JOURNAL OF THE ROYAL LAUREATES ACADEMY

www.rlaindia.org

AN ADAPTIVE MACHINE LEARNING FRAMEWORK FOR REAL-TIME NETWORK THREAT IDENTIFICATION AND CYBERATTACK DETECTION

Adewar Swapnil Prabhakar

Research Scholar, Dept. of Computer Science and Engineering, Sabarmati University, Ahmedabad, Gujarat

Dr. Amit Kishor

Research Supervisor, Dept. of Computer Science and Engineering, Sabarmati University, Ahmedabad, Gujarat

ABSTRACT

The continuous evolution of cyberattacks requires intrusion detection systems capable of identifying malicious network behaviour rapidly and adapting to changing threat patterns. Conventional detection mechanisms may struggle with previously unseen attacks, rapidly changing traffic characteristics, and large-scale network environments. Adaptive machine learning provides an opportunity to develop intrusion detection frameworks that continuously learn from evolving network data while maintaining real-time detection capabilities. This paper examines an adaptive machine learning framework for network threat identification and cyberattack detection, emphasizing data preprocessing, dynamic feature selection, supervised classification, concept-drift management, and continuous model evaluation. The proposed conceptual framework combines real-time traffic monitoring with machine learning to distinguish benign and malicious behaviour and periodically update the detection model. It also considers challenges associated with false positives, class imbalance, adversarial behaviour, computational limitations, and model degradation. The paper argues that effective real-time cybersecurity requires more than high initial classification accuracy. Instead, adaptive IDS architecture should combine rapid inference, continuous monitoring, dynamic learning, robust evaluation, and human oversight to remain effective against changing cyber threats.

Keywords: Adaptive Machine Learning, Cybersecurity, Intrusion Detection, Real-Time Detection, Network Security, Concept Drift, Cyberattack Detection, Anomaly Detection

I. INTRODUCTION

The modern cybersecurity environment is characterized by rapidly changing attack techniques, increasing network complexity, and continuous growth in connected devices. Cloud computing, Internet of Things systems, mobile platforms, and distributed applications generate enormous quantities of network traffic that must be monitored for malicious behaviour. Attackers continuously modify their techniques to bypass established security controls, creating a need for intrusion detection systems that can identify both known and emerging threats. Recent research describes machine learning as an increasingly important component of IDS because it can learn patterns from network traffic and improve detection capabilities beyond traditional signature-based mechanisms.

A major limitation of static machine-learning IDS is that their performance may deteriorate when the statistical characteristics of network traffic change. A model trained using historical data assumes that future traffic will have sufficiently similar characteristics. In practice, however, new applications, protocols, devices, user behaviours, and attack techniques continually change network distributions. This phenomenon is commonly described as concept drift. Recent research specifically identifies concept drift and feature dynamics as underexplored but important challenges for intrusion detection, emphasizing the need for dynamic and adaptable IDS solutions. Adaptive machine learning addresses this problem by enabling models or feature representations to change in response to new information.

Real-time detection introduces additional requirements. A model may achieve high offline accuracy but still be unsuitable for operational deployment if it requires excessive computational resources or produces decisions too slowly. Real-time IDS must process traffic efficiently, minimize latency, manage high-volume data streams, and produce alerts that security analysts can act upon. Detection quality must therefore be considered together with computational efficiency, false-positive rates, scalability, and adaptability. Datasets such as UNSW-NB15 and CICIDS2017 provide useful benchmarks for developing and evaluating these capabilities. UNSW-NB15 contains over 2.5 million records and nine attack categories, providing a relatively diverse environment for intrusion detection research.

An adaptive machine-learning framework should therefore integrate several components

rather than rely on a single algorithm. These include continuous data acquisition, preprocessing, feature extraction and selection, classification, concept-drift detection, model updating, and performance monitoring. The framework should distinguish between normal changes in network behaviour and genuinely malicious activity. It should also avoid uncontrolled automatic retraining because poisoned or manipulated data could degrade the model. This paper examines the conceptual development of an adaptive machine-learning framework for real-time network threat identification and cyberattack detection, focusing on adaptive architecture, real-time classification, and challenges associated with continuously changing cybersecurity environments.

II. ADAPTIVE MACHINE LEARNING ARCHITECTURE FOR EVOLVING NETWORK THREATS

An adaptive intrusion detection architecture begins with continuous collection of network traffic and security events. Network flows can be monitored through sensors, routers, switches, endpoint systems, cloud environments, or network monitoring platforms. The collected information can include packet counts, flow duration, protocols, source and destination characteristics, traffic rates, timing information, and connection states. A real-time system must transform raw traffic into structured features efficiently because directly processing every raw packet may impose substantial computational costs.

The preprocessing layer is responsible for cleaning and transforming incoming data before classification. Missing values, invalid records, duplicate observations, inconsistent formats, and extreme values can affect model performance. Categorical variables may need encoding, while numerical variables may require scaling. In a streaming environment, preprocessing must be computationally efficient because data continuously arrive rather than appearing as a single static dataset. The preprocessing pipeline should therefore be designed as an automated component capable of maintaining consistent transformations over time.

Dynamic feature selection is particularly important for adaptive IDS. The relevance of network features can change as applications and attacks evolve. A feature that was highly predictive during one period may become less informative later, while a previously unimportant variable may become valuable for detecting a new attack. Recent research on concept drift and feature dynamics argues for dynamic feature selection as part of future IDS architectures. An adaptive system can periodically reassess feature importance and adjust the

feature subset used by the classifier.

Concept-drift detection forms another central component. The system can monitor changes in prediction distributions, feature distributions, classification errors, or other statistical indicators. When significant changes are detected, the framework can trigger model evaluation or retraining. Drift detection should distinguish between temporary fluctuations and persistent changes. Excessive model updates can increase computational costs and introduce instability, whereas insufficient updates can allow performance degradation to continue.

Adaptive learning strategies can include incremental learning, sliding-window approaches, periodic retraining, ensemble updating, or hybrid models. Incremental learning updates the existing model using newly labelled observations. Sliding-window approaches emphasize recent observations while reducing the influence of outdated traffic. Ensemble approaches can maintain multiple models trained under different conditions and adjust their contributions as traffic patterns change. The appropriate strategy depends on the operational environment, data volume, availability of labels, and computational resources.

The complete architecture should also include governance and human oversight. Automated adaptation must not be allowed to change security controls without appropriate safeguards. Newly detected patterns should be validated before they are incorporated into the trusted training data. Security analysts can review uncertain or high-impact alerts and confirm whether traffic represents a genuine threat. Thus, adaptive IDS should combine machine speed with human verification. Such a framework can reduce the risk that attackers manipulate the learning process and can help maintain reliable performance over time.

III. REAL-TIME THREAT IDENTIFICATION AND CYBERATTACK DETECTION

Real-time threat identification requires a detection pipeline capable of converting network activity into actionable security information with minimal delay. Incoming traffic can be transformed into flow-level features and passed through a trained classifier. The classifier then determines whether an observation is benign or suspicious and, when appropriate, assigns it to an attack category. A binary first-stage classifier can be followed by a multi-class classifier to provide more detailed threat identification. Such hierarchical architectures can help prioritize rapid detection while maintaining attack categorization capabilities.

Model selection should consider both accuracy and computational efficiency. Traditional algorithms such as logistic regression, decision trees, random forests, and SVM remain useful because they can provide relatively efficient inference compared with some complex deep-learning architectures. Research using UNSW-NB15 has demonstrated the continued evaluation of classical machine-learning models for network intrusion detection. More computationally demanding models may be useful in centralized analysis, while lightweight models may be more appropriate for edge or high-throughput environments.

Real-time detection must also manage class imbalance. In operational networks, benign traffic may substantially exceed certain types of malicious activity. A classifier trained without appropriate balancing may become biased toward the majority class. Techniques such as class weighting, carefully designed sampling, threshold optimization, and synthetic oversampling can be considered. However, synthetic examples should not be allowed to contaminate evaluation data. Performance must be measured using attack-specific recall, precision, F1-score, and false-positive rates rather than accuracy alone.

False positives are particularly important in real-time cybersecurity. An IDS that generates large numbers of incorrect alerts can overwhelm security analysts and reduce confidence in the system. Alert prioritization can therefore be incorporated into the framework. The system can assign risk scores based on the confidence of the classifier, attack category, affected assets, and contextual information. High-risk alerts can be escalated immediately, while lower-confidence cases can be placed in a review queue. This approach can make machine-learning detection more useful to security operations teams.

Real-time systems must also account for unknown attacks. Supervised classifiers are trained using known categories and may fail when traffic patterns differ significantly from training data. An adaptive framework can supplement supervised classification with anomaly detection to identify observations that do not resemble established patterns. Such observations can be flagged for investigation and potentially used to improve future training data after validation. This hybrid approach can increase the ability of an IDS to identify novel or previously underrepresented behaviour.

Finally, real-time detection should be evaluated under realistic operational constraints. Offline random train-test splits may overestimate performance when traffic patterns are temporally dependent. Time-based evaluation, cross-environment testing, and evaluation on

different datasets can provide stronger evidence of generalization. The objective is to determine whether the system continues to detect threats when network conditions change. An adaptive IDS should therefore be evaluated not only on static benchmark accuracy but also on detection latency, drift response, resource consumption, false alarms, and performance stability over time.

IV. ADAPTIVE EVALUATION, SECURITY CHALLENGES, AND DEPLOYMENT

Continuous evaluation is essential for maintaining an adaptive intrusion detection system. A model that initially performs well may gradually lose effectiveness as attack patterns and network behaviour change. Monitoring systems can track precision, recall, F1-score, false-positive rate, detection latency, and class-specific performance. Sudden performance changes can indicate concept drift, data-quality problems, or adversarial manipulation. Automated monitoring can then trigger investigation or model-update procedures.

One of the major challenges is obtaining reliable labels. Supervised adaptive learning requires information about whether new observations represent attacks or legitimate behaviour. In operational environments, labels may be delayed or incomplete. Security analysts may need to investigate alerts before confirming their classification. Semi-supervised learning, active learning, anomaly detection, and analyst feedback can help address this problem. Active learning can prioritize uncertain observations for human labelling, thereby concentrating expert effort on cases that provide the greatest information value.

Adversarial machine learning is another concern. Attackers may deliberately manipulate network traffic to evade detection or poison the learning process. If an adaptive IDS automatically incorporates maliciously manipulated observations into its training data, its future performance may deteriorate. Data validation, trusted feedback channels, robust learning, anomaly checks, and controlled retraining are therefore essential. Adaptation should never mean accepting all newly observed traffic as reliable training information.

Scalability is also important for real-time deployment. Large organizations may generate enormous quantities of network events across cloud services, endpoints, applications, and data centres. Centralized processing can create bottlenecks, while distributed detection may introduce coordination challenges. Edge-based inference can reduce latency by analysing

traffic closer to its source, while centralized systems can perform more computationally intensive analysis. Hybrid architectures can combine lightweight local detection with centralized model management.

Interpretability is another important consideration. Security analysts need to understand why a system generated an alert, particularly when automated responses could affect business operations. Feature importance, confidence scores, local explanations, and rule-based supporting evidence can improve analyst understanding. Explainable AI research in intrusion detection has increasingly emphasized the importance of transparency and interpretability for practical deployment. An adaptive model should therefore provide not only a prediction but also meaningful contextual information.

Finally, successful deployment requires integration with existing cybersecurity infrastructure. The adaptive IDS should exchange information with security information and event management platforms, endpoint security tools, threat-intelligence systems, and incident-response workflows. Model updates should be version-controlled and validated before production deployment. Security teams should establish rollback procedures in case a new model performs poorly. Thus, adaptability must be governed as carefully as detection itself. A mature real-time IDS should be continuously monitored, periodically evaluated, securely updated, and supported by human expertise.

V. CONCLUSION

The continuous evolution of cyber threats has created a need for intrusion detection systems that are not only accurate but also adaptive, scalable, and capable of operating in real time. Static models can become less effective when network behaviour changes, new applications appear, or attackers modify their techniques. Adaptive machine learning provides a promising approach by enabling detection systems to monitor changing patterns, identify concept drift, update models, and adjust feature representations.

An effective adaptive framework should integrate continuous data acquisition, preprocessing, dynamic feature selection, supervised classification, anomaly detection, concept-drift monitoring, and controlled model updating. Dynamic feature selection can ensure that the classifier continues to use informative network characteristics as traffic evolves. Concept-drift mechanisms can identify changes that require model adaptation, while hybrid supervised and anomaly-based approaches can provide complementary capabilities for known and

emerging threats. Recent research confirms that concept drift and feature dynamics deserve greater attention in future IDS research.

Real-time cyberattack detection also requires attention to practical operational requirements. Accuracy alone cannot determine the usefulness of an IDS. Detection latency, false-positive rate, computational efficiency, scalability, interpretability, and resilience against adversarial manipulation must also be considered. Benchmark datasets such as UNSW-NB15 provide useful research environments, but real-world deployment requires testing across different networks and changing temporal conditions. Continuous evaluation and controlled adaptation are therefore essential.

In conclusion, an adaptive machine-learning framework can provide a strong foundation for next-generation cybersecurity intrusion detection. Its greatest value lies in combining machine learning with dynamic feature management, concept-drift detection, real-time analytics, explainability, and human oversight. Future research should investigate online learning, federated and distributed detection, adversarially robust models, dynamic feature selection, and cross-dataset generalization. The ultimate goal should be a cybersecurity system capable of learning from evolving network environments while maintaining reliability, transparency, and operational security.

REFERENCES

Aldweesh, A., Derhab, A., & Emam, A. Z. (2024). Unveiling machine learning strategies and considerations in intrusion detection systems: A comprehensive survey. Frontiers in Computer Science, 6, 1387354.

Buczak, A. L., & Guven, E. (2016). A survey of data mining and machine learning methods for cyber security intrusion detection. IEEE Communications Surveys & Tutorials, 18(2), 1153–1176.

Khraisat, A., Gondal, I., Vamplew, P., &Kamruzzaman, J. (2019). Survey of intrusion detection systems: Techniques, datasets and challenges. Cybersecurity, 2, 20.

Liu, H., &Motoda, H. (Eds.). (2007). Computational methods of feature selection. Chapman & Hall/CRC.

Moustafa, N., & Slay, J. (2015). UNSW-NB15: A comprehensive data set for network

intrusion detection systems (UNSW-NB15 network data set). In 2015 Military Communications and Information Systems Conference (MilCIS) (pp. 1–6). IEEE.

Moustafa, N., & Slay, J. (2016). The evaluation of Network Anomaly Detection Systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set. Information Security Journal: A Global Perspective, 25(1–3), 18–31.

Patel, A., Taghavi, M., Bakhtiyari, K., &Júnior, J. C. (2013). An intrusion detection and prevention system in cloud computing: A systematic review. Journal of Network and Computer Applications, 36(1), 25–41.

Sarker, I. H. (2021). Machine learning: Algorithms, real-world applications and research directions. SN Computer Science, 2, 160.

Zoghi, A., &Serpen, G. (2024). Building an intrusion detection system on UNSW-NB15: Reducing the margin of error to deal with data overlap and imbalance. Concurrency and Computation: Practice and Experience.

Zhou, Y., Cheng, G., Jiang, S., & Dai, M. (2020). Building an efficient intrusion detection system based on feature selection and ensemble classifier. Computer Networks, 174, 107247.