International Journal of Research in Engineering and Science (IJRES)
ISSN (Online): 2320-9364, ISSN (Print): 2320-9356
www.ijres.org Volume 13 Issue 5 ǁ May 2025 ǁ PP. 190-198

A Comprehensive Review of Network Intrusion Detection Systems

Augustine Ugbari
Computer Science
University of Port Harcourt

Samuel Oluwafemi Adebayo
Network Systems
Huawei Technologies Co., Nig. Ltd

Abstract
The rising frequency and sophistication of cyberattacks have driven significant advancements in network security technologies. Among these, Intrusion Detection Systems (IDS) serve as critical components for identifying unauthorized access and anomalies. This paper presents a comprehensive review of traditional IDS approaches alongside modern machine learning (ML)-based systems. Drawing from an implementation case study—the Network Intrusion Detection and Mitigation Framework (NIDMF)—we compare detection strategies, highlight performance metrics, and identify key challenges including scalability, false positives, and evolving threats. Results from the NIDMF project, including Random Forest and LSTM models trained on the CIC-IDS2017 dataset, show promising accuracies above 99%, demonstrating the strength of ML in enhancing IDS performance. The review concludes with recommendations for future research, emphasizing the need for hybrid systems, explainability, and real-time responsiveness.

Keywords*: Intrusion Detection Systems (IDS), Network Security, Machine Learning (ML), Deep Learning (DL), Random Forest, Long Short-Term Memory (LSTM), DDoS Detection, Cybersecurity, Anomaly Detection, Real-Time Threat Mitigation, Network Traffic Analysis, Explainable AI (XAI), Cloud-Based Security, False Positives in IDS, Hybrid Detection Frameworks.*

Date of Submission: 12-05-2025 Date of acceptance: 26-05-2025

I. Introduction

With the increasing digitization of services, network infrastructures face persistent threats from attackers seeking to exploit system vulnerabilities. While traditional defense mechanisms such as firewalls provide baseline security, they often fail to detect sophisticated threats in real-time. Intrusion Detection Systems (IDS) are designed to fill this gap by continuously monitoring network traffic and flagging malicious activities. IDS techniques are traditionally classified into two broad categories: signature-based and anomaly-based systems. However, the growing complexity of network environments necessitates the adoption of intelligent, adaptive detection systems. This review examines the evolution of IDS technologies, highlighting the limitations of rule-based systems and showcasing the transformative potential of machine learning (ML) in intrusion detection, particularly in the context of Distributed Denial of Service (DDoS) detection.

II. Literature Review

2.1 Traditional Intrusion Detection Systems

Traditional Intrusion Detection Systems (IDS) have formed the backbone of network security for over two decades. These systems typically fall into two categories: signature-based IDS and anomaly-based IDS. Signature-based systems, such as Snort and Suricata, operate by matching network traffic against a database of known attack signatures (Khraisat et al., 2019). These signatures are predefined patterns of malicious activity, such as specific packet sequences or header anomalies, that correspond to previously identified threats. This method is highly efficient in detecting well-documented attacks with minimal false positives and low computational overhead.

However, a major limitation of signature-based IDS is their inability to detect unknown or zero-day attacks—threats for which no signature currently exists. The constant evolution of malware and the ingenuity of attackers in crafting new exploitation techniques render these systems reactive rather than proactive.

To address the limitations of signature-based IDS, anomaly-based systems were developed. These systems establish a baseline of "normal" network behavior and raise alerts when deviations from this baseline are detected (Xu et al., 2019). Such behavior could include unexpected port usage, abnormal packet sizes, or unusual connection rates. Despite offering improved detection of novel threats, anomaly-based IDS suffer from high false

positive rates, often misclassifying legitimate traffic as malicious due to the dynamic nature of modern networks. Moreover, configuring a reliable normal behavior profile can be extremely challenging in heterogeneous environments such as cloud or IoT-based networks.

Additionally, both traditional approaches require constant rule and profile updates, increasing the administrative burden and making them less suitable for dynamic or large-scale environments where traffic patterns fluctuate frequently.

2.2 Machine Learning in Intrusion Detection Systems

With the proliferation of high-volume network data and the need for more adaptive detection methods, machine learning (ML) has emerged as a promising approach for developing intelligent IDS. Unlike traditional systems that rely on static rules, ML-based IDS use statistical learning to identify patterns and relationships within the data that signify malicious behavior.

Supervised learning algorithms such as Support Vector Machines (SVM), Decision Trees (DT), Random Forests (RF), and Gradient Boosting Machines (GBM) have been widely adopted for intrusion detection tasks (Kocher & Kumar, 2021). These models are trained on labeled datasets that distinguish between benign and malicious traffic, enabling them to classify incoming traffic based on learned patterns. For instance, Random Forests have been shown to perform well in high-dimensional data scenarios by reducing variance through ensemble learning, making them suitable for classifying diverse attack vectors.

In parallel, deep learning (DL) methods have gained traction due to their ability to automatically extract hierarchical features from raw data. Models such as Convolutional Neural Networks (CNNs) are effective at detecting spatial correlations in packet flows, while Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks are adept at identifying temporal sequences in network traffic (Upadhyay & Patel, 2024). For example, LSTM networks can detect multistep attacks like advanced persistent threats (APTs) by analyzing packet behavior over time.

Despite their advantages, ML and DL models present several implementation challenges:

- They often require large, diverse, and well-labeled datasets, which are difficult to obtain in cybersecurity due to privacy concerns.
- The training process is computationally intensive, especially for deep learning models, demanding substantial processing power and memory.
- These systems are sensitive to concept drift, where the statistical properties of network traffic evolve over time, necessitating periodic retraining to maintain accuracy.
- Moreover, the interpretability of complex models remains a concern for security analysts who need to understand the reasoning behind each decision, particularly in high-stakes environments like critical infrastructure.

Nonetheless, the adaptability and predictive capabilities of ML-based IDS make them well-suited to modern network environments characterized by variability, scale, and sophisticated threat tactics.

2.3 Comparative Analysis

The shift from traditional to machine learning-based IDS marks a fundamental transition in cybersecurity from rule-based detection to data-driven intelligence. Each approach presents distinct strengths and limitations.

Traditional IDS, with their predefined rule sets and deterministic logic, excel in:

- Simplicity and low overhead.
- High precision in detecting known threats.
- Ease of deployment in static environments with well-understood traffic patterns.

However, they suffer from:

- Inability to detect novel attacks (zero-day vulnerabilities).
- High maintenance requirements due to constant rule updates.
- Limited contextual awareness of evolving threats.
- In contrast, ML-based IDS offer:
- High adaptability to new and unknown attack types.
- Automated feature extraction and classification.
- Scalability across large, complex networks.
- Yet, they also introduce:
- Dependence on high-quality datasets.
- Model complexity and lack of interpretability.
- Vulnerability to adversarial attacks (e.g., poisoning and evasion).

As noted by Wijekoon (2024), the future of IDS lies in hybrid models that combine the deterministic confidence of traditional IDS with the adaptive intelligence of ML systems. Such systems could use signature-based modules

for fast filtering of known threats, while ML models analyze anomalous or ambiguous traffic. These hybrid approaches can also leverage multi-source data fusion, incorporating logs from endpoints, servers, and firewalls to enrich the decision-making process.

Moreover, explainable AI (XAI) techniques are emerging to bridge the transparency gap in ML-based IDS, allowing security professionals to trace decisions back to specific data patterns or features—an essential requirement in incident response and forensic analysis.

III. Case Study: Network Intrusion Detection and Mitigation Framework (NIDMF)

The **Network Intrusion Detection and Mitigation Framework (NIDMF)** was developed as a practical solution to address the limitations of traditional IDS by integrating machine learning and deep learning methodologies with modern deployment and automation tools. Designed to detect and respond to **Distributed Denial of Service (DDoS)** attacks in real time, the framework exemplifies how artificial intelligence (AI) can be leveraged to enhance network defense mechanisms.

3.1 System Design

The NIDMF adopts a **modular and scalable architecture** that facilitates detection, classification, and mitigation of malicious traffic with minimal human intervention. The system uses the **CIC-IDS2017** dataset, which contains labeled records of benign and malicious traffic scenarios including DDoS, port scans, and brute force attacks (Strgenix, 2023). The modular design allows for seamless integration of multiple subsystems including:

- **Data ingestion and preprocessing**: CSV-formatted datasets are loaded, cleaned, normalized, and transformed using Pandas and NumPy.
- **Feature engineering**: Statistical, temporal, and protocol-based features are extracted and selected using correlation analysis and Principal Component Analysis (PCA).
- **Machine learning model inference**: Preprocessed data is passed to trained models for classification.
- **Dashboard visualization**: Real-time results and traffic metrics are visualized using tools like Grafana and Wireshark to assist network administrators.
- **Automated mitigation module**: On detection of malicious flows, pre-programmed actions such as IP blocking or alert dispatching are triggered using AWS Lambda functions.

This design ensures that the framework not only identifies threats but also responds to them autonomously—closing the loop from detection to mitigation.

3.2 Models Used

To achieve optimal detection accuracy and operational versatility, two supervised learning models were developed and evaluated:

Random Forest Classifier

The Random Forest (RF) model, an ensemble-based learning algorithm, was selected for its robustness, ease of interpretability, and strong performance on tabular data. The model was trained using 70% of the preprocessed dataset and validated on the remaining 30%. Random Forest Classifier Performance Metrics are:

i. Accuracy: 100%
ii. Precision: 1.00
iii. Recall: 1.00
iv. F1-Score: 1.00

These results suggest that the Random Forest model was able to perfectly classify benign and DDoS traffic in the dataset. The model's decision trees effectively capture the feature interactions that characterize malicious flows.

Long Short-Term Memory (LSTM) Neural Network

LSTM, a type of Recurrent Neural Network (RNN), was implemented to capture **temporal dependencies** in sequential network traffic data. LSTM is particularly suited to identifying **patterns that unfold over time**, such as coordinated multi-step DDoS attacks. Long Short-Term Memory (LSTM) Neural Network Performance Metrics are:

i. Validation Accuracy: 99.95%
ii. Test Accuracy: 99.94%
iii. Test Loss: 0.02

Despite the slightly lower accuracy compared to the Random Forest model, the LSTM proved highly effective in identifying nuanced and time-based attack behaviors, offering a complementary approach to traditional feature-based classification.

The choice of two diverse models demonstrates the framework's flexibility and highlights the importance of model selection based on operational context—Random Forest for resource-limited environments and LSTM for environments requiring temporal pattern recognition.

3.3 Real-Time Testing

To validate the system's deployment readiness, a **RESTful API** was built using **Flask**, enabling real-time traffic classification via HTTP requests. The API endpoint accepts JSON-formatted inputs representing network flows and returns classification results along with confidence scores.

Testing was conducted using **Postman**, a popular API development tool. A sample payload mimicking a typical benign flow was submitted, and the model successfully responded with:

1. **Label**: "BENIGN"
2. **Confidence Score**: 99.87%

This test confirmed that the deployed model could classify real-time traffic accurately and efficiently. The API's architecture supports integration with broader network monitoring systems, enabling immediate responses to detected threats.

3.4 System Architecture

The NIDMF framework is designed for **scalable, fault-tolerant deployment**, primarily using **Docker containers** and **Amazon Web Services (AWS)**:

Docker Containerization

Each system component—data preprocessing scripts, ML models, API services, and dashboards—is encapsulated in a Docker container. Benefits include:

i. Portability across environments
ii. Ease of maintenance and updates
iii. Consistent runtime behavior

Docker Compose or Kubernetes orchestrates the containers, ensuring resource-efficient scaling and inter-service communication.

Cloud Deployment on AWS

Deployment on **AWS infrastructure** provides the system with resilience, scalability, and global accessibility. Key AWS services include:

1. **EC2 Instances**: Host containerized applications.
2. **S3 Storage**: Archive logs, datasets, and model snapshots.
3. **CloudWatch**: Monitor system performance and trigger alerts.
4. **Lambda Functions**: Execute automated threat mitigation tasks such as updating firewall rules or sending alerts.

The use of cloud-native services allows the system to dynamically scale based on network traffic volume and computational load, making it suitable for enterprise-level implementations.

IV. Challenges in Network Intrusion Detection Systems

Despite significant advancements in intrusion detection technologies, especially with the integration of machine learning and automation, several **critical challenges** continue to hinder the effectiveness and widespread adoption of IDS, particularly in large-scale and dynamic network environments. These challenges are multifaceted, encompassing technical, operational, and architectural concerns.

4.1 Scalability

Modern networks generate **massive volumes of data**, especially in cloud environments, enterprise data centers, and IoT infrastructures. Traditional IDS architectures often struggle to **scale horizontally** to process and analyze this high-velocity traffic in real-time. The ability to inspect millions of packets per second without introducing significant latency is a major requirement for today's detection systems.

Scalability issues arise from:

1. **Throughput limitations** of packet inspection engines.
2. **I/O bottlenecks** in data pipelines.
3. Inability to maintain **stateful inspection** at scale.

To address this, IDS must be designed using **distributed processing frameworks**, containerized microservices, and cloud-native architectures. Frameworks like Apache Kafka, Kubernetes, and AWS Auto Scaling can help manage traffic bursts, but they also introduce complexity in deployment and monitoring.

4.2 False Positives

One of the most persistent and **operationally costly challenges** in intrusion detection is the high rate of **false positives**—benign traffic incorrectly flagged as malicious. Excessive false alarms lead to:

1. **Alert fatigue** among analysts.
2. Missed real threats due to **desensitization**.
3. Reduced trust in the IDS output.

This problem is exacerbated in **anomaly-based detection systems**, where deviations from established norms are flagged, even if they are harmless. Machine learning systems, while better at pattern recognition, also require **careful tuning and context-aware feature engineering** to reduce false positives.
Techniques such as **ensemble learning**, **contextual threat intelligence**, and **feedback loops from SOC (Security Operations Center) analysts** can help improve detection precision. However, there remains a trade-off between sensitivity and specificity that must be balanced based on organizational risk tolerance.

4.3 Evolving Threats
The **cyber threat landscape is dynamic**, with adversaries constantly developing new tactics, techniques, and procedures (TTPs) to evade detection. Traditional IDS solutions, especially signature-based ones, are **static** and require frequent updates to remain relevant. Even ML-based IDS can become outdated if **training datasets** are not refreshed regularly to reflect the latest attack behaviors.
Some emerging evasion strategies include:
1. **Polymorphic malware**, which changes its code signature.
2. **Living-off-the-land attacks**, which use legitimate system tools.
3. **Encrypted traffic tunneling**, which hides malicious payloads in SSL/TLS.

To counter this, IDS solutions must incorporate **continuous learning mechanisms**, **adaptive threat modeling**, and **behavioral analytics**. However, implementing such capabilities without compromising system stability or increasing complexity remains a significant hurdle.

4.4 Interoperability
In large organizations, IDS must **integrate seamlessly** with various tools in the cybersecurity ecosystem, including:
1. Firewalls and routers
2. Security Information and Event Management (SIEM) platforms
3. Endpoint Detection and Response (EDR) systems
4. Cloud workload protection platforms

Each of these tools may use **different data formats, APIs, and communication protocols**, making interoperability a challenge. For instance, correlating alerts between Suricata logs, Grafana dashboards, and a custom ML classifier may require middleware components, custom connectors, or data normalization pipelines.
Moreover, achieving **data fusion across systems** for holistic threat detection and incident response is a non-trivial task, especially when proprietary software or legacy systems are involved.

4.5 Resource Constraints
Advanced intrusion detection systems, particularly those powered by deep learning models like LSTMs or CNNs, are **computationally intensive**. Training and deploying these models requires:
1. **High-performance GPUs**
2. **Large memory and storage capacities**
3. Efficient **batch processing and inference mechanisms**

These requirements can strain IT budgets, especially in small to medium-sized enterprises (SMEs) or environments with **limited infrastructure**. Furthermore, the **energy consumption and carbon footprint** of training large models is a growing concern in sustainable computing.
Some mitigations include:
1. Using **model compression** and **quantization** techniques.
2. Leveraging **cloud-based inference APIs** with elastic billing.
3. Implementing **hybrid architectures**, where lightweight models are deployed at the edge while heavier processing occurs in centralized data centers.

Nevertheless, the tension between **model complexity and deployability** remains an ongoing research and engineering challenge.

V. Research Gaps and Contributions of the Network Intrusion Detection and Mitigation Framework (NIDMF)

While significant strides have been made in the development of Intrusion Detection Systems (IDS), many existing solutions fall short in terms of adaptability, real-time responsiveness, and system integration. Traditional and even some modern ML-based IDS still suffer from limitations in automated mitigation, scalability, interpretability, and actionable analytics. The **Network Intrusion Detection and Mitigation Framework (NIDMF)** was specifically designed to address these deficiencies through a multi-layered architecture combining machine learning, cloud-native deployment, and operational automation.

5.1 Integrated Detection and Mitigation

One of the most prominent gaps in IDS research and practice is the **separation of detection from mitigation**. Most systems focus solely on identifying malicious activity, leaving the response to manual intervention or external tools. This **disconnect introduces latency** between threat detection and action, increasing the potential damage from attacks such as DDoS or malware propagation.

NIDMF bridges this gap by embedding a real-time mitigation module within its architecture. Upon identifying a threat, the system:

1. Triggers predefined mitigation actions (e.g., blocking an IP address, terminating suspicious sessions).
2. Sends alerts or updates to firewalls and other network security appliances using automation tools such as **AWS Lambda**.
3. Supports extensibility to include custom workflows such as isolation of affected devices or dynamic re-routing of traffic.

This **closed-loop design** enables a swift and automated threat response, significantly reducing Mean Time to Detect (MTTD) and Mean Time to Respond (MTTR), two critical metrics in cybersecurity operations.

5.2 Accuracy and Low False Positives

Achieving high detection accuracy while minimizing false positives is a **longstanding challenge** in IDS. Traditional rule-based systems are deterministic but often rigid, while many ML-based systems are flexible but prone to overfitting or misclassification without sufficient tuning.

NIDMF employs **tailored machine learning pipelines** to overcome this issue. Key contributions include:

1. **Feature engineering** that captures temporal, statistical, and protocol-based attributes from raw traffic data.
2. **Careful algorithm selection**, combining the interpretability of ensemble models like Random Forests with the temporal detection capabilities of LSTM neural networks.
3. **Preprocessing strategies** such as SMOTE (Synthetic Minority Over-sampling Technique) and normalization to balance class distributions and improve model generalizability.

In empirical evaluations using the CIC-IDS2017 dataset, the Random Forest model achieved **100% accuracy**, while the LSTM model achieved **99.94%**, with extremely low false positive rates. These outcomes demonstrate the effectiveness of the framework in accurately classifying traffic without overwhelming analysts with irrelevant alerts.

5.3 Cloud-Scalable Design

Most legacy IDS are deployed on-premises and are **not optimized for distributed or cloud-native architectures**. In contrast, today's network environments are highly dynamic, often spanning hybrid, multi-cloud, and edge infrastructures. IDS solutions must therefore be **elastic**, **portable**, and **infrastructure-agnostic**.

NIDMF addresses this challenge through a **containerized deployment architecture** using **Docker** and **AWS** cloud infrastructure:

1. Components such as the data preprocessing engine, model inference services, and dashboards are deployed in isolated containers.
2. **Horizontal scaling** is achieved through orchestration tools like Docker Compose or Kubernetes.
3. **Cloud-native tools** like AWS EC2, S3, and CloudWatch are used for hosting, data storage, and monitoring, respectively.

This design ensures that NIDMF can be deployed across various operating environments with minimal configuration effort, supporting **enterprise-grade scalability** and performance.

5.4 Unified Monitoring and Visualization

A common limitation in many IDS implementations is the **lack of a unified interface** for monitoring, analysis, and decision-making. Security analysts often rely on disparate tools to correlate alerts, visualize network activity, and assess system health, leading to operational inefficiencies and missed threats.

NIDMF enhances usability and situational awareness by integrating **Grafana dashboards** for unified monitoring. Features include:

1. Real-time visualization of traffic patterns, threat alerts, and model performance.
2. Drill-down analytics that allow users to inspect individual traffic flows or anomaly scores.
3. Customizable panels for tracking system health, such as CPU usage, detection latency, and data ingestion rates.

By offering a **centralized operational view**, NIDMF empowers security teams with actionable intelligence, helping them move from reactive monitoring to proactive threat management.

VI. Discussion

The implementation and evaluation of the **Network Intrusion Detection and Mitigation Framework (NIDMF)** yielded valuable insights into the practical strengths, trade-offs, and deployment considerations associated with machine learning-based intrusion detection systems. This section synthesizes findings from the comparative performance of the two implemented models—**Random Forest** and **Long Short-Term Memory (LSTM)**—and explores their implications for real-world deployment and future enhancement of intelligent network security systems.

6.1 Model Effectiveness and Operational Trade-Offs

The **Random Forest (RF) classifier** emerged as a highly effective and interpretable model for intrusion detection. Its ability to:

1. Handle high-dimensional feature sets,
2. Provide feature importance rankings,
3. And maintain high performance on modest computing resources

makes it a strong candidate for **resource-constrained environments** such as small enterprise networks or edge computing platforms. The RF model achieved **100% accuracy, precision, and recall** on the CIC-IDS2017 dataset, confirming its robustness in identifying DDoS patterns within pre-engineered traffic features.

On the other hand, the **LSTM neural network**, though slightly less accurate (99.94% test accuracy), demonstrated unique strengths in **capturing sequential and temporal patterns**—a critical capability for identifying **multi-step attacks** and detecting **anomalous behaviors that evolve over time**. However, its deployment presents challenges:

1. **High computational demand** for training and inference.
2. **Greater sensitivity** to hyperparameter configurations.
3. Longer inference times compared to decision tree-based models.

These trade-offs underscore the importance of **contextual model selection**, where system architecture, available resources, and security objectives guide whether interpretability (RF) or temporal sensitivity (LSTM) takes precedence.

6.2 System Integration and Deployment Considerations

A significant achievement of NIDMF was the **successful integration of the trained models into a RESTful API service** using Flask, tested in a simulated environment via Postman. This approach demonstrated that **machine learning-driven IDS can be modularized and exposed as services**, making them highly portable and compatible with existing security infrastructure. Key benefits of this deployment model include:

1. Easy incorporation into **SIEMs, firewalls, or traffic gateways**.
2. Potential for **cross-platform deployment** using container technologies like Docker.
3. Support for **scalable and distributed deployment** via cloud platforms such as AWS.

This architectural choice aligns with current trends in **cybersecurity automation** and **microservices-based security orchestration**, enhancing operational flexibility.

6.3 Toward a Hybrid and Explainable IDS Architecture

The findings suggest that neither RF nor LSTM alone can fully address the multifaceted nature of modern cyber threats. A promising direction for future enhancement is the adoption of a **hybrid IDS architecture**, which leverages:

1. **Ensemble learning techniques**, combining outputs from multiple models to improve robustness and coverage.
2. **Model stacking**, where predictions from base models (e.g., RF, LSTM) are combined through a meta-classifier.
3. **Multi-layer analysis pipelines**, where signature-based filtering precedes ML-driven classification and behavior analysis.

Additionally, the growing emphasis on **explainable AI (XAI)** in cybersecurity calls for integrating interpretability tools such as:

- SHAP (SHapley Additive exPlanations)
- LIME (Local Interpretable Model-agnostic Explanations)

These tools can help bridge the trust gap in black-box models, providing security analysts with insights into why specific traffic was flagged as suspicious—an essential feature in incident response and compliance reporting.

6.4 Expanding Threat Coverage

While the current implementation of NIDMF is optimized for **DDoS detection**, the same pipeline can be extended to handle a broader range of attack types, such as:

i. **SQL Injection**
ii. **Phishing**
iii. **Man-in-the-Middle (MITM) attacks**
iv. **Insider threats**

This would require:

i. Collecting or generating labeled datasets for each attack class.
ii. Retraining or fine-tuning models to support **multi-class classification**.
iii. Redesigning the mitigation module to trigger appropriate responses for each specific threat category.

By expanding the framework's coverage, NIDMF can evolve from a **single-threat IDS** to a **comprehensive, multi-threat detection and mitigation platform**.

VII. Conclusion

The field of **Intrusion Detection Systems (IDS)** has undergone a substantial transformation, moving from early **signature-based, rule-driven mechanisms** to today's increasingly **intelligent, adaptive, and automated frameworks**. This evolution has been driven by the complexity of modern network environments and the dynamic, evasive nature of cyber threats. Traditional IDS, while still relevant for detecting known attack patterns with high precision, are no longer sufficient on their own to handle zero-day threats, advanced persistent threats (APTs), or the vast data scales typical of enterprise and cloud networks.

The development and evaluation of the **Network Intrusion Detection and Mitigation Framework (NIDMF)** presented in this study underscore the potential of **machine learning (ML)** and **deep learning (DL)** to augment and extend the capabilities of IDS. Through its dual-model approach—leveraging the **Random Forest classifier** for interpretable, high-speed detection and the **LSTM neural network** for capturing complex temporal patterns—the NIDMF demonstrates how tailored algorithms can be selected and optimized based on specific operational contexts and threat landscapes.

Key contributions of NIDMF include:

1. **Real-time detection and mitigation**, achieved through the integration of model inference pipelines with automated response mechanisms.
2. **High classification accuracy** with minimal false positives, achieved through rigorous feature engineering, preprocessing, and model selection.
3. **Scalable, cloud-native deployment**, made possible through containerization and orchestration across AWS infrastructure.
4. **Unified monitoring and visualization**, facilitated by Grafana dashboards for comprehensive situational awareness.

The framework not only detects threats but also acts upon them, reflecting a **paradigm shift from passive monitoring to proactive security orchestration**. Its modularity ensures that it can be extended to cover new threat types, integrated into diverse architectures, and continually improved through retraining and feedback.

Future Directions

As the cybersecurity domain continues to evolve, future research and development efforts should focus on:

1. **Explainable AI (XAI)** to improve trust and interpretability of complex ML/DL models.
2. **Ensemble and hybrid models** to combine the strengths of different algorithms.
3. **Federated learning and privacy-preserving training techniques** to support model generalization without compromising sensitive data.
4. **Broader attack detection** covering phishing, insider threats, data exfiltration, and cross-domain exploits.
5. **Integration with SIEM and SOAR platforms**, enabling complete incident response pipelines.

References

[1]. CertiSEC. (n.d.). What is an Intrusion Detection System and Why is it Important? CertiSEC. https://certisec.org/what-is-an-intrusion-detection-system-and-why-is-it-important/
[2]. Kästner, C. (2022, February 24). Scaling ML-Enabled Systems. Medium. https://ckaestne.medium.com/scaling-ml-enabled-systems-b5c6b1527bc
[3]. Khraisat, A., Gondal, I., Vamplew, P., & Kamruzzaman, J. (2019). Survey of intrusion detection systems: techniques, datasets and challenges. Cybersecurity, 2(1), 1–22. https://doi.org/10.1186/s42400-019-0038-7
[4]. Kocher, G., & Kumar, G. (2021). Machine learning and deep learning methods for intrusion detection systems: recent developments and challenges. Soft Computing, 25, 9731–9763. https://doi.org/10.1007/s00500-021-05893-0
[5]. Rayhan, A. (2024). Cybersecurity in the Digital Age: Assessing Threats and Strengthening Defenses. https://doi.org/10.13140/RG.2.2.31480.25607
[6]. Strgenix. (2023). CIC-IDS2017 Dataset. Kaggle. https://www.kaggle.com/datasets/dhoogla/cicids2017

[7]. Upadhyay, D., & Patel, P. (2024). Machine Learning-Based and Deep Learning-Based Intrusion Detection System: A Systematic Review. In S. D. P. Ragavendiran et al. (Eds.), Innovations and Advances in Cognitive Systems (pp. 379–394). Springer. https://doi.org/10.1007/978-3-031-69201-7_31
[8]. Wijekoon, V. B. (2024, September 6). AI in Intrusion Detection Systems: A Game Changer in Network Security. Medium. https://medium.com/@ViduraAI/ai-in-intrusion-detection-systems-ids-a-game-changer-in-network-security-1e514c5753f1
[9]. Xu, X., Liu, H., & Yao, M. (2019). Recent Progress of Anomaly Detection. Complexity, 2019, 1–11. https://doi.org/10.1155/2019/2686378
[10]. their Applications, XV(5), 22-37.