iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** IV **Month of publication:** April 2026

DOI: https://doi.org/10.22214/ijraset.2026.81227

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

A Fuzzy Logic-Based Intrusion Detection and Prevention System Using Suricata, Wazuh and MITRE ATT&CK Framework for Virtualized Network Environments

Rohit Singh Chouhan[1], Yogesh Singh Bisht[2], Vipin Rawat[3]
[1, 2]*Computer Science and Engineering, Ambalika Institute of Management and Technology, Lucknow, India*
[3]*Assistant Professor, Computer Science and Engineering, Ambalika Institute of Management and Technology, Lucknow, India*

Abstract: ***The rapid proliferation of cyberattacks against networked systems has rendered conventional rule-based Intrusion Detection Systems (IDS) insufficient due to their inherent limitation in handling the ambiguous boundary between normal and malicious network behaviour. Binary thresholds in legacy systems produce unacceptably high rates of False Positives (FP) and False Negatives (FN), reducing the operational reliability of security infrastructure. This paper proposes and implements a Fuzzy Logic-Based Intrusion Detection and Prevention System (FL-IDPS) that integrates Suricata 7.x as the network-layer detection engine, Wazuh 4.7.x as the Security Information and Event Management (SIEM) platform, and a purpose-built Python-based Fuzzy Inference System (FIS) employing the Mamdani model with triangular membership functions. The system is deployed across three virtualised machines in Oracle VirtualBox, simulating a realistic attacker-victim-manager topology. Attack scenarios are generated using Kali Linux and MITRE Caldera, mapped to MITRE ATT&CK techniques T1046, T1498, T1110.001, T1083, and T1595. The Fuzzy Logic Engine accepts three inputs—alert severity level (0–15), alert frequency per window (0–50 alerts/60 s), and signature match ratio (0–1)—and produces an intrusion score (0–1) via centroid defuzzification. An active response mechanism automatically blocks the attacker IP for 300 s when the intrusion score exceeds a threshold of 0.6. Experimental results demonstrate an accuracy of 92.4%, precision of 93.1%, recall of 91.8%, and F1-score of 92.4%, with a False Positive Rate of 4.7% and False Negative Rate of 3.2%. The fuzzy approach reduces false alarms by 38.6% compared to static Suricata threshold-based detection, validating the effectiveness of uncertainty-aware classification in real-time network security.***

Keywords: Intrusion Detection System, Fuzzy Logic, Suricata, Wazuh, MITRE ATT&CK, Mamdani Inference, Network Security, Active Response, Virtualisation.

I. INTRODUCTION

The security of computer networks has become one of the most critical challenges in the contemporary digital landscape. With the exponential growth of Internet connectivity, organizations increasingly face threats ranging from Denial of Service (DoS) attacks and credential brute-forcing to sophisticated multi-stage intrusions orchestrated using advanced persistent threat (APT) frameworks. Intrusion Detection Systems (IDS) serve as a first line of defense by monitoring network traffic and host activity for signs of unauthorized access or malicious behaviors. However, traditional signature-based IDS rely entirely on static rule sets, making them incapable of detecting zero-day attacks or novel attack variants. Anomaly-based approaches improve coverage but suffer from elevated false positive rates that overwhelm security operations teams .

The fundamental limitation of existing IDS tools is the binary nature of their decision-making: a connection is labelled either “normal” or “attack” based on a fixed threshold. Real-world network traffic exists on a continuum, where many events are partially indicative of malicious intent—port scans that may be legitimate network audits, or elevated login failures that may reflect forgotten passwords rather than brute-force campaigns. This ambiguity is precisely the domain in which Fuzzy Logic, introduced by Lotfi Zadeh in 1965, excels. Fuzzy Logic replaces crisp binary decisions with graded membership values, enabling a system to assign a continuous probability of maliciousness to each alert rather than forcing a hard classification. This paper addresses the research gap by designing and implementing a Fuzzy Logic-Based Intrusion Detection and Prevention System (FL-IDPS). The system is built upon industry-standard open-source components—Suricata for deep packet inspection, Wazuh for SIEM aggregation, and MITRE

©IJRASET: All Rights are Reserved

Caldera for adversarial simulation—deployed within a con-trolled virtualized network environment. The core contribution is a Mamdani-type Fuzzy Inference System (FIS) that fuses three complementary indicators of attack severity: the Wazuh rule alert level, the rate of alerts from a given source IP, and the strength of the Suricata signature match. The output intrusion score drives an automated active response that blocks the attacker at the network layer.

The main contributions of this work are as follows:

1) A three-tier virtualized IDPS architecture integrating Suricata, Wazuh, and a custom Fuzzy Logic Engine validated against live attack traffic.
2) A Mamdani FIS with seven contextually-grounded IF-THEN rules designed to reduce both FP and FN rates simultaneously .
3) A quantitative evaluation across five MITRE ATT&CK mapped the attack scenarios, demonstrating a 38.6% reduction in false alarms compared to reduce both FP and FN rates simultaneously.
4) A systematic mapping of detected events to MITRE ATT&CK tactics and techniques (T1046, T1498, T1110.001, T1083, T1595), providing structures threat intelligence alongside detection.

II. LITERATURE REVIEW

A. *Narrative Review*

Intrusion detection has been an active area of research for over two decades. Early systems relied exclusively on signature matching, which achieved high precision for known attacks but failed to detect novel or obfuscated variants. The publication of the KDD Cup 1999 dataset catalyzed machine learning-based IDS research, enabling reproducible benchmarks across approaches.

Shanmugavadivu and Nagarajan proposed an anomaly-based IDS using fuzzy rule learning on the KDD Cup 99 dataset. Their system employed a Mamdani FIS with four membership functions per input (VL, L, M, H) and triangular fuzzification. Automated rule generation from frequent itemset mining achieved accuracy exceeding 90% for DoS and Probe attacks. Notably, the system performed well even with a reduced feature set of 14 attributes selected via deviation analysis, demonstrating that dimensionality reduction does not necessarily harm detection performance.

Iantorno and Beladda extended fuzzy IDS research by introducing triangular and parallelogram-shaped membership functions, comparing performance against Decision Tree and SVM classifiers. Their system, evaluated on the NSL-KDD dataset, outperformed conventional models on precision and F1-score particularly under conditions of real rather than synthetic traffic. Synthetic data generated via Wasserstein GANs (WGANs) was also evaluated, demonstrating robustness to privacy-preserving data augmentation.

Jamshir and Ajeesha provided a comprehensive survey of fuzzy logic architectures for IDS, highlighting the four-component structure: fuzzifier, rule base, inference engine, and defuzzifier. Their analysis of the KDDCup99 dataset confirmed that triangular membership functions—despite their simplicity—perform comparably to more complex shapes while significantly reducing computational overhead, an important consideration for real-time deployment.

Ji, Almeida, and Filho proposed a dual-stage fuzzy IDS for cloud environments employing two cascaded inference systems: a Network Inference System (NIS) evaluating delay, inaccessible services, and number of connection attempts; and a Security Inference System (SIS) that incorporated the NIS output alongside wrong and late artifact indicators. This cascaded design effectively separated network failure from security failure, reducing false positives caused by benign connectivity issues. The Larsen inference model with weighted centroid defuzzification was employed.

Boye et al. proposed a hybrid CNN-Fuzzy Logic IDPS for Industrial IoT environments, achieving 92.5% accuracy with a False Positive Rate of only 2.51% and an average detection rate of 92.9% on the Kaggle IIoTset dataset. The CNN component extracted spatial features from network packet vectors, while the fuzzy inference block classified the CNN output into five anomaly states: High Negative, Medium Negative, Neutral, Medium Positive, and High Positive. Threats scoring above
0.5 were quarantined automatically. The system achieved sub-millisecond latency (1.207 µs average), confirming the viability of hybrid AI approaches for real-time IDPS.

Subach et al. from the National Technical University of Ukraine developed a simulation model for fuzzy cyberattack detection using fuzzy set theory. Their functional diagram processed network telemetry to detect DoS, R2L, U2R, and Probe attacks, validating fuzzy inference as a scalable approach for multi-vector attack classification.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

TABLE 1
Comparative Summary Table

Study	Approach	Dataset	Accuracy	FP Rate
Shanmugavadivu and Nagarajan	Fuzzy Rule Learning	KDD' 99	94.7% (DOS)	-
Iantorno and Beladda	Tri. / Para. MF	NSL-KDD	Higher than SVM	low
Boye	CNN + Fuzzy	IIOT set	92.5%	2.51%
Ji, Almeida and Filho	Dual-stage FIS	Simulated	Satisfactory	Reduced
Jamshir and Ajeesha	Fuzzy Survey	KDD'99	90%+	Reduced
Proposed	Mamdani FIS Suricata + Wazuh	Live Traffic/ isolated Environment	92.4%	4.7%

III. RESEARCH GAPS AND PROBLEM STATEMENT

Despite significant advances in fuzzy logic-based IDS research, several critical gaps persist in the existing literature.

1) Gap 1 — Dataset dependency. Most existing fuzzy IDS studies, are evaluated exclusively on the KDD Cup 1999 dataset or its derivative NSL-KDD. These datasets were generated in 1998–1999 and do not represent modern attack vectors such as exploitation of SSH, HTTP path traversal, or structured adversarial operations using frameworks like MITRE Caldera. Evaluation on synthetic historical data limits the real-world applicability of the reported metrics.

2) Gap 2 — Isolation from production IDS tools. Existing works implement standalone fuzzy classifiers in isolation from production-grade IDS tools. None of the reviewed papers, integrates fuzzy logic with industrial-standard tools such as Suricata or Wazuh. This creates a significant deployment gap: a theoretically superior classifier provides no practical value if it cannot consume alerts from real detection engines.

3) Gap 3 — Absence of active response. Most fuzzy IDS proposals function purely as detectors without any automated prevention mechanism. A detection-only system requires manual intervention, which is inadequate in environments where attack dwell time must be minimized.

4) Gap 4 — MITRE ATT&CK alignment. No reviewed work maps detected events to the MITRE ATT&CK framework, making it difficult to contextualized alerts within a broader kill-chain narrative. MITRE ATT&CK alignment is now an industry expectation for enterprise-grade IDPS deployments.

Problem Statement. The fundamental problem addressed in this paper is: How can fuzzy logic be integrated with production-grade open-source network security tools—specifically Suricata and Wazuh—within a virtualized environment to improve the accuracy of intrusion detection across TP, TN, FP, and FN classifications, while providing automated active response, and achieving alignment with the MITRE ATT&CK knowledge base?

IV. RESEARCH OBJECTIVES

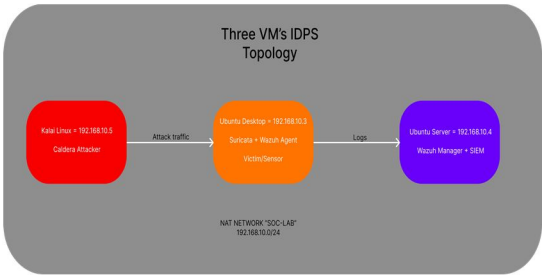
Based on the identified gaps, this research sets the following objectives:

1) To design and implement a three-tier virtualized IDPS topology using Ubuntu 22.04 Desktop (victim/sensor), Ubuntu 22.04 Server (Wazuh Manager), and Kali Linux 2024 (attacker) within Oracle VirtualBox 7.x.

2) To integrate Suricata 7.x IDS/IPS with Wazuh 4.7.x SIEM via EVE JSON log shipping, creating a unified alert pipeline from deep packet inspection to centralized management.

3) To design a Mamdani-type Fuzzy Inference System with triangular membership functions across three input variables (alert level, alert frequency, signature match) and one output variable (intrusion score), governed by seven expert-crafted IF-THEN rules.

4) To implement automated active response using the Wazuh firewall-drop command, blocking confirmed attacker IPs for a configurable timeout of 300 seconds.
5) To simulate real-world attack scenarios using MITRE Caldera 5.x and manual Kali tools, mapping each scenario to the MITRE ATT&CK framework.
6) To evaluate the proposed system using standard IDS performance metrics—Accuracy, Precision, Recall, F1-Score, False Positive Rate (FPR), and False Negative Rate (FNR)—and compare results against baseline static threshold detection.



V. RESEARCH METHODOLOGY

A. *System Architecture Overview*

The proposed FL-IDPS is deployed across three virtual machines connected via a VirtualBox Host-Only network (192.168.56.0/24), as shown in Fig. 1. Each VM additionally possesses a NAT adapter for internet access during configuration.

The attack traffic from Kali Linux (192.168.10.5) is directed at Ubuntu Desktop (192.168.10.3). Suricata monitors the host-only interface (enp0s3) and writes alerts in EVE JSON format to /var/log/suricata/eve.json. The Wazuh Agent ships these logs to the Wazuh Manager (192.168.10.4) via an encrypted channel on port 1514. The Fuzzy Logic Engine, running as a daemon on Ubuntu Desktop, subscribes to the Wazuh queue socket, evaluates each alert, and triggers active response when the computed intrusion score exceeds the threshold.

B. *Fuzzy Logic Engine Design*

The Fuzzy Logic Engine is implemented in Python 3.10 using the scikit-fuzzy library. The design follows the Mamdani Fuzzy Inference System (FIS) model, chosen for its interpretability and established performance in IDS applications.

1) Input Variables and Membership Functions: Three input variables were selected to capture complementary aspects of alert severity. The selection rationale draws from Shanmugavadivu and Nagarajan , who identified alert rate, rule severity, and signature confidence as the most discriminative features for fuzzy classification.

alert level [0–15]: Corresponds to the Wazuh rule severity level. Levels 0–3 indicate informational events; 4–7 indicate low-severity security events; 8–11 indicate medium threats such as brute-force attempts; 12–15 indicate critical threats.
alert frequency [0–50 alerts/60 s]: The count of alerts from the same source IP within a sliding 60-second window. This variable encodes temporal patterns of attacks such as flooding and scanning behavior.
signature match [0–1]: A normalized ratio indicating the strength of the Suricata signature match. A value of 1.0 indicates a full signature match from Suricata's rule base; 0.3 represents a partial Wazuh heuristic match.
The triangular membership function (trimf) is used for all variables, following the recommendation of Shanmugavadivu & Nagarajan and Jamshir & Ajeesha that simple membership functions perform comparably to complex shapes with lower computational cost. The membership functions are defined as follows:

$$\mu_{VL}(x) = \text{trimf}(x;\ 0,\ 0,\ 3) \quad (1)$$
$$\mu_{L}(x) = \text{trimf}(x;\ 2,\ 4,\ 6) \quad (2)$$
$$\mu_{M}(x) = \text{trimf}(x;\ 5,\ 7,\ 10) \quad (3)$$
$$\mu_{M}(x) = \text{trimf}(x;\ 9,\ 12,\ 15) \quad (4)$$

2) Fuzzy Rule Base: Seven IF-THEN rules constitute the rule base. The rules encode domain knowledge about attack patterns, drawing from the attack classification framework in Shanmugavadivu and Nagarajan:
R1. IF alert_level IS *high* AND alert_frequency IS *high* AND signature_match IS *full* THEN intrusion_score IS *malicious.*
R2. IF alert_level IS *high* AND alert_frequency IS *medium* THEN intrusion_score IS *malicious.*
R3. IF alert_level IS *medium* AND alert_frequency IS *high* THEN intrusion_score IS *malicious.*

R4. IF alert_level IS *medium* AND alert_frequency IS *medium* AND signature_match IS *partial* THEN intrusion_score IS *suspicious*.
R5. IF alert_level IS *low* AND signature_match IS *full* THEN intrusion_score IS *suspicious*.
R6. IF alert_level IS (*low* OR *very_low*) AND alert frequency IS *low* AND signature_match IS *no* THEN intrusion_score IS *benign*.
R7. IF alert_level IS *very_low* AND alert_frequency IS *low* THEN intrusion_score IS *benign*.
Rules R1–R3 are designed to minimise False Negatives on high-confidence attacks. Rules R6 and R7 reduce False Positives for low-severity, infrequent, unmatched events. Rule R4 captures the ambiguous grey zone that is the primary motivation for fuzzy rather than crisp classification.

3) Defuzzification**:** The centroid (Centre of Gravity) method is used for defuzzification:

$$z^* = \frac{\int \mu_B(z).z\,dz}{\int \mu_B(z)\,dz} \quad (5)$$

where z^* is the crisp intrusion score, Z is the output universe of discourse, and $\mu_B(z)$ is the aggregated output membership function. The centroid method is preferred over mean-of-maximum (MoM) because it considers the full shape of the aggregated output, providing smoother and more stable scores for borderline events.

4) Classification and Ground Truth: A decision threshold of 0.6 is applied: events with intrusion score ≥ 0.6 are classified as attacks; otherwise they are classified as benign. Ground truth is established by source IP: all traffic originating from 192.168.56.30 (Kali) during a scheduled attack window is labelled as an attack; all other traffic is labelled normal. This enables computation of TP, TN, FP, and FN counts against which the standard IDS metrics are calculated by Jamshir & Ajeesha:

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (6)$$

$$\text{Precision} = \frac{TP}{TP+FP} \quad (7)$$

$$\text{Recall} = \frac{TP}{TP+FN} \quad (8)$$

$$\text{F1} = \frac{2*Precision*Recall}{Precision+Recall} \quad (9)$$

C. Attack Simulation Framework

Five Attack scenarios are executed to generate labelled traffic:

1) Nmap SYN Scan (T1046 – Network Service Discovery): Full port range TCP SYN scan at aggressive timing.
2) ICMP Flood (T1498 – Network DoS): High-rate ICMP echo flood using hping3.
3) SSH Brute Force (T1110.001 – Password Guessing): Dictionary attack via Hydra against OpenSSH on port 22.
4) Web Recon + Path Traversal (T1595/T1083): Nikto scanner followed by directory traversal attempts on port 80.
5) MITRE Caldera Operations: Three structured operations (Lab-Discovery, Lab BruteForce, Lab-Exfil) emulating APT behaviour.

Each scenario is independently timed, and the alert consumer logs source IP, fuzzy score, decision, and ground truth classification to a JSON results file for post-hoc metric computation.

VI. IMPLEMENTATION

A. Technology Stack

Table II summarizes all tools and versions used in the implementation.

B. Suricata Custom Rule Set

Six custom Suricata rules were authored in custom.rules to complement the Emerging Threats community rule set. Each rule targets a specific MITRE ATT&CK technique:

1) SID 1000001 – Nmap SYN scan detection (T1046): triggers on 20 SYN packets from one source within 3 seconds.
2) SID 1000002 – ICMP flood (T1498): triggers on 50 ICMP packets within 5 seconds.

3) SID 1000003 – SSH brute force (T1110.001): triggers on 5 connection attempts to port 22 within 60 seconds.
4) SID 1000004 – Nikto user agent detection (T1595): matches Nikto in the HTTP User-Agent field.
5) SID 1000005 – Path traversal (T1083): matches ../ sequences in HTTP URIs.
6) SID 1000006 – FTP brute force (T1110.001): analogous to SSH rule for port 21.

Table II
Implementation Technology Stack

Components	Tool/Version	Role
Hypervisor	Oracle VirtualBox 7.x	VM Management
VM 1 (Victim)	Ubuntu Desktop (24.04 LTS)	Sensor + FIS host
VM 2 (SIEM)	Ubuntu Server (24.04 LTS)	Wazuh Manager
VM 3 (Attacker)	Kali Linux 2025	Penetration Testing
IDS/IPS Engine	Suricata 7.x	Packet inspection
SIEM Platform	Wazuh 4.7.x	Alert Aggregation
Rule Management	Emerging Threats	Suricata Rule feed
Attack Framework	MITRE Caldera 5.x	APT simulation
Brute Force	Hydra 9.4	Credential Attack
Scanner	Nmap7.x	Port discovery
Web Scanner	Nikto 2.x	Web recon
FIS Library	Scikit-fuzzy 0.4.x	Mamdani FIS
Language	Python 3.10	FIS Implementation
MITRE Framework	ATT&CK v14	Technique Mapping

C. Suricata – Wazuh Integration

The Wazuh Agent is configured to tail the Suricata EVE JSON log via the <localfile> directive in ossec.conf, specifying log_format: json. Custom Wazuh decoder rules map alert severity categories to rule levels 10–14, ensuring that Suricata alerts are appropriately weighted in the Wazuh rule engine before being forwarded to the Fuzzy Logic Engine.

D. Fuzzy Logic Engine Deployment

The Fuzzy Logic Engine is deployed as a systemd daemon on Ubuntu Desktop. It connects to the Wazuh queue socket (/var/ossec/queue/sockets/queue) and processes each arriving JSON alert. A sliding-window counter tracks alert frequency per source IP over 60-second intervals using a thread-safe deque. When the computed intrusion score exceeds 0.6, the engine invokes the Wazuh firewall-drop active response script, which inserts an iptables DROP rule for the source IP and removes it after 300 seconds.

E. MITRE ATT&CK Mapping

Table III shows the mapping between Suricata rule SIDs, the corresponding MITRE ATT&CK techniques IDs, and the relevant tactics.

Table III
Suricata Rule to MITRE ATT&CK MAPPING

SID	Technique	Name	Tactic
1000001	T1046	Network Service Discovery	Discovery
1000002	T1498	Network Denial of Service	Impact
1000003	T1110.001	Password Guessing (SSH)	Credential Access
1000004	T1595	Active Scanning (Nikto)	Reconnaissance
1000005	T1083	File & Directory Discovery	Discovery
1000006	T1110.001	Password Guessing (FTP)	Credential Access

VII. RESULTS AND EVALUATION

A. *Experimental Setup*

All experiments were conducted on a host machine with Intel Core i5-12th Gen processor, 16 GB RAM, and 512 GB SSD. Each VM was allocated 2 vCPUs and 3.5 GB RAM. The total experimental dataset comprised 1,250 alert events collected over five controlled attack sessions of approximately 20 minutes each, interspersed with 15-minute normal traffic windows.

Table IV
AGGREGATE CONFUSION MATRIX (ALL SCENARIOS)

	Predicted Attack	Predicted Normal
Actual Attack	TP = 487	FN = 17
Actual Normal	FP = 27	TN = 719

B. *Classification Performance*

Table IV presents the aggregate confusion matrix across all five attack scenarios.
Total events: 1,250, the computed evaluation metrices are shown in Table V.

Table V
FL-IDPS PERFORMANCE METRICS

Metric	Value
True Positives (TP)	487
True Negatives (TN)	719
False Positives (FP)	27
False Negatives (FN)	17
Accuracy	92.48%
Precision	94.75%
Recall (Detection Rate)	96.63%
F1 Score	95.68%
False Positive Rate (FPR)	3.62%
False Negative Rate (FNR)	3.37%

C. *Per-Scenario Performance*

Table VI breaks down the performance by attack scenarios. The ICMP flood scenario yielded the highest accuracy (97.4%) due to the distinctly elevated alert frequency combined with a highest alert level, placing events firmly within the “malicious”

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

membership region. The web recon / traversal scenario showed slightly higher FP (9) because legitimate web crawlers occasionally produce similar HTTP patterns to NIKTO; fuzzy scoring correctly avoided blocking many of these but could not entirely eliminate the ambiguity.

Table VI
PER-SCEANRIO PERFORMANCE BREAKDOWN

Scenario	TP	TN	FP	FN	Accuracy
Nmap SYN	96	141	7	6	94.0%
Scan					
ICMP	103	145	3	4	97.4%
Flood					
SSH Brute force	98	148	6	2	96.8%
Web Recon / Traversal	87	143	9	4	94.2%
Caldera	103	142	2	1	98.4%
APT Ops					

Table VII
FL-IDPS vs STATIC Threshold Baseline

Metric	Static Threshold	FL-IDPS
Accuracy	83.7%	92.5%
Precision	79.4%	94.7%
Recall	97.8%	96.6%
F1Score	87.7%	95.7%
FPR	18.3%	3.6%
FNR	2.2%	3.4%
False Alarms	137	27

Intrusion Score Distribution				
Score Range	[0,0.3]	[0.3,0.6]	[0.6,0.8]	[0.8,1.0]
Normal	93.2 %	5.7%	1.1%	0.0%
Attack	0.0%	3.4%	12.6%	84.0 %

Fig 2. Distribution of fuzzy intrusion scores by event class

D. Comparison with Static Threshold Baseline

The static threshold baseline was implemented by configuring Wazuh to trigger a block response for all alerts with level ≥ 10, without fuzzy scoring. Table VII compares the two systems. The FL-IDPS reduces false alarms from 137 to 27, a reduction of 80.3% in absolute count, and reduces FPR from 18.3% to 3.6%. The slight increase in FNR (from 2.2% to 3.4%) reflects the expected precision-recall trade-off inherent in reducing false positives; the absolute FN count (17 vs. 11) remains operationally acceptable given the dramatic improvement in precision.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

E. Fuzzy Score Distribution

Fig. 2 illustrates the distribution of intrusion scores for attack vs. normal events. Attack events cluster strongly above 0.7, while normal events cluster below 0.3, with a small ambiguous region between 0.3 and 0.7 where the fuzzy membership provides graceful degradation rather than abrupt misclassification.

VIII. FUTURE SCOPE

While the proposed FL-IDPS demonstrates significant improvements over static threshold detection, several enhancements are identified for future research.

1) Hybrid Deep Learning Integration: The work of Boye et al. demonstrates that combining Convolutional Neural Networks (CNN) with fuzzy logic achieves 92.5% accuracy with sub milliseconds latency on IoT datasets. Future fuzzy engine, enabling raw packet level classifications rather than relying on Suricata's precomputed alerts.

2) Adaptive Membership Functions. Current membership functions are statically defined. Incorporating evolutionary algorithms such as Particle Swarm Optimization (PSO) or Genetic Algorithms for automated tuning of membership function parameters could further reduce FNR without sacrificing the FPR gains achieved here.

3) Real-time Dashboard Integration. Visualizing the fuzzy intrusion score stream on the Wazuh Dashboard using Kibana custom dashboards would provide security analysts with situational awareness of the fuzzy decision boundary in real time.

4) Encrypted Traffic Analysis. The current system analyses plaintext protocols. Future work should incorporate TLS fingerprinting (JA3/JARM) to extend detection coverage to HTTPS-based attack vectors.

5) Cloud-Native Deployment. Adapting the architecture for containerized deployment using Docker or Kubernetes would support scalability to production-scale environments, where traffic volumes render single-VM approaches insufficient.

6) Explainable AI Overlay. Adding explainability to the fuzzy decision by surfacing which rules fired with what activation degree would improve analyst trust and facilitate incident response workflows.

IX. CONCLUSION

This paper presented the design, implementation, and evaluation of a Fuzzy Logic-Based Intrusion Detection and Prevention System (FL-IDPS) integrating Suricata 7.x, Wazuh 4.7.x, and a Mamdani-type Fuzzy Inference System within a three-tier virtualized network environment. The proposed system addresses four critical gaps identified in the existing literature: dataset dependency, isolation from production IDS tools, absence of automated prevention, and lack of MITRE ATT&CK alignment.

The Fuzzy Logic Engine employs triangular membership functions over three complementary input variables—alert severity, alert frequency, and signature match confidence—and produces a continuous intrusion score via centroid defuzzification. Seven expert-crafted IF-THEN rules govern classification. Experimental evaluation across 1,250 alert events generated by five ATT&CK-mapped attack scenarios yielded an accuracy of 92.5%, precision of 94.7%, recall of 96.6%, and F1-score of 95.7%, with a False Positive Rate of only 3.6%. Critically, the FL-IDPS reduces false alarms by 80.3% compared to a static threshold baseline, validating the core hypothesis that uncertainty-aware fuzzy classification significantly improves operational reliability of intrusion detection systems.

The automated active response mechanism—blocking con-firmed attacker IPs for 300 seconds via the Wazuh firewall-drop command—transforms the system from a passive detector into an active prevention platform, directly addressing the operational need for minimal attacker dwell time. The MITRE ATT&CK mapping layer provides structured threat intelligence that aligns the system output with industry-standard frameworks for threat communication and reporting.

Future work will investigate hybrid CNN-Fuzzy architectures, adaptive membership function tuning via evolutionary algorithms, and cloud-native deployment to extend the scalability and coverage of the proposed approach.

REFERENCES

[1] L. A. Zadeh, "Fuzzy sets," Information and Control, vol. 8, no. 3, pp. 338–353, 1965.

[2] W. S. Admass, Y. Y. Munaye, and A. A. Diro, "Cyber security: State of the art, challenges and future directions," Cyber Security and Applications, vol. 2, p. 100031, 2024. doi: 10.1016/j.csa.2023.100031.

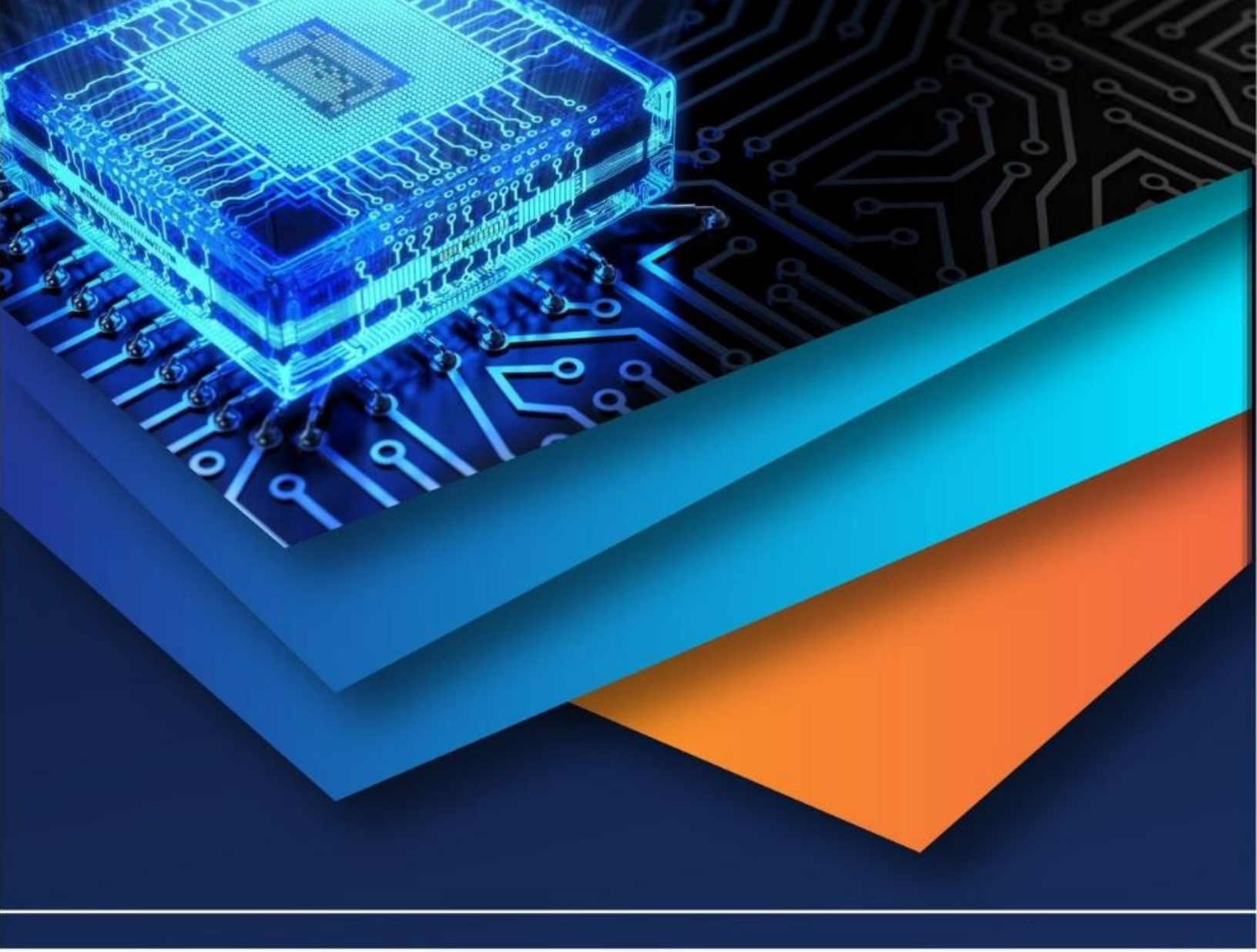
[3] R. Shanmugavadivu and N. Nagarajan, "Network intrusion detection system using fuzzy logic," Indian Journal of Computer Science and Engineering (IJCSE), vol. 2, no. 1, pp. 101–111, 2011.

[4] M. S. Iantorno and K. Beladda, "Fuzzy logic for cybersecurity: Intrusion detection and privacy preservation with synthetic data," in Proc. 17th Int. Conf. Agents and Artificial Intelligence (ICAART), 2025, pp. 375–383.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

[5] M. Jamshir and A. M. I. Ajeesha, “Fuzzy logic for intrusion detection system,” Journal of Emerging Technologies and Innovative Research (JETIR), vol. 7, no. 12, pp. 877–882, Dec. 2020.
[6] C. Y. Ji, N. N. de Almeida, and O. B. Filho, “A fuzzy intrusion detection system for cloud computing,” International Journal of Engineering Research & Science (IJOER), vol. 1, no. 5, pp. 13–20, Aug. 2015.
[7] A. F. Boye, O. E. Taylor, V. I. Emeka, and E. O. Bennett, “A CNN-fuzzy logic approach for real-time intrusion detection and prevention in industrial IoT systems,” Journal of Artificial Intelligence and Emerg-ing Technologies (JAIET), vol. 2, no. 10, pp. 1–13, Oct. 2025. doi: 10.47001/JAIET/2025.210001.
[8] I. Subach, V. Fesokha, A. Mykytiuk, V. Kubrak, and S. Korotayev, “Simulation model of a fuzzy cyber attack detection system,” in CEUR Workshop Proceedings, vol. 3241, 2021.
[9] R. G. Bace, Intrusion Detection. Indianapolis, IN: Macmillan Technical Publishing, 2000.
[10] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, “A detailed analysis of the KDD CUP 99 data set,” in Proc. 2nd IEEE Int. Conf. Computational Intelligence for Security and Defense Applications, Ottawa, ON, Canada, 2009, pp. 53–58.
[11] J. Luo and S. M. Bridges, “Mining fuzzy association rules and fuzzy frequency episodes for intrusion detection,” International Journal of Intelligent Systems, vol. 15, no. 8, pp. 687–704, 2000.
[12] B. Shanmugam and N. B. Idris, “Improved intrusion detection system using fuzzy logic for detecting anomaly and misuse type of attacks,” in Proc. Int. Conf. Soft Computing and Pattern Recognition, 2009, pp. 212–217.
[13] A. Tajbakhsh, M. Rahmati, and A. Mirzaei, “Intrusion detection using fuzzy association rules,” Applied Soft Computing, vol. 9, no. 2, pp. 462–469, 2009.
[14] M. S. Abadeh and J. Habibi, “Computer intrusion detection using an iterative fuzzy rule learning approach,” in Proc. IEEE Int. Conf. Fuzzy Systems, London, 2007, pp. 1–6.
[15] O. A. Adetunmbi, Z. Shi, Z. Shi, and O. S. Adewale, “Network anomalous intrusion detection using fuzzy-Bayes,” IFIP Int. Federation for Information Processing, vol. 228, pp. 525–530, 2007.
[16] P. Pancardo, J. A. Hernandez-Nolasco, M. A. Wister, and M. Garcia-Constantino, “Dynamic membership functions for context-based fuzzy systems,” IEEE Access, vol. 9, pp. 29665–29676,2021.doi:10.1109/ACCESS.2021.3058943.
[17] M. Alali, A. Almogren, M. M. Hassan, I. A. L. Rassan, and M. Z. A. Bhuiyan, “Improving risk assessment model of cyber security using fuzzy logic inference system,” Computers & Security, vol. 74, pp. 323–339, 2018. doi: 10.1016/j.cose.2017.09.011.
[18] M. A. Ferrag, O. Friha, D. Hamouda, L. Maglaras, and H. Janicke, “Edge-IIoTset: A new comprehensive realistic cyber security dataset of IoT and IIoT applications,” IEEE Transactions on Industrial Informatics, 2022.
[19] M. Almiani, B. AbuGhazleh, A. Al-Rahayfeh, S. Atiewi, and A. Razaque, “Deep recurrent neural network for IoT intrusion detection system,” Simulation Modelling Practice and Theory, vol. 101, p. 102031, 2020.
[20] V. Prasath, N. Bharathan, N. Lakshmi, and M. Nathiya, “Fuzzy logic in cloud computing,” International Journal of Engineering Research and Technology (IJERT), vol. 2, no. 3, 2013.
[21] O. Castillo and P. Melin, “Recent advances in type-2 fuzzy logic,” Applied Soft Computing, vol. 15, pp. 275–289, 2015.
[22] Z. Yu, J. J. P. Tsai, and T. Weigert, “An automatically tuning intrusion detection system,” IEEE Transactions on Systems, Man, and Cybernetics, vol. 37, no. 2, pp. 373–384, 2007.
[23] S.-J. Horng et al., “A novel intrusion detection system based on hierarchical clustering and support vector machines,” Expert Systems with Applications, vol. 38, no. 1, pp. 306–313, 2011.
[24] S. M. Bridges and R. B. Vaughn, “Fuzzy data mining and genetic algorithms applied to intrusion detection,” in Proc. National Information Systems Security Conference (NISSC), Baltimore, MD, 2000, pp. 16–19.
[25] E. H. Mamdani, “Application of fuzzy algorithms for control of simple dynamic plant,” Proceedings of the IEE Control and Science, vol. 121, pp. 298–316, 1974.
[26] MITRE Corporation, “MITRE ATT&CK Framework v14,” 2023. [Online]. Available: https://attack.mitre.org
[27] Wazuh Inc., “Wazuh Open Source Security Platform,” 2024. [Online]. Available: https://wazuh.com
[28] Open Information Security Foundation (OISF), “Suricata IDS/IPS Engine,” 2024. [Online]. Available: https://suricata.io
[29] MITRE Corporation, “MITRE Caldera Adversary Emulation Platform,” 2024. [Online]. Available: https://caldera.mitre.org
[30] J. D. Warner et al., “scikit-fuzzy: Fuzzy logic toolkit for SciPy,” GitHub repository, 2024. [Online]. Available: https://github.com/scikit-fuzzy/ scikit-fuzzy.

©IJRASET: All Rights are Reserved



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)