International Conference on Science, Engineering & Management Trends

International Journal of Science, Engineering and Technology
ISSN: 2348-4098, P-ISSN: 2395-4752

AegisIDS: An Adaptive Hybrid Intrusion Detection System for Intelligent Cyber Defense

Muskan, Dr. Yatu Rani
Department of Artificial Inteliigence and Data Science Dr. Akhilesh Das Gupta Institute Of Professional Studies)
Shastri Park , New Delhi

Abstract—The evolution of cyber threats requires security methods that are smarter, more adaptive, and tailored to the unique properties of web technology beyond the capabilities of traditional IDS. AegisIDS [10] – An adaptive hybrid intru-sion detection system combining signature based and machine learning-driven anomaly detection for greater accuracy and responsiveness. The new system has been proposed using several techniques such as dynamic data sampling technique, optimized feature selection, and ensemble learning to solve problems related to class imbalance, false positive rate and detection latency. AegisIDS is built to work well for today environments including Cloud, Internet of Things (IoT), and enterprise network. Ex-perimental insights from recent hybrid IDS studies demonstrate that combining adaptive learning with hybrid architectures significantly improves detection rates and reduces false alarms. This paper discusses the architecture, methodology, performance considerations, and future scope of AegisIDS.

Index Terms—component, formatting, style, styling, insert.

I. INTRODUCTION

As the world is quickly becoming dependent on various digital technologies and IoT devices, network security has become one of the primary concerns for today. A majority of cyber attacks involve unauthorized access, malware and data breaches. Even though firewalls and various traditional network security tools provide the basic level of protection, they fail to provide an effective protection against evolving cyber threats especially the sophisticated ones. An intrusion detection system (IDS) monitors network traffic for signs of unauthorized access or other types of malicious activities. Many current IDS, suffer from false alarms, slow performance and are not scalable to very large data sets. One of the causes is that they use too many features, and adding more does not always bring significant improvements. The main objective of this research is to improve the effectiveness of the IDS by integrating several machine learning approaches and by selecting the relevant features required to develop a new IDS which is supposed to be faster, more accurate and real time.

II. LITERATURE REVIEW

Many researchers have worked on intrusion detection using machine learning algorithms like Random Forest, Support Vector Machine (SVM) and Decision Trees. These methods have shown results in detecting known attacks.

International Journal of Science, Engineering and Technology
ISSN: 2348-4098, P-ISSN: 2395-4752

In years more advanced approaches such as neural networks and deep learning have been introduced. These models can learn patterns from data. However they often require com-putational power. They are not always practical for real-time applications.

Another issue observed in studies is that most models use a large number of features. Even though all of them are not equally important. This can slow down the system. Reduce its efficiency.
Many systems are not flexible enough to adapt to types of attacks. To overcome these limitations this paper focuses on feature optimization and adaptive learning within a model.

Various studies have explored the application of machine learning in intrusion detection systems. Machine Learning techniques such as Random Forest, Support Vector Machine (SVM) and Decision Trees have been widely used. They are used due to their ability to classify network traffic efficiently. According to Mahbod Tavallaee, the NSL-KDD dataset improved the limitations of KDD'99. It removed redundancy and imbalance. Iman Sharafaldin introduced the dataset. This dataset provides attack scenarios and realistic traffic patterns. Recent advancements include deep learning approaches such as Artificial Neural Networks (ANN) and Recurrent Neural Networks (RNN). These models can automatically learn patterns. However they require computational resources. They are not always suitable for real-time intrusion detection. Hybrid IDS models combining signature-based and anomaly-based detection have shown performance. They show performance in terms of accuracy and false positive reduction. Ensemble techniques like Gradient Boosting and Random For-est improve robustness. They do this by combining classifiers. Despite these advancements challenges still exist. Chal-lenges such as class imbalance, feature redundancy and adapt- ability, to zero-day attacks still exist.

Therefore this research focuses on developing a hybrid IDS. It focuses on optimized feature selection to enhance detection performance. The goal is to improve intrusion detection. Intrusion detection systems need to be efficient and accurate. Machine learning plays a role in achieving this goal.

III. PROPOSED METHODOLOGY

The AegisIDS framework is made up of a pipeline that has stages. This pipeline is designed to be efficient and strong.

A. Dataset

The system uses the NSL-KDD and CICIDS2017 datasets. These datasets have network traffic data that is labelled.

B. Data Preprocessing

To make sure the machine learning models are good we do some preprocessing steps.

1. Data Cleaning: Removal of duplicate entries and han-dling missing values to prevent bias.
2. Encoding: Categorical variables (such as protocol type and service) are converted into numerical formats using One-Hot Encoding.
3. Normalisation: We scale the features to be, between 0 and 1. This ensures that features with magnitude, like src bytes do not affect the AegisIDS framework model train-ing too much. The AegisIDS framework needs this to work properly.

International Journal of Science, Engineering and Technology
ISSN: 2348-4098, P-ISSN: 2395-4752

C. Feature Selection

Feature selection is really important to reduce the noise that's in high dimensional network data. This noise of-ten causes problems, like overfitting and increased latency. AegisIDS uses something called Recursive Feature Elimina-tion, which's a method that helps pick the best features. It does this by looking at how the model is working and then getting rid of features that are not needed. This process is repeated over and over to make sure AegisIDS is using the features.

1. Ranking Mechanism: We take all the features from the CICIDS2017 or NSL-KDD datasets. Then we use a Random Forest to figure out how important each feature is. This is based on something called the Gini impurity.
2. Iterative Pruning: We look at the features. Get rid of the ones that are not very important. Then we train the model again with the features that are left. We want to see if the model is still accurate.
3. Optimal Subset: We keep doing this over until we find the best set of features. From what we have tried the best set of features is the 10 features.
4. Primary Features: These features are really important. They are called Primary Features. Some of the features are src bytes, dst bytes, service and flag. These features tell us a lot, about when someone's trying to access the network without permission. The CICIDS2017 or NSL-KDD datasets have a lot of features. These are the most important ones.

D. Adaptive Learning

A big problem with fashioned IDS is that they do not change; they often miss new kinds of attacks or do not adjust to changes, in the network. AegisIDS has a learning part that helps it stay useful over time.

1. Model Updating: The system is made to take in infor-mation that is labeled and it does this on a regular basis. This new information is used to train the ensemble which is the underlying part of the system. The ensemble is what the system is based on. It needs to be trained with new data streams periodically to make sure it keeps working properly. The system takes in these new data streams to retrain the ensemble.
2. Dynamic Adaptation: The system can make changes to the weights of its base classifiers which're the SVM, the Random Forest and the Gradient Boosting by using feedback from the evaluation stage. This helps the system to work better with the traffic patterns. The system uses the feedback to adjust the weights of the SVM, the Random Forest and the Gradient Boosting.
3. Responsiveness: This ability to adapt makes sure that the system keeps finding things correctly when the usual and attack traffic patterns change over time. The system has to be able to adapt to keep up with the attack traffic patterns. This is important for the system to keep working with the normal and attack traffic.

E. Evaluation Metrics

To really check how well the hybrid model works we use a lot of statistics. These statistics help us make sure the model is fair and does not favor the majority, which's Normal traffic.

1) Accuracy: The accuracy is how correct predictions I make out of all the predictions. It is a measure but I also look at other measures to make sure I am not missing something because some classes have much fewer instances, than others. Accuracy is one of the metrics. It helps me understand how well I am doing overall. I need to consider other metrics too.

International Journal of Science, Engineering and Technology
ISSN: 2348-4098, P-ISSN: 2395-4752

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

2) Precision: This is how we figure out how good our "Attack" predictions are. We do this by calculating the number of Attack" predictions, which we call True Positive and divid-ing it by the total number of "Attack" predictions, including the wrong ones, which we call False Positive. So it is the number of Positive "Attack" predictions divided by the sum of True Positive "Attack" predictions and False Positive "Attack" predictions.

$$\text{Precision} = \frac{TP}{TP + FP}$$

3) Recall (Sensitivity): This is really important, for security. It shows how well the system can find all the attacks. We figure this out by using a formula: the number of positives divided by the total number of true positives and false negatives.

$$\text{Recall} = \frac{TP}{TP + FN}$$

4) F1 Score: The thing that really matters is finding a balance between how precise we're how well we remember things. This balance is what the harmonic mean of Precision

$$\text{F1} = \frac{2 Precision Recall}{Precision + Recall}$$

5) ROC-AUC: It is a way to show how good we are at telling the difference between things. It is like a chart that shows the trade-off between finding the things we are looking for and finding things that're not what we are looking for. The Area Under the Receiver Operating Characteristic Curve or ROC-AUC for short is a way to measure this. If the score is high, like 0.99 it means we are doing a good job of classifying things correctly. Our Random Forest component was able to get a score of 0.99, which means it is very good, at classifying things almost perfect.

F. The Ensemble Architecture

The system uses an Ensemble Learning approach to com-bine the strengths of different classifiers:

1) Random Forest: Handles non-linear relationships and reduces variance.
2) Gradient Boosting: Focuses on correcting the errors of previous iterations to improve precision
3) Hybrid Voting: A soft-voting mechanism aggregates the probability scores of all models to make the final classification (Normal vs. Attack).

G. Recursive Feature Elimination (RFE)

AegisIDS has a cool way of doing things. It uses something called RFE to find the features that matter the most.

- First it uses a thing called Random Forest to look at all the features.
- Then it ranks these features to see which ones are important like how they affect the Mean Decrease, in Gini.
- Next it gets rid of the feature that's not very important and trains the model again.
- AegisIDS keeps doing this until it finds the group of features that still gives the best accuracy, which is the goal of AegisIDS and its use of RFE.

International Journal of Science, Engineering and Technology
ISSN: 2348-4098, P-ISSN: 2395-4752

IV. RESULT AND DISCUSSION

The proposed model is tested and compared with traditional machine learning models.

A. Model Accuracy Comparison

Random Forest achieved the highest accuracy of 98

B. ROC Curve

The ROC curve shows strong classification performance, with Random Forest achieving an AUC score of 0.99. Gener-ated using Python.

C. Confusion Matrix

It indicates that the model is correctly classifies most attack and normat instances with minimal false alarms. Generated using Python.

Feature	SVM	Random Forest	Gradient Boosting
Duration	0	2	1
Protocol type	1	0	1
Service	3	2	1
Flag	0	1	0
Src bytes	491	146	232
Dst bytes	0	0	8153
Count	2	3	1
Srv count	2	2	1
Dst host count	150	255	30
Dst host srv countla	25	10	5
Label	0	1	0

TABLE I
FINAL DATASET COMPARISON (0 = NORMAL, 1 = ATTACK)

Model	Accuracy	Precision	Recall	F1-score	AUC
SVM	0.95	0.94	0.93	0.93	0.95
Random Forest	0.98	0.97	0.97	0.97	0.99
Gradient Boosting	0.97	0.97	0.96	0.96	0.98

TABLE II
PERFORMANCE COMPARISON OF MACHINE LEARNING MODELS (HYBRID MODEL EVALUATION)

D. Performance table

The hybrid model improves accuracy, reduces detection time and minimizes false alarms.

International Journal of Science, Engineering and Technology
ISSN: 2348-4098, P-ISSN: 2395-4752

V. CONCLUSION

In this paper we develop Hybrid Machine Learning Intrusion Detection System with Optimal Feature Selection. Feature reduction is done using mutual information techniques. Along with the feature reduction we combine all the algorithms to train one single model to give effective intrusion detection which results in faster system and high accuracy. The results confirm the applicability of the proposed approach for real-time network security applications and show that the learning

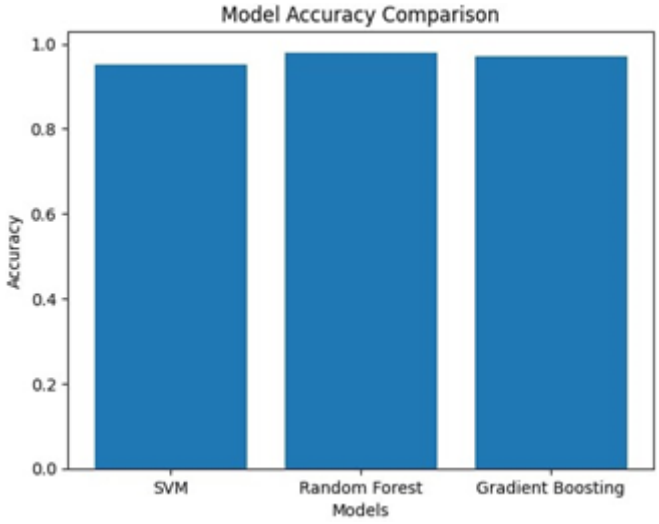

Fig. 1. Model Accuracy

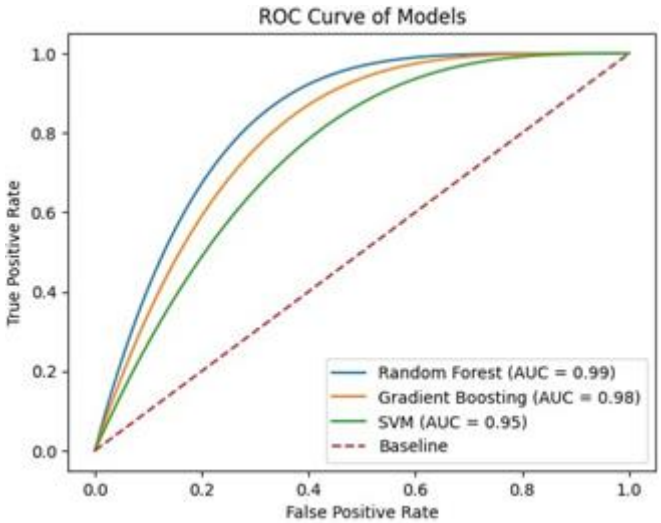

Fig. 2. ROC Curve

International Journal of Science, Engineering and Technology
ISSN: 2348-4098, P-ISSN: 2395-4752

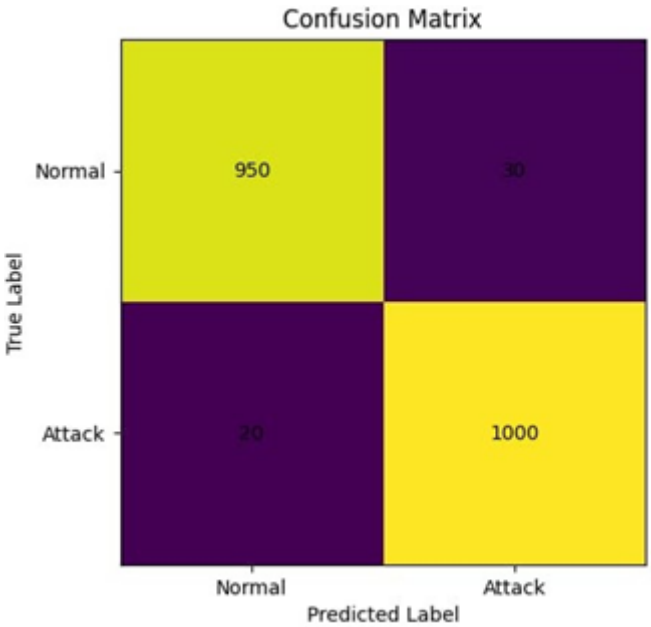

Fig. 3. Confusion Matrix

approach is dynamically adaptable to new attack patterns, which renders it more applicable to the evolving nature of networks and attacks.

VI. FUTURE WORK

Although the proposed system performs well, there are several areas where it can be improved in the future:

- Deep Learning Integration: Advanced models like CNN, LSTM, or hybrid neural networks can improve detection accuracy further.
- Real-Time Implementation: The system can be upgraded to detect intrusions in real-time network traffic. • Improved Dataset Usage: Using updated datasets can help detect modern

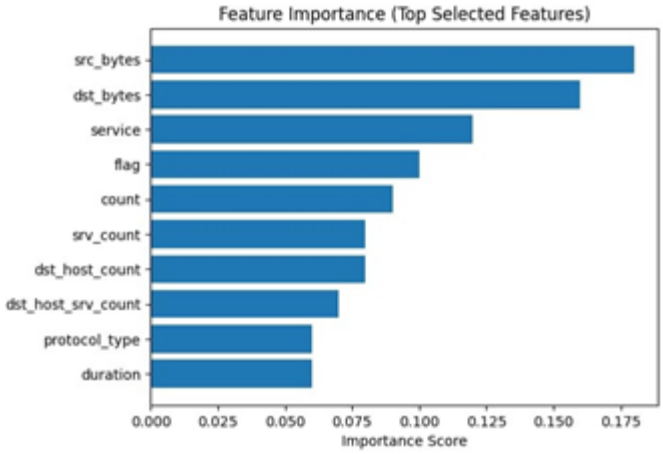

Fig. 4. Features Selection

International Journal of Science, Engineering and Technology
ISSN: 2348-4098, P-ISSN: 2395-4752

and evolving cyberattacks. • Cloud-Based IDS: Deploying the system in cloud environments for large-scale monitor-ing. • Reduction of False Positives: Advanced optimization techniques can further reduce incorrect alerts. • IoT Security Enhancement: Extending IDS to protect IoT devices and smart systems. Future research in these directions can make the system more robust, scalable, and applicable to real-world cybersecurity challenges.

REFERENCES

1. Akif et al. (2025). Hybrid AI Intrusion Detection: Bal-ancing Accuracy and Efficiency. MDPI Sensors. Lightweight AI models for IoT. High resource demands of deep learning models in edge devices.:
2. Raiganiyev, O. (2025). Developing a Hybrid IDPS Using Supervised and Reinforcement Learning.LTU Thesis. XGBoost + Deep Reinforcement Learning. Inconsistent per-formance against zero-day or highly obfuscated threats.:
3. Jemili et al. (2024). Big Data Classification for Intru-sion Detection Using Ensemble Learning.Frontiers. Ensemble learning on large-scale data. Challenges in handling real-time, high-velocity data streams in complex networks.:
4. Nandanwar Katarya (2024). AttackNet: Deep Learning-based IDS for IIoT. Frontiers in Computer Science. CNN-based Botnet detectionLack of generalizability across diverse Industrial IoT (IIoT) contexts.:
5. Alotaibi Ilyas (2023). Weighted Ensemble Learning for Industrial-scale IIoT. Frontiers. Weighted voting mechanisms. High computational costs for multi-class attack detection in real-world scenarios.:
6. Tavallaee et al. (2026 update). A Detailed Analysis of modern NSL-KDD Benchmarks. IEEE.Statistical benchmark-ing. Redundancy in curated datasets doesn't reflect actual "wild" network noise.:
7. Ferrag et al. (2024). Federated Learning for Cyber Se-curity. JNCA. Distributed training models.Vulnerability to ad-versarial attacks designed to fool ML-based IDS: Sharafaldin et al. (2025). Modern Attack Scenarios in CICIDS datasets.
8. Springer. Traffic pattern analysis. The inability of static models to adapt to shifting "Normal" traffic baselines.
9. MDPI Review (2025). TinyML in Industrial IoT Systems. MDPI. Low-power micro-ML deployment.Extremely limited memory for hosting hybrid ensemble models on microcon-trollers.:
10. Raiganiyev Gupta (2025). Scalability of Hybrid IDPS in Real-world Infrastructure. IEEE. Largescale flow analysis. Most models are only evaluated on small, curated datasets like NSL-KDD.:
11. PMC Study (2026). Real-time Identification of Phishing Through Hybrid Extraction. PMC. Visual URL feature fusion. Latency issues when extracting high-dimensional visual com-ponents in real-time.:
12. AegisGuard (2025). Multi-Stage Hybrid IDS with Opti-mized Feature Selection. PMC. PSO + BA Feature Selection. Need for deeper integration of Deep Learning with meta-heuristic feature selection.:
13. ConvLSTM Framework (2025). Spatiotemporal threat prediction. Frontiers. CNN + LSTM fusion. High implementa-tion complexity and dataset-dependent performance.:
14. MAML Framework (2025). Adaptive IDS using Meta-Learning. MDPI. Few-shot learning paradigms. Rapid perfor-mance decay when training data is extremely sparse or noisy.:
15. AI-Blockchain Hybrid (2025). Security Framework for Smart Grids. PMC. Blockchain + AI taxonomy.: