BODHIVRUKSHA JOURNAL OF DIVERSE DISCIPLINE
E-ISSN : 3139-1486
An Open Access, Peer-Reviewed, Multidisciplinary Scholarly Journal
Bi-monthly | Multilingual (English, Hindi, Marathi) | Academic Research
@ Volume 2, Issue 5 (September-October 2026) (Special Issue 1) ISSN: 3139-1486 (Online)

AI-Powered Cybersecurity: A Review of Intelligent Threat Detection, Automated Blocking, and Incident Reporting

[1]Yash Vijendra Vadak, [2]Prof. Varsha Babar

[1]Student, TE-C (AI & DS), PRN: 72426518B, [2]Mentor, SPPU Technical Seminar ELC342CSD
[1]Dept. of Artificial Intelligence and Data Science
[1]Ajeenkya DY Patil School of Engineering, Pune, India

Abstract

Cyber attacks are increasing with many being very targeted and difficult to detect using signature based approaches. Signature based approaches are useful for detecting known threats but have weaknesses when trying to detect unknown threats like zero day attacks, mutated malware and anomalies. This paper explores the different types of AI and ML techniques that can be used for intelligent threat detection, blocking and reporting. A review of eleven AI/ML papers related to cybersecurity was undertaken with a range of different techniques including supervised classification for web attacks, tree based intrusion detection, ensemble techniques, DDoS detection, online Isolation Forest, deep learning for wireless and web intrusion detection, and general cybersecurity surveys. By reviewing the literature it has been found that using learning based approaches can increase flexibility within the system and provide real time monitoring. However there are limitations to these approaches when applied to real world cases including old and limited benchmark data sets, false positives, cost of models, class imbalance and poor integration between detection and blocking. This paper suggests an end-to-end concept framework for capturing and pre-processing telemetry data, using supervised classification, anomaly detection, assessing the risk severity, applying policies to contain the threat and creating an audit-able report of the incident. The framework has not been implemented or validated. The framework will be validated using up-to-date data sets, precision, recall, false positive rate, detection latency, throughput and response time. High impact actions will be required to be approved by humans. The framework will focus on using AI as part of a wider system to assist with decision making.

Keywords: Artificial intelligence, Machine learning, Cybersecurity, Threat detection, Intrusion detection system, Anomaly detection, Automatic blocking, Incident reporting, DDoS detection.

I. Introduction

Modern enterprise environments depend on cloud, web applications, wireless and endpoints meaning that there is more opportunities for attacks to be carried out. These attacks can be caused by malicious code such as malware, ransomware, phishing, DDoS, web attacks, credential abuse and unauthorised access causing disruption or leaking of sensitive data [2], [8], [9]. The attacks have evolved to use dynamic payloads, infrastructure, published exploits and realistic social engineering.

BODHIVRUKSHA JOURNAL OF DIVERSE DISCIPLINE
E-ISSN : 3139-1486
An Open Access, Peer-Reviewed, Multidisciplinary Scholarly Journal
Bi-monthly | Multilingual (English, Hindi, Marathi) | Academic Research
@ Volume 2, Issue 5 (September-October 2026) (Special Issue 1) ISSN: 3139-1486 (Online)

Traditional detection methods such as firewalls, antivirus and Intrusion Detection Systems (IDS) are vital for the system. When attacks match previously known attacks the detection methods offer the best protection. However, when new variants are used such as using low and slow, encryption or unusual combinations, the attacks can be difficult to detect. In addition, keeping the rules up to date can cause issues [3], [9].

AI and ML can be used to help detect. A model can be trained to identify statistical relationships between features in the traffic/events and known attack types. Anomaly detection can be used to identify anomalies [6], [9]. Deep learning can be used to find complex patterns in the data and ensemble methods can be used where multiple classifiers are used [4], [7], [8], [10]. Detection still needs to be done. Alerts need to be investigated, prioritised, safely managed and logged.

This review shows that 11 studies between 2019 and 2024 can be categorised into different detection methods and environments. Issues were identified such as having a representative data set, false positives, resource intensive and issues with integrating with response. A conceptual framework was also proposed to link detection to policy based containment and reporting. This is a design solution based on the literature reviewed and has not been implemented or tested.

The rest of this paper is organised as follows. Section II outlines the threat landscape and relevant ML techniques. Section III gives an overview of the selected studies. Section IV compares advantages and disadvantages of each study. Section V discusses research gaps. Section VI provides a unified framework and how it will be evaluated. Finally Section VII concludes with future work.

II. Background

A. Threat Landscape

There are numerous attacks which must be considered to detect threats. For web based attacks this may include injection based attacks, malformed requests and logical flaws. For network attacks it can include unwanted access, unusual connections and odd traffic patterns. DDoS attacks aim to saturate the bandwidth, connections or application. Wireless environments will have variation due to mobility of devices. Additional information can be obtained from endpoints and event logs to provide more information to identify suspicious sequences [2], [7], [8], [11].

It is rare to actually see the attack itself. Instead, you will receive information relating to the attack including packet/flow statistics, request rates, protocol fields, connection times, sequences of events, successful/unsuccessful authentications and resource changes. This depends on the quality of the features, how often they were sampled, if they are encrypted, any missing fields and how much variation there is in normal user behaviour. Therefore, it cannot be assumed that models will generalise between different environments.

B. Limits of Rule-Based Security

Signature based methods compare an event to known attack signatures. This can provide a good level of efficiency and ease of understanding, however if the signatures are not kept up to date or there are not enough distinguishing features between the different events then there can be a low coverage. Behavioural rules can help increase this coverage, however the strict threshold can cause a high false positive rate. Attackers can evade detection by using the same pattern used within a signature but varying the request

BODHIVRUKSHA JOURNAL OF DIVERSE DISCIPLINE
E-ISSN : 3139-1486
An Open Access, Peer-Reviewed, Multidisciplinary Scholarly Journal
Bi-monthly | Multilingual (English, Hindi, Marathi) | Academic Research
@ Volume 2, Issue 5 (September-October 2026) (Special Issue 1) ISSN: 3139-1486 (Online)

order, time, source address or encoding within the payload. These problems are still faced when using ML, but the way the data is represented will be different and this will affect the decision boundary.

C. AI/ML Approaches

Supervised classification uses labelled examples to learn a mapping from features to benign or malicious classes. Its performance depends on label quality, the range of attacks in training, and the degree to which operational data resembles the training distribution. Tree-based models can provide understandable decision paths and are often useful where analysts need interpretable reasons for an alert [5].

Ensemble learning combines multiple estimators to reduce dependence on any single model. It may improve robustness, but inference cost and operational complexity can increase [4]. Anomaly detection instead models normality or isolates unusual observations; it can be valuable when labelled examples of emerging attacks are scarce. Online Isolation Forest is one approach reported for low-latency SD-WAN traffic monitoring [6]. Deep neural networks can learn complex feature interactions but may require substantial training data, hardware, tuning, and monitoring [7], [8], [10], [11].

D. Detection, Response, and Reporting

Once detected you'll have a score/category/incident. How this is responded to can vary. If it's a low confidence incident it can be logged to investigate. High confidence high severity incidents can lead to rate limiting, blocking and quarantining. Automated actions should be reversible, only affect the asset that was impacted and be policy driven. When creating an incident you get information such as time of detection, evidence, systems impacted, confidence score, severity, action taken and results. This can aid with triages, auditing, threat hunting and help improve your models in the future.

E. Operational Requirements

A deployable AI security component should meet requirements beyond raw classification accuracy. It should handle expected traffic volume, make decisions within an acceptable latency budget, expose useful explanations, protect telemetry, and fail safely when the model or data pipeline is unavailable. It should also be monitored for data drift and changing false-positive rates. A technically accurate model may still be unsuitable if it blocks legitimate users or consumes more resources than the protected system can spare.

III. Literature Review

The selected research has a progression from classical ML classification techniques to ensemble, online anomaly detection and Deep Learning. Each has a different focus on threat types, network infrastructure requirements, testing environment and ease of implementation in an overall workflow.

A. DDoS Mitigation and Traffic Classification

Zhao et al. [2] uses ML to classify and shape traffic using in network mitigation for DDoS attacks. This work has a key finding of moving from just detection in traffic to network mitigation. This could help reduce time between detection and mitigation but requires a network infrastructure that can be programmed and the policies must be implemented carefully. Classification could lead to false positive issues where legitimate bursts are slowed down. This method would not necessarily be implementable in networks which cannot be controlled.

BODHIVRUKSHA JOURNAL OF DIVERSE DISCIPLINE
E-ISSN : 3139-1486
An Open Access, Peer-Reviewed, Multidisciplinary Scholarly Journal
Bi-monthly | Multilingual (English, Hindi, Marathi) | Academic Research
@ Volume 2, Issue 5 (September-October 2026) (Special Issue 1) ISSN: 3139-1486 (Online)

Zhang et al. [6] uses an online version of Isolation Forest for detecting malicious traffic in an SD-WAN network. The online aspect is useful for situations where traffic changes over time and decisions need to be made quickly. By using anomaly detection, less attack data is required with complete labels but anomalies do not always mean there is malicious intent. Anomalies can arise from changes in application, maintenance periods, new devices or out of the ordinary business activity. This technique would be specific to SD-WAN networks and would need to be looked at further to see if it can implemented in other network architectures.

B. Classical and Ensemble Machine Learning

This work by Abdou Vadhil et al. [1] uses supervised ML classifiers to detect web attacks. This shows how ML classification can be used to detect common malicious traffic in web attacks. The work is focussed on application traffic specifically. If the detector is trained on web traffic it may not detect DDoS behaviour or detect suspicious events at endpoints.

Nath et al. [5] looks at using tree based algorithms for detecting intrusion. Tree based models can be very interpretable showing how different features contribute to the prediction. However there can be issues when trying to predict in highly imbalanced classes. Rare attack types may not be present enough in the training set meaning accuracy could be good but recall may not be so good for those specific attacks that we would want to detect.

Hossain and Islam [4] look at using ensemble techniques for intrusion detection. Combining multiple models can increase robustness if they each make different mistakes but comes at extra cost in terms of computation, memory and managing multiple models. If using the model in real time, the increase in performance would need to be considered against how much traffic the system can process in the given time.

Yadav and Ningshen [3] discusses improving ML based IDS with feature selection and preprocessing. Using feature selection can help reduce complexity of the training and prediction by removing irrelevant or duplicated features. Preprocessing can also help reduce noise in the data that could cause false positives. The drawback of this paper is that they used restricted or old benchmark data sets. If the features were learned from a small data set they may not be optimal when traffic changes or attack techniques change.

C. Deep Learning for Web and Wireless Intrusion

Yang et al. [7] propose using Deep Learning to detect intrusions in real time in wireless networks. Using ML to learn traffic patterns can help find complex relationships in the data which are hard to specify using manual rules. However training and prediction can be expensive and may not be suited to low resource edge devices. There can also be legitimate variation in wireless networks meaning the model must be able to differentiate between the changes and hostile intrusions.

Kim et al. [8] propose using AI-IDS to detect web intrusions in real time using Deep Learning. This supports the use of ML to learn patterns in web traffic but is very specific to web attacks. Generalisation, computational expense and continuous validation would need to be considered.

Ravi et al. [10] look at using Deep Learning for intelligent intrusion detection and classifying network intrusion. This shows how Neural techniques can be used to model complex patterns in events and traffic. However Deep Learning models usually require representative training data and can be computationally

BODHIVRUKSHA JOURNAL OF DIVERSE DISCIPLINE
E-ISSN : 3139-1486
An Open Access, Peer-Reviewed, Multidisciplinary Scholarly Journal
Bi-monthly | Multilingual (English, Hindi, Marathi) | Academic Research
@ Volume 2, Issue 5 (September-October 2026) (Special Issue 1) ISSN: 3139-1486 (Online)

expensive. Just because a high performance was reported in one experimental scenario doesn't mean it would be the same in different traffic scenarios.

Lee et al. [11] uses Artificial Neural Networks to detect suspicious events based on event profiles. Using event profiles gives more context to the events than individual packets could. However the reliability of the event profiles and generalisation to new threats is an issue. Often event profiles will not be complete, have inconsistent times and can differ between systems.

D. General Survey

Shaukat et al. [9] have produced a survey of ML techniques used in Cybersecurity over the last decade. This shows how varied the techniques can be. It also highlights how the results would depend on what data sets, features and testing were used. This would not guarantee that an algorithm would be the best choice for a new scenario but would help to understand what tradeoffs need to be made.

E. Summary of Literature

None of the techniques discussed in the literature can solve all the requirements for an operational system. Supervised techniques can have good performance on classes that are well represented. Anomaly detection can help identify unknown behaviour. Tree based models can be easier to explain. Ensemble techniques can increase robustness. Deep Learning can help with complex patterns. However there can be issues with bias in the data sets, imbalanced classes, infrastructure requirements and resource limitations. Most of the work discussed is focused on detection or a specific mitigation technique and not looking to cover the whole life cycle including decision making for responses and consistent reporting.

IV. Comparative Analysis

Table I summarizes the reviewed studies in terms of method, reported contribution, and key limitation. The comparison is qualitative and follows the descriptions in the source paper; it should not be interpreted as a controlled benchmark because the studies use different tasks, datasets, metrics, and environments.

Table I: Qualitative Comparison of the Reviewed Studies

Study	Method / focus	Contribution described	Limitation described
Abdou Vadhil et al. [1], 2024	Supervised ML for web attacks	Detects common web-attack traffic	Web-focused; limited coverage of network-level threats
Zhao et al. [2], 2024	ML-driven in-network traffic shaping	DDoS mitigation through traffic adjustment	Needs programmable network infrastructure
Yadav and Ningshen [3], 2023	ML IDS with feature selection and preprocessing	Improved classification and fewer false alarms	Limited or older benchmark datasets
Hossain and Islam [4], 2023	Ensemble-based ML IDS	Robustness through model combination	Higher computational cost

BODHIVRUKSHA JOURNAL OF DIVERSE DISCIPLINE
E-ISSN : 3139-1486
An Open Access, Peer-Reviewed, Multidisciplinary Scholarly Journal
Bi-monthly | Multilingual (English, Hindi, Marathi) | Academic Research
@ Volume 2, Issue 5 (September-October 2026) (Special Issue 1) ISSN: 3139-1486 (Online)

Study	Method / focus	Contribution described	Limitation described
Nath et al. [5], 2023	Tree-based ML	Accuracy with interpretable rules	Imbalanced attack classes
Zhang et al. [6], 2023	Online Isolation Forest over SD-WAN	Low-latency anomaly detection	SD-WAN-specific assumptions
Yang et al. [7], 2020	Deep learning for wireless IDS	Real-time wireless intrusion detection	Training and compute overhead
Kim et al. [8], 2020	Deep learning for web IDS	Real-time web intrusion detection	Web scope and generalization concerns
Shaukat et al. [9], 2020	Cybersecurity ML survey	Synthesizes ML techniques and use cases	Findings depend on underlying studies
Ravi et al. [10], 2019	Deep-learning IDS	Intrusion identification and classification	Data and compute requirements
Lee et al. [11], 2019	ANN on cyber-event profiles	Identifies suspicious event patterns	Profile quality and generalization

A common comparison should consider more than accuracy. Precision indicates how many alerts are correct, while recall indicates how many actual malicious events are detected. The false-positive rate estimates how often benign activity is incorrectly flagged. Latency and throughput determine whether a model can keep up with operational traffic. Resource use affects deployment on gateways and edge devices. Finally, interpretability and actionability determine whether an alert can be investigated and converted into a safe response.

The literature also shows a distinction between model performance and system-level effectiveness. A classifier may identify an attack but fail to reduce its impact if the alert reaches an analyst too late. Conversely, automated blocking can reduce response delay but amplify the consequences of a false positive. Therefore, a practical design needs a calibrated decision policy, an auditable action mechanism, and a recovery path alongside the detection model.

V. Research Gaps and Challenges

A. Data Set Currency and Representativeness

Many techniques utilise data sets which may not reflect current technologies, applications, protocols and attackers [3]. As new versions of the software are released, there may be new normal traffic patterns and attacks meaning the data becomes outdated. There may also be issues with balancing classes where there are more benign events than attack events. The attack events may also be rare meaning they only occur a few times. In future evaluations details should be provided about the source and date of the data. Details should also be provided about the classes of data and what features were available. The data should be split using time or environment based splits to improve generalisation.

BODHIVRUKSHA JOURNAL OF DIVERSE DISCIPLINE
E-ISSN : 3139-1486
An Open Access, Peer-Reviewed, Multidisciplinary Scholarly Journal
Bi-monthly | Multilingual (English, Hindi, Marathi) | Academic Research
@ Volume 2, Issue 5 (September-October 2026) (Special Issue 1) ISSN: 3139-1486 (Online)

B. False Positives and Class Imbalance

Anomaly detection aims to flag when anomalous events occur. However this can cause false positives where the data changes legitimately [6]. This can cause the analyst to waste time and cause distrust of automated systems. The accuracy may be high however the recall may be low for the under represented classes. Accuracy should be provided per class. Confusion matrices, false positive rate and precision and recall should be provided. Thresholds should be tuned using a validation set.

C. Computational Overhead

Deep learning models can have a large computational overhead and cost to train and store [4], [7], [10]. Many decisions in security have to be made at the edge of the network meaning there are limited resources. The overall budget of the system should be considered when choosing a model. Dimensionality reduction techniques, model compression, sampling, multi-stage detection and complex models being used selectively should be experimented with.

D. Escalation

Many techniques focus on the classification aspect. Details should be provided about how to escalate the alert. Automatically blocking may cause additional problems. If there is a false positive then it may cause disruption to a legitimate service. Attackers may also try to trick the model into blocking the victim. A response should be provided. Thresholds need to be defined for different levels, actions that can be taken, time before expiry, rollback and what needs to be approved. High impact actions should still be manually verified unless the system has been thoroughly validated and it is allowed by policy.

E. Generalisation, Explainability and Drift

The model may not generalise well to new environments. The model should be explainable to provide a concise explanation for why the decision was made. Monitoring should be in place to detect drift in the data. If drift occurs investigations should be carried out or it should be controlled with re-training. Training data, model version, thresholds and response policy version should be noted.

VI. Proposed Unified Framework

An overview of how an AI-based threat detection framework can be designed. Below describes the architecture and processes.

A. Architecture & Process Flow

1) Data collection and preprocessing

Collect data from various sources such as network flow summaries, web request metadata, authentication logs, endpoint logs, relevant telemetry. Normalise the data to ensure the timestamps are consistent. Fill in missing values and remove duplicates. Extract features to ensure no unnecessary sensitive data is being collected. Ensure the data is of a good quality.

2) Threat Detection

Use a lightweight supervised learning approach to detect known attack patterns. Perform anomaly detection to determine anomalies within the data. This can then be combined with deterministic rules and reputation

BODHIVRUKSHA JOURNAL OF DIVERSE DISCIPLINE
E-ISSN : 3139-1486
An Open Access, Peer-Reviewed, Multidisciplinary Scholarly Journal
Bi-monthly | Multilingual (English, Hindi, Marathi) | Academic Research
@ Volume 2, Issue 5 (September-October 2026) (Special Issue 1) ISSN: 3139-1486 (Online)

scores. There's no set way of choosing the model, this depends on the threat types, available telemetry and computational constraints. Output confidence and anomaly scores. Include features used to trigger the alert.

3) Risk scoring

Consider confidence within the model, how impactful the threat is likely to be, criticality of assets, contextual information and evidence. Decide how severe the threat should be classified (informational / low / medium / high / critical). The risk score should not be based purely on the model output. For example if the anomaly occurred on a low risk test host it may simply be logged. If there are many malicious requests to a public service it may be throttled.

4) Policy Based Automated Mitigation

Decide what can be done to mitigate the threat. This depends on the severity and confidence. Examples are logging, monitoring, throttling, blocking the source and terminating sessions. Include why these actions were taken, severity of action, duration, how to rollback. Any actions with high impact should require analyst approval. If there is no telemetry available or the model cannot be used due to low confidence then the system should adopt a cautious approach.

5) Incidents

Every alert should be logged in detail. This should include a unique ID, date/time, asset affected, type of event, model version, confidence, severity, summary of evidence, policy which was applied, action taken, duration and resolution. Analysts may add notes to the event such as true/false, outcome and comments. This will aid with evaluation in the future. However it should not automatically be used to retrain models in production without quality checks and approvals.

B. Policy

Create a sensible policy. Low confidence or low impact can just be logged and correlated. Medium confidence can be further monitored, alerted on or throttled depending on the risk. These actions will be rolled back. High confidence and high impact can be mitigated temporarily. Include clear permissions which can be rolled back. Critical actions or high model uncertainty, asset criticality and business impact should be reviewed by a human analyst. This ensures a balance between speed and errors.

C. Incident Logging

Ensure the incident logging is concise to help with triage but includes enough detail for auditing. Incident ID, date/time of event, host/account/service affected, indicators, model prediction and confidence, reason for severity, alerts, action taken, policy, start time and end time. Don't include too much sensitive information or mask. Ensure all times are consistent and include links to protected evidence.

D. Evaluation Plan

Evaluate each section of the framework. Offline: Compare different models. Use supervised, tree based, ensemble and anomaly detection models. Ensure the data is split correctly so there is no data leakage. Note class distribution. If possible split the data over time to account for changes in traffic. Compare against simple models and current rules. Compare results using precision, recall, F1-score, false positive rate,

BODHIVRUKSHA JOURNAL OF DIVERSE DISCIPLINE
E-ISSN : 3139-1486
An Open Access, Peer-Reviewed, Multidisciplinary Scholarly Journal
Bi-monthly | Multilingual (English, Hindi, Marathi) | Academic Research
@ Volume 2, Issue 5 (September-October 2026) (Special Issue 1) ISSN: 3139-1486 (Online)

breakdown by class and confusion matrices. Include performance metrics such as latency, events per second, memory usage, CPU usage and detection to action time.

Mitigation: Test to see if mitigation has reduced the impact of attacks without affecting any legitimate traffic. Incident Logging: Consider completeness, traceable and analyst understanding / resolution time.

Shadow Experiment: Allows analysts to understand the alerts that have been raised and what has actually happened. It will also help spot any repeated false positives. After enough validation has been done a few reversible actions can be enabled for specific situations. It is important to implement this incrementally. Include a dashboard to monitor, kill switch and rollback. No numerical results have been included as the framework has not yet been implemented or tested.

E. Security

Ensure your detection model and data pipeline are secure. Attackers may attempt to poison the training data, manipulate features, cause too many alerts or attack the model. Secure your model files, feature pipelines, thresholds and mitigation policies. Log all changes. Ensure everything is versioned, verified and reversible. Mitigation controller should be independent to ensure allowlisting and safety rules are in place. This means the model can't perform any random commands based on its predictions or cause the block to go beyond what it is allowed.

F. Benefits & Tradeoffs

This framework could improve response times to alerts. As there are no unnecessary handovers for low risk well understood events. The structured logging makes auditing easier and helps spot patterns. Using supervised learning and anomalies could give broad coverage. These are theories which would need to be tested. More integration may be needed, false positives contained, model drift, telemetry stored and policies and evaluation methods maintained.

VII. Conclusion and Future Work

Eleven AI/ML papers were reviewed which were published between 2019 and 2024. These include examples of learning-based solutions for detecting cyber-threats using supervised learning for classifying web attacks, tree-based and ensemble intrusion detection, DDoS traffic shaping, online Isolation Forest, deep learning for wireless and web environments, event profiling and a general overview of machine learning approaches. This shows how learning-based solutions can be used to provide flexible detection mechanisms. However, they must be able to use realistic data, produce accurate results, be computationally efficient and have few false positives to be useful.

A key issue found was the lack of integration between threat detection and response. This has been resolved with a unified framework that includes data preprocessing, multiple methods for detection, threat scoring, containment using policies and structured reporting. It includes elements such as being able to undo processes, having clear thresholds, having human involvement for high impact decisions, and ensuring privacy of telemetry. This is purely theoretical with no experimental results.

In future work, a prototype should be created. This should be tested using modern and varied datasets with temporal testing and imbalanced attack classes. This should include testing lightweight and more complex models, recording detection and system level metrics. It should also test if automation can reduce the time

BODHIVRUKSHA JOURNAL OF DIVERSE DISCIPLINE
E-ISSN : 3139-1486
An Open Access, Peer-Reviewed, Multidisciplinary Scholarly Journal
Bi-monthly | Multilingual (English, Hindi, Marathi) | Academic Research
@ Volume 2, Issue 5 (September-October 2026) (Special Issue 1) ISSN: 3139-1486 (Online)

taken to respond to threats without increasing service interruption. Further work should be done to account for new social engineering threats, modelling drift, adversarial attacks and privacy of telemetry. A successful solution will not just detect malicious events but also help defenders to accurately, transparently and safely respond.

References

[1] F. Abdou Vadhil, M. Lemine Salihi, and M. Farouk Nanne, "Machine learning-based intrusion detection system for detecting web attacks," *IAES International Journal of Artificial Intelligence*, vol. 13, no. 1, 2024.

[2] Z. Zhao, Z. Liu, H. Chen, F. Zhang, Z. Song, and Z. Li, "Effective DDoS mitigation via ML-driven in-network traffic shaping," *IEEE Transactions on Dependable and Secure Computing*, vol. 21, no. 4, pp. 4271–4289, 2024.

[3] M. K. Yadav and M. Ningshen, "Enhancement of intrusion detection system using machine learning," *International Journal of Engineering Research & Technology*, vol. 12, no. 1, 2023.

[4] M. A. Hossain and M. S. Islam, "Ensuring network security with a robust intrusion detection system using ensemble-based machine learning," *Array*, vol. 19, 2023.

[5] S. Nath, D. Pal, and S. Chimphlee, "Intrusion detection system (IDS) development using tree-based machine learning algorithms," *International Journal of Computer Networks and Communications*, vol. 15, no. 4, 2023.

[6] P. Zhang, F. He, H. Zhang, J. Hu, X. Huang, J. Wang, X. Yin, H. Zhu, and Y. Li, "Real-time malicious traffic detection with online Isolation Forest over SD-WAN," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 2076–2090, 2023.

[7] L. Yang, J. Li, L. Yin, Z. Sun, Y. Zhao, and Z. Li, "Real-time intrusion detection in wireless network: A deep learning-based intelligent mechanism," *IEEE Access*, vol. 8, pp. 170128–170139, 2020.

[8] A. Kim, M. Park, and D. H. Lee, "AI-IDS: Application of deep learning to real-time web intrusion detection," *IEEE Access*, vol. 8, pp. 70245–70261, 2020, doi: 10.1109/ACCESS.2020.2986882.

[9] K. Shaukat, S. Luo, V. Varadharajan, I. A. Hameed, and M. Xu, "A survey on machine learning techniques for cyber security in the last decade," *IEEE Access*, vol. 8, pp. 222310–222354, 2020, doi: 10.1109/ACCESS.2020.3041951.

[10] V. Ravi, M. Alazab, S. Kp, P. Poornachandran, and A. M. Al-Nemrat, "Deep learning approach for intelligent intrusion detection system," *IEEE Access*, vol. 7, pp. 41525–41550, 2019, doi: 10.1109/ACCESS.2019.2895334.

[11] J. Lee, J. Kim, I. Kim, and K. Han, "Cyber threat detection based on artificial neural networks using event profiles," *IEEE Access*, vol. 7, pp. 165607–165626, 2019, doi: 10.1109/ACCESS.2019.2953095.