IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 5, October 2026
Impact Factor: 8.2

Advanced AI-Powered Intrusion Detection System for Cybersecurity and Network Protection

Pawar Sandhya Nanasaheb[1], Prof. P. T. Dhotre[2], Prof. G. R. Sarode[3], Dr. H. B. Jadhav[4]
Department of Computer Engineering[1,2,3,4]
Adusul Technical Campus, Chas, India

Abstract: *Modern computer networks face a wide range of threats, including denial-of-service attacks, port scanning, brute-force attempts, botnet activity, and previously unseen malicious behaviour. Traditional signature-based intrusion detection systems can identify known patterns but may struggle when attack behaviour changes or has no existing signature. This paper reviews how Artificial Intelligence (AI), machine learning (ML), and deep learning can strengthen intrusion detection and network protection. It presents a conceptual layered framework combining network-flow collection, data preparation, AI-based classification, anomaly detection, explainable alerts, and human-supervised response. The paper discusses commonly used approaches such as decision trees, random forests, support vector machines, neural networks, and anomaly-detection models, alongside evaluation measures including precision, recall, F1-score, false-positive rate, and detection latency. Public benchmark datasets such as CICIDS2017 can support controlled evaluation, but their limitations mean that results must not be assumed to transfer directly to live networks. This is a literature-based conceptual study; it does not claim to have trained or experimentally validated a new model. The paper concludes that an effective AI-powered IDS requires reliable data, careful validation, privacy protection, analyst oversight, and continuous monitoring.*

Keywords: Artificial Intelligence, Intrusion Detection System, Cybersecurity, Network Security, Machine Learning, Anomaly Detection, Deep Learning, Threat Detection

I. INTRODUCTION

Digital services, cloud platforms, mobile devices, and connected systems have increased the amount and complexity of network traffic. This growth also creates opportunities for attackers to exploit weak configurations, stolen credentials, vulnerable applications, and exposed services. An Intrusion Detection System (IDS) monitors network or host activity and identifies events that may indicate a security incident. An IDS can generate alerts for investigation; an Intrusion Prevention System (IPS) may additionally take configured actions to block or limit traffic [1].

Conventional signature-based detection compares observed activity with known attack patterns. It can be efficient for recognized threats, but signatures must be updated and may not identify new variants. Anomaly-based detection instead learns or defines expected behaviour and flags deviations. AI and ML can help analyse many traffic features and uncover relationships that are difficult to express as fixed rules. However, AI is not a complete replacement for established security controls: inaccurate data, changing traffic, adversarial manipulation, and excessive false alarms can reduce its value [2], [3].

This paper outlines a practical conceptual design for an AI-powered IDS intended to support cybersecurity and network protection. It focuses on architecture, model choices, evaluation, deployment considerations, and research limitations rather than presenting unsupported experimental results.

Copyright to IJARSCT
www.ijarsct.co.in
DOI: 10.48175/IJARSCT-40150

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 5, October 2026
Impact Factor: 8.2

II. OBJECTIVES AND SCOPE

A. Objectives

- To explain the role of AI and machine learning in network intrusion detection.
- To describe a layered architecture for collecting, processing, analysing, and responding to network events.
- To compare suitable detection approaches and evaluation metrics at a conceptual level.
- To identify implementation challenges, including false alarms, data imbalance, privacy, model drift, and explainability.
- To recommend a responsible evaluation and deployment workflow.

B. Scope and Approach

This paper is a narrative, literature-based review and conceptual synthesis. It draws on established guidance for intrusion detection and published work on machine-learning-based detection. No new network traffic was collected, no model was trained, and no performance results are claimed. Any future implementation should be tested using documented datasets and, where possible, representative local traffic in a controlled environment.

III. INTRUSION DETECTION AND AI METHODS

A. Types of Intrusion Detection

Network-based IDS (NIDS) examines traffic moving across network segments. Host-based IDS (HIDS) monitors events on individual devices, such as system logs, process activity, and file changes. Wireless IDS focuses on wireless network behaviour, while network behaviour analysis looks for unusual traffic patterns. These approaches can complement each other because a single monitoring point may not provide enough context to identify every incident [1].

B. Signature-Based and Anomaly-Based Detection

Signature-based detection matches traffic or events against known rules and indicators. It is useful for well-understood attacks and often produces interpretable alerts, but it depends on timely signature updates. Anomaly-based detection establishes a baseline of normal or expected behaviour and flags deviations. It may identify unfamiliar activity, but legitimate changes—such as a sudden workload increase—can also appear anomalous. A hybrid system can combine rules, supervised classification, and anomaly scoring to balance coverage and precision.

C. Machine Learning and Deep Learning

Decision trees and random forests can model non-linear relationships in flow features and provide feature-importance estimates. Support vector machines can separate classes in feature space, while k-nearest neighbours classifies activity based on similar examples. Neural networks can learn more complex patterns from larger datasets, and sequence models may be considered when order and timing matter. Unsupervised methods can help identify unusual events where labels are limited. Model selection should consider accuracy, inference speed, interpretability, computing cost, and the quality of the available labels—not complexity alone [2], [3].

IV. PROPOSED AI-POWERED IDS FRAMEWORK

The proposed framework contains seven connected layers. Security policy, access control, logging, and human oversight apply across the entire pipeline rather than only at the response stage.

1. NETWORK DATA COLLECTION	Flow records, packet metadata, DNS, firewall and system logs
2. DATA PREPARATION	Remove duplicates, handle missing values, normalize features, label and document data

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 5, October 2026
Impact Factor: 8.2

3. FEATURE ENGINEERING	Flow duration, packet counts, byte rates, protocol, connection frequency and timing
4. AI DETECTION ENGINE	Known-attack classifier + anomaly score + rule-based checks
5. ALERT PRIORITIZATION	Risk score, confidence, affected asset, event context and explanation
6. ANALYST REVIEW / RESPONSE	Investigate alert; apply approved containment or blocking policy when justified
7. FEEDBACK AND MONITORING	Review false alarms, detect drift, update models, audit actions and evaluate outcomes

Fig. 1. Proposed conceptual architecture for an AI-powered intrusion detection system.

A. Network Data Collection and Preparation

The system collects approved metadata from network flows and security logs. Full packet capture may contain sensitive information, so collection should follow organizational policy and data-minimization principles. Preparation includes checking data types, removing duplicate records, handling missing or infinite values, encoding categorical fields, and scaling features where required. Data splitting must avoid leakage: near-duplicate flows or records from the same event should not unintentionally appear in both training and test sets.

B. Feature Engineering and AI Detection

Potential features include connection duration, packets and bytes transferred, packet-length statistics, protocol, connection rate, failed connection counts, and time-based behaviour. The system can use a supervised model to classify known categories and an anomaly detector to flag unusual activity. A rules engine can add context from threat intelligence or local policy. The combination should be evaluated as a whole because multiple independent alerts may refer to the same underlying event.

C. Explainable Alerts and Response

An alert should contain the event time, affected asset, relevant features, predicted class, confidence or anomaly score, and a short reason for prioritization where possible. The response layer may recommend investigation, increase monitoring, or request a policy-approved containment action. High-impact actions should require safeguards such as confidence thresholds, allow-lists, rollback procedures, and analyst approval. Automatic blocking without adequate validation can interrupt legitimate services.

V. DATASETS, TRAINING AND EVALUATION

A. Public Datasets

CICIDS2017 is a commonly used labelled intrusion-detection dataset containing benign traffic and multiple attack categories, including denial-of-service, brute-force, web attacks, and botnet activity [4]. It can support reproducible experiments, but it reflects a particular collection environment and period. Dataset labels, class imbalance, duplicated or unrealistic patterns, and differences between benchmark and operational traffic can affect results. A second dataset or a time-separated test set can help assess generalization.

B. Training Workflow

A responsible workflow defines the threat classes and intended deployment environment first. Next, it inspects label quality, partitions data into training, validation, and test sets, and applies preprocessing only using training-set information. Candidate models are trained and tuned on the validation data. The final test set is used once for an unbiased estimate, and the full pipeline—including preprocessing, thresholds, and alert grouping—is documented. Testing should include benign traffic from different periods and, where feasible, external datasets.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 5, October 2026
Impact Factor: 8.2

C. Performance Metrics

Accuracy measures the overall proportion of correct predictions, but can be misleading when attack classes are rare. Precision measures how many predicted alerts are truly malicious; recall measures how many malicious events are detected. The F1-score balances precision and recall. False-positive rate matters because too many incorrect alerts can overwhelm analysts, while false negatives represent missed threats. Detection latency, throughput, memory use, and model-update cost are also important for near-real-time use. Results should be reported per attack class and with the test protocol clearly stated.

Metric	Purpose
Precision	Indicates how many raised alerts are correct.
Recall / Detection Rate	Indicates how many actual attacks are detected.
F1-score	Balances precision and recall.
False-positive rate	Measures benign events incorrectly flagged as attacks.
Detection latency / throughput	Measures timeliness and ability to handle traffic volume.

VI. CHALLENGES AND MITIGATION STRATEGIES

A. False Positives and Class Imbalance

Attack events are often much less frequent than normal traffic. A model may therefore appear accurate while missing rare but important attacks. Threshold tuning, class-aware metrics, careful sampling, and per-class reporting can provide a clearer picture. Alert grouping and severity levels can reduce repeated notifications, but should not hide distinct incidents.

B. Evolving and Adversarial Behaviour

Network patterns change as applications, users, devices, and attacker techniques evolve. This model drift can lower detection quality after deployment. Monitoring should track feature distributions, alert rates, and analyst-confirmed outcomes. Retraining must use reviewed data and a controlled release process. AI models may also be manipulated by crafted inputs or poisoned training data, so access to training pipelines and model artifacts must be protected.

C. Explainability and Trust

Security analysts need enough context to decide whether an alert is actionable. Feature importance, local explanations, matched rules, and examples of similar events can help, but explanations should not be treated as proof that a prediction is correct. Model versions, thresholds, data sources, and response actions should be logged so decisions can be audited.

D. Privacy, Security and Governance

Traffic records can reveal user behaviour, device identities, and business-sensitive information. The system should collect only what is necessary, restrict access by role, protect data in transit and at rest, define retention periods, and maintain audit logs. The model service itself must be secured against unauthorized access, tampering, and misuse. Deployment should follow applicable laws and organizational security policies.

E. Integration, Cost and Operational Skills

A technically strong model may fail if it cannot integrate with existing firewalls, SIEM platforms, ticketing systems, and incident-response procedures. Organizations should begin with a limited pilot, define ownership and escalation rules, train analysts, and measure operational benefits. Total cost includes data pipelines, compute, storage, maintenance, staff time, and incident handling—not only model development.

VII. DISCUSSION AND FUTURE DIRECTIONS

AI can extend conventional intrusion detection by combining known-threat classification with anomaly detection and prioritization. The strongest design is likely to be hybrid: deterministic rules remain useful for clear policies, while ML

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 5, October 2026
Impact Factor: 8.2

supports pattern recognition across high-volume data. The value of the system depends not only on classification metrics but also on whether alerts arrive in time, are understandable, and lead to appropriate action.

Future work may explore privacy-preserving collaborative learning, graph-based analysis of relationships between hosts, sequence models for time-dependent traffic, and robust detection under distribution shift. Explainable AI can assist analyst review, while automated response can be introduced gradually under strict safeguards. Any claimed improvement should be demonstrated through a reproducible experiment, transparent baseline comparisons, independent testing, and operational evaluation.

VIII. LIMITATIONS

This paper proposes a conceptual framework based on published literature and guidance. It does not present a new algorithm, implementation, live-network deployment, or experimental accuracy values. The suggested architecture and evaluation workflow are recommendations rather than validated outcomes. Actual effectiveness will depend on the organization, traffic patterns, attack coverage, data quality, and response processes. A future study should implement the framework and compare candidate models on documented datasets and a carefully governed test environment.

IX. CONCLUSION

An advanced AI-powered intrusion detection system can support cybersecurity by analysing network behaviour, identifying suspicious activity, and helping security teams prioritize incidents. Machine learning and deep learning offer useful tools, but they also introduce risks involving false alarms, data bias, model drift, explainability, and adversarial manipulation. A layered architecture that combines data preparation, hybrid detection, contextual alerts, supervised response, and continuous monitoring provides a practical starting point. Reliable deployment requires secure data handling, representative testing, transparent evaluation, and human oversight. Future implementation should report measured results honestly and verify performance in environments that reflect intended use.

ACKNOWLEDGMENT

The author expresses sincere gratitude to the faculty and institution for their guidance, encouragement, and academic support during the preparation of this research paper. Appreciation is extended to Prof. P. T. Dhotre, Prof. G. R. Sarode, and Dr. H. B. Jadhav for their support and valuable direction.

REFERENCES

1. K. Scarfone and P. Mell, “Guide to Intrusion Detection and Prevention Systems (IDPS),” NIST Special Publication 800-94, National Institute of Standards and Technology, 2007. doi: 10.6028/NIST.SP.800-94.
2. “Benchmarking of Machine Learning for Anomaly Based Intrusion Detection Systems in the CICIDS2017 Dataset,” IEEE Access, vol. 9, pp. 22351–22370, 2021. doi: 10.1109/ACCESS.2021.3056614.
3. “Adversarial attacks against supervised machine learning based network intrusion detection systems,” 2022. Available via PubMed Central: PMC9565394.
4. I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, “Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization,” in Proc. International Conference on Information Systems Security and Privacy (ICISSP), 2018.