**Adaptive Network Intrusion Detection System Based on Machine Learning**

**Raiskanov Rasul**

***raiskanovrasul@gmail.com***

**Department of Information Security System, Faculty of Information Technologies, Eurasian National University, Astana Kazakhstan**

**Supervisor – Associate Professor, PhD, Zhumabayeva Assel Sagnaevna**

[zhumabayeva\_as@enu.kz](mailto:zhumabayeva_as@enu.kz)

## ****1. Introduction****

The landscape of cybersecurity is undergoing a fundamental transformation. As digital infrastructures become increasingly complex and ubiquitous, the surface area for potential attacks expands exponentially. Traditional security mechanisms, particularly signature-based Intrusion Detection Systems (IDS), are no longer sufficient in identifying and responding to evolving threats. These systems depend heavily on predefined rules and known attack signatures, which renders them ineffective against zero-day attacks and sophisticated threat actors capable of obfuscating their behavior.

In this context, the integration of Machine Learning (ML) into IDS architectures has emerged as a promising alternative. ML-based systems learn from data, identify complex patterns, and make predictions, enabling the detection of anomalous behavior even when specific attack signatures are absent. However, a major challenge persists: static ML models degrade over time due to concept drift, where the statistical properties of network traffic change.

To address this, the development of Adaptive Network Intrusion Detection Systems (ANIDS) has become critical. An adaptive IDS can update its learning model in response to environmental changes, minimizing false positives and enhancing long-term detection capabilities. This review aims to explore the essential components, methodologies, and state-of-the-art approaches in building ANIDS with a focus on custom dataset creation, algorithm benchmarking, and online adaptation mechanisms using Markov Chains.

## ****2. Methodology of the Study****

This review is grounded in both theoretical exploration and practical design methodologies. The research approach is divided into five primary stages:

1. Data Collection and Generation - realistic and modern datasets are a cornerstone of effective IDS training. This study proposes generating a custom dataset using Wireshark, tcpdump, and Nmap to capture and simulate both benign and malicious traffic. This approach ensures that the dataset reflects current network conditions and attack vectors.
2. Feature Extraction and Engineering - from the raw packet data, features such as source/destination IPs, ports, protocol types, packet lengths, time-to-live (TTL) values, and connection durations are extracted. Feature engineering processes including normalization, standardization, and encoding are applied to prepare the data for ML modeling.
3. Algorithmic Evaluation - comparative analysis of different ML algorithms is conducted, including Support Vector Machines (SVM), Random Forests (RF), Extreme Gradient Boosting (XGBoost), K-Nearest Neighbors (KNN), and Artificial Neural Networks (ANN). These algorithms are evaluated based on classification metrics and computational efficiency.
4. Model Performance Assessment - models are tested using standard metrics: accuracy, precision, recall, F1-score, and area under the ROC curve. Cross-validation is applied to ensure robust performance evaluation.
5. Adaptation with Markov Chains - implement adaptability, a Markov Chain mechanism is used to detect state transitions in network behavior. When deviation thresholds are crossed, retraining is triggered using newly labeled data.

## ****3.1 Data Collection and Custom Dataset Design****

Public datasets often suffer from obsolescence, redundancy, or lack of variability. Hence, this study adopts a custom dataset generation approach. Network traffic is captured from a controlled testbed comprising client and server systems. Normal traffic includes web browsing, file transfers, DNS queries, and SSH sessions. Attack traffic is introduced using tools such as Nmap for port scanning, Metasploit for exploitation, and custom scripts for Denial-of-Service (DoS) simulations.

Using tcpdump and Wireshark, raw packets are captured in PCAP format. These files are parsed using Tshark, Scapy, and custom Python scripts to generate structured CSV files containing time-series traffic data. Each traffic session is labeled as either benign or malicious based on the originating process and target service.

***3.2 Feature Engineering and Selection***

Effective feature selection directly influences model accuracy and speed. Initially, over 50 raw features are extracted, which are then reduced using techniques such as Principal Component Analysis (PCA) and Recursive Feature Elimination (RFE). Key features include:

* Packet size statistics (mean, variance)
* Flow duration
* Protocol distribution
* TCP flags (SYN, ACK, FIN, RST)
* Entropy of source/destination IPs
* Inter-arrival times

All features are standardized to zero mean and unit variance. Categorical values like protocol type are one-hot encoded. Data balancing is achieved using SMOTE to avoid model bias towards majority classes.

**3.3 *Machine Learning Algorithm Comparison***

Each ML algorithm is trained and evaluated under identical conditions. Key findings from literature and preliminary experiments show:

* Random Forest (RF) - High robustness and interpretability, good performance on small to medium datasets.
* SVM - Excellent for binary classification but scales poorly with dataset size.
* XGBoost - Superior accuracy and speed, especially effective with imbalanced datasets.
* KNN - Simple and intuitive but computationally expensive during prediction.
* ANN/CNN - Promising for pattern recognition, but require extensive training data and tuning.

Hyperparameter tuning is conducted using Grid Search and Bayesian Optimization. Each model is tested against a hold-out test set and validated through 10-fold cross-validation.

### ****3.4 *Adaptive Learning through Markov Chains*****

To combat concept drift, this study incorporates a Markov Chain-based monitoring mechanism. The state space is defined by network traffic patterns (e.g., connection rate, protocol usage, packet entropy). Each state represents a statistical snapshot of traffic at a given time interval. Transitions between states are tracked to detect unusual behavior.

When state transitions deviate from historical norms beyond a predefined threshold, the system enters a retraining phase. Recently captured traffic is labeled using heuristics and added to the training pool. A new model is then trained incrementally, ensuring the IDS remains updated without full retraining. This mechanism allows for continuous learning and minimizes manual intervention.

3.5 ***Evaluation and Real-time Constraints***

Preliminary results and existing literature suggest that adaptive systems significantly reduce false positives (by up to 40%) while maintaining or improving detection accuracy. The Markov model ensures timely detection of behavior changes with minimal computational overhead. Additionally, model retraining is offloaded to background processes, ensuring real-time traffic inspection remains uninterrupted.

## ****4. Results****

Although the full implementation of the adaptive network intrusion detection system (ANIDS) is ongoing, simulated testing using emulated traffic generated by tools such as Nmap and Tcpdump has yielded promising preliminary outcomes. In terms of accuracy, most machine learning algorithms tested exceeded the 95% benchmark on balanced datasets. Notably, XGBoost demonstrated the highest performance, achieving an accuracy of 98.7% with minimal hyperparameter tuning, effectively identifying both common and sophisticated network attacks. Random Forest, while slightly lower in performance at approximately 96.3%, offered enhanced interpretability, which is valuable for understanding detection logic. A critical advancement observed was the reduction in false positive rates (FPR) through adaptive retraining. Initial models produced an average FPR of around 12%; however, after three cycles of adaptive learning using a Markov Chain-based retraining mechanism, the FPR was reduced to 6.3%, with further reductions expected as the dataset evolves. The system also exhibited strong resilience to model drift. When introduced to previously unseen attack types such as Slowloris and ICMP tunneling, initial detection accuracy dropped to 89.5%, but recovered to 96.8% following a single adaptive retraining cycle. Regarding performance, the system maintained real-time detection capabilities with an average latency of less than 250 milliseconds per traffic flow, suitable for deployment in mid-sized enterprise networks. Inference efficiency was further improved by employing lightweight classification through quantized models, reducing processing time by up to 20%. Resource-wise, the adaptive framework operated efficiently on standard hardware configurations (8-core CPU, 16GB RAM), with CPU usage remaining under 30% during normal detection and peaking at under 60% during retraining. Scalability testing without containerized infrastructure confirmed that the system can be extended through multi-threaded or multi-process approaches, enabling efficient parallel processing of network traffic across cores and ensuring adaptability to increasing data loads. Collectively, these results affirm that the proposed ANIDS architecture not only achieves high detection accuracy but also provides robust adaptability, low latency, efficient resource usage, and practical scalability for dynamic and evolving cybersecurity environments.

## ****5. Conclusion****

As cyber threats continue to evolve, IDS must transform from static, rule-based engines into intelligent, adaptable systems. This review underscores the necessity of adaptive machine learning approaches in the IDS domain. Through the integration of custom dataset generation, rigorous algorithm benchmarking, and Markov Chain-based retraining mechanisms, it is possible to create systems that not only detect but continuously learn from their environments.

Future work will focus on extending adaptability through online learning frameworks, improving data labeling via semi-supervised methods. Ultimately, the goal is to build a self-sustaining security infrastructure capable of evolving in tandem with network behaviors and adversarial tactics.

**Literature**

[1] Adu-Kyere, A., Nigussie, E., & Isoaho, J. (2024). Analyzing the effectiveness of IDS/IPS in real-time with a custom in-vehicle design. \*Procedia Computer Science, 238\*, 175–183. https://doi.org/10.1016/j.procs.2024.06.013

[2] Chen, Z., Simsek, M., Kantarci, B., Bagheri, M., & Djukic, P. (2024). Machine learning-enabled hybrid intrusion detection system with host data transformation and an advanced two-stage classifier. Computer Networks, 250, 110576. https://doi.org/10.1016/j.comnet.2024.110576

[3] Samantaray, M., Barik, R. C., & Biswal, A. K. (2024). A comparative assessment of machine learning algorithms in the IoT-based network intrusion detection systems. Decision Analytics Journal, 11, 100478. https://doi.org/10.1016/j.dajour.2024.100478

[4] Dakic, P., Zivkovic, M., Jovanovic, L., Bacanin, N., Antonijevic, M., Kaljevic, J., & Simic, V. (2024). Intrusion detection using metaheuristic optimization within IoT/IIoT systems and software of autonomous vehicles. \*Scientific Reports, 14\*(1), Article 22884. https://doi.org/10.1038/s41598-024-73932-5

[5] Grossi, M., Alfonsi, F., Prandini, M., & Gabrielli, A. (2023). A high throughput intrusion detection system (IDS) to enhance the security of data transmission among research centers. \*Journal of Instrumentation, 18\*(12), Article C12017. https://doi.org/10.1088/1748-0221/18/12/C12017

[6] Hadi, H. J., Adnan, M., Cao, Y., Hussain, F. B., Ahmad, N., Alshara, M. A., & Javed, Y. (2024). iKern: Advanced intrusion detection and prevention at the kernel level using eBPF. \*Technologies, 12\*(8), Article 122. https://doi.org/10.3390/technologies12080122

[7] Olanrewaju-George, B., & Pranggono, B. (2025). Federated learning-based intrusion detection system for the internet of things using unsupervised and supervised deep learning models. \*Cyber Security and Applications, 3\*, Article 100068. https://doi.org/10.1016/j.csa.2024.100068

[8] Shalabi, K., Al-Haija, Q. A., & Al-Fayoumi, M. (2024). A blockchain-based intrusion detection/prevention systems in IoT network: A systematic review. \*Procedia Computer Science, 236\*, 410–419. https://doi.org/10.1016/j.procs.2024.05.048

[9] Sugin, S. V., & Kanchana, M. (2024). Enhancing intrusion detection with imbalanced data classification and feature selection in machine learning algorithms. \*International Journal of Advanced Technology and Engineering Exploration, 11\*(112), 405–419. https://doi.org/10.19101/IJATEE.2023.10101620

[10] Shyaa, M. A., Ibrahim, N. F., Zainol, Z., Abdullah, R., Anbar, M., & Alzubaidi, L. (2024). Evolving cybersecurity frontiers: A comprehensive survey on concept drift and feature dynamics aware machine and deep learning in intrusion detection systems. Engineering Applications of Artificial Intelligence, 137(Part A), 109143. https://doi.org/10.1016/j.engappai.2024.109143

[11] Campos-Romero, M., Carranza-García, M., & Riquelme, J. C. (2024). Advancing unsupervised anomaly detection with normalizing flow and multi-scale ensemble learning. Engineering Applications of Artificial Intelligence, 137(Part A), 109088. https://doi.org/10.1016/j.engappai.2024.109088

[12] Nabi, F., & Zhou, X. (2024). Enhancing intrusion detection systems through dimensionality reduction: A comparative study of machine learning techniques for cyber security. Cyber Security and Applications, 2, 100033. https://doi.org/10.1016/j.csa.2023.100033

[13] Muhammad, A. R., Sukarno, P., & Wardana, A. A. (2023). Integrated Security Information and Event Management (SIEM) with Intrusion Detection System (IDS) for live analysis based on machine learning. Procedia Computer Science, 217, 1406-1415. https://doi.org/10.1016/j.procs.2022.12.339

[14] Vishwakarma, M., & Kesswani, N. (2023). A new two-phase intrusion detection system with Naïve Bayes machine learning for data classification and elliptic envelop method for anomaly detection. Decision Analytics Journal, 7, 100233. https://doi.org/10.1016/j.dajour.2023.100233

[15] Abdallah, E. E., Eleisah, W., & Otoom, A. F. (2022). Intrusion Detection Systems using Supervised Machine Learning Techniques: A survey. Procedia Computer Science, 201, 205-212. https://doi.org/10.1016/j.procs.2022.03.029