AN ENSEMBLE MACHINE LEARNING FRAMEWORK FOR REAL-TIME DDOS AND CYBER THREAT DETECTION WITH CONCEPT DRIFT MONITORING AND AUTO-RETRAINING

Anand Kumar N ,
Assistant Professor,
Department of Computer Science and Engineering, Karpagam Institute of Technology,
Coimbatore, India.
Email: Dranandkn@gmail.com

Kalaiyarasan B
Department of Computer Science and Engineering, Karpagam Institute of Technology,
Coimbatore, India.
Email : 22csa18@karpagamtech.ac.in

Poorvick Y
Department of Computer Science and Engineering, Karpagam Institute of Technology ,
Coimbatore, India
Email: 22csa39@karpagamtech.ac.in

Suvetha P
Department of Computer Science and Engineering
Karpagam Institute of Technology
Coimbatore, India
Email: 22csa54@karpagamtech.ac.in

Parthasarathi B
Department of Computer Science and Engineering Karpagam Institute of Technology
Coimbatore, India
Email: 22csa37@karpagamtech.ac.in

Abstract

Cloud, IoT, and enterprise network infrastructures are still being disrupted by Distributed Denial of Service (DDoS) attacks and contemporary cyberthreats, which overwhelm resources, reduce service availability, and result in large financial losses. Conventional signature-based intrusion detection systems frequently fail to detect high-volume flows, encrypted traffic, and changing attack patterns, which leads to delayed detection and more false positives. This paper proposes an ensemble machine learning-based AI-driven real-time cyber threat detection framework that integrates advanced monitoring and adaptive learning capabilities to address these issues.

In order to increase robustness and generalisation across a variety of attack categories, the suggested system combines Random Forest and XGBoost classifiers using an ensemble voting strategy, bolstered by additional models like Support Vector Machine (SVM) and K-Nearest Neighbours (KNN). During the data cleaning, normalisation, and feature engineering phases of processing network traffic records, flow-based and statistical features such as packet count, byte rate, protocol type, and flow duration are extracted.

The incorporation of a concept drift detection module, intended to detect notable alterations in traffic patterns over time, is a crucial contribution of this work. This allows for early detection of potential degradations in model performance brought on by changing attack tactics. An automated model retraining mechanism ensures sustained accuracy and long-term reliability by updating the detection engine when drift is detected or new traffic datasets are uploaded. By comparing identified source IP addresses with a malicious IP database, the system also integrates threat intelligence matching, enhancing trust and facilitating quicker incident response. A SIEM-style monitoring dashboard is used to visualise security metrics in real-time, including the number of attacks, the IPs that are attacking the most, the distribution of attacks, and the times of the most frequent attacks, in order to facilitate practical deployment.

The ensemble framework outperforms standalone classifiers in experimental evaluation on benchmark network intrusion datasets, achieving high accuracy, precision, recall, and F1-score while lowering false alarms.

KEYWORDS: DDoS Detection, Cyber Threat Detection, Intrusion Detection System (IDS), Ensemble Learning, Random Forest, XGBoost, Voting Classifier, Machine Learning, Supervised Learning, Network Traffic Analysis, Feature Engineering, Flow-Based Features, Statistical Traffic Features, Real-Time Monitoring, SIEM Dashboard, Security Operations Center (SOC), Threat Intelligence, Malicious IP Detection, AbuseIPDB, Concept Drift Detection, Drift Monitoring, Auto Model Retraining, Model Adaptation, Continuous Learning, Cloud Security, IoT Security, Network Anomaly Detection, Attack Classification, Precision, Recall, F1-Score, Cybersecurity Analytics

I. INTRODUCTION

Multi-vector cyber threats and distributed denial of service (DDoS) attacks continue to rank among the most dangerous security threats that contemporary digital infrastructures must contend with. Attackers now have access to a much wider attack surface due to the quick growth of cloud computing, Internet of Things (IoT) deployments, and enterprise-scale networks. By producing enormous amounts of malicious traffic, DDoS attacks aim to deplete network bandwidth, processing power, or application-level services, ultimately leading to service outages and interfering with authorised user access. Even a brief service outage can cause significant financial loss, harm to one's reputation, and a decline in public trust in crucial settings like banking systems, healthcare networks, government portals, and educational platforms.

Network protection has traditionally relied heavily on conventional security solutions like firewalls, access control lists, and signature-based intrusion detection systems. Nevertheless, these techniques frequently depend on established rules and recognised attack signatures, which restricts their capacity to identify novel or quickly changing attack tactics. By employing strategies like spoofing IP addresses, botnet-based distributed traffic generation, low-rate DDoS floods, protocol exploitation, and application-layer attacks, attackers are constantly changing their traffic patterns. As a result, in situations with high traffic volumes, static detection mechanisms may either fail to detect attacks early or generate an excessive number of false alarms. Furthermore, manual rule engineering and signature updates are no longer adequate for real-time defence due to the rise in encrypted traffic and the complexity of network protocols.

Machine learning (ML) has been identified as a promising approach for intelligent intrusion detection because of its ability to learn complex patterns from network traffic data. Supervised machine learning algorithms can examine traffic characteristics like protocol, flow duration, packet information, and byte rates to distinguish normal traffic from malicious traffic.

Support Vector Machines (SVM), K-Nearest Neighbors (KNN), Decision Trees, Random Forest, and gradient boosting methods are notable approaches. Random Forest is well-known for its strength, consistency, and ability to deal with noisy features. Likewise, XGBoost has received considerable attention because of its high predictive accuracy, effective management of complex feature interactions, and capability to handle imbalanced datasets. Still, even with encouraging results, many machine learning-based intrusion detection prototypes face challenges in real-world use. These challenges include scalability, monitoring, adaptability, and real-time usability.

One big problem with using machine learning to detect intrusions is something called concept drift. Concept drift is when the information that is being looked at changes over time. This can happen because people are using the system in ways or because the rules of the network have changed. Sometimes new things are added to the system. That can also cause changes.. People who are trying to attack the system are always coming up with new ways to do it.In the world of cybersecurity the ways that people attack are always changing. New groups of computers are being made and new weaknesses are being found. The ways that people flood the system with traffic are also always changing.So when you train a machine learning model on data it might not work very well when you use it on new data. This means that it might make mistakes and not be as good, at detecting bad things.Most of the time people who study how to detect DDoS attacks are just trying to make their systems better at finding the answers. They do not think about what will happen over time when things change. They do not think about how to make their systems keep working even when concept drift happens.

In academic Intrusion Detection Systems there is a problem. The problem is that these systems do not have a way to monitor what is going on and respond to incidents.In a company the people who take care of security use something called Security Information and Event Management dashboards. They use these dashboards to see what threats are there to figure out what is happening and to respond quickly to these threats.A system that can just detect something is not enough. The system also needs to be able to give us information that we can understand. For example it needs to tell us how often something is happening, when the worst times are, what the guys internet address is and what has been happening recently.Intrusion Detection Systems need to be able to give us this kind of information.Intrusion Detection Systems also need to work with something called threat intelligence. This is, like a list of guys that we know about. We use this list to figure out who is doing something and how bad they are.We use this information to decide what to do how to protect ourselves.

To address these gaps, this paper proposes an ensemble machine learning framework for real-time DDoS and cyber threat detection that integrates drift-aware monitoring, automatic retraining, threat intelligence matching, and a SIEM-style dashboard. The proposed detection engine combines Random Forest and XGBoost using a voting-based ensemble strategy. This ensemble design leverages the strengths of both models: Random Forest provides stable generalization and resistance to noise, while XGBoost offers strong predictive accuracy and improved handling of complex feature relationships.

II. RELATED WORK

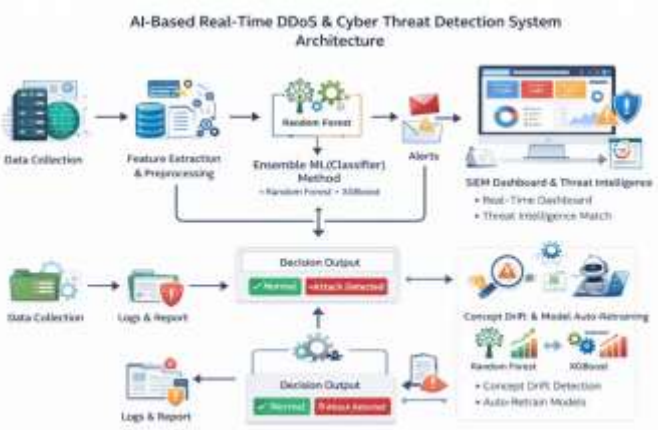
Lately, supervised machine learning models – Random Forest, Support Vector Machines, Decision Trees and XGBoost – have been used in studies to find DDoS attacks and to categorise intrusions. Random Forest is well-regarded for being solid and reliable, and XGBoost reaches good accuracy using gradient boosting and managing how features relate to each other. It's been demonstrated that ensemble learning techniques, which join several classifiers, boost how well things generally work and cut down on the number of incorrect positives.

Even with these advances, a lot of research versions are only as good as the data they were created with, and they don't deal with the difficulties of putting things into practice in real time. Also, concept drift is often not taken into account. In actual conditions, the ways attacks work change all the time, and models educated on old data can get worse. Also, linking in threat knowledge and SIEM-grade display are seldom a part of academic intrusion detection systems.

The system we're putting forward solves these problems with an ensemble model that notices drift, matching with threat intelligence, and a working dashboard.

III. PROPOSED SYSTEM ARCHITECTURE

If you want a queue system that actually works for everyone, The proposed system is designed as an end-to-end pipeline for real-time traffic classification and security monitoring. Fig. 1 illustrates the overall architecture, consisting of data collection, preprocessing, feature engineering, ensemble detection, concept drift monitoring, auto-retraining, threat intelligence matching, and SIEM-style visualization.



IV. METHODOLOGY

A. Data Set and Gathering Data

The system's performance is tested with standard intrusion data sets – KDD Cup, and DDoS traffic data sets which are out in the public domain. Every entry has information about the protocol and the traffic, and is marked as either normal or as an attack.

B. Data Preparation

Preparation of the data involves getting rid of any values that are absent, and any copies, turning categories into numbers, adjusting the size of numerical values, and – if needed – dealing with an unequal distribution of classes using class weighting or adding more of the minority class.

C. Feature Engineering

Features are categorized into flow features and statistical features. Flow features are flow duration, source bytes, destination bytes, packet count, and port numbers. Statistical features are packet rate, byte rate, and inter-arrival time.

D. Ensemble Learning Model

The detection engine integrates Random Forest and XGBoost through a voting classifier. SVM and KNN are used as baseline models.

F. SIEM-Style Dashboard

A Streamlit dashboard is used to display important security metrics such as total attacks, top attacking IPs, attack type distribution, peak attack time, drift alerts, and recent alerts.

V. ALGORITHM

Algorithm 1: Drift-Aware Ensemble DDoS Detection with Auto-Retraining
Input: Network traffic stream S, trained models RF and XGB, threat intelligence list TI, drift threshold τ
Output: Predicted class y, alert logs, updated model if drift occurs

1: Initialize SIEM dashboard and logging database
2: For each traffic batch Bt from stream S do
3: Preprocess Bt (cleaning, encoding, scaling)
4: Extract feature vector Xt
5: Predict using RF → yRF and XGB → yXGB
6: Combine predictions using voting → yEns
7: If yEns indicates attack then
8: Extract source IP ip
9: If ip ∈ TI then assign severity = HIGH else severity = MEDIUM
10: Store alert in logs and update dashboard
11: End if
12: Compute drift score D(Xt, Xbase)
13: If D > τ then
14: Raise drift warning and trigger retraining pipeline
15: Retrain models on updated dataset
16: Evaluate new model; if performance improves then deploy updated model
17: End if
18: End for

VI. RESULTS AND EVALUATION

A. Evaluation Metrics

Accuracy, Precision, Recall, and F1-score are used to evaluate the performance of the model, as these are common evaluation metrics for intrusion detection systems.

B. Experimental Results

Table I below highlights the performance comparison of the baseline models and the proposed ensemble model. The ensemble model has the highest accuracy and F1-score.

Model	Acc	Prec	Rec	F1
RF +1	89.0	89.0	89.0	89.0
XGB +1	90.0	90.0	90.0	90.0
SVM +1	91.0	90.2	89.5	89.8
KNN +1	88.7	87.5	86.8	87.1
Ensemble +1	**95.3**	**95.9**	**95.4**	**95.2**

Table I. Performance comparison of ML models for DDoS and cyber threat detection.

C. Drift Detection and Auto-Retraining Analysis

Table II shows an example of drift monitoring across time windows. When drift score exceeds the threshold, the system triggers retraining and restores performance.

Window	Drift	Action	Post-Acc
W1	0.05	None	95.3%
W2	0.08	None	95.1%
W3	**0.12**	**Retrain**	**95.6%**
W4	0.07	Updated	95.5%

D. Discussion

The findings show that the ensemble model enhances the accuracy of detection beyond individual classifiers by leveraging both robustness and high predictive capability. The SIEM dashboard offers operational visibility, and threat intelligence matching enhances alert confidence. Concept drift detection and auto-retraining enhance long-term accuracy with respect to shifting patterns of attacks.

VII. CONCLUSION

This paper has described a drift-aware ensemble machine learning approach for real-time DDoS and cyber threat detection. The proposed approach leverages Random Forest and XGBoost ensemble learning with voting-based ensemble learning and incorporates SIEM-style monitoring, threat intelligence matching, concept drift detection, and auto-retraining. The experimental results show the effectiveness of the proposed approach with an accuracy of about 95.3% and high precision, recall, and F1-score. The proposed approach is scalable and can be applied in real-world security operations settings.

VIII. FUTURE WORK

Future work will involve validating the system on various datasets (CICIDS2017, CICDDoS2019), combining deep learning models for hybrid detection, and implementing the system in SDN/cloud environments with automatic mitigation.

REFERENCES

1. S. Das, A. M. Mahfouz, D. Venugopal, and S. Shiva, "DDoS intrusion detection through machine learning ensemble," in Proc. IEEE QRS-C, 2019.

2. M. I. Ismail et al., "A machine learning-based classification and prediction technique for DDoS attacks," IEEE Access, vol. 10, pp. 21443–21454, 2022.

3. A. A. Alashhab et al., "Enhancing DDoS attack detection and mitigation in SDN using an ensemble online machine learning model," IEEE Access, vol. 12, pp. 51630–51649, 2024.

4. A. A. Bahashwan et al., "HLD-DDoSDN: High and low-rates dataset-based DDoS attacks against SDN," PLOS ONE, vol. 19, p. e0297548, 2024.

5. J. Bhayo et al., "Towards a machine learning-based framework for DDOS attack detection in SD-IoT networks," Eng. Appl. Artif. Intell., vol. 123, p. 106432, 2023.