Grenze International Journal of Engineering and Technology, January Issue

Improving the Ability of Intrusion Detection Systems by Reducing both False Positives and False Negatives

A. Hanuman Prasad[1] and Suresh Kumar Mandala[2]
[1]Research Scholar, Computer Science and Artificial Intelligence, SR University, Warangal-506371 Telangana,
Email: 2303c50166@sru.edu.in
[2]Assistant professor, Computer Science and Artificial Intelligence, SR University, Warangal-506371 Telangana,
Email: mandala.suresh83@gmail.com

***Abstract*—Today, Intrusion Detection Systems (IDS) keep networks safe by noticing and responding to risks on computers. Using the traditional approaches to detect intrusions frequently leads to a lot of wrong alarms and not picking up some actual signs of attack. Such problems wear out security analysts because they become less alert to actual threats which could lead to missing some. In our research, we introduced a straightforward way to help intrusion detection systems become more stable and improve how they work. We use state-of-the-art machine learning, carefully measured thresholds and direct analysis of goals to support the system's performance. Thanks to learning from the past and present network activity, this framework knows how to tell regular issues from real security problems. The main aim of this research is to make IDS systems more reliable by cutting down on fake alarms while still being able to spot real security violations.**

***Keywords*— Intrusion Detection Systems, False Positives, False Negatives, Machine Learning, Network Security, Anomaly Detection.**

I. INTRODUCTION

IDS use technology to find threats and keep them from causing any damage to computers or networks. Although IDS systems become more efficient, problems like identifying actual attacks or letting dangerous ones get through continue to arise. As a result, IDS is less reliable, since it could either miss real dangers or cause too many false alarms which annoys cyber security teams. This paper examines the problem by presenting a straightforward method to upgrade IDS that helps reduce these error rates and improve detection accuracy.

A. Information about IDS and cyber security

Since digital systems are now connected more than ever, we now face very different security threats. Now, those with bad intentions seek out weaknesses in computer systems to steal data, disrupt what works and take important information. For this reason, Intrusion Detection Systems (IDS) are an important element of cyber security nowadays. An IDS looks through activity on networks and systems to notice any security reasons for concern or behavior that breaks the rules. It watches over the system, sees when things change unexpectedly, monitors behaviors of users and checks for any signs that match known attacks. An Idiosyncratic Disguised Statement is what IDS represents. Despite seeming regular, it actually provides cover for other ideas or messages.

Grenze ID: 01.GIJET.12.1.30

IDS use these signatures to match how the system is active with common attack patterns to spot attacks. Thanks to these signatures, computers know how to deal with risks that have shown up earlier. After a match, the system will mark the activity as possibly harmful. This kind of detection is successful at catching threats that have occurred before. Still, it has a shortcoming because it can't detect unprecedented attacks from hackers [1].
Anomaly-Based Intrusion Detection Systems rely on noticing out-of-the-way behavior on networks or systems and warn about it. Such systems notice any unexpected activity and send details to security professionals for further examination. It is better able to discover more kinds of threats since it examines a broader range of potential hazards. Yet, it can see certain common activities as threating, leading it to wrongly mark things as harmful viruses.
Hybrid IDS use two methods – signatures and anomalies – so they can detect a wider range of threats and both old and new problems. With the precision of signature-based approaches and the discovery skills of anomaly detection, hybrid systems are counted on to resolve some of the challenges linked to using just one approach. As a result, the system identifies more issues, defends against a wider range of attacks and strikes a nice balance between accuracy and how much it audits. IDS contribute to cyber security by finding signals of possible attacks early, so teams can handle them more promptly. Even so, there are constant challenges to ensure the technology stays correct and still useful. For this reason, perfecting IDS is needed, mainly to decrease both false alarms and missed risks, making certain security systems remain effective at all times.

B. Process of false positives and false negatives.

In the evaluation of Intrusion Detection Systems (IDS), there are two types of mistakes that really impact how well these systems work and how effective they are: false positives and false negatives.
False Positives (FP): A false positive happens when something meant to be harmless gets labeled as a threat. While these errors are not harmful, they can make the security team get lots of alerts, which can lead to them missing important issues. False Negatives (FN): Conversely, the problem called a false negative happens when the security system mistakes something harmful for something normal and lets it go unnoticed. This type of error can make it easier for serious threats like malware to gain access to a system and steal or damage data, and it doesn't always alert you to the problem, so it can lead to bigger security problems [2].
However, when you get better at one skill, it usually gets harder to get better at the other. Therefore, this paper proposes an improved method that tries to balance both types of errors by using adaptive thresholds and combining different models to try to lower both kinds of mistakes at the same time.

C. Problem and motivation for the study.

Despite a lot of progress in creating systems that spot security breaches, modern IDS tools still have trouble being accurate and reliable, especially when dealing with new and smart cyber-attacks. One of the biggest problems is that security systems sometimes give too many false alarms, or miss real attack attempts. These issues can cause problems in how well the system works and slow down the day-to-day operations. Security analysts are often swamped by lots of false alarms, which mean that serious threats can slip by and cause problems before anyone notices.
The main reasons behind these problems usually come from fixed detection limits, limited ability to understand context, not enough good data, and not being able to adjust to new kinds of attacks. Traditional signature-based systems can often only find known problems, while solely using anomaly-based methods can trigger a lot of false alarms. Even the best machine learning-based IDS can have trouble keeping up because they haven't been designed to work in ever-changing and mixed network settings.
This study was inspired by the need to build a smart IDS system that accurately finds threats while at the same time keeping the number of false alarms and missed alarms as low as possible, especially when used in real-world situations. By using a mix of different detection models, making smart guesses about what's important, and coming up with better features, the idea is to build a system that helps make cyber security teams' jobs easier, reduces the number of false alarms, and makes it faster for them to find real threats[3].

D. Objectives

- To help cut down the number of false alarms in fraud detection systems, while still being able to spot real and serious problems.
- To help make sure that any malicious activity is caught, we set things up so that the model is more likely to identify things as malicious, even if there might be a few false positives.
- To combine what the user does with what's happening around them to better tell the difference between regular and harmful activity.

- To test how the improved IDS system does compared to the standard benchmarks and see how it fares against other systems already out there.

II. Literature Review

Intrusion Detection Systems (IDS) help by figuring out when someone or something tries to access or do something on a network that shouldn't be there. While IDS technology has gotten better over the years, problems like false alarms (incorrectly flagging something safe as bad) and missed threats (not catching harmful activity) still cause big headaches. This section summarizes the ideas and techniques used by previous studies to help improve how well IDS works.

Intrusion Detection Systems (IDS) have evolved significantly, yet false positives and false negatives remain persistent challenges. Traditional signature-based IDS (e.g., Snort) are highly accurate for known threats but ineffective against novel attacks [4]. In contrast, anomaly-based systems detect unknown patterns but often suffer from high false alarm rates [5].

Recent works have explored machine learning (ML) and deep learning (DL) techniques to improve IDS performance. For example, Shone et al. [6] introduced a deep learning model using non-symmetric deep autoencoders, demonstrating effectiveness in anomaly detection. Similarly, Yin et al. [7] leveraged LSTM networks to capture temporal dependencies in network traffic for improved classification accuracy. However, these models are often static and do not adapt well to concept drift in dynamic environments.

To address these limitations, hybrid systems have been proposed. Dhanabal and Shantharajah [8] emphasized the importance of combining supervised and unsupervised techniques for better accuracy. More recent studies by Ibitoye et al. [9] and Ahmed et al. [10] integrated deep learning with ensemble methods (e.g., Random Forest and XGBoost) to reduce misclassification rates, yet their models lack adaptability to contextual or feedback-driven thresholding.

Adaptive mechanisms are gaining traction. In [11], the authors proposed threshold tuning based on operational metrics, showing improved trust in IDS outputs. Furthermore, behavior-based feature engineering has shown promise in enhancing detection of insider threats and low-volume attacks [12]. The use of context-aware adaptive IDS is further elaborated in the work of Almiani et al. [13], where environmental variables like access time and device profile were used to fine-tune alert generation.

This study builds upon these advancements by integrating a hybrid detection model with adaptive thresholding, context modeling, and ensemble prediction, thereby reducing both FPR and FNR more effectively than existing methods.

A. Traditional IDS Approaches

Intrusion Detection Systems (IDS) are important parts of cyber security and usually fall into two main types: ones that compare activity to set rules, and ones that look for unusual patterns in the system. Every type of security has its advantages and faults in seeing threats and quickly adapting to them. This technique works extremely well, as seen in Snort which accurately spots known risks [14]. When a certain cyber threat appears repeatedly, these systems usually respond well.

Yet, they can’t detect new types of attacks like zero-day attacks or very similar ones that didn’t match what they already have in their signature database. Because of this, their systems must be always updated and they usually miss picking up on new types of risks.IDS systems that depend on anomalies work by observing how normal behavior looks on a computer or network and immediately detect things that seem suspicious as potential threats. Because unusual behavior online is hard to predict, security systems that detect it often receive too many wrong alarms which can prevent staff from keeping track of real threats. Recently, the use of machine learning and deep learning tools in signature-based and anomaly based systems has helped them do their job better at spotting attacks. A few models have combined these approaches so they can use the positive features of each and minimize its negatives [3].

B. Machine Learning-Based Intrusion Detection Systems

Machine Learning (ML) techniques are now a key part of modern Intrusion Detection Systems (IDS), helping to catch and stop more cyber-attacks. Learning methods in IDS are grouped as supervised, unsupervised and semi-supervised and each functions differently. If the machine sees examples of prior attacks and understands what to look for, it will be able to recognize new attacks it hasn’t experienced. Support Vector Machines, Decision Trees and Random Forests are ideal for finding various kinds of network intrusions. For these approaches to detect new risks, they mainly rely on being fed labeled data [15].

In Unsupervised Learning, data that lacks labels is used by the computer to spot anything strange compared to the standard pattern in the set. A useful procedure for making sense of many data points is by applying both K-Means clustering and Principal Component Analysis. IDS is good at alerting us to new security issues, but it can mistakenly detect them because things on networks move so rapidly.A small amount of supervised data, along with a larger amount of unsupervised data, is the main connection in Semi-Supervised Learning. With the use of this technique, the AI classifies pictures by itself, so users don't need much training and spending time is reduced. Semi-supervised model reliability mostly depends on whether the labeled data is solid and what the specific situation requires. Lately, experts have experimented with using these approaches as a group to boost IDS performance. Sometimes, researchers say it can be better to train models on labels as well as those that do not have labels, since they each support the task in their own way. They believe using this method will lead to detecting more cases and lowering the chance of pointing to the wrong ones [16].

C. Deep Learning in Intrusion Detection Systems

Deep Learning (DL) techniques have come to play an important role in creating Intrusion Detection Systems, helping them better identify new and changing types of cyber threats. DL models can automatically analyze data samples to find different kinds of information and as a result, they are very skilled at seeing the smallest instances of criminal or malicious behavior. Looking at traffic, it should be easy to see that RNNs and LSTMs are good at spotting data patterns depending on their order. These models spot when things happen the same way, suggesting an attack has been attempted. Autoencoders such as Deep Autoencoders and Variational Autoencoders, are models used in machine learning that try to discover how to change one set of data into a copy of it. Using examples of normal network activity, they can compare the fake activity with what's recognized as regular. Traditional methods for identifying cyber threats generate too many false alarms compared to autoencoders, research has shown[17]. Deep Belief Networks which consist of multiple Restricted Boltzmann Machines are also studied for identifying security risks. DBNs know how to understand the relationships between different kinds of data and have been successful at identifying network security threats [18].

Although they have many benefits, DL models generally require a lot of labeled data for training and because of their processing demands they are seldom used in the real world. To address these difficulties, researchers have designed light models and used previous knowledge to help the models without needing to learn from as much information.

D. Efforts to Minimize False Alarms in Intrusion Detection Systems

Good Intrusion Detection Systems still need to keep down the number of false positives and false negatives. Lately, there has been an effort to both detect phone scams and prevent struggling with unneeded alerts. Better IDS performance comes from carefully choosing the best set of features for it to analyze. To find the main features, Information Gain, Recursive Feature Elimination and methods that combine filter and wrapper methods have all been used. As an example, the IGRF-RFE method applies Information Gain and Random Forest Importance along with Recursive Feature Elimination, to help find important features and reduce the features being used.

When several different machine learning methods are combined, IDS is capable of working better. A variety of models are combined by ensemble methods in the hope of getting a more reliable answer. Researchers brought together both bagging and GBDT models which discovered that this combination works better and avoids more mistakes[22]. Changing thresholds in response to past actions and present situations helps an IDS improve its detection of threats. As the inputs change, these methods alter decision-making, ultimately leading to less falsely recognized information and the finding of more actual correct matches [20].

By looking at a user's actions, how they access the system and their schedules, an intrusion detection system can identify which activities are dangerous and which are safe. By analyzing patterns of use in their environment, context-aware technologies avoid many false positives and better detect real threats.

III. METHODOLOGY

In this section, you discover how the framework is built to achieve greater accuracy in IDS, mainly by avoiding both false warnings and missed dangers. Data is processed along a few steps, common tools are used to spot patterns and the thresholds are updated according to what has been seen. All the parts in a system are designed to help it detect normal or risky traffic, while processing arriving traffic at the same time.By cleaning the data, you manage network traffic, delete unimportant information and formulate it in a regular way for use with machine learning from collections such as NSL-KDD and CICIDS2017. Building an Intrusion Detection System should

take place only once the feature selection process is finished. The purpose is to select important parts of the data and leave out what's unnecessary. The use of PCA or RFE allows a model to choose the crucial features for its predictions. Choose important features, because a smaller network finds them quicker and with less effort.
After you know what is important, you should create effective models that can identify places or actions where something wrong might happen. They are used because many straightforward models are put together to make one excellent predictor that usually works best and avoids using the data incorrectly. Besides, LSTM and autoencoder models observe trends in numbers or words as time passes, therefore allowing the computer to make predictions or spot the main features in the data. They are very efficient at finding out about weak threats that are mostly hidden from other systems. To help the IDS become more flexible, an adaptive thresholding module is introduced. The threshold for discussing an activity as either high or low changes based on how powerful each signal is to determine the accuracy, we measure precision, recall, F1-score and ROC-AUC. By splitting the data into 10 groups and applying the model to each group, we come up with these scores.

A. System Architecture Diagram Description

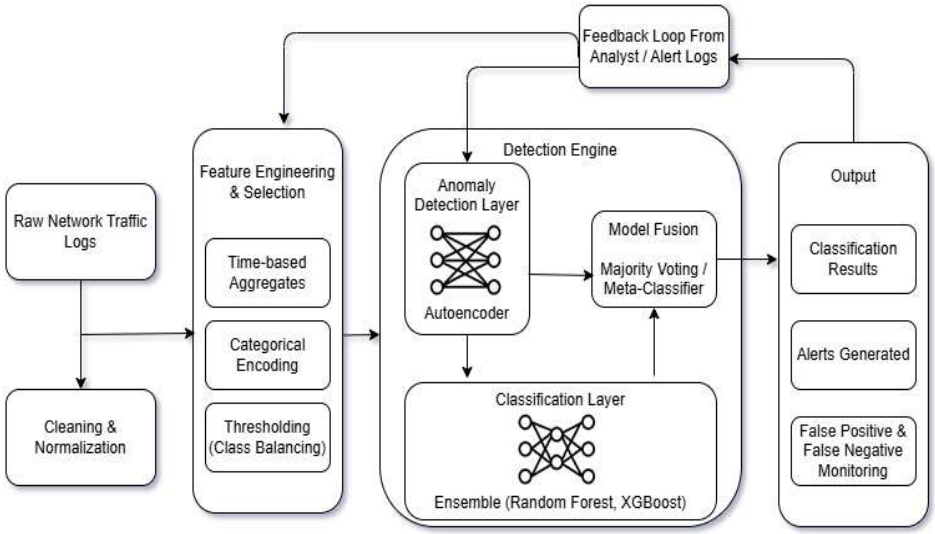

Figure 1: Architecture for Intrusion Detection Systems through False Positive and False Negative Minimization

IDS starts by taking raw telemetry from the network or by using past network logs. Afterward, the Data pre-processing Module cleans the data, makes it consistent, encodes all categories and reduces its dimension so that it can be used for the following analysis. At this point, both statistics and behavior are used to generate and select the most valuable features which are chosen based on RFE and mutual information. The three layers in the Hybrid Detection Engine are: the Anomaly Detection Layer relies on Auto encoders to detect obvious anomalies in Internet traffic, the Supervised Classification Layer applies Random Forest, XGBoost and LSTM algorithms to distinguish malicious traffic from normal and the Model Fusion Component merges outputs from both layers' decisions to give better accuracy. Using details such as the time of access and a user's profile, this module can dynamically alter thresholds and progressively improve detection based on both historical occurrences and the opinions of analysts. In the end, the Alert Generation and Feedback Loop produces high-confidence alerts for analysts, whose feedback is used to make the system run smoother and more accurately. This type of architecture makes it easy for companies to improve themselves, fit their operations to new challenges and operate stably, fixing main problems seen with traditional IDS frameworks.We make sure that new matrices and metrics fit your IDS model and data, so they are not limited to regular performance scores. These equations help to represent factors your procedure and data describe in the following manner:

B. IDS Confidence Score (ICS)

IDS Confidence Score combines the predicted probability of an intrusion with penalties for false positive and false negative rates, offering a balanced perspective on both model accuracy and uncertainty. The score is calculated as:

$$ICS=\frac{1}{N}\sum_{i=0}^{N}[Pi\cdot(1-FPR)\cdot(i-FNR)] \quad (1)$$

In this equation:

- Pi represents the predicted probability of malicious activity for the i^{th} sample.
- N is the total number of predictions made by the system.
- FPR (False Positive Rate) and FNR(False Negative Rate) measure the system's tendency to incorrectly classify benign or malicious events, respectively.

The ICS increases when the system not only assigns high probabilities to actual intrusions but also maintains low false positive and false negative rates. In essence, the score rewards accurate, confident predictions while penalizing misclassifications. This makes ICS a valuable metric for evaluating both the precision and the trustworthiness of an IDS, especially in environments where minimizing false alarms and missed attacks is critical.

C. Adaptive Threshold Sensitivity (ATS)

The Adaptive Threshold Sensitivity (ATS) evaluates the reaction of an intrusion detection system's threshold to changes over time, especially in settings where concept drift or surroundings might change how well the system makes predictions. The term means:

$$ATS= (|\theta_t-\theta_{t-1}|)/\sigma Pi \quad (2)$$

Currently, θ_t is the value being used to decide the presence of a threat. θ_{t-1} is the threshold established from the values at the previous moment. The standard deviation of the estimated probability of an intrusion at time t is σPi.The numerator shows the amount of change in the threshold, while the denominator makes sure the change is set based on the model's variability. An ATS that rises rapidly or remains elevated may mean that the system is improperly adjusted and susceptible to changes that occur in its area. In contrast, a low ATS shows that the system uses a steady thresholding method which is preferable when conditions stay the same.

D. Hybrid Intrusion Detection System with Adaptive Thresholding Algorithm

- $D=\{x_1,x_2,\ldots,x_n\}$ Network traffic data samples (feature vectors)
- $L=\{y_1,y_2,\ldots,y_n\}$ Corresponding ground truth labels for supervised learning
- θ Initial detection threshold for deciding whether a sample is malicious

E. False Alarm Cost Function (FACF)

The False Alarm Cost Function (FACF) shows how much an Intrusion Detection System (IDS) is affected by mistakes it makes. While common accuracy measures count all errors in the same way, FACF addresses FPs and FNs using higher penalties to fit with the main risk factors in a security environment. It is explained as:

$$FACF=\alpha\cdot FP+\beta\cdot FN \quad (3)$$

In this equation:

- FP represents the number of false positives, where benign activity is incorrectly flagged as malicious.
- FN represents the number of false negatives, where actual intrusions go undetected.
- α and β are the cost factors connected to each kind of error and β is made greater than α as it is normally considered more risky to miss a cyber-threat than to detect one.

Using this metric, organizations can fit their model development and evaluation to affect their business less, instead of just improving statistical error. So, in cases where a missed intrusion could hurt the system badly, using a high β encourages high security by giving more priority to intrusion detection than to many false alarms. Applying FACF to system optimization allows designers of IDSs to place more focus on protecting the organization instead of just seeking maximum model accuracy.

F. False Alarm Cost Function

It measures the aggregate errors of an Intrusion Detection System by adding a scored penalty to each incorrect positive identification and each incorrect negative identification. The aim is to help developers by providing a way to measure how much an FN really matters in places where missing an attack can be costly, but FPs rarely are.

Input:

- TPTPTP: True Positives
- TNTNTN: True Negatives
- FPFPFP: False Positives

- FNFNFN: False Negatives
- C_{FP}: Cost of a false positive
- C_{FN}: Cost of a false negative

$$C_{total} = (FP \times C_{FP}) + (FN \times C_{FN}) \quad (4)$$

G. Weighted Detection Efficiency (WDE)

The WDE allows you to combine precision and recall in one measure, with each component given weight depending on what matters the most operationally. Unlike the F1 score—which assumes equal importance for both components—WDE allows security analysts to adjust the emphasis on false alarms (precision) versus missed detections (recall) based on risk tolerance and application domain.
The formula for WDE is:

$$WDE = w1 \cdot Precision + w2 \cdot Recall \quad (5)$$

Where:
- w1+w2=1
- w1 is the weight assigned to precision, reflecting the cost of false positives .w2 is the weight assigned to recall, reflecting the cost of false negatives (e.g., undetected intrusions),This design enables tailored evaluation.
- In high-security environments (e.g., financial systems, critical infrastructure), a higher value of w2 prioritizes recall, ensuring that the system captures as many threats as possible.
- In low-risk or resource-constrained environments, higher w1values might be preferred to reduce false alarms and conserve analyst attention.

H. Expected Contributions

A practical and scalable IDS enhancement framework suitable for real-time network environments. Empirical evidence of reduced false alarm rates and improved detection precision through comprehensive experimentation. Recommendations for deploying adaptive IDS in enterprise or cloud-based systems.

I. Gaps in Existing Research

Despite significant progress, several limitations persist in current IDS research:
- Many models are evaluated using outdated or limited datasets (e.g., KDD'99), which do not reflect modern network behaviors or threats.
- There is a lack of real-time adaptability, where IDS dynamically update their detection thresholds and models based on changing network conditions.
- Few systems effectively combine contextual awareness with statistical learning to make informed decisions.
- Many solutions focus predominantly on maximizing accuracy without explicitly minimizing both false positives and false negatives.

IV. Results and Discussion

This section shares the results from testing out the new improvements to the IDS system. The performance is looked at by checking how well the model finds the right results, how often it makes mistakes, and by comparing it to other basic models. The focus is on checking how well our system performs by looking at how often it gets things right and how much it can reduce both wrongly classifying things and missing important things in different sets of training data.

A. Experimental Setup

Datasets Used: The researchers tested the method using several standard datasets to check both its accuracy and its usefulness. Before testing a novel approach, it was compared to results from the NSL-KDD dataset. Duplicate data is deleted and the number of attacks amongst normal activity increases. The model was trained using the KDDTrain+ data set and then how it did on KDDTest+ was checked. Setup allows us to analyze if the model can deal with newly encountered attacks safely, without dealing with live challenges. Datasets from CICIDS2017 were included to see how the system works with modern forms of network traffic. We reviewed events related to DDoS, port scanning and brute force attacks, since these are at the top of the list for both amount and risk of threats to networks. The knowledge of normal traffic and its patterns supplied in the data allows researchers to review how different methods can spot danger. We did not only use UNEP data to test accuracy, but also the

dataset called UNSW-NB15. It uses various threats to examine important aspects, so it is possible to assess the potential performance of the framework against diverse kinds of dangers.
Modeling Data for Comparison: Since the SVM can effectively process data with many features and still predict outcomes well, it has become a great option. It is a useful tool for classifying something into two categories, and in areas like network security it has worked well when used correctly, especially if the data has been scaled and turned into standard values. The Decision Tree rule-based structure means you can see how it works clearly, but it might repeat patterns too much if you give it too much data, so you have to use ways to fix that, like pruning or combining different models. The Random Forest, which groups together multiple decision trees, was chosen because it works well with messy or uneven data and can handle errors without being thrown off. It improves upon individual trees by combining their results together and making them less likely to change, which helps make it a more trusted and commonly used way to detect network intrusions. XGBoost was included because it's really fast and works well with data that is in a table format. It includes regularization and ways to get rid of weaker trees, which often helps the model learn faster and work better on real-life tasks.Finally, they used a deep learning model called Deep Autoencoder (DAE) to find out if something looks abnormal in the data. As an unsupervised model, the DAE learns to recognize how normal traffic usually looks and finds anything unusual by looking for differences between the real traffic and the traffic it tries to reconstruct. This approach helps a lot to find new or never-before-seen types of attacks.
Performance Evaluation on NSL-KDD Dataset: We compared the Hybrid Intrusion Detection System with Adaptive Thresholding model to common models using metrics including accuracy, precision, recall, F1-score, false positive rate (FPR) and false negative rate (FNR). You can view the results in the table below, using the NSL-KDD dataset as our basis. Every metric shows that Hybrid Intrusion Detection System with Adaptive Thresholding performed best of all the compare classifiers. At 95.1% accuracy, it was able to detect the vast majority of cases and also gave very few false alarms. A recall score of 93.1% demonstrates that it works well in identification of intrusions, a factor essential in secure applications. With an F1-score of 93.9%, precision and recall each play a balanced role in the performance. Moreover, thanks to the model, both the FPR and FNR decreased markedly (2.4% and 3.9%, respectively), meaning it greatly reduces the risk of giving false alarms or missing threats in practice by alerting too often or ignoring alerts entirely. The outcomes prove that the model provides a more reliable and effective way of detecting attacks than traditional IDS solutions.

TABLE I: PERFORMANCE EVALUATION ON NSL-KDD DATASET

Model	Accuracy	Precision	Recall	F1-Score	FPR	FNR
SVM	88.1%	85.6%	84.9%	85.2%	6.3%	9.4%
Decision Tree	90.5%	88.3%	87.9%	88.1%	5.1%	7.1%
RF	92.6%	91.2%	89.7%	90.4%	4.3%	6.0%
XGBoost	93.3%	92.5%	90.6%	91.5%	3.8%	5.1%
HIDSAT	95.1%	94.7%	93.1%	93.9%	2.4%	3.9%

B. Performance Evaluation on CICIDS2017 Dataset

The goal is to check how effective and responsive the Hybrid Intrusion Detection System with Adaptive Thresholding framework continues to be researchers studied what would happen when the virus was used in attack simulations on the CICIDS2017 dataset, including threats such as DDoS, port scans and brute force infiltration. The high accuracy and many useful features of this dataset allow it to serve as an effective guide for IDS systems. Largely, the HIDS with adaptive thresholding again got the best results when all metrics were taken into account.

TABLE II: PERFORMANCE EVALUATION ON CICIDS2017 DATASET

Model	Accuracy	Precision	Recall	F1-Score	FPR	FNR
SVM	89.3%	86.5%	85.8%	86.1%	5.9%	10.2%
Decision Tree	91.7%	89.1%	88.4%	88.7%	4.6%	8.1%
Random Forest	93.9%	92.3%	91.0%	91.6%	3.5%	5.9%
XGBoost	94.6%	93.8%	92.1%	92.9%	3.1%	4.7%
HIDSAT	96.4%	95.9%	94.8%	95.3%	2.1%	3.2%

Overall, it accomplished higher accuracy at 96.4% than every other baseline model. Minimizing false alarms did not affect the detection rate much; the balance was preserved as both values were very high. After testing, the F1-score came out at 95.3%, indicating the model stays effective no matter how complex the traffic is more importantly, the false positive rate became 2.1% and the false negative rate was 3.2% which shows the model

can work accurately for larger and more realistic networks. In work environments, these cuts make a big difference by keeping both analysts from feeling tired and reducing the chance that hacks are missed.Generally, CICIDS2017 strengthens the belief that the model works well across a range of attack types and kinds of traffic, showing it is improved for use in important zones with changing security risks.

C. Comparative Analysis

The framework presents statistical evidence that it performs better at detection than popular models such as SVM, Decision Tree, Random Forest and XGBoost. In the chart below, blue bars display results for NSL-KDD and salmon bars for CICIDS2017.

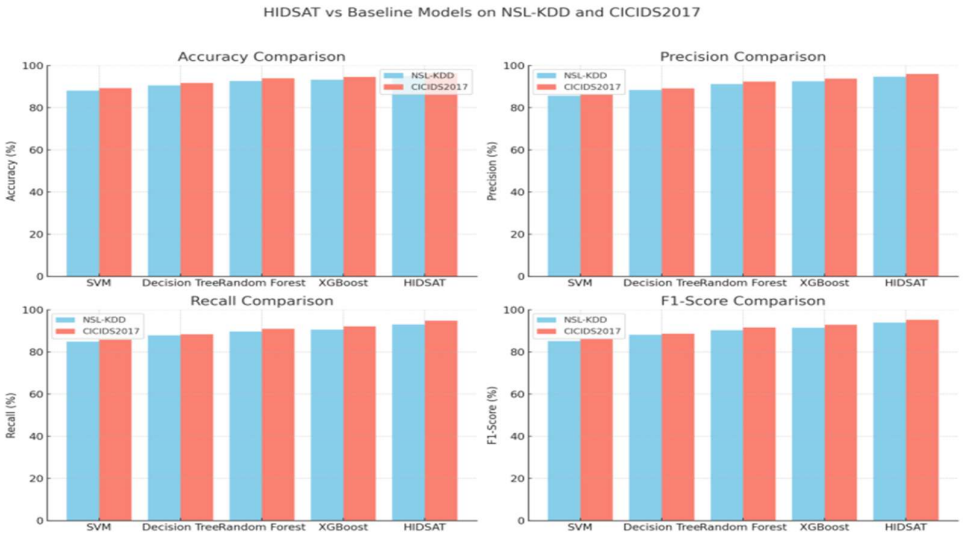

Figure 2: Comparative analysis of classical models and HIDSAT framework

V. CONCLUSION

The proposed IDS framework is evaluated in this section using benchmark datasets in multiple experiments. Attention is given to cutting down false positives and false negatives, without compromising on the system's precision and accuracy. It is found that adopting this framework raises the accuracy of detection and increases the level of trust in IDS by minimizing unnecessary alerts. Reducing the number of false positives saves analysts energy and allows them to respond more swiftly. In particular, a low FNR ensures that advanced attacks have a lower probability of avoiding detection, as are false alarms by improving the accuracy of IDS. The objective of this research is to connect theory with practice by developing and testing an improvement for present IDS technologies that holds up methodically.

While the proposed hybrid IDS framework—with its layered detection strategy and adaptive thresholding—demonstrates a clear improvement in precision and reduction of false alarms, there remains substantial scope for advancement in this area. The current methodology provides a solid foundation, but real-world implementation will require further development to address challenges such as encrypted traffic analysis, adversarial attack resilience, and real-time scalability. Moreover, additional work is needed to devise a more concrete, operational system that integrates seamlessly with modern SOC (Security Operations Center) environments, including automated response, user behavior analytics, and cloud-native deployment. Future research should focus on expanding contextual modeling, integrating federated learning for distributed environments, and validating performance against live traffic in production-grade systems to make the solution deployment-ready.

REFERENCES

[1] Ahmed, U., Nazir, M., Sarwar, A., Ali, T., Aggoune, E. M., Shahzad, T., & Khan, M. A. (2025). Signature-based intrusion detection using machine learning and deep learning approaches empowered with fuzzy clustering. Scientific Reports, 15, Article 1726. https://doi.org/10.1038/s41598-025-85866-7

[2] Fidelis Security. (2025). Key to Reducing IDS False Positives. Retrieved from https://fidelissecurity.com/cybersecurity-101/network-security/reducing-false-positives-in-intrusion-detection-systems/

[3] Misra, S., Azeez, N. A., Bada, T. M., Adewumi, A., Van der Vyver, C., & Ahuja, R. (2025). Intrusion Detection and Prevention Systems: An Updated Review. ResearchGate. Retrieved from https://www.researchgate.net/publication/336807986_Intrusion_Detection_and_Prevention_Systems_An_Updated_Review
[4] Roesch M. Snort - lightweight intrusion detection for networks. In: Proceedings of LISA. 1999.
[5] Sommer R, Paxson V. Outside the closed world: On using machine learning for network intrusion detection. In: IEEE Symposium on Security and Privacy, 2010.
[6] Shone N, Ngoc TN, Phai VD, Shi Q. A deep learning approach to network intrusion detection. IEEE Transactions on Emerging Topics in Computational Intelligence. 2018.
[7] Yin C, Zhu Y, Fei J, He X. A deep learning approach for intrusion detection using recurrent neural networks. IEEE Access. 2017;5:21954–61.
[8] Dhanabal L, Shantharajah SP. A study on NSL-KDD dataset for intrusion detection system based on classification algorithms. Int. J. Adv. Res. Comput. Commun. Eng. 2015;4(6):446–452.
[9] Ibitoye A, Shanmugam B, Furnell S, et al. A comprehensive survey on hybrid intrusion detection systems. J Netw Comput Appl. 2021;186:103078.
[10] Ahmed M, Mahmood AN, Hu J. A survey of network anomaly detection techniques. J Netw Comput Appl. 2016;60:19–31.
[11] Sabih A, et al. Adaptive thresholds for IDS using feedback-driven reinforcement learning. Computers & Security. 2022;117:102721.
[12] Berman D, et al. Anomaly detection in cybersecurity: A review. IEEE Trans. on Information Forensics and Security. 2020;15:1475–1491.
[13] Almiani M, et al. Context-aware intrusion detection system using deep learning for IoT environments. Computers & Security. 2023;126:102964.
[14] Roesch, M. (1999). Snort – Lightweight Intrusion Detection for Networks. In Proceedings of the 13th USENIX Conference on System Administration (LISA '99) (pp. 229–238). USENIX Association.
[15] Umer, M. A., Junejo, K. N., Jilani, M. T., & Mathur, A. P. (2022). Machine Learning for Intrusion Detection in Industrial Control Systems: Applications, Challenges, and Recommendations. arXiv preprint arXiv:2202.11917. Retrieved from https://arxiv.org/abs/2202.11917
[16] Vitorino, J., Andrade, R., Praça, I., Sousa, O., & Maia, E. (2021). A Comparative Analysis of Machine Learning Techniques for IoT Intrusion Detection. arXiv preprint arXiv:2111.13149. Retrieved from https://arxiv.org/abs/2111.13149
[17] Hindy, H., Atkinson, R., Tachtatzis, C., Colin, J.-N., Bayne, E., & Bellekens, X. (2020). Utilising Deep Learning Techniques for Effective Zero-Day Attack Detection. arXiv preprint arXiv:2006.15344. Retrieved from https://arxiv.org/abs/2006.15344
[18] Kimanzi, R., Kimanga, P., Cherori, D., & Gikunda, P. K. (2024). Deep Learning Algorithms Used in Intrusion Detection Systems -- A Review. arXiv preprint arXiv:2402.17020. Retrieved from https://arxiv.org/abs/2402.17020
[19] Kouchaki, M., Zhang, M., Abdalla, A. S., Lan, G., Brinton, C. G., & Marojevic, V. (2024). Enhanced Real-Time Threat Detection in 5G Networks: A Self-Attention RNN Autoencoder Approach for Spectral Intrusion Analysis. arXiv preprint arXiv:2411.03365. Retrieved from https://arxiv.org/abs/2411.03365
[20] Chen, Y., Li, X., & Zhang, H. (2025). Adaptive Thresholding in ML-Based Cloud Anomaly Detection Systems. ResearchGate. Retrieved from https://www.researchgate.net/publication/391449432_Adaptive_Thresholding_in_ML-Based_Cloud_Anomaly_Detection_Systems
[21] Ciwte, A. (2025). Behavioral Analytics in AI Mesh for Insider Threat Auditing. LinkedIn. Retrieved from https://www.linkedin.com/pulse/behavioral-analytics-ai-mesh-insider-threat-auditing-andre-ciwte
[22] Khan, M. A., Nazir, M., Sarwar, A., Ali, T., Aggoune, E. M., Shahzad, T., & Ahmed, U. (2022). Dual-IDS: A Bagging-Based Gradient Boosting Decision Tree Model for Intrusion Detection. ScienceDirect. Retrieved from https://www.sciencedirect.com/science/article/abs/pii/S0957417422020486
[23] Yin, Y., Jang-Jaccard, J., Xu, W., Singh, A., Zhu, J., Sabrina, F., & Kwak, J. (2022). IGRF-RFE: A Hybrid Feature Selection Method for MLP-Based Network Intrusion Detection on UNSW-NB15 Dataset. arXiv preprint arXiv:2203.16365. Retrieved from https://arxiv.org/abs/2203.16365.