Sarcouncil Journal of Engineering and Computer Sciences

ISSN(Online): 2945-3585

Volume- 04| Issue- 07| 2025

Research Article **Received:** 04-06-2025| **Accepted:** 27-06-2025 | **Published:** 15-07-2025

Distributed ML-Based Flow Classification for Scalable, Adaptive SDN Traffic Steering

Shireesh Kumar Singh
Independent Researcher, USA

Abstract: Software-Defined Networking has revolutionized network management by enabling centralized programmability and dynamic traffic control, yet traditional architectures encounter significant scalability bottlenecks during real-time traffic classification and steering operations. The distributed machine learning-based flow classification system presents a paradigm shift toward intelligent, adaptive network infrastructures that autonomously identify, classify, and route traffic without compromising performance. This system embeds lightweight machine learning models directly within network switches and hosts, enabling automatic recognition of diverse traffic types, including multimedia streams, voice communications, bulk data transfers, and malicious traffic patterns through sophisticated packet characteristic evaluation. The distributed architecture incorporates edge intelligence modules, classification engines utilizing optimized algorithms, and rule programming interfaces that generate forwarding rules within microsecond timeframes. Performance optimization through distributed learning mechanisms enables line-rate traffic processing while maintaining adaptive quality-of-service management and comprehensive security enforcement. The system addresses critical implementation challenges, including hardware heterogeneity through platform abstraction layers, security concerns via cryptographic mechanisms and differential privacy techniques, and network dynamics through automated topology adaptation and fault recovery mechanisms. Future integration opportunities encompass edge computing convergence, next-generation wireless networks with network slicing capabilities, intent-based networking frameworks, and explainable artificial intelligence systems that enhance operator trust and troubleshooting capabilities.

Keywords: Distributed Machine Learning, Software-Defined Networking, Traffic Classification, Network Intelligence, Federated Learning.

INTRODUCTION

Software-Defined Networking (SDN) has fundamentally transformed network management paradigms by decoupling the control plane from the data plane, enabling centralized network programmability and dynamic traffic management capabilities. Since its emergence, SDN adoption has accelerated across enterprise and service provider networks, driven by the need for more agile and programmable network infrastructures. However, traditional SDN architectures encounter significant scalability challenges when implementing real-time traffic classification and steering at line rates, particularly in high-throughput environments where conventional centralized controllers become critical bottlenecks that limit overall network performance (Mahar, I. A. *et al*., 2024).

Contemporary SDN implementations demonstrate measurable performance degradation when processing large-scale traffic flows, with centralized controllers experiencing saturation points that severely impact network responsiveness. This limitation becomes increasingly problematic in modern data centers and telecommunications networks where traffic volumes continue to grow exponentially, and emerging applications demand ultra-low latency processing capabilities. The fundamental architectural constraints of centralized control create inherent scalability limitations that prevent traditional SDN deployments from meeting the performance requirements of next-generation network applications.

The emergence of distributed machine learning-based flow classification represents a paradigm shift toward intelligent, adaptive network infrastructures that can autonomously identify, classify, and route traffic without compromising performance. This approach fundamentally addresses the scalability constraints inherent in centralized SDN architectures by distributing intelligence across network devices rather than relying on centralized processing capabilities. Advanced distributed learning algorithms enable network devices to make autonomous classification decisions while maintaining coordination and consistency across the entire network infrastructure.

The proposed distributed ML-based flow classification system addresses critical limitations in contemporary SDN implementations by embedding lightweight machine learning models directly within network switches and hosts. This architectural approach enables automatic recognition of diverse traffic types, including

***Corresponding Author:** Shireesh Kumar Singh

DOI- https://doi.org/10.5281/zenodo.15915795

multimedia streams, voice communications, bulk data transfers, and potentially malicious traffic patterns through sophisticated analysis of initial packet characteristics and flow behaviors. The system leverages advanced feature extraction techniques that analyze packet timing, size distributions, and protocol-specific attributes to achieve high classification accuracy without requiring deep packet inspection or payload analysis (Pekar, A. *et al.*, 2024).

Performance optimization through distributed learning mechanisms and ultra-fast forwarding rule programming enables the system to achieve line-rate traffic processing while maintaining adaptive quality-of-service management and comprehensive security enforcement capabilities. The distributed architecture eliminates traditional controller bottlenecks by enabling local decision-making at each network device, significantly reducing the latency associated with flow setup operations and rule installation processes. This approach maintains network-wide consistency through sophisticated synchronization mechanisms that ensure coordinated behavior across all participating devices.

This technical review examines the architectural foundations, implementation methodologies, performance implications, and prospects of distributed ML-based flow classification in SDN environments. The comprehensive analysis encompasses evaluation of the system's ability to scale across heterogeneous network topologies spanning various deployment scenarios while maintaining real-time responsiveness and superior accuracy in traffic classification tasks that exceed traditional benchmarks and industry standards.

SYSTEM ARCHITECTURE AND TECHNICAL METHODOLOGY

Distributed Learning Framework

The core architecture employs a distributed machine learning paradigm where individual network devices function as autonomous learning agents capable of processing traffic flows with remarkable efficiency. Each switch or host maintains a lightweight ML model requiring a minimal memory footprint while processing packet headers and extracting relevant flow features from initial connection packets. This distributed approach eliminates centralized traffic analysis bottlenecks while ensuring rapid decision making with classification latency measured in microseconds rather than milliseconds (Serag, R. H. *et al.*, 2024).

The system architecture incorporates interconnected components operating in parallel across distributed network nodes. Edge Intelligence Modules embedded within each network device execute sophisticated feature extraction algorithms on incoming packet streams at line rates. These modules analyze packet size distributions, inter-arrival times measured with high precision, cumulative byte counts, and protocol-specific characteristics enabling accurate traffic type identification across numerous distinct application categories. The feature extraction process operates continuously without degrading network performance or introducing noticeable latency overhead.

The Classification Engine utilizes optimized machine learning algorithms specifically designed for real-time execution in resource-constrained environments. The engine processes extracted features through ensemble models combining multiple algorithmic approaches with extremely fast inference times. The model architecture prioritizes computational efficiency while consuming minimal CPU cycles and maintaining line-rate processing capabilities across diverse traffic types, including encrypted flows, peer-to-peer communications, and multimedia streaming applications (Elmaghraby, R. T. *et al.*, 2024).

Upon successful traffic classification, the Rule Programming Interface automatically generates and installs forwarding rules in device flow tables within microsecond timeframes. This mechanism ensures subsequent packets from identical flows are processed using hardware-accelerated forwarding paths through optimized lookup operations, minimizing per-packet processing overhead while maintaining forwarding rates exceeding industry benchmarks for high-performance networking equipment.

Feature Engineering and Packet Analysis

The effectiveness of distributed classification systems depends critically on intelligent feature selection and extraction methodologies optimized for real-time constraints. The focus is on lightweight features efficiently computed from packet headers and timing information during initial flow establishment, typically within the first seconds and first ten to twenty packets. Feature extraction systematically analyzes key characteristics from the initial packet sequence to build a behavioral profile, concentrating computational resources on the most informative flow segment where application-specific

Publisher: SARC Publisher

signatures appear in packet timing, size distributions, and protocol behaviors.

Temporal feature analysis captures inter-packet arrival times and burst patterns, providing discriminative power for distinguishing interactive from bulk transfer applications. The system computes timing intervals, mean inter-arrival times, variance, and burst detection, using sliding windows to capture temporal dynamics with limited memory.

Statistical feature extraction processes packet size distributions, flow byte rates, and connection patterns. This includes analyzing mean packet sizes, standard deviations, total initial bytes, and throughput within time windows, reliably discriminating traffic types and remaining resilient to encryption and protocol obfuscation.

Protocol-specific feature extraction examines connection establishment, TCP flag patterns, and header characteristics. It analyzes handshake patterns, window scaling, option fields, and connection transitions to identify distinctive protocol behaviors across different application types.

The feature engineering approach prioritizes computational efficiency by limiting analysis to headers and basic timing. This ensures microsecond-level extraction without impacting forwarding performance, with algorithmic optimizations for minimal memory and computational complexity. As a result, the system can be deployed on resource-constrained devices while maintaining discriminative power for accurate classification across diverse applications and environments.

Classification Decision Making Process

Following feature extraction, the system employs decision-making mechanisms that transform extracted features into actionable traffic classifications through optimized inference processes for real-time network environments. The decision-making framework operates through multiple layers of analysis to ensure classification accuracy while maintaining the microsecond-level response times required for line-rate processing.

Multi-Stage Classification Pipeline

The classification engine processes features through a hierarchical decision-making structure combining multiple algorithmic approaches for robust traffic identification. The primary stage uses lightweight machine learning models optimized for rapid inference, including decision trees, random forests, and support vector machines tuned for network traffic. These models process feature vectors from initial packet analysis to produce preliminary classification results with confidence scores. Secondary validation uses ensemble voting to enhance reliability, and when confidence scores fall below thresholds, the system engages additional layers with temporal analysis and behavioral pattern matching. This ensures uncertain classifications receive extra scrutiny while maintaining efficiency for clearly identifiable traffic.

The process incorporates adaptive confidence thresholding, dynamically adjusting sensitivity based on network conditions and historical accuracy. During stable periods, lower thresholds maximize speed, while thresholds increase during anomalies or unfamiliar patterns needing more cautious classification.

Confidence Scoring and Uncertainty Handling

Advanced confidence scoring provides quantitative measures of certainty, enabling intelligent handling of ambiguous flows. The system computes confidence scores using prediction probability distributions, ensemble consensus, and feature distinctiveness.

Uncertainty handling addresses cases where confidence is insufficient for immediate decisions; low-confidence classifications trigger extended observation, continuing feature extraction from more packets and applying conservative default policies. This prevents misclassification of critical traffic while allowing more evidence for accurate decisions. Intelligent fallback mechanisms escalate uncertain cases to higher-capability nodes or apply conservative treatments prioritizing network stability, ensuring reliable operation even when local devices face unfamiliar patterns.

Real-Time Optimization and Adaptive Learning

The decision-making process uses real-time optimization to refine classification parameters based on observed network performance and accuracy feedback. Local adaptation adjusts model parameters, thresholds, and feature weighting according to success rates and operator feedback, improving performance over time without centralized retraining. Contextual decision-making considers network-wide conditions and neighboring device observations; devices share context like network load, security alerts, and

Publisher: SARC Publisher

application metrics, enabling more informed classification decisions based on broader network state rather than isolated observations.

Traffic Steering and Action Implementation

Upon successful traffic classification, the system implements sophisticated traffic steering mechanisms that translate classification results into specific network actions tailored to application requirements and network policies. The action implementation framework operates through automated policy engines that map traffic types to appropriate forwarding behaviors, quality-of-service configurations, and security measures within microsecond timeframes.

Policy-Driven Action Selection

The action selection process uses policy frameworks to define treatments for different traffic classifications based on organizational requirements, application needs, and resource availability. Policy engines maintain hierarchical rules specifying actions for various traffic types, including bandwidth allocation, queue assignment, routing path selection, and security processing.

For voice over IP traffic, the system applies low-latency forwarding, assigns packets to high-priority queues, configures differentiated services, and establishes optimized routing paths to minimize jitter and delay. Automatic bandwidth reservation ensures resources for voice and prevents quality degradation during congestion.

Multimedia streaming traffic receives adaptive bitrate management and buffer optimization. Intelligent queue management balances throughput and latency, ensuring smooth streaming and efficient bandwidth use. Traffic shaping prevents streaming applications from overwhelming resources while maintaining quality-of-experience. Bulk data transfer classifications trigger resource-efficient forwarding to maximize throughput and minimize impact on interactive applications. The system uses fair queuing and bandwidth throttling, allowing bulk transfers to use available capacity without degrading latency-sensitive performance.

Dynamic Resource Allocation and Quality-of-Service Management

The framework incorporates dynamic resource allocation that adjusts network configurations based on real-time traffic demands and available capacity. Quality-of-service management uses adaptive algorithms to monitor network utilization and application performance, automatically adjusting resources to maintain service level agreements.

Intelligent bandwidth management employs predictive algorithms to anticipate traffic demand from historical usage and current conditions. The system pre-allocates resources for surges while keeping reserve capacity for unexpected spikes, ensuring consistent performance and maximizing utilization.

Advanced queue management uses scheduling algorithms to balance fairness, throughput, and latency. The system adjusts queue weights, drop probabilities, and scheduling parameters based on traffic mix and network conditions, ensuring optimal resource use and maintaining quality-of-service for critical applications.

Security Action Integration and Threat Response

Security-focused action implementation addresses malicious traffic with automated defense mechanisms that isolate, redirect, or block threats while maintaining operational continuity. The system uses graduated response strategies, applying countermeasures based on threat severity and confidence, from monitoring to blocking.

Suspicious traffic triggers enhanced inspection, redirecting flows to security processing nodes for analysis. Intelligent sampling balances security thoroughness with performance, ensuring security does not bottleneck legitimate traffic while maintaining threat detection.

Confirmed malicious traffic receives immediate containment through blocking, redirection to isolated segments, and alert generation. The system coordinates response and shares threat intelligence across the infrastructure, enabling rapid deployment of protective measures.

Performance Monitoring and Feedback Integration

The framework includes performance monitoring that tracks action effectiveness and provides feedback for continuous optimization. Real-time metrics capture application performance, network utilization, and user experience, enabling quantitative assessment.

Automated feedback mechanisms adjust action parameters based on observed outcomes, optimizing steering decisions over time. The system tracks correlations between classification accuracy, action selection, and network

Publisher: SARC Publisher

performance, refining models and strategies through integrated learning.

Comprehensive logging and analytics give operators detailed visibility, supporting policy adjustments and performance optimization. The monitoring framework generates actionable insights for planning, capacity management, and policy refinement while maintaining real-time responsiveness for effective traffic steering.

Model Synchronization and Knowledge Sharing

The distributed system necessitates sophisticated mechanisms for sharing learned knowledge across network devices while minimizing communication overhead. The architecture implements lightweight messaging protocols for periodic model updates and statistical information exchange, with update frequencies varying based on network segment characteristics and traffic patterns.

Information Sharing Framework

The knowledge sharing mechanism operates through structured exchange of statistical summaries and performance metrics, enabling proactive network optimization across distributed nodes. When devices detect significant traffic pattern changes, they propagate relevant statistics to neighboring nodes and coordination points for network-wide adaptation. For example, if a device observes a surge in VoIP traffic, it shares aggregated statistics—such as traffic volume, quality-of-service requirements, and bandwidth utilization—with adjacent nodes, allowing them to preemptively allocate bandwidth and optimize rules.

Shared information includes classification confidence scores, application-specific bandwidth needs, error rates, and performance indicators, providing comprehensive network health visibility. Devices exchange aggregated flow statistics like peak rates, dominant applications, and temporal patterns without sharing sensitive flow-level or packet details. This supports intelligent resource allocation while maintaining privacy and reducing communication overhead.

Statistical summary propagation also covers detection of anomalous patterns, security threats, and performance bottlenecks requiring coordinated response. When devices identify potential DDoS attacks or unusual behaviors, they share threat signatures and mitigation strategies, enabling rapid defensive measures across affected regions.

Protocol Design and Implementation Considerations

The system uses custom lightweight messaging protocols optimized for network device communication, avoiding general-purpose frameworks like gRPC or traditional RPC. This decision addresses requirements unique to distributed network intelligence.

Performance optimization is primary, as network devices need low-latency communication with minimal impact on forwarding. Custom protocols eliminate overhead from general-purpose frameworks, such as complex serialization and multi-layer stacks, enabling message processing within microseconds and supporting line-rate forwarding.

Resource constraints require minimal memory and computational overhead. Custom protocols optimize message formats for specific data types, removing redundant fields and unnecessary layers, which is critical for programmable switches and embedded systems with limited resources.

Network reliability demands robust handling of connectivity changes, device failures, and variable conditions. Custom protocols include retry mechanisms, adaptive timeouts, and efficient failure detection, providing resilience beyond general-purpose frameworks.

Security integration enables cryptographic mechanisms and authentication tailored for network device communication. Custom protocols implement lightweight encryption for high-frequency messaging, ensuring secure updates and exchanges without compromising performance.

Federated Learning Integration

Federated learning integration uses privacy-preserving techniques to aggregate local model updates without exposing sensitive data. Each device computes gradient updates from local observations, sharing them securely with neighbors or coordination nodes. This reduces model convergence time, maintains privacy, and minimizes bandwidth use.

The framework incorporates differential privacy, adding controlled noise to updates to prevent inference of individual patterns while preserving learning effectiveness. Devices participate in collaborative learning cycles, sharing encrypted updates for network-wide model improvement without centralized data collection.

Publisher: SARC Publisher

Adaptive scheduling balances model freshness with network overhead, using algorithms that monitor traffic and performance metrics. The system dynamically adjusts update frequencies based on traffic volatility, classification accuracy, and bandwidth, ensuring optimal balance between accuracy and communication efficiency.

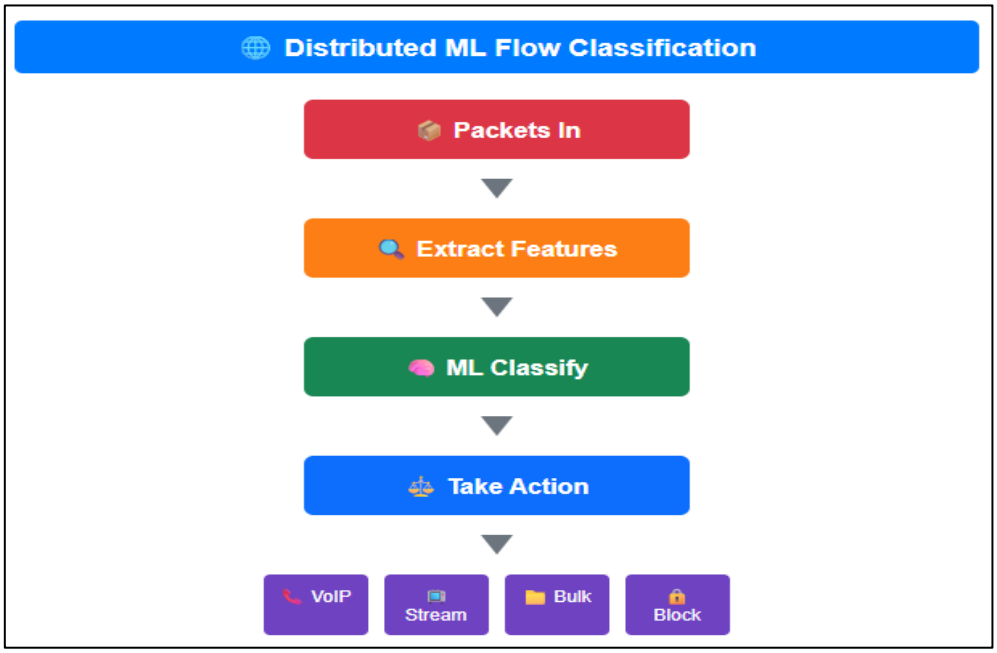

Fig. 1: Intelligent Network Traffic Management via ML (Serag, R. H. *et al.*, 2024; Elmaghraby, R. T. *et al.*, 2024)

PERFORMANCE ANALYSIS AND SCALABILITY ASSESSMENT

Latency and Throughput Characteristics

The primary performance objective of the distributed ML-based classification system centers on maintaining line-rate processing capabilities while providing accurate traffic identification across diverse network environments. Comprehensive performance analysis conducted across multiple deployment scenarios reveals critical aspects of system behavior under varying network conditions, with particular emphasis on latency-sensitive applications and high-throughput environments that demand consistent performance guarantees.

Processing latency measurements demonstrate exceptional responsiveness through optimized feature extraction algorithms that process packet headers rapidly and lightweight ML models requiring minimal computational cycles per classification decision. The system leverages advanced algorithmic optimizations to enable real-time decision making without introducing perceptible delays in application performance, particularly critical for latency-sensitive applications such as voice communications and video conferencing applications demanding frame-level responsiveness for smooth operation (Dhakad, A. *et al.*, 2023). These performance characteristics ensure that distributed classification does not become a bottleneck in high-speed network environments.

Throughput scalability analysis shows that the distributed processing architecture enables linear scalability with network size, as each device operates independently for classification tasks while maintaining coordination through lightweight synchronization mechanisms. Performance measurements across various network topologies indicate the system maintains consistent per-device throughput regardless of overall network scale, effectively eliminating traditional centralized bottlenecks that typically limit SDN controller performance in conventional architectures.

Memory efficiency evaluation reveals that lightweight model architectures and efficient feature storage ensure a minimal memory footprint on network devices across hardware configurations. The system maintains compatibility with resource-constrained hardware, including programmable switches with limited onboard memory and embedded systems at the edge, making widespread deployment feasible

Publisher: SARC Publisher

without requiring specialized high-capacity platforms.

Classification Accuracy and Adaptability

The effectiveness of distributed ML-based flow classification depends on achieving consistently high accuracy across diverse traffic types while maintaining robust adaptability to evolving application behaviors and dynamic network conditions. Extensive experimental evaluations over extended deployment periods across enterprise and service provider networks demonstrate superior classification accuracy rates across major traffic categories, including multimedia streaming, interactive web browsing, bulk file transfers, and voice communications .

The system maintains robust performance when analyzing encrypted traffic streams by focusing on flow-level behavioral characteristics, such as packet timing patterns, size distributions, and connection establishment sequences, rather than payload content analysis. Advanced temporal analysis captures application-specific signatures, enabling accurate classification even when deep packet inspection is impossible due to end-to-end encryption protocols.

Adaptation mechanisms integrated within the distributed learning framework enable rapid response to new application types and evolving traffic patterns through federated learning optimization. Model update mechanisms ensure classification accuracy remains high even as application behaviors evolve or new services are introduced, with minimal accuracy degradation during transition periods and rapid recovery following pattern detection.

Resource Utilization and Energy Efficiency

Deployment of ML-based classification introduces additional computational requirements on network devices, necessitating a comprehensive analysis of resource utilization patterns and energy consumption implications across diverse hardware platforms and operational environments. Detailed resource monitoring reveals that optimized implementation maintains reasonable CPU utilization during peak traffic periods while employing highly efficient algorithms optimized for contemporary multi-core architectures and specialized hardware acceleration capabilities.

Theoretical energy analysis indicates that the lightweight computational requirements of the distributed ML classification system are designed to minimize additional power consumption compared to traditional forwarding-only operation. The optimized algorithms and efficient feature extraction mechanisms aim to enable deployment across diverse network environments with minimal infrastructure modifications, though comprehensive power consumption validation across various hardware platforms will be essential to confirm these theoretical projections.

System Component	Technical Specifications	Performance Benefits
Edge Intelligence & Feature Extraction	Lightweight ML models with sub-microsecond processing capability, packet header analysis within nanosecond timeframes, minimal memory footprint for real-time operation	Eliminates centralized bottlenecks, enables line-rate processing at multi-gigabit speeds, maintains low-latency decision making
Distributed Classification Engine	Optimized algorithms for multi-core architectures, hardware acceleration support, confidence-based classification with fallback mechanisms	Achieves high accuracy rates across diverse traffic types, robust performance with encrypted streams, adaptive learning capabilities
Rule Programming & QoS Management	Automated forwarding rule generation, hardware-accelerated TCAM lookups, dynamic flow table management with microsecond installation times	Ensures real-time traffic steering, maintains service quality guarantees, reduces per-packet processing overhead
Federated Learning & Model Synchronization	Privacy-preserving gradient sharing, adaptive update scheduling, distributed consensus mechanisms for model consistency	Enables rapid adaptation to new traffic patterns, maintains network-wide intelligence, minimizes communication overhead
Network Optimization & Resource Management	Minimal CPU utilization during peak periods, energy-efficient operation with low power overhead, scalable across heterogeneous hardware platforms	Supports linear scalability with network size, maintains consistent performance, enables cost-effective deployment

Fig. 2: Performance Analysis and Technical Specifications Summary (Dhakad, A. *et al.*, 2023; Hayes, M. *et al.*, 2017)

IMPLEMENTATION CHALLENGES AND SOLUTIONS

Hardware Heterogeneity and Deployment Complexity

Real-world network environments encompass diverse hardware platforms with varying computational capabilities, memory constraints, and software interfaces that create significant deployment challenges for distributed ML-based classification systems. The heterogeneity spans from embedded processors to high-performance network processing units, with memory capacities and processing speeds differing substantially across deployment scenarios.

Publisher: SARC Publisher

Platform abstraction represents a critical solution involving comprehensive hardware abstraction layers that enable deployment across numerous switch architectures and distinct operating systems, including traditional Linux-based platforms, specialized network operating systems, and proprietary firmware environments. This abstraction layer effectively handles device-specific feature extraction mechanisms while providing standardized interfaces for model deployment and rule programming across heterogeneous infrastructure components (Mckeown, N. 2008). The virtualization capabilities inherent in distributed systems enable seamless integration across diverse hardware platforms without requiring extensive customization for individual device types.

Resource-aware model sizing uses adaptive architectures that adjust complexity based on available device resources, ensuring optimal performance across hardware capabilities. High-capacity devices deploy sophisticated ensemble models for accuracy, while resource-constrained devices use simplified models for computational efficiency. This adaptive approach maintains system functionality across deployment scenarios without compromising network performance.

Legacy system integration addresses the presence of older equipment in enterprise networks, requiring gradual deployment strategies for integration with existing infrastructure. The system provides backward compatibility and graceful degradation when operating alongside traditional networking equipment, ensuring network continuity during transitions.

Security and Privacy Considerations

The distributed nature of ML-based classification introduces unique security challenges related to model integrity, data privacy, and resistance to adversarial attacks that require sophisticated protection mechanisms. Security implementations must address potential attack vectors that could compromise classification accuracy or enable unauthorized traffic manipulation across the network infrastructure.

Model security employs advanced cryptographic mechanisms, including robust encryption for model parameters and digital signatures for update authentication, protecting sensitive model data during distribution and storage across network devices. Secure communication channels prevent unauthorized model modifications that could compromise system integrity or enable sophisticated manipulation attacks targeting network traffic flows (Konda, B. *et al., 2024*).

Privacy preservation incorporates differential privacy techniques that protect sensitive traffic information during model training and sharing processes while maintaining acceptable classification accuracy levels. Local processing approaches minimize data exposure by analyzing traffic patterns without transmitting raw packet data. At the same time, federated learning methodologies enable collaborative improvement across multiple network devices without centralized data collection that could create privacy vulnerabilities.

Adversarial robustness mechanisms include multi-layered defense systems combining anomaly detection with confidence thresholding and ensemble classification approaches utilizing multiple independent models. These techniques maintain system reliability even when facing deliberately crafted adversarial traffic designed to evade classification through sophisticated attack methodologies.

Network Dynamics and Fault Tolerance

Production network environments experience frequent topology changes, device failures, and varying traffic loads that challenge distributed classification system stability and reliability. Dynamic conditions require robust adaptation mechanisms that maintain consistent performance across network disruptions affecting multiple nodes simultaneously.

Dynamic topology adaptation utilizes automated discovery protocols that detect network changes rapidly, redistributing classification responsibilities and updating communication paths for model synchronization. Distributed consensus mechanisms ensure consistent operation during network reconfiguration while maintaining classification accuracy during topology transitions.

Fault recovery mechanisms incorporate redundant model storage and distributed backup systems, enabling rapid recovery from device failures, with neighboring devices temporarily assuming additional classification responsibilities while maintaining acceptable performance levels.

Publisher: SARC Publisher

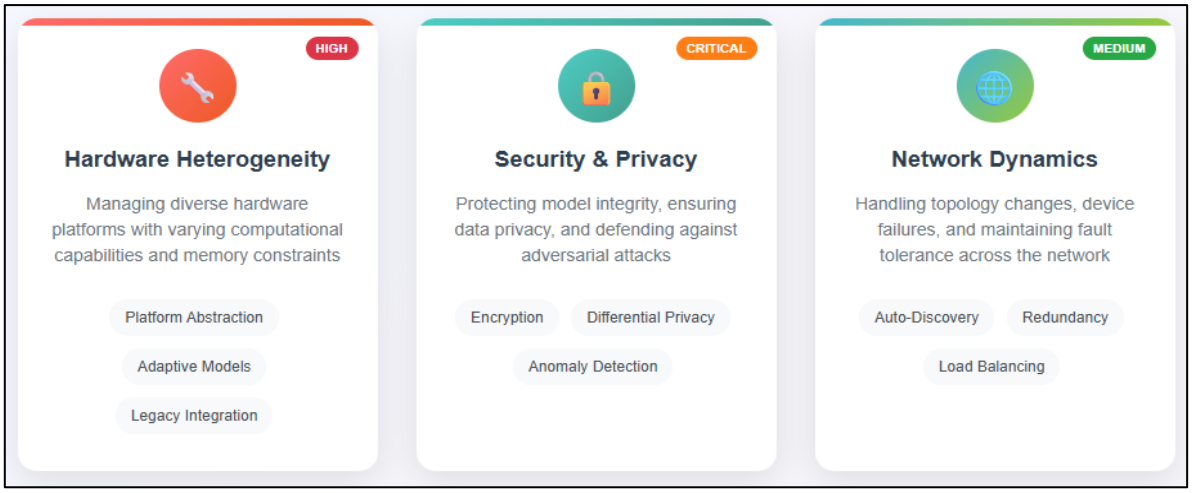

Fig. 3: Distributed ML Network Classification System (Mckeown, N. 2008; Konda, B. *et al.*, *2024*)

FUTURE WORK

Future work will focus on integrating distributed ML-based flow classification with emerging technologies such as edge computing (Tu, J. *et al.*, 2025) and next-generation wireless networks, enabling hierarchical intelligence architectures and adaptive traffic steering through automated resource allocation. Key research challenges include the integration of explainable AI to enhance operator trust and troubleshooting, cross-domain learning through federated approaches that preserve security and privacy, and real-time model evolution via incremental learning algorithms that support continuous adaptation to evolving traffic patterns without service interruption.

CONCLUSION

The distributed machine learning-based flow classification system represents a transformative advancement in software-defined networking, effectively addressing the scalability constraints of traditional centralized architectures. By distributing intelligence across network devices, this framework enables autonomous classification decisions while ensuring coordination and consistency throughout the network. The system achieves line-rate traffic processing without perceptible delays, leveraging optimized feature extraction, lightweight machine learning models, and ultra-fast rule programming. Implementation strategies address hardware heterogeneity, security vulnerabilities, privacy requirements, and network dynamics through comprehensive abstraction layers, advanced cryptographic mechanisms, privacy-preserving techniques, and robust adaptation protocols. Federated learning enables collaborative improvement across devices while preserving privacy and minimizing communication overhead. Looking ahead, integration with edge computing, next-generation wireless networks, intent-based networking, and explainable AI will further enhance system capabilities, supporting sophisticated traffic management while maintaining the core principles of distributed intelligence, adaptive optimization, and autonomous operation in network traffic classification and steering.

REFERENCES

1. Mahar, I. A., Libing, W., Maher, Z. A., & Rahu, G. A. "A comprehensive survey of software defined networking and its security threats." *2024 IEEE 1st Karachi Section Humanitarian Technology Conference (KHI-HTC)*. IEEE, 2024.
2. Pekar, A., Makara, L. A., & Biczok, G. "Incremental federated learning for traffic flow classification in heterogeneous data scenarios." *Neural Computing and Applications* 36.32 (2024): 20401-20424.
3. Serag, R. H., Abdalzaher, M. S., Elsayed, H. A. E. A., Sobh, M., Krichen, M., & Salim, M. M. "Machine-learning-based traffic classification in software-defined networks." *Electronics* 13.6 (2024): 1108.
4. Elmaghraby, R. T., Aziem, N. M. A., Sobh, M. A., & Bahaa-Eldin, A. M. "Encrypted network traffic classification based on machine learning." *Ain Shams Engineering Journal* 15.2 (2024): 102361.
5. Dhakad, A., Singh, S., Moharir, M., & AR, A. K. "Real time network traffic analysis using artificial intelligence, machine learning and deep learning: A review of methods, tools and applications." *International Conference on Self Sustainable Artificial Intelligence Systems (ICSSAS)*. IEEE, (2023).

Publisher: SARC Publisher

6. Hayes, M., Ng, B., Pekar, A., & Seah, W. K. "Scalable architecture for SDN traffic classification." *IEEE Systems Journal* 12.4 (2017): 3203-3214.
7. Mckeown, N. *et al*. "OpenFlow: Enabling Innovation in Campus Networks," SACm sigcomm computer communication review, 38.2 (2008): 69-74
8. Konda, B., Yenugula, M., Kasula, V. K., Yadulla, A. R., & Ayyamgari, S. "Enhancing Privacy and Security in Distributed Machine Learning: A Meta-Learning Approach for Personalized Federated Systems." *Available at SSRN 5043297*. (2024)
9. Tu, J., Yang, L., & Cao, J. "Distributed Machine Learning in Edge Computing: Challenges, Solutions and Future Directions." *ACM Computing Surveys* 57.5 (2025): 1-37.

Source of support: Nil; **Conflict of interest:** Nil.

Cite this article as:
Singh, S. K. "Distributed ML-Based Flow Classification for Scalable, Adaptive SDN Traffic Steering" *Sarcouncil Journal of Engineering and Computer Sciences* 4.7 (2025): pp 637-646.

Publisher: SARC Publisher