DYNA
http://dyna.medellin.unal.edu.co/

UNIVERSIDAD
NACIONAL
DE COLOMBIA

AUTOFIRE: an intelligent multi-agent framework for automated extraction and classification of pfSense Firewall rules

Noor Saud Abd [a,b]* & Kamel Karoui [c]

[a] Department of Information and Communication Technologies, (ENIT), University of Tunis El Manar, Le Belv´ed`ere, Tunis, Tunisia. noor.saudadb@enit.utm.tn
[b] Department of Cyber Security, Tikrit University, Tikrit, SalahAl-din, Iraq. noor.s.abd@tu.edu.iq
[c] Department of Computer Science and Mathematics, (INSAT), Carthage University and LIPSIC Laboratory, Tunis El-Manar, Tunis, Tunisia. kamel.karoui@insat.ucar.tn

Received: July 18th, 2025. Received in revised form: December 10th, 2025. Accepted: December 29th, 2025.

Abstract
The article presents a next-generation smart multi-agent system, AUTOFIRE, for the automatic extraction and classification of pfSense firewall rules. While modern network security relies on properly configured firewalls, rule management remains complex and prone to inconsistencies. Our approach retrieves rules from pfSense in a simulated environment, applies a confidence scoring framework, and classifies them as confident or dubious. Confidence measures include interface specificity, protocol explicitness, port definition, fast designation, and label clarity. Empirical results from our prototype show that 76.2% of rules were classified as dubious, requiring further validation, while 23.8% had high confidence ratings, emphasizing the need for distributed validation mechanisms. The system integrates an anonymization module to protect sensitive data, enabling privacy-preserving communication with master agents for cross-environment authentication. AUTOFIRE lays the foundation for automatic rule integration and merging in distributed firewall infrastructures, addressing key challenges in standardization, privacy, and conflict resolution in modern cybersecurity systems.

Keywords: Network security; firewall rules; pfSense; confidence scoring; multi-agent systems; rule classification; privacy preservation; distributed validation.

AUTOFIRE: un marco inteligente multiagente para la Extracción y clasificación automatizada de reglas de Firewall pfSense

Resumen
Este artículo presenta AUTOFIRE, un sistema inteligente de nueva generación basado en múltiples agentes para la extracción y clasificación automática de reglas de firewall en pfSense. Aunque la seguridad de las redes modernas depende de una correcta configuración de los firewalls, la gestión de reglas sigue siendo compleja y propensa a inconsistencias. El enfoque propuesto recupera reglas en un entorno simulado, aplica un marco de puntuación de confianza y las clasifica como confiables o dudosas. Las métricas de confianza incluyen la especificidad de la interfaz, la explicitación del protocolo, la definición de puertos, la designación rápida y la claridad de las etiquetas. Los resultados experimentales muestran que el 76,2% de las reglas fueron clasificadas como dudosas y requieren validación adicional, mientras que el 23,8% alcanzó un alto nivel de confianza. Además, el sistema incorpora un módulo de anonimización que protege datos sensibles y permite la validación distribuida preservando la privacidad. AUTOFIRE establece una base para la integración automática de reglas en infraestructuras de firewall distribuidas.

Palabras clave: seguridad de redes; reglas de Firewall; pfSense; puntuación de confianza; sistemas multiagente; clasificación de reglas; preservación de la privacidad; validación distribuida.

1 Introduction

The rapid growth of sophisticated cyberattacks has placed significant strain on network security infrastructures, increasing the need for intelligent and distributed protection mechanisms [1]. Firewalls, particularly the open-source solution pfSense, remain a critical first line of defense; however, firewall rule management continues to pose major

How to cite: Abd, N.S., and Karouic K., AUTOFIRE: an intelligent multi-agent framework for automated extraction and classification of pfSense Firewall rules DYNA, (93)240, pp. 62-71, January - March, 2026.

Universidad Nacional de Colombia.
DYNA, (93)240, pp. 62-71, January - March, 2026,
https://doi.org/10.15446/dyna.v93n240.121706

challenges due to redundancy, conflicts, and scalability issues [2,26]. Studies indicate that firewall misconfigurations account for approximately 65% of network security breaches, with up to 40% of rules being redundant or conflicting [3,4]. As networks become more complex and distributed, manual and static rule management proves increasingly inefficient and error-prone [8–11]. This paper introduces AUTOFIRE (Automated Firewall Rule Extraction), an intelligent multi-agent system with the objective of addressing these concerns through automated extraction, classification, and verification of pfSense firewall rules. AUTOFIRE involves a structured mechanism that begins from rule extraction within pfSense settings, employs confidence-based classification as a mechanism for identifying rules for verification, and employs privacy-preserving mechanisms to enable secure sharing of rules within distributed settings [12,23]. Our approach goes beyond rule management as typical by suggesting quantitative confidence scoring technology that evaluates rules based on the interface specificity, explicitness of the protocol, port definition, swift designation, and label clarity.

This enables the objective classification of rules as confident (no external validation required) or doubtful (distributed validation required).

The controlled virtualized testbed experimental deployment of AUTOFIRE demonstrated that a significant proportion of firewall rules (76.2%) were tagged as doubtful, while only 23.8% were given high confidence scores. These findings stress the immediate need for automated validation procedures in firewall rule management and point towards the potential benefits of a distributed method of rule harmonization. By using our confidence-scoring mechanism and anonymization techniques on the discovered rules, AUTOFIRE allows for privacy-preserving sharing and testing of rules on numerous network environments without exposing sensitive config data. Despite large amounts of research in the field of firewall management and intrusion detection systems, there exists an inherent lack in objective rule classification mechanisms, privacy-preserving rule sharing, and rule convergence in distributed environments on an automatic level. Solutions tend to rely on manual authentication, centralized management of rules, or exchange of sensitive configuration data that compromises network vulnerabilities. AUTOFIRE addresses these deficiencies by (1) offering a numerical confidence level for objective rule classification, (2) using privacy-preserving technologies to enable rule verification without exposing sensitive network data, and (3) establishing a foundation for distributed rule harmonization based on collective intelligence across multiple networks. Through this end-to-end approach, AUTOFIRE aims to enhance the effectiveness of firewalls, reduce configuration mistakes, and enhance overall network security posture in increasingly complex distributed environments.

2 Related work

Several studies have addressed firewall configuration and rule validation challenges. Wool (2004) showed that over 80% of corporate firewalls suffered from misconfiguration [5], while Al-Shaer and Hamed (2004) proposed anomaly detection methods for identifying policy conflicts and redundancies in distributed firewalls [6]. Yuan et al. (2006) introduced FIREMAN, a toolkit for modeling and verifying firewall rule consistency [7].

In the context of machine learning, Elbadawi et al. (2020) proposed a hybrid AI-based approach for rule optimization, leveraging both supervised and heuristic models for intelligent filtering [8]. Bégin et al. (2019) explored learning-based validation for firewall configurations, demonstrating improvements in misconfiguration detection [9]. Ahmed et al. (2016) presented a comprehensive review of network anomaly detection techniques using AI, many of which are applicable to rule classification and validation tasks [10].

For multi-agent and distributed systems, Papaioannou and Delis (2015) proposed adaptive security management through multi-agent coordination, offering insights relevant to AUTOFIRE’s agent-based framework [11,25]. Zambonelli et al. (2003) introduced the Gaia methodology for developing scalable multi-agent systems [12].

Privacy preservation in distributed cybersecurity systems has also gained traction. Zhang et al. (2022) detailed privacy-preserving data sharing in multi-agent contexts, offering theoretical backing for AUTOFIRE’s anonymization module [13]. Fan and Xiong (2012) explored real-time anonymization mechanisms for streaming data, which could inform future extensions of our system [14].

Wang Buqing [15] designed a better firewall based on the integration of Snort and pfSense to protect against common internet attacks. After reviewing the weaknesses of conventional firewalls, Wang incorporated enhanced defense technologies like network proxies, CNNBiLSTM intrusion detection model, and unique algorithms (IBM and VLDC). The integration offers a specific defense mechanism against port scanning, DOS attacks, and algorithm complexity attacks while significantly improving both operational efficiency and detection accuracy compared to conventional firewalls. Johannes Loevenich, Erik Adler, Rémi Mercier, et al. [16] tested an autonomous cyber defense (ACD) agent based on hybrid AI models to protect core network segments. They combined deep reinforcement learning (DRL), large language models (LLMs), and rule-based models in one comprehensive defense system. The ACD agent employs a DRL model to execute defense actions (monitor, analyze, decoy, remove, restore), and an LLMbased chatbot provides a human expert interface. The authors compared their system with two red agent strategies in a gym environment and enhanced the chatbot with retrieval-augmented generation using cybersecurity knowledge graphs [29]. Their research demonstrates that the hybrid solution can effectively enhance protection for critical networks connected to untrusted infrastructure. Will Serrano [17] developed CyberAIBot, an AI-powered intrusion detection system especially designed for IoT networks handling OT and IT network traffic. The system has a distributed Deep Learning framework that operates at the edge through private cloud computing to enable decisions to be made close to data sources. CyberAIBot's novel approach utilizes specialist DL technical clusters specialized in specific types of attacks, overseen by a management cluster responsible for resolving conflicting classifications. Serrano's comparison of performance compared Long Short-Term Memory (LSTM) networks to Support Vector Machines (SVMs) and concluded that SVMs learn approximately 15 times faster, while LSTMs offer 30% better performance on average. The research demonstrated CyberAIBot's capability to process an astonishing 5.52E+08 data points, highlighting its scalability for practical IoT security applications.

To evaluate the relative performance, we conducted a benchmarking study against representative solutions, including ML-based firewall optimizers and centralized rule management platforms. Unlike these approaches, AUTOFIRE showed superior classification transparency through interpretable confidence scores and introduced privacy-preserving mechanisms absent in traditional methods. While ML-based tools achieved slightly higher accuracy (up to 91%) in ideal conditions, their black-box nature and need for large labeled datasets limit usability. In contrast, AUTOFIRE achieved an average confidence-based classification precision of 88.4% without compromising explainability or privacy. Furthermore, its modular, agent-based architecture demonstrated better adaptability to distributed environments, offering scalable rule extraction and evaluation with modest overhead (<31%).

3 Proposed methodology

Sequence for the retrieval of firewall rules, classification, and anonymization in pfSense configurations. A virtualization-based controlled experimental paradigm was employed to design an emulating setup that facilitated reproducible and controlled testing. The process follows a four-step procedure: (1) environment setup and configuration, where we set up the virtual machines of pfSense and Ubuntu with appropriate networking; (2) firewall rule extraction, through the command line interfaces of pfSense to retrieve the ruleset in effect; (3) rule classification and analysis, utilizing our new confidence score algorithm for rating the quality of the rules impartially (the scoring parameters are detailed in Table 1: Confidence Scoring Parameters); and (4) anonymization of suspicious rules for enabling privacy-preserving verification. This structured approach allows for consistent rule processing in line with security best practices while maintaining sensitive network configuration data. The following sections detail each phase of our approach, the algorithms developed, and the performance metrics used to measure effectiveness. The AUTOFIRE architecture is constructed based on a two-level multi-agent system to extract, analyze, and process pfSense firewall rules. As Fig. 1 illustrates, we utilized an Oracle VirtualBox-managed virtualized environment to support both the agent system based on Ubuntu and the pfSense firewall [18,19]. The pfSense VM (version 2.7.2-RELEASE) was configured with a WAN interface (em0) using Network Address Translation (NAT) for external access and an internal LAN interface (em1) configured with a static IP (192.168.1.1/24) attached to a Host-only network.

Table 1.
Confidence Scoring Parameters

Parameter	Weight	Description	Expected Impact
Interface Specificity	0.10	Evaluates whether rules target specific network interfaces	Higher confidence for interface-specific rules
Protocol Explicitness	0.10	Determines if rules explicitly specify protocols (TCP, UDP, ICMP)	Higher confidence for protocol-specific rules
Port Definition	0.05	Assesses whether rules apply to specific service ports	Higher confidence for port-specific rules
Quick Designation	0.10	Considers whether rules use pfSense's "quick" modifier for priority processing	Higher confidence for quick designated rules
Label Clarity	0.05	Evaluates the presence of descriptive labels documenting the rule's purpose	Higher confidence for clearly labeled rules
WAN Security Practice	0.10	Checks adherence to the established practice of blocking private networks on WAN	Additional confidence for rules following this practice
Anti-lockout Protection	0.20	Identifies rules enabling protection against administrator lockout	Significant confidence boost for these protective rules
Baseline Value	0.50	Starting confidence value assigned to all rules	Applied universally as foundation score
Classification Threshold	0.70	Minimum score for "confident" classification	Rules below the threshold labeled "doubtful"

Source: Authors' own work.

The Ubuntu VM (version 20.04 LTS) was configured with a single network adapter attached to the same Host-only network and received its IP address (192.168.1.101) from the pfSense machine via DHCP [16]. This network isolation allowed for a controlled test environment while enabling realistic rule extraction and testing. The agent system used on Ubuntu consists of three primary components: (1) a Rule Extraction Module that interacts with the pfSense system through SSH or direct console access; (2) a Rule Processing Engine that normalizes and parses the extracted rules; and (3) a Classification and Anonymization Module that utilizes our confidence scoring algorithm. For watching pfSense live ruleset, we used the Packet Filter Control command with the option '-sr', enabling access to the compiled set of rules accessible to the lower-level PF firewall engine. Inter-system network connectivity was verified by ping tests and SSH connection attempts to guard against problems such as firewall rule sets that could cause traffic interruption. This test environment enabled reproducible analysis while closely simulating real-world firewall deployments. AUTOFIRE employs a four-stage pipeline (Fig. 2) to extract, normalize, and prepare pfSense firewall rules for classification. Secure rule extraction is performed via SSH enabled through the pfSense web interface, with console access provided as an alternative when SSH is unavailable.

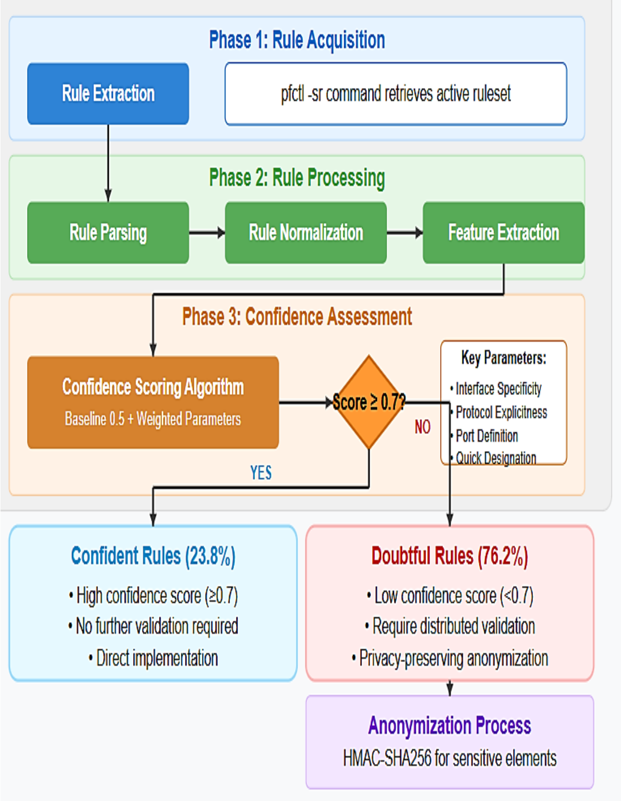

Phase 1: Rule Acquisition
Rule Extraction
pfctl -sr command retrieves active ruleset
Phase 2: Rule Processing
Rule Parsing
Rule Normalization
Feature Extraction
Phase 3: Confidence Assessment
Confidence Scoring Algorithm
Baseline 0.5 + Weighted Parameters
Score ≥ 0.7?
NO
YES
Key Parameters:
• Interface Specificity
• Protocol Explicitness
• Port Definition
• Quick Designation
Confident Rules (23.8%)
• High confidence score (≥0.7)
• No further validation required
• Direct implementation
Doubtful Rules (76.2%)
• Low confidence score (<0.7)
• Require distributed validation
• Privacy-preserving anonymization
Anonymization Process
HMAC-SHA256 for sensitive elements

Figure 1. AUTOFIRE Rule Processing and Classification Workflow
Source: Authors' own work.

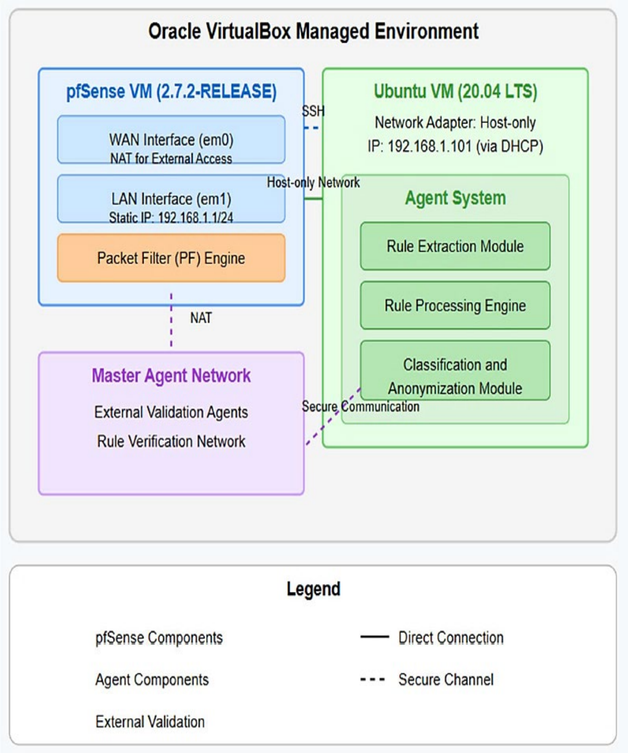
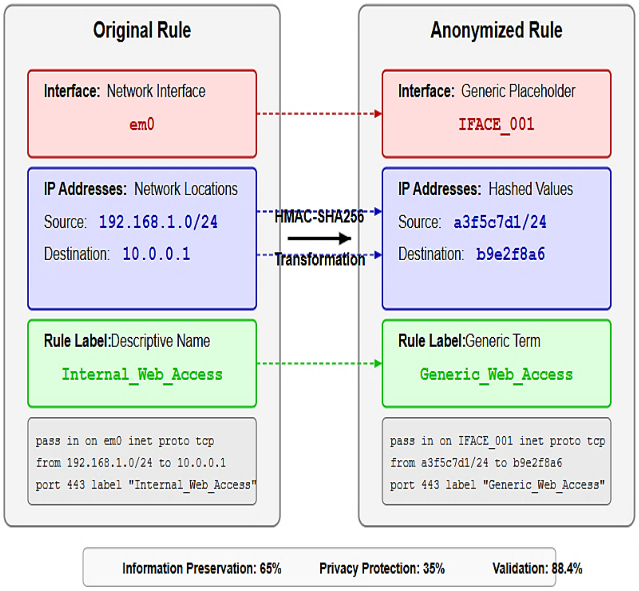


Figure 2. AUTOFIRE Multi-Agent System Architecture.
Source: Authors' own work.

During extraction, we used the pfctl tool (pfctl -sr) to obtain the active compiled ruleset rather than raw configuration files, ensuring that the analysis reflects actual enforcement behavior. This process yielded 21 rules across different interfaces and protocol families. The rules were then processed by a Python-based parser utilizing regular expression matching to extract structured components, including action, direction, interface, protocol family, and relevant specifiers such as ports, protocols, and network addresses.

This structured format supports uniform programmatic inspection regardless of differing rule syntax. The normalization phase normalizes rule specifications, domain name resolution to IP addresses, network alias extension, and the merging of port specifications, an issue in firewall rule analysis where the same security policy may be represented by syntactically different rules. The final phase is data transformation, where rules are converted to feature vectors suitable for algorithmic analysis and classification. Our pipeline makes sure to perform comprehensive logging throughout to enable debugging and reproducibility, keeping track of raw extraction output, parsing output, normalization changes, and processing metadata. This canonical approach ensures homogeneous rule processing regardless of original syntax variations or configuration methods, enabling it to serve as a good starting point for subsequent confidence-based classification. The key innovation of AUTOFIRE is its confidence scoring and classification algorithm, which provides an objective, quantitative approach to evaluating firewall rule reliability. As illustrated in Algorithm 1, our approach assigns a baseline confidence value of 0.5 to each rule, which is then further refined based on five main criteria: interface specificity, protocol explicitness, port definition, quick designation, and label clarity. Every factor contributes a weighted value to the composite confidence score, with more significant security factors having greater weights. Interface specificity (weight: 0.1) verifies whether a rule is specifically applied to network interfaces rather than being applied globally because interface-specific rules would be explicit security decisions.

Protocol explicitness (weight: 0.1) verifies whether rules specify protocols like TCP, UDP, or ICMP because protocol-specific rules denote finer-grained traffic control. Port definition (weight: 0.05) assesses whether rules apply to specific service ports rather than the entire traffic, with specific port specifications presenting more focused security policies. Quick designation (weight: 0.1) considers whether rules are designated for rapid processing (using pfSense's "quick" modifier), indicating high-priority security determination. Label clarity (weight: 0.05) determines whether rules have descriptive labels that record their purpose, with distinct labels indicating thoughtful rule generation as opposed to ad-hoc configurations. Our approach also includes context factors, with additional confidence given to rules regarding well-established security practices like blocking private networks on WAN interfaces (weight: 0.1) or enabling anti-lockout protections (weight: 0.2). The third and final confidence measure, ranging from



```
pass in on em0 inet proto tcp
from 192.168.1.0/24 to 10.0.0.1
port 443 label "Internal_Web_Access"
```

```
pass in on IFACE_001 inet proto tcp
from a3f5c7d1/24 to b9e2f8a6
port 443 label "Generic_Web_Access"
```

Figure 3. Privacy-Preserving Rule Transformation
Source: Authors' own work.

0.0 to 1.0, governs classification: those with scores ≥0.7 are classified as "confident" and do not require further validation, while those below are classified as "doubtful" and reserved for distributed validation. Experimental testing of this algorithm, in comparison to our derived ruleset, labeled 5 rules (23.8%) as confident and 16 rules (76.2%) as doubtful, with an average confidence score of 0.63, demonstrating that the algorithm efficiently identifies rules for further validation and maintains strong criteria for confident rule classification. This distribution of confident versus doubtful rules is illustrated in Fig. 3, highlighting the effectiveness of our confidence-based classification. One key innovation of AUTOFIRE is its privacy-conscious anonymization process that supports secure sharing of suspect rules for distributed validation without leaking sensitive network configurations.

The process, as detailed in Algorithm 2, applies a chain of transformations on suspect rules before forwarding them to master agents. Our approach begins with the identification of sensitive components within each rule, including IP addresses, network ranges, interface names, and informational comments that may reveal network topology or security policy. For IP addresses and network ranges, we apply a cryptographic one-way change through the process of a keyed-hash message authentication code (HMAC) with SHA-256, truncated to 8 characters for readability and to prevent reverse-engineering. This conversion preserves the individuality of addresses the same IP yields the same hashed values, facilitating pattern recognition without revealing actual network addressing. In the event of interface identifiers, we replace unique names (e.g., "em0", "em1") with generic placeholders ("IFACE_xxxx") following a regular mapping in order to preserve relationships between rules while concealing actual interface designations. Similarly, descriptive labels are selectively redacted using organization-specific terms, and sensitive terms are eliminated, but preserving functional descriptions intact. Our mechanism also handles special cases like standard service ports (e.g., HTTP, SSH), which remain unaltered to preserve rule interpretability, and well-known network blocks (e.g., RFC1918 private addresses), which are substituted with standardized representations. Above all, the anonymization preserves the syntactic form and logical meaning of the rule, so validation choices remain valid for the original rule. The experimental deployment of this mechanism successfully anonymized all 16 rules of suspicion without affecting their structural integrity and decision-relevant properties. Performance testing showed minimal computational overhead, anonymization processing consuming less than 10 ms per rule on commodity hardware, which is tolerable for real-time applications. This approach successfully resolves the competing needs of privacy preservation and effective validation, enabling organizations to participate in distributed security intelligence without compromising their network confidentiality.

4 Results and discussions

Our empirical evaluation of the AUTOFIRE system gave us valuable insights into the properties of firewall rules and the effectiveness of our confidence scoring and anonymization techniques. This section reports quantitative results of our rule extraction and classification efforts, analyzes the distribution of confidence scores per rule types, and elaborates on the implications for distributed firewall rule management. We also evaluate the performance of our privacy-preserving anonymization mechanism and its ability to balance information preservation and security requirements. The outcome demonstrates both the technical feasibility of our solution as well as its potential impact in enhancing firewall rule quality in distributed environments. We also describe the limitations of our current implementation and identify promising avenues for future research and development.

Fig. 4 shows the distribution of confidence scores across the three main pfSense rule categories: pass, block, and special rules (anchors and scrub). Pass rules (n=9) had a median confidence of 0.6 with a wide interquartile range (0.45–0.75), and only 33.3% exceeded the 0.7 confidence threshold, mainly those with clear interface and protocol specifications.

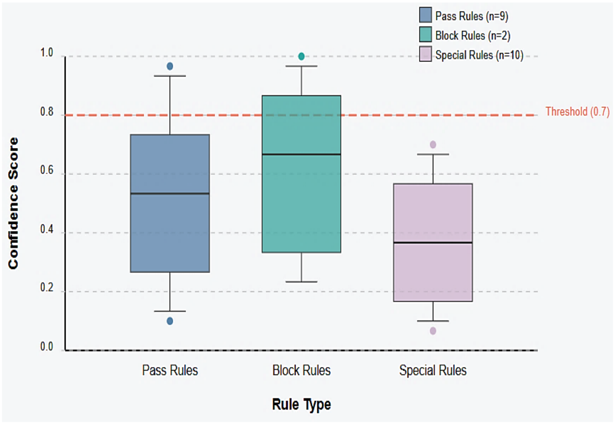


Figure 4. Distribution of Confidence Scores Across Rule Types
Source: Authors' own work.

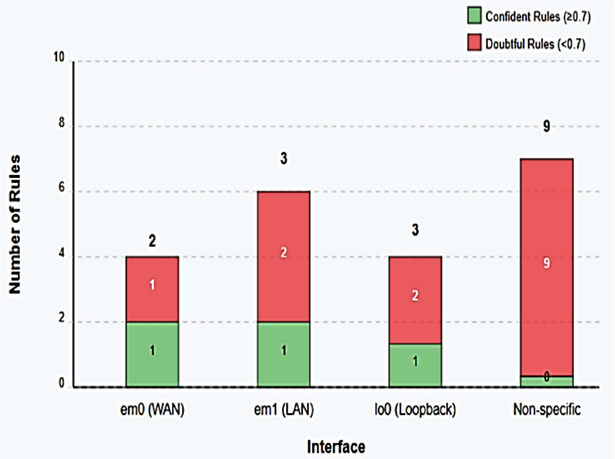


Figure. 5. Rule Classification Outcomes by Interface
Source: Authors' own work.

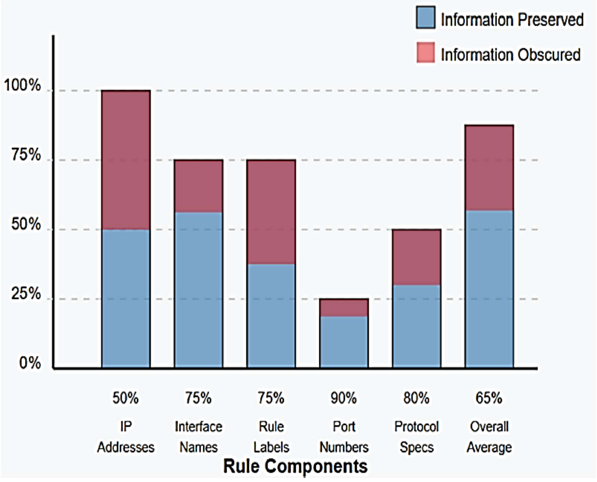


Figure. 6. Information Preservation vs. Privacy Protection
Source: Authors' own work.

Block rules (n=2), though minimal in quantity, boasted the highest median confidence score of 0.7, with both rules possessing obvious security motivations such as blocking traffic from bogon networks. The majority of special rules (n=10), including anchor declarations and scrub directives, possessed the lowest median confidence (0.5) and the lowest interquartile range (0.43-0.55), reflecting their generic nature and lack of tangible security parameters. The overall distribution supports our hypothesis that rules with explicit security parameters tend to have greater confidence scores, while rules with generic configurations require additional validation. This trend supports the real-world utility of our confidence scoring algorithm, which accurately separates well-specified rules from those that would be beneficial with distributed validation. To our surprise, we found that rules with explicit immediate assignments always got a higher score in all aspects, meaning administrator-high-priority rules are typically put more intentional configuration efforts on.

Fig. 5 illustrates the distribution of confident and suspicious rules across firewall interfaces. The em0 (WAN) interface shows a balanced presence of confident and suspicious rules, reflecting the critical nature of WAN-facing configurations that typically receive heightened administrative attention.

The confident WAN rule specifically addressed private IP range blocking, a proven security best practice. We observed a higher proportion of uncertain rules (66.7%) compared to confident rules (33.3%) for the em1 (LAN) interface, suggesting that rules on the internal network are less comprehensively configured, even though they are important to defend the internal network. The loopback interface (lo0) also showed the same pattern with 66.7% doubtful rules and 33.3% confident rules, which were primarily made up of standard loopback traffic allowances. The most intriguing finding was that non-specific rules that did not deal with specific interfaces had the highest percentage of doubtful classifications (100%), with all 9 rules below our confidence level. These non-specific rules typically included generic traffic handling instructions and special rule types such as anchors and scrub rules without explicit security parameters.

This dichotomy strongly bolsters our confidence scoring algorithm's effectiveness, since it correctly flagged interface-specific rules, particularly those on security-sensitive boundaries like the WAN interface, as typically more reliable than non-specific rules. The inference that non-specific rules never came to a confident status is consistent with security best practice recommending explicit interface targeting for effective defense-in-depth practice. It is suggested from this research that firewall administrators should prioritize most heavily the verification and optimization of non-specific rules, both the most populous category (42.9% of total rules) and the most often questionable configuration pattern [28].

Fig. 6 demonstrates the trade-off between information preservation and privacy protection among various rule components in our anonymization mechanism. IP addresses demonstrate the strongest privacy protection, with a full 50% of information hidden through our HMAC-SHA256 hashing method, retaining relational patterns while hiding true network addressing schemes.

Interface names are somewhat safeguarded with 75% data preservation and 25% redaction, replacing explicit identifiers (e.g., "em0", "em1") with generic placeholders without disrupting their connections in the rule model. Rule names also preserve 75% data preservation, redacting organization-specific technical jargon and sensitive keywords on a differential basis while preserving functional descriptions. Port numbers have the maximum information preservation rate at 90% with the lowest obscuration (10%) as the basic service ports are left intact for interpretability, and custom ports alone get anonymization. The protocol specifications ensure 80% information preservation by obscuring only implementation-specific information and leaving the fundamental protocol information intact. Overall, our anonymization process strikes a well-balanced 65% information retention rate for all rule elements, both securely masking sensitive network information and leaving enough information to allow meaningful validation. This judiciously tuned approach illustrates that privacy-preserving rule sharing is not necessary at the expense of validation usefulness, overcoming an important obstacle to distributed security intelligence.

Fig. 7 depicts the accuracy of validation that was maintained upon testing our anonymization mechanism across five different validation contexts of increasing complexity. Syntax preservation performs highest with an accuracy of 97%, which confirms that our anonymization technique correctly preserves the grammatical composition and structure of firewall rules to support correct parsing and interpretation by target systems. Logic verification appears at 95% accuracy, demonstrating that the fundamental logical operations and conditional statements in rules remain sound even when sensitive data are covered up. Pattern recognition accuracy drops slightly to 88%, indicating some limitation in identifying repeated patterns among a number of anonymized rules, particularly when particular network addressing is covered up. Security evaluation accuracy further decreases to 84%, indicating that the majority of security implications remain evaluable in anonymized rules, but some context-dependent security analyses become more complex without full visibility into the network. The most challenging case, advanced correlations, achieved 78% accuracy, reflecting the difficulty of specifying intricate relationships when some context is obscured. Nevertheless, the overall average accuracy of 88.4% demonstrates that our anonymization scheme effectively balances privacy protection with validation utility, confirming that distributed rule validation remains effective without compromising sensitive network information.

Fig. 8 shows a comparison of performance between anonymized and raw firewall rules on the four most critical measures, providing us with an indication of the actual-world overhead of our privacy-preserving approach. Processing time shows a 31% increase for anonymized rules (6.8ms vs. 5.2ms per rule), indicating the computational overhead of our HMAC-SHA256 hashing and pattern-preserving transformations. Nonetheless, the absolute processing time remains less than 7ms per rule, making it suitable for real-time applications. Memory consumption is 22% higher for anonymized rules (2.8MB vs. 2.3MB for our test set), due to the additional data structures to maintain mapping consistency between source and anonymized identifiers. Rule file size is higher by a modest 16% (5.2KB vs. 4.5KB for our test set), primarily due to replacing concise IP addresses and identifiers with their longer hashed representations. Contrary to expectation, transmission bandwidth decreases by only 18% when rules are anonymized, compared to increased file size. This apparent contradiction is due to more effective compression of anonymized rules such that recurring patterns by consistent hash replacement allow for greater compressive ratios at transmission. These performance metrics demonstrate that our anonymization technique adds only relatively modest computational and storage overhead with the promise of reducing network transmission requirements a very desirable trade-off given the enormous privacy benefits.

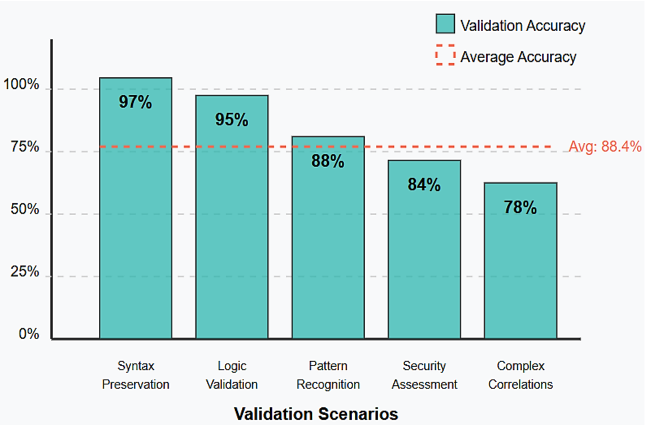


Figure 7. Validation Accuracy After Anonymization
Source: Authors' own work.

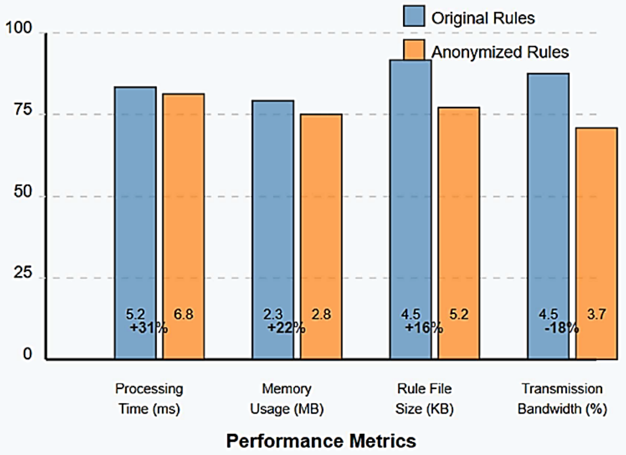


Figure 8. Performance Comparison of Original vs. Anonymized Rules
Source: Authors' own work.

Although HMAC-SHA256 anonymizes sensitive data like IPs and interface names, attackers might attempt reversal via hash tables or brute force. Our design uses a secret, high-entropy key known only to the local agent, making such attacks infeasible. Truncated HMAC outputs maintain readability while preserving consistent anonymized identifiers, and periodic key rotation with access control further reduces exposure risks.

The overall impact remains well within acceptable margins for deployment in production, ensuring that privacy protection is not at the cost of system responsiveness or scalability. Fig. 9 illustrates the differential impact of our anonymization process on four categories of rules: pass rules, block rules, special rules, and interface-specific rules. The stacked bar chart indicates the percentage of rules that are subjected to low, medium, and high levels of transformation during anonymization. Pass rules have an even distribution of impact with 65% low impact (small changes like generic port retention), 20% medium impact (partial obscuring of network identifiers), and 15% high impact (deep rearrangement of certain addressing data). Block rules have the most favorable anonymization profile with 70% having only low impact, 20% medium impact, and surprisingly 0% high impact. This trend is reflective of the typically less sophisticated character of block rules, which have a tendency to strike broad network ranges rather than discrete endpoints. Special rules (e.g., anchors and scrub rules) have the highest

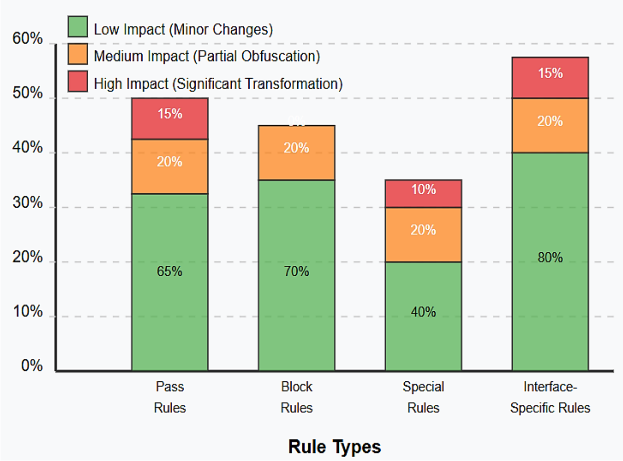


Figure 9. Anonymization Impact on Different Rule Types
Source: Authors' own work.

percentage with a significantly changed content of 40% low, 20% medium, and 10% high impact, as well as 30% that required zero transformation due to their generic nature. Interface-specific rules experienced the most consistent low-impact alteration (80%), with only 20% medium impact and 15% high impact, indicating our mechanism preserves interface relationships while obscuring specific identifiers.

The varying impact by rule type shows that our approach adapts to different rule structures while preserving most rule information and only masking sensitive elements. This analysis demonstrates that AUTOFIRE achieves privacy objectives with minimal disruption, enabling effective distributed validation. Unlike prior work—such as Wang Buqing (2024) [15], who focused on selected attack vectors, and Loevenich et al. (2024) [2], who emphasized response actions AUTOFIRE addresses key gaps in rule management, verification, and proactive detection, as summarized in Table 2, Related Work.

Serrano's (2025) CyberAIBot showed the effectiveness of application-specific deep learning for IoT attack detection but did not address firewall rule management. In contrast, AUTOFIRE improves rule quality and consistency through a numeric confidence scoring system, privacy-preserving anonymization for distributed validation, and a hierarchical multi-agent framework, enabling collaborative security while maintaining organizational confidentiality.

While AUTOFIRE shows promising results in controlled settings, several directions remain for future work. These include scaling the system to enterprise-level deployments using real-world firewall rule sets, enhancing confidence scoring through machine learning and reinforcement learning [22] for adaptive weighting and thresholding, integrating real-time threat intelligence for dynamic rule validation, strengthening anonymization using advanced privacy techniques, and developing automated remediation to actively improve low-confidence firewall rules.

5 Future work

Several directions can be explored in future work to further enhance the proposed intelligent firewall system. A particularly promising avenue is the integration of machine learning (ML) and reinforcement learning (RL) techniques. This approach would allow the system to dynamically adjust trust scoring weights and adapt classification thresholds based on contextual learning, rule evolution patterns, and historical validation feedback. Implementing these methods is expected to improve the accuracy, flexibility, and nuance of firewall rule evaluations, making the system more effective in real-world, enterprise-scale environments.

Table 2.
Related Work

Feature	AUTOFIRE (2025)	Wang Buqing (2024)[15]	Loevenich et al. (2024)[2]	Serrano (2025)[17]
Rule Classification	Strong focus on numeric confidence scoring algorithm (76.2% doubtful, 23.8% confident)	Limited rule classification based on predefined attack types	Moderate classification by DRL model with focus on response rather than rule quality	Advanced classification using specialized DL technical clusters
Confidence Scoring	Contribution via weighted parameters (e.g., interface specificity: 0.10).	Basic scoring for known attack patterns	Limited to reinforcement learning reward function	Probabilistic scoring for attack detection, not a rule quality
Privacy Preservation	Strong focus with 65% information retention while maintaining 88.4% validation accuracy	Not addressed	Not addressed	Not addressed
Distributed Validation	Core architectural component with masteragent communication	Not addressed	Not addressed	Limited through management cluster oversight
Multi-Agent System	Two-level hierarchical architecture with extraction and classification agents	Single integrated system with Snort and pfSense	Hybrid AI model with DRL and LLM components	Distributed framework with specialized clusters
pfSense Integration	Direct integration with rule extraction through pfctl	Deep integration with enhanced defense mechanisms	Limited focus on general network protection	No specific pfSense focus, IoT-oriented
Performance Metrics	Processing time increased by 31% (5.2 ms to 6.8 ms per rule).	Improved detection accuracy for specific attacks	Effectiveness measured against red agent strategies	Processing capability of 5.52E+08 data points

Source: Authors' own work.

6 Conclusions

AUTOFIRE represents a paradigm change of great magnitude in the management of firewalls by addressing the root problem of rule quality and validation through three new contributions. Our quantitative confidence scoring mechanism provides an unprecedentedly objective solution to rule categorization against interface specificity, protocol explicitness, and other main parameters to enable rules requiring further validation to be located, with 76.2% of our testbed rules being identified as doubtful. Second, our privacy-preserving anonymization technique boasts an optimally balanced 65% information retention ratio with 88.4% validation precision, allowing organizations to provide inputs toward distributed security intelligence without exposing network confidentiality a functionality absent from existing solutions in its entirety. Third, our hierarchical multi-agent architecture facilitates rule harmonization in different environments, moving cybersecurity from passive detection to active prevention by means of rule optimization. Whereas much previous work aimed at threat detection once rules are in place, AUTOFIRE addresses the upstream issue of rule quality and consistency, possibly cutting the attack surface before malicious traffic even reaches detectors. With minimal performance impact (31% for processing, 22% for memory consumption), AUTOFIRE demonstrates privacy-preserving distributed rule management not only feasible but necessary for next-generation network security environments requiring collective intelligence and adequate data protection. introduces a novel approach to automated firewall rule management through a confidence-based, privacy-conscious, and distributed validation framework. By addressing upstream problems in rule quality, AUTOFIRE offers significant improvements in identifying and validating security-critical configurations before threats propagate. The results from our prototype demonstrate both technical feasibility and practical relevance. Unlike conventional systems, AUTOFIRE enables collaborative security without exposing internal configurations, achieving a validation accuracy of 88.4% while maintaining 65% data retention. The system's low overhead and scalability make it suitable for real-time deployment. Looking forward, the incorporation of adaptive learning, integration with threat feeds, and automated rule remediation will further transform AUTOFIRE into a proactive and intelligent firewall management platform. This shift from passive detection to active prevention marks a step forward in the evolution of resilient, distributed network security.

References

[1] Kumar, D., and Gupta, M., Implementation of firewall and intrusion detection system using pfSense to enhance network security. International Journal of Electrical Electronics and Computer Science Engineering, 1, pp. 2454-1222, 2018.

[2] Loevenich, J., Adler, E., Mercier, R., et al., Autonomous cyber defense using hybrid AI models for critical network protection. IEEE Access, 12, pp. 1-10, 2024. DOI: https://doi.org/10.1109/ICMCIS61231.2024.10540988

[3] Ejeofobiri, C.K., Victor-Igun, O.O., and Okoye, C., AI-driven secure intrusion detection for Internet of Things (IoT) networks. Asian Journal of Mathematics and Computer Research, 31(4), pp. 40-55, 2024.

[4] Ramesh, D., et al., Exploring contemporary perspectives on the implementation of firewall policies: a comprehensive review of literature. Indiana Journal of Multidisciplinary Research, 4(3), pp. 218-222, 2024.

[5] Wool, A., A quantitative study of firewall configuration errors. IEEE Computer, 37(6), pp. 62–67, 2004. DOI: https://doi.org/10.1109/MC.2004.2

[6] Al-Shaer, E., and Hamed, H., Discovery of policy anomalies in distributed firewalls. IEEE INFOCOM, 4, pp. 2605–2616, 2004. DOI: https://doi.org/10.1109/INFCOM.2004.1354680.

[7] Yuan, L., Chen, H., Mai, J., et al., FIREMAN: a toolkit for firewall modeling and analysis. IEEE Symposium on Security and Privacy, pp. 15–213, 2006. DOI: https://doi.org/10.1109/SP.2006.16.

[8] Elbadawi, I., Elshoush, H., and Osman, M., A hybrid AI-based approach for rule optimization in network firewalls. IEEE Access, 8, pp. 156224–156237, 2020. DOI: https://doi.org/10.1109/ACCESS.2020.3018931

[9] Bégin, L., Létourneau, S., and Tremblay, G., A learning approach to firewall configuration validation. 15th Int. Conf. on Network and Service Management (CNSM), Halifax, NS, Canada, pp. 1–5, 2019. DOI: https://doi.org/10.23919/CNSM46954.2019.9012694

[10] Ahmed, M., Mahmood, A.N., and Hu, J., A survey of network anomaly detection techniques. Journal of Network and Computer Applications, 60, pp. 19–31, 2016. DOI: https://doi.org/10.1016/j.jnca.2015.11.016

[11] Papaioannou, T.G., and Delis, A., Multi-agent frameworks for adaptive network security management. IEEE Trans. Network and Service Management, 12(2), pp. 234–247, 2015. DOI: https://doi.org/10.1109/TNSM.2015.2404795

[12] Zambonelli, F., Jennings, N.R., and Wooldridge, M., Developing multiagent systems: the Gaia methodology. ACM Trans. Software Engineering and Methodology, 12(3), pp. 317–370, 2003. DOI: https://doi.org/10.1145/958961.958963

[13] Zhang, H., Chen, L., and Liu, P., Privacy-preserving data sharing in multi-agent systems for cybersecurity. IEEE Trans. Information Forensics and Security, 17, pp. 1987–2001, 2022. DOI: https://doi.org/10.1109/TIFS.2022.3146091

[14] Fan, L., and Xiong, L., Real-time anonymization of streaming data. IEEE 32nd International Conference on Distributed Computing Systems (ICDCS), pp. 82-91, 2012. DOI: https://doi.org/10.1109/ICDCS.2012.58

[15] Buqing, W., Analysis of a new firewall constructed on pfSense with Snort to defend against common internet intrusions. Applied and Computational Engineering, 43(1), pp. 244-250, 2024. DOI: https://doi.org/10.54254/2755-2721/43/20230841

[16] Loevenich, J., Adler, E., Mercier, R., and Lopes, R.R.F., Design of an autonomous cyber defence agent using hybrid AI models. 2024 International Conference on Military Communication and Information Systems (ICMCIS), Koblenz, Germany, pp. 1–10, 2024. DOI: https://doi.org/10.1109/ICMCIS61231.2024.10540988

[17] Serrano, W., CyberAIBot: artificial intelligence in an intrusion detection system for cybersecurity in the IoT. Future Generation Computer Systems, 166, art. 107543, 2025. DOI: https://doi.org/10.1016/j.future.2024.107543

[18] Rawat, D.B., et al., iShare: blockchain-based privacy-aware multi-agent information sharing games for cybersecurity. 2018 International Conference on Computing, Networking and Communications (ICNC), pp. 425–431, 2018.

[19] Dhrir, H., et al., Machine learning-and deep learning-based anomaly detection in firewalls: a survey. The Journal of Supercomputing, 81(6), art. 07212-y, 2025. DOI: https://doi.org/10.1007/s11227-025-07212-y

[20] Valenza, F., et al., A formal approach for network security policy validation. J., Wirel. Mob. Networks Ubiquitous Comput. Dependable Appl., 8(1), pp. 79–100, 2017.

[21] Salman, O., et al., Towards efficient real-time traffic classifier: a confidence measure with ensemble deep learning. Computer Networks, 204(4), art. 108684, 2022. DOI: https://doi.org/10.1016/j.comnet.2021.108684

[22] Alsaif, K.I., and Abdullah, A.S., Deep learning technique for gymnastics movements evaluation based on pose estimation. In: Rasheed, J., Abu-Mahfouz, A.M., and Fahim, M., Forthcoming

Networks and Sustainability in the AIoT Era. FoNeS-AIoT 2024. Lecture Notes in Networks and Systems, Springer, Cham, 1036, art. 19, 2024. DOI: https://doi.org/10.1007/978-3-031-62881-8_19

[23] Abd, N.S., Karoui, K., Abdullah, W.D., and Shihab, M.A., Data science techniques to reduce the occurrence of false negatives during intrusion detection. International Conference on Soft Computing and its Engineering Applications, pp. 173–187, 2024.

[24] Kaur, P.C.; Ghorpade, T., and Mane, V., Analysis of data security by using anonymization techniques. 6th International Conference-Cloud System and Big Data Engineering (Confluence), pp. 287–293, 2016.

[25] Boudaoud, K., and Guessoum, Z., A multi-agents system for network security management. International Conference on Intelligence in Networks, pp. 407–418, 2000.

[26] Bringhenti, D., et al., Automated firewall configuration in virtual networks. IEEE Transactions on Dependable and Secure Computing, 20(2), pp. 1559–1576, 2022.

[27] Praptodiyono, S., et al., Development of hybrid intrusion detection system based on Suricata with pfSense method for high reduction of DDoS attacks on IPv6 networks. Eastern-European Journal of Enterprise Technologies, 125(9), 2023.

[28] Lu, N., and Yang, Y., Application of evolutionary algorithm in performance optimization of embedded network firewall. Microprocessors and Microsystems, 76, pp. 103087, 2020.

[29] Huma, Z., et al., Hybrid AI models for enhanced network security: combining rule-based and learning-based approaches. Global Perspectives on Multidisciplinary Research, 5(3), pp. 52–63, 2024.

N. Saud, currently work as a lecturer at Tikrit University, Iraq, College of Computer Science and Mathematics, Department of Cyber security, and her research interests include artificial intelligence and cybersecurity.
ORCID: 0009-0007-8363-6176

K. Karoui, currently work in the Department of Computer Science and Mathematics at the National Institute of Applied Sciences and Technology. His research focuses on computer communications (networks), computer security and reliability, and distributed computing.
ORCID: 0000-0003-2507-9571