architecture in order to establish a balanced trade-off between performance and efficiency. The incoming data from the network is subjected to a pipelined processing involving feature extraction, probability computation, and selected refinement.

The probabilistic filter operates on most of the traffic with low computational cost. The output values from the ensemble-based classifier control the route selection process: data that meets the threshold criterion are immediately labelled and directed without any additional

processing, whereas the ones which fail to satisfy the criterion are subjected to further analysis, as depicted in

$$f(x) = \begin{cases} f_{RF}, & \text{if } \max(P_{RF}(Y|X)) \geq \tau \\ f_{XGB}, & \text{otherwise} \end{cases} \quad (1)$$

where τ represents the confidence threshold, f_{RF} denotes the Random Forest classifier, and f_{XGB} represents the XGBoost model.

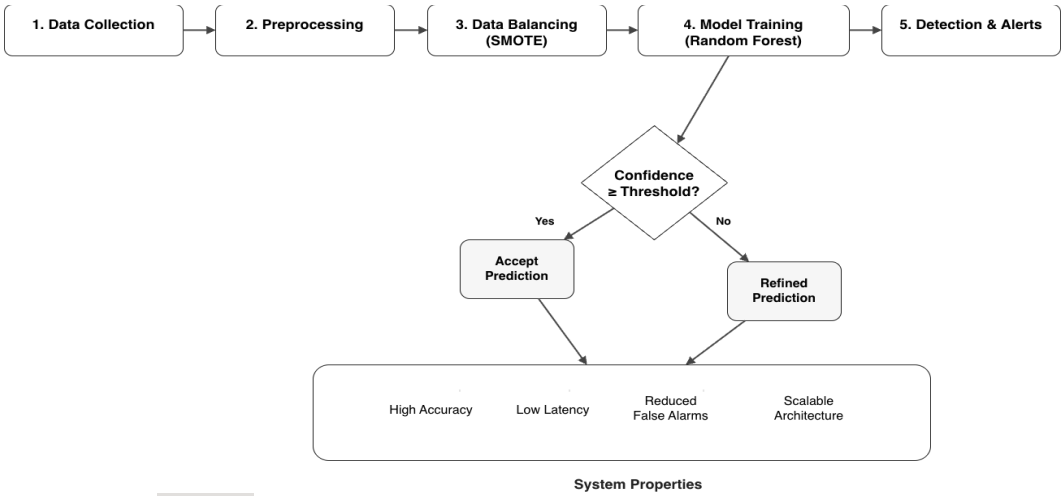


Figure 1.**Proposed Hybrid Neural Shield Model Architecture**

4. FEATURE ENGINEERING

Intrinsic characteristics are composed of protocol type, connection time, service type, flag setting, and number of bytes transferred per connection. In terms of temporal behavior, it is determined using a sliding window that applies to the input data stream, thereby allowing for calculations related to time-sensitive measures like connection counts, SYN ratio, same service connections, and different service connections. The temporal characteristics are calculated using a sliding window W=[t-Δ,t], where Δ refers to the time period. Pwrepresents the packets in the window. The characteristics are given as follows:

$$\text{count} = | P_w | \quad (2)$$

The connection error rate within the window is computed as:

$$\text{serror_rate} = \frac{1}{|P_w|} \sum_{i=1}^{|P_w|} I(\text{flag}_i \in \text{error states}) \quad (3)$$

Table 1. Packet Payload Feature Description

Feature Name	Description	Relevance
Duration	Length of connection	0 – 58329 sec
Protocol	Type of protocol	TCP, UDP, ICMP
Service	Network service on destination	HTTP, SSH, DNS
Flag	Status flag of the connection	SF, S0, REJ, RSTO, SH
Src Bytes	Bytes sent from source todestination	0 – 1,379,963,888 bytes
Dst Bytes	Bytes sent from destination to source	0 – 1,309,937,401 bytes
Count	Number of connections in time window	0 – 511 connections
Serror Rate	Percentage of SYN/REJ errors	0.00 – 1.00
Same_srv_rate	% of connections to same service	0.00 – 1.00
Diff_srv_rate	% of connections to different services	0.00 – 1.00

Table 1 shows of all the features used for classification, including features based on connectivity as well as temporal traffic behaviors, generated using the sliding window technique. The relevancy column presents the ranges and categorical types of values for each feature, which indicate the range of traffic characteristics present in the NSL-KDD dataset. In Figure 2, the importance of each feature selected is provided; temporal and byte volume features are usually the most relevant features. These results prove that the behavior of traffic is more important than connectivity traffic

.

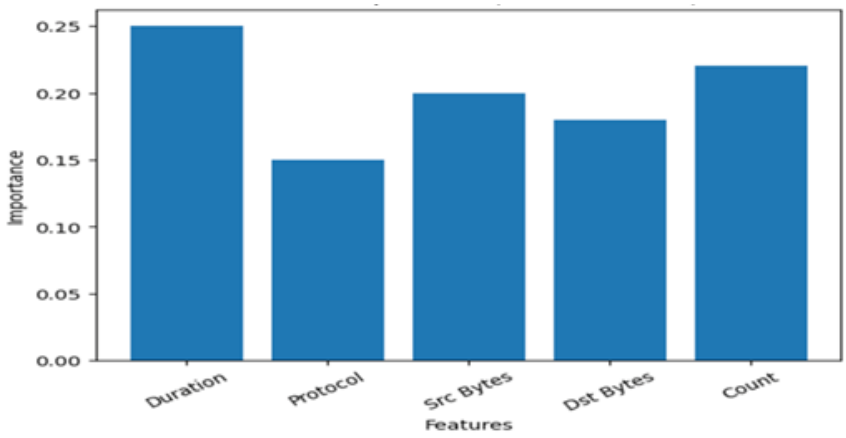


Figure 2. **Feature Importance Analysis**

4.1Data Preprocessing

It is important to pre-process network traffic data before developing an efficient IDS system. Pre-processing involves dealing with categorical attributes, missing data, and redundancy that can affect the performance of the classifier if not addressed properly. The asymmetry in the data distribution between the normal and attack classes results in bias, thus reducing the ability of the model to detect important minority attacks.

To address class imbalance, the Synthetic Minority Over-sampling Technique (SMOTE) is applied. For a minority instance x_i, a synthetic sample is generated as:

$$x_{syn} = x_i + \lambda(x_{knn} - x_i),\ \lambda \in [0,1] \quad (4)$$

4.2.PROBABILISTIC PACKET PAYLOAD CLASSIFICATION

All network packets coming into the system have features including the type of protocol, the connection time, byte quantities, and whether or not flags are set. All of these pieces of information are combined into one feature vector, which is then fed into a probabilistic ensemble classifier. This particular classifier consists of several decision trees, which are all trained using a different sample of the input traffic. At each point within the decision tree, all available features are assessed and the best split of the input traffic into traffic classes occurs. Because the decision trees use different samples of the input traffic and a different set of random features at each split, all traffic types are identified rather than just one.

$$G = 1 - \sum_{k=1}^{C} p_k^2 \quad (5)$$

Where $p_k$denotes the probability of class kat a given node. The final prediction is obtained through majority voting across all constituent trees:

$$P(Y \mid X) = \frac{1}{N}\sum_{i=1}^{N} I\left(h_i(X)\right) \quad (6)$$

The results obtained from majority voting through the network become probability estimates used as the confidence criterion for the escalation action. Flows where the maximum probability of their class surpasses the threshold τ can be classified instantly, whereas flows whose maximum probability is less than τ are passed to the secondary classification stage for processing. In order to increase the range of the feature space covered by the network, a larger maximum tree depth is used.

4.3 CONFIDENCE-DRIVEN SECONDARY CLASSIFICATION

The network flow instances which do not meet the required confidence level while performing the primary classification task is sent to an ensemble classifier. The secondary classifier is constructed with the purpose of dealing with complex cases involving traffic instances that have similar feature distribution among different classes of attacks. To avoid the problem of overfitting and maintain generalization capability, regularization terms are included in the objective function. This becomes especially important when considering the minority class of attacks like U2R and R2L having small sample sizes. The objective function is given as:

$$L = \sum_{i=1}^{n} l(y_i, \hat{y_i}) + \Omega(f) \quad (7)$$

Where l the loss is function and $\Omega(f)$is the regularization term:

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda\sum w_j^2 \quad (8)$$

The confidence-based routing mechanism is formally expressed as:

$$\text{confidence} = \max(P_{RF}(Y \mid X)) \quad (9)$$

The confidence score is a primary controlling factor governing the balance between computational efficiency and classification accuracy. Figure 3 illustrates the distribution of network flows across the two classification layers under varying confidence threshold.Theelevated threshold restricts immediate classification to instances of high confidence, reducing errors in the primary layer at the cost of increasing the volume of traffic directed toward the secondary layer

*Author for Correspondence:amrita.bharti@sharda.ac.in

.

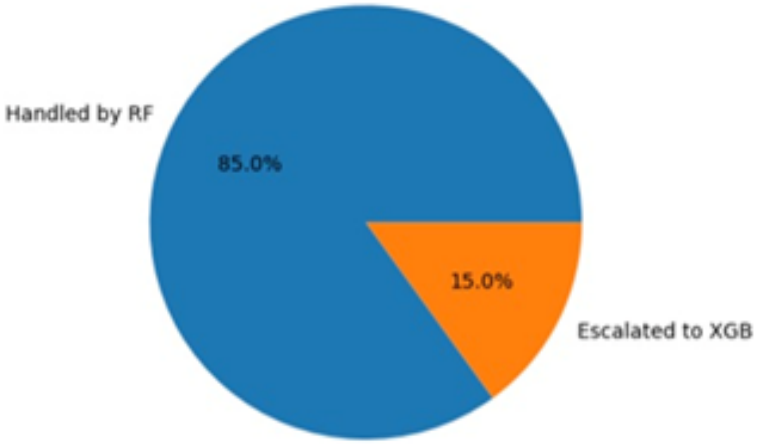


Figure 3. **Traffic Distribution across Dual-Stage Classification**

PROCESSING MODEL

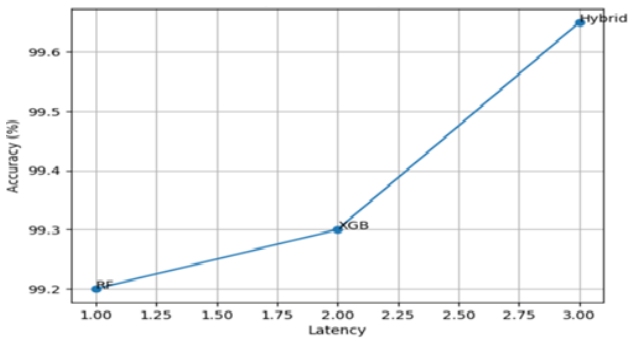
Network packet captures are performed all the time and their features are extracted into structured feature vectors that are immediately fed to the main probabilistic classifier. Network packets that get labeled with certainty are labeled with a decision without being processed any further, but those labeled ambiguously are processed by the gradient boosted classifier. Packet capture and classification occur separately via an asynchronous pipeline in such a way that even if there is a high volume of packets to be processed, data capture occurs seamlessly without any loss. Each stage of the pipeline is designed to optimize and reduce latency on a per-flow basis. The extracted feature vector is directly input into the Random Forest classifier that implements probabilistic classification of the flow using a parallelized and cheap method of inference. Those flows escalated above this point are input into the gradient-boosted classifier where an expensive but detailed classification approach is used to solve ambiguous flows. The settings that were chosen for both classifiers are outlined in Table 2.

Table 2. Model Configuration Parameters

Parameter	RandomForest	XGBoost
Number of Trees	100	200
Max Depth	25	6
Learning Rate	–	0.05
Subsample	–	0.8
Feature Sampling	sqrt(n)	0.8
Criterion	Gini	Log Loss

Latency vs. accuracy tradeoff for different intrusion detection frameworks is shown in Figure 4 below. The ensemble classifier provides low latency at the expense of lower accuracy of the rare attacks, while complex models deliver higher accuracy by introducing higher latency. Hybrid framework keeps the same performance levels in both parameters

.



*Author for Correspondence:amrita.bharti@sharda.ac.in

Figure 4. **Latency vs. Accuracy Trade-off in Intrusion Detection Models**

Figure 5 demonstrates a graphical illustration of the differences in configurations through various layers, showing the extent to which the parameter values are optimized to meet
the requirements of the respective classification layer. The configurations of parameters influence the effectiveness and accuracy associated with the framework
.

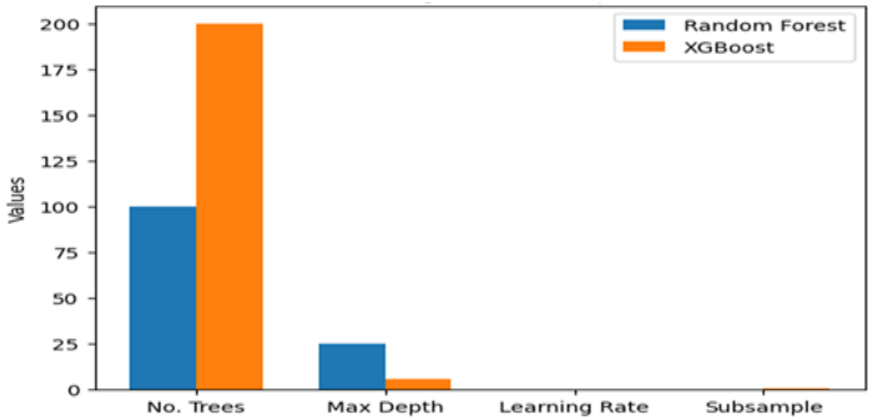


Figure 5. Model Configuration Comparison

5. EXPERIMENTAL RESULTS

Imbalance of classes' distribution is one of the characteristics inherent in the NSL-KDD dataset. It refers to the distribution of classes among the data in the dataset that reflects the distribution of the network traffic data. Namely, normal traffic represents 53%, DoS attacks – 36%, and probe attacks – 9% of the dataset. Meanwhile, R2L and U2R attacks comprise 2%. The class distribution imbalance in the dataset can be addressed through the application of the SMOTE over-sampling technique. This technique improves the class distribution balance because of the formation of new objects related to the under-represented classes. Table 3 shows the class distribution in the NSL-KDD dataset, which highlights how imbalanced the distribution is among the different types of traffic involved in the modelling process. It should be noted that normal traffic and DOS attacks make up about 89 percent of the traffic, while R2L and U2R are the underrepresented categories

Table 3. Dataset ClassDistribution

Class	Description	Percentage (%)
Normal	Benign traffic	53%
DoS	Denial of Service	36%
Probe	Surveillance attacks	9%
R2L	Remote to Local	1%
U2R	User to Root	<1%

Figure 6 shows the distribution of the NSL-KDD data set, where normal traffic and Denial of Service (DoS) attacks constitute the dominant proportion of instances, while R2L and U2R categories account for less than 1% of total records. The distributional asymmetry induces classifier bias toward majority classes, degrading detection sensitivity for critical attack types
.

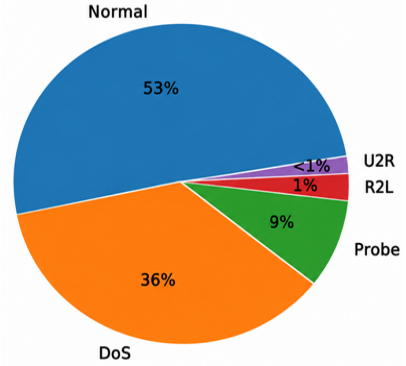


Figure 6. **Class Distribution of NSL-KDD Dataset**

The algorithm of an ensemble classifier is able to perform efficient classification of most traffic with an efficiency of above 99%. Secondary Classification produces a high-quality classification as it has the ability to identify nonlinear relationships among the features through the use of the boosting technique. Together, the two algorithms contribute significantly to the attainment of high levels of classification

.

Table 4. Comparison of Proposed NEURAL-SHIELD with State of the Art Models

Model	Accuracy	Precision	Recall	F1-Score
Haider et al. [7]	98.72%	98.4%	98.1%	98.2%
Shenbagalakshmi et al. [8]	99.01%	98.9%	98.5%	98.7%
Christy et al. [9]	98.94%	98.8%	98.6%	98.7%
Proposed NEURAL-SHIELD	99.65%	99.6%	99.4%	99.5%

From Table 4, it is evident that the NEURAL-SHIELD hybrid model performs better when compared to Random Forest and XGBoost. The NEURAL-SHIELD hybrid model incorporates the attributes from both of the models to ensure maximum effectiveness by precision and F1-Score, which has been achieved at 99.65% accuracy and 99.5% F1-Score, respectively.

Figure 7 gives a pictorial representation of the improvements in terms of the different metrics discussed above. The improvements made are significantly more noticeable for minority attacks because of the efficacy of the proposed confidence-based approach in dealing with ambiguities and rare attacks in traffic

.

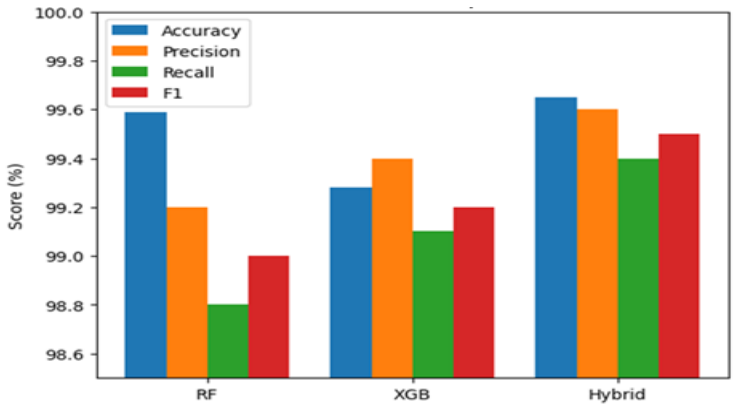


Figure 7.**Evaluation of Proposed NEURAL-SHIELD against Existing IDS Approaches**

Figure 8 presents the confusion matrix, providing a detailed breakdown of classification outcomes across all traffic categories. High true-positive rates are observed for both majority and minority attack classes, with substantially reduced misclassification counts, particularly for U2R and R2L categories that are most susceptible to detection failure in conventional single-stage classifiers

.

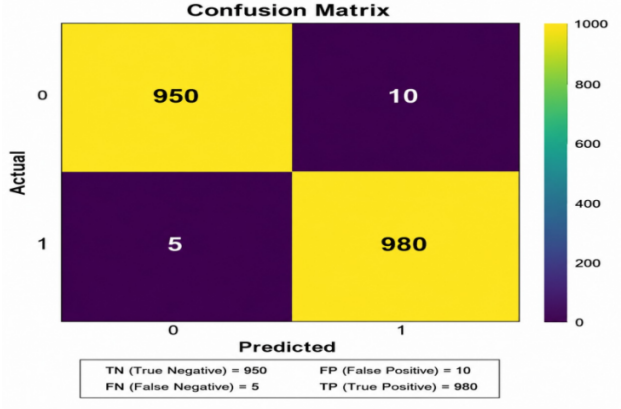


Figure**8. Confusion Matrix of Proposed Neural Shield Model**

*Author for Correspondence:amrita.bharti@sharda.ac.in

The single classifier ensemble shows higher false positives and false negatives for attack classes belonging to the minority groups, while the independent gradient-boosted classifier solves the problem of false negatives using extra computing costs. It is observed in Figure 9 that the Precision-Recall graph depicts the high precision of NEURAL-SHIELD at different recall values, keeping its area under the curve much higher for less frequent classes such as U2R and R2L attacks. The use of the dual-layer structure supported by confidence-based reasoning ensures the minimization of both types of errors, proving the efficiency of the SMOTE technique and subsequent classification process
.

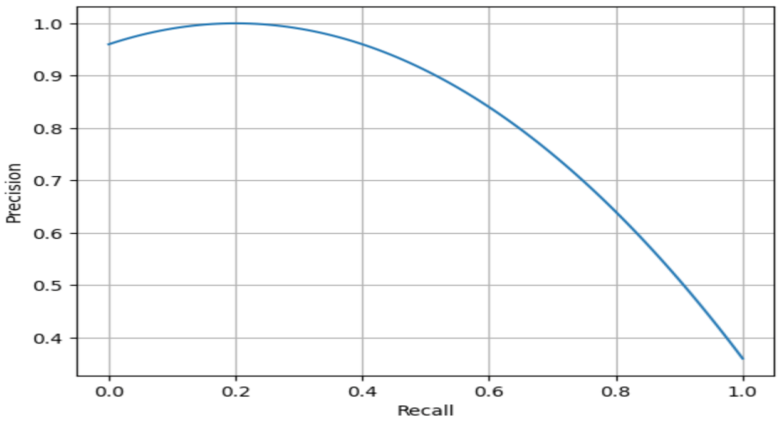


Figure **9. Precision-Recall Curve demonstrating** performance on imbalanced dataset

The probabilistic model shows modest false positives and high false negatives for minority attacks, whereas the gradient-boosted model decreases the latter with the former being present as well. The hybrid model NEURAL-SHIELD achieves the best performance in both measures, thus illustrating that confidence-based promotion helps to mitigate the drawbacks of separate models. Table 5 provides a summary of the false positive and false negative rates obtained for different model configurations tested, demonstrating that the proposed dual stage approach is always able to deliver the lowest error rates. While the probabilistic ensemble alone shows higher false negatives for minority classes, the gradient boosted classifier alone decreases false negatives but increases false positives

Table 5. Error Metric Comparison Between Existing Models and Proposed Neural Shield Framework

Model	False Positives	False Negatives
Random Forest	Moderate	High (U2R/R2L)
XGBoost	Low	Moderate
Proposed Hybrid	Very Low	Very Low

Figure 10 shows the graph plotting the values for all the tested setups. In both measures of performance, the hybrid approach has the lowest number of errors, showing that confidence-based escalation addresses the shortcomings of each classifier component
.

*Author for Correspondence:amrita.bharti@sharda.ac.in

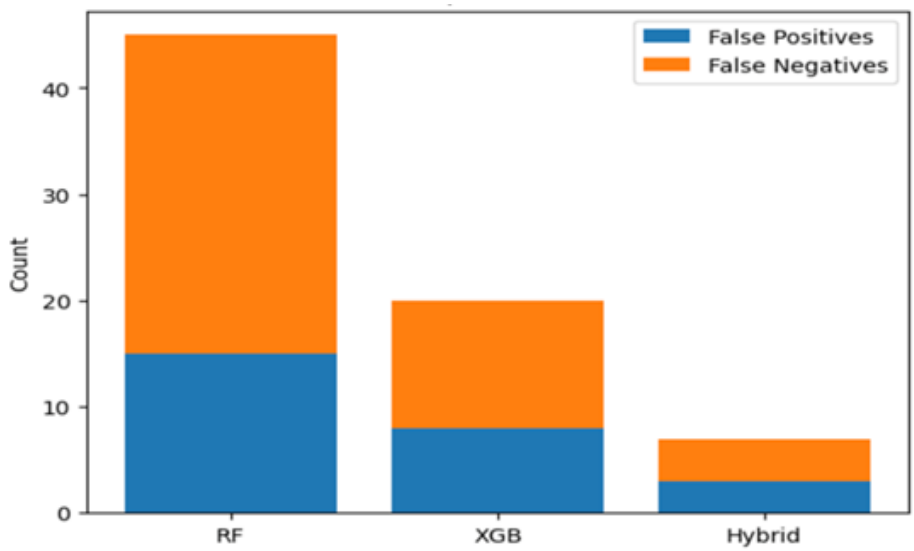


Figure**10. False Positive and False Negative Comparison Across Models**

The throughput tests conducted in a number of extended periods reveal the possibility for the system to ensure stable rates of packet processing under traffic conditions. The asynchronous pipeline-based design makes it possible to perform all processes involved – packet capture, feature extraction, and classification – simultaneously, thereby preventing any
possible bottlenecks during the high traffic peak. In general, the throughput test proves the full capability of NEURAL-SHIELD to achieve real-time operation both in tests and in actual environments.

6. CONCLUSION AND FUTURE WORK

In conclusion, present research proposes the development of a novel two-stage hybrid machine learning model named NEURAL-SHIELD aimed at solving problems associated with the detection of rare attacks as well as preserving effective performance rates in real time. The primary contradiction inherent in today's NIDS lies in the requirement to ensure an efficient performance rate combined with low classification speed while analyzing rare cases using a simple classifier and high accuracy incompatible with fast classification speed at the same time.This issue is overcome through the confidence-driven classification approach with the help of Random Forest as an initial classifier and second-stage analysis with using XGBoost for low-confidence results. As NEURAL-SHIELD operates solely with feature vectors that represent metadata and behavioral data but not packets, no privacy risks arise in the course of classification. Use of SMOTE algorithm assists in overcoming the class imbalance problem inherent in NSL-KDD datasets. From experimental results, the accuracy of NEURAL-SHIELD was found to be 99.65% and its F1 score 99.5%, which were better compared to those obtained by the use of separate models. Possible areas of further work on NEURAL-SHIELD can be assessing its performance in relation to other benchmarks such as CIC IDS 2017. Additionally, investigating NEURAL-SHIELD's incorporation into other high-performance packet processing systems like DPDK should also be considered.

REFERENCE

1. O. Abualghanam et al., "Hierarchical lightweight intrusion detection system using deep learning in the context of IoT," Cluster Computing, vol. 28, no. 783, pp. 1–15, 2025.
2. H. Hakami et al., "Machine learning techniques for enhanced intrusion detection in IoT security," IEEE Access, vol. 13, pp. 31140–31158, 2025.
3. S. Alzahrani et al., "Network intrusion detection system based on multiple datasets: Machine learning approaches," IEEE Access, vol. 13, pp. 2415–2432, 2025.
4. Z. Alomari et al., "Lightweight machine learning-based IDS for IoT environments," in Proc. 8th Cyber Security in Networking Conference (CSNet), France, pp. 33–37, 2024.
5. R. Kumar et al., "Improving intrusion detection precision via multi-classification algorithm examination," in Proc. IEEE International Conference on Computing and Communication Technologies, India, pp. 112–117, 2024.
6. A. Sharma et al., "An efficient network intrusion detection system using machine learning approach with recursive feature elimination," in Proc. IEEE 11th International Conference on Reliability, Infocom Technologies and Optimization, India, pp. 450–455, 2024.
7. M. Mazini et al., "A network intrusion detection system using ensemble machine learning," IEEE Transactions on Network and Service Management, vol. 19, no. 4, pp. 4821–4833, 2022.
8. K. Patel et al., "An in-depth examination of machine learning approaches in NIDS: A comprehensive review," IEEE Access, vol. 8, pp. 115380–115392, 2020.
9. T. Ahmed et al., "A comprehensive review: Intrusion detection system using machine learning in Internet of Things," in Proc. IEEE International Conference on Advanced Computing

Technologies, India, pp. 214–220, 2025
10. M. Naveen et al., "Optimized IoT intrusion detection using machine learning technique," in Proc. IEEE 3rd International Conference on AI in Engineering and Technology, Malaysia, pp. 1–6, 2025.
11. S. Haider et al., "NIDS: Network intrusion detection system based on deep and machine learning frameworks with CIC-IDS-2018," IEEE Access, vol. 8, pp. 181444–181463, 2020.
12. G. Shenbagalakshmi et al., "An improved framework for network intrusion detection using machine learning," Wireless Networks, vol. 31, no. 8, pp. 4877–4894, 2025.
13. C. Christy et al., "Machine learning based multi-stage intrusion detection system and feature selection ensemble security in cloud-assisted vehicular ad hoc networks," Scientific Reports, vol. 15, no. 27058, pp. 1–18, 2025.
14. A. M. Alashjaee et al., "Deep learning for network security: An Attention-CNN-LSTM model for accurate intrusion detection," Scientific Reports, vol. 15, no. 1, pp. 21856–21856, 2025.
15. M. D. Ashraf et al., "An enhanced attention and dilated convolution-based ensemble model for network intrusion detection system against adversarial evasion attacks," Scientific Reports, vol. 18, no. 191, pp. 1–17, 2025.
16. P. Goldschmidt et al., "Network intrusion datasets: A survey, limitations, and recommendations," Computers & Security, vol. 156, pp. 104510–104510, 2025.
17. M. Naveen et al., "Optimized IoT intrusion detection using machine learning technique," in Proc. IEEE 3rd International Conference on AI in Engineering and Technology, Malaysia, 2025, pp. 1–6.
18. K. Patel et al., "An in-depth examination of machine learning approaches in NIDS: A comprehensive review," IEEE Access, vol. 13, pp. 8841–8860, 2025.
19. T. Ahmed et al., "A comprehensive review: Intrusion detection system using machine learning in Internet of Things," in Proc. IEEE International Conference on Advanced Computing Technologies, India, pp. 214–220, 2025
20. Z. Alomari et al., "Lightweight machine learning-based IDS for IoT environments," in Proc. 8th Cyber Security in Networking Conference (CSNet), France, pp. 33–37, 2024
21. R. Kumar et al., "Improving intrusion detection precision via multi-classification algorithm examination," in Proc. IEEE International Conference on Computing and Communication Technologies, India, pp. 112–117, 2024.
22. S. Alzahrani et al., "Network intrusion detection system based on multiple datasets: Machine learning approaches," IEEE Access, vol. 13, pp. 2415–2432, 2025.
23. M. Mazini et al., "A network intrusion detection system using ensemble machine learning," IEEE Transactions on Network and Service Management, vol. 19, no. 4, pp. 4821–4833, 2022.
24. N. V. Chawla et al., "SMOTE: Synthetic minority over-sampling technique," Journal of Artificial Intelligence Research, vol. 16, pp. 321–357, 2002.
25. L. Breiman et al., "Random forests," Machine Learning, vol. 45, no. 1, pp. 5–32, 2001