International Journal of Advanced Multidisciplinary Research and Educational Development
Volume 2, Issue 1 | January – February 2026 | www.ijamred.com
ISSN: 3107-6513

AISafeHoneynet: AI-Orchestrated System-Level Defenses for Future-Proof Honeynets Against Botnet Hijacking and DDoS Exploitation with Federated Threat Intelligence

Swapnil Rajesh Bhopi[1], Aarohi Chandrakant Keny[2], Jayesh Shinde[3]
[1]Msc-IT(Cybersecurity) ,[2]Msc-IT (Cybersecurity) ,[3]Professor
University Department of Information Technology, University of Mumbai, Kalina, Maharashtra, India
[1]sbhopi18@gmail.com, [2]aarohikeny14@gmail.com, [3]jayeshshinde@udit.mu.ac.in

Abstract:

AISafeHoneynet introduces a shift - by weaving AI insights into core safeguards, it counters misuse before harm spreads. Most setups gather logs but leave exits unchecked; danger slips through those gaps. Instead of just logging attacks, this approach watches for escape routes attackers might take. Automation often lags, leaving systems open when threats evolve fast. Here, responses adjust in real time, shaped by live traffic patterns. Old models collect data passively; this one act early, guided by behavioral clues. Programmable networks enable tighter rules that adapt without manual input. Safety improves not by adding layers, but by making each layer respond wisely. The result? A honeynet less likely to turn against its owner. Observations from honeypot activity feed into machine learning systems that detect suspicious patterns as they happen. Instead of relying on fixed protocols, responses adapt based on live data, shaping a more responsive defense layer. Unusual outbound traffic gets blocked without disrupting the appearance of vulnerability. Insights gathered during attacks travel across network nodes, strengthening each segment through shared experience. Automation handles threat isolation and reporting, freeing personnel from repetitive oversight tasks. Because judgments emerge from ongoing signals, not pre-set conditions, the system supports deployments that stay resilient amid evolving threats.

Keywords— Honeynet security, botnet hijacking, distributed denial-of-service (DDoS), artificial intelligence, machine learning, software-defined networking (SDN), federated learning, threat intelligence sharing, system-level defense, network security

I. INTRODUCTION

A trap by design, honeynets exist to attract cyber intrusions - all while staying tightly monitored. Rather than repel assaults, they open doors to artificial environments that feel genuine, letting hostile behavior play out under quiet observation. Appearing vulnerable, these systems lure attackers who operate freely, unaware their techniques are laid bare. What unfolds inside reveals not just movement patterns, but also command pathways embedded deep within compromised machines. Over time, insights pulled from these dummies have refined security responses, boosted detection timing, broken apart malware structures, and followed breach trails further than conventional shields usually manage [1], [2].

A trap designed to study hackers works differently than usual security tools - it invites intrusion rather than blocking it immediately. Once inside, attackers face no resistance, allowing close observation of their actions over time. This reveals not only how they operate, but what drives them during different stages of an attack. Information gathered in such settings sharpens systems meant to detect emerging threats. When traps detect activity, security teams update their defenses accordingly. Warnings appear early, stopping threats well ahead of the main network.

Even though they offer advantages, classic honeynet setups usually focus on gathering information and watching activity, yet pay little attention to safe operations or stopping abuse. When rules for monitoring stay fixed and automation is minimal, along with poor control over outgoing connections, hacked honeypots might get used without permission - joining networks of infected machines or helping flood targets during DDoS assaults. This kind of exploitation brings legal questions and moral concerns for those running the systems, weakening their role in defense by causing actual damage [3], [4]. At worst, badly protected honeynets could end up feeding the threats they were built to examine.

Botnets keep changing fast, making existing problems worse. Instead of old methods, many now use smart home gadgets, hidden data paths, signals that shift form, avoiding easy spotting. Today's strikes decide moves on their own, spread quickly between systems, stay quiet after breaking in - old pattern checks or hand-set network guards can't handle such threats [5]. Because of this, fake networks set up for research draw unwanted attention; attackers see them as free tools for more power, probing other machines, boosting assault reach.

Meanwhile, progress in artificial intelligence, machine learning, and adaptive network systems opens fresh paths for reimagining honeynet structures. Instead of relying on fixed rules, behavior-based models powered by AI assess intruder moves as a whole, telling apart harmless access, probing, attack trials, and harmful follow-up actions through live data. On top of that, software-controlled networks paired with deep system safeguards give precise oversight of outgoing behaviors, limiting misuse while preserving the authentic feel crucial for meaningful interaction with attackers [6], [7].

Even as collaboration grows in cybersecurity, sharing threat data across distant honeynets highlights a growing need for reliable, scalable communication paths. Although centralized collection points serve a purpose, concerns about privacy, capacity, and trust emerge - particularly when handling sensitive intrusion records. Using distributed approaches such as federated learning enables independent systems to improve detection jointly without moving raw data, merging shared progress with individual oversight [8].

AISafeHoneynet takes shape where ongoing vulnerabilities meet fresh tech opportunities. Driven by artificial intelligence, it links live activity monitoring with smart detection that evolves over time. Containment begins at the system level, slowing damage long before human response is needed. With decentralized protocols guiding traffic, warnings move fast across nodes - preparedness grows even without central control. What keeps it effective is how quietly it adapts to new threats over time. Built-in automation takes care of warnings, replies, and information sharing, needing little human oversight. The structure resists misuse even as monitoring runs continuously. Rather than locking things down tightly, clarity and protection guide its behavior - allowing fake systems to stay useful yet shielded.

II. PROBLEM DEFINITION

Though built to mimic weak points, today's honeynets serve mainly as tools for tracking how attackers move and what tricks they use. Instead of just blocking threats, they invite intrusion under watchful conditions so experts can follow each step a hacker takes. Because their main job is gathering evidence, protection once breached often gets overlooked. This blind spot opens doors - not only for hijacked traffic but also for lawsuits, misuse, or public scrutiny. Watching closely comes at a cost when defenses stop where observation begins.

Out there, after a breach, honeypots can start sending out harmful software or flooding remote systems with sudden surges of information. With outbound signals often slipping past careful checks - especially when responses lag and rules resist change - it gets easier for intruders to push deeper into networks. Instead of protecting assets, the setup begins assisting threats, flipping its original purpose upside down. Damage reaches farther, pulling uninvolved machines into chaos, making people question whether such setups are truly useful for honest study.

Even now, most threat containment approaches in honeynet systems react only once an incident has unfolded, depending heavily on manual oversight. Due to this need for constant monitoring, they struggle to match the speed of evolving attacks that use concealed pathways, autonomous decision-making, or dynamic command infrastructures. Rather than operating as a unified system, typical security components - firewalls, intrusion detection platforms, and sandboxed testing zones - often run independently. Without coordination, identifying risks, enforcing policies, and exchanging information happens unevenly across layers. Isolated mechanisms weaken post-incident analysis, making it harder to follow attacker moves while response times stretch out.

One problem stands out: poor links between isolated honeynet networks. Though certain main nodes gather attack information, they struggle when scaling up, facing tighter privacy rules and doubts over accuracy - especially with sensitive leak reports. On the flip side, independent systems lack access to collective insights, duplicating efforts while threats linger unseen. So it happens that valuable results remain scattered across groups and organizations, limiting how far one insight can boost broader protection measures.

Ultimately, what this study tackles lies in the lack of an integrated system - automated, focused on safety - for honeynets that spot post-breach threats instantly. Such a setup would also block misuse through hijacking into botnets or flooding networks. At the same time, it allows protected exchange of useful attack data between separate nodes while keeping private information hidden. Solving these issues keeps honeynets trustworthy, strong, and fair within today's digital protection strategies.

III. LITERATURE SURVEY

For years, researchers have used honeynets and honeypots to observe how attackers act, spread malware, or exploit weaknesses - all within carefully managed settings. Initiated by Spitzner, early efforts framed honeynets as rich interaction platforms that record extensive details about intrusions, supporting thorough investigation after events [1]. Later research broadened their role - extending into gathering unknown malware samples and spotting previously unseen attacks [2], [3].

Yet some research highlights weaknesses in typical honeynet designs - on theoretical and practical levels. When containment breaks down, malicious actors could seize control of these decoys, using them to fuel more attacks, like expanding networks of infected devices or overwhelming services with data flow [4], [5]. A documented case showed that poor defenses led simulated environments to shift from monitoring roles to serving real criminal operations [6].

From infected gadgets onward, today's botnets run via scattered systems that disrupt standard security setups. Rather than depending on central command hubs, several now exchange data directly between nodes or create domain addresses dynamically to mask operations [7],[8]. Such shifts make tracing and disabling them far more difficult. Signs point to rapid conversion of weak devices into weapons for assaults. A study revealed minimal decoy traps transforming into sources for massive wave attacks in just a few hours [9].

Honeywell tried shaping outbound traffic by using set filters together with speed limits, cutting off risky connections outside the network [10]. While this worked to a degree, reliance on static rules and manual tweaks made it less effective against smart attacks that change form quickly [11]. Early tests showed promise - yet adaptation lagged behind new AI-driven risks. Fixed logic struggles where threat patterns shift mid-cycle.

Facing such gaps, scientists started combining machine learning with systems that spot intrusions or monitor fake networks meant to trap attackers. Instead of relying only on known signs of attacks, models trained through labeled examples or left to find structure on their own help flag suspicious data flows, unusual actions, or activity after a breach [12], [13]. Work reviewed by Ahmed and colleagues, along with analysis from Sommer and Paxson, shows how these methods catch new kinds of threats that older rule-based tools miss [14], [15]. Still, most efforts prioritize spotting threats correctly, yet pay little attention to stopping them once found or limiting their spread.

Although software-defined networking has introduced flexible, code-driven management of networks, its security models can guide data flow instantly while separating threats and applying rules on the fly - this adaptability proves useful when protecting honeynets [16], [17]. Research shows these systems help block large-scale flooding attempts and manage harmful connections as they emerge [18]. Still, many current SDN-powered protections do not learn from how attackers behave once inside decoy environments.

Sharing information about cyber threats now plays a key role in how organizations protect digital systems. Instead of pooling actual attack data, some platforms collect signs of breaches from many places - yet questions remain around who can be trusted, how private details stay protected, and whether such setups work at scale [19]. One way to tackle these issues involves using federated learning, a method that builds shared models while keeping data local [20]. In areas like spotting intrusions or sorting malicious software, this technique has already delivered measurable progress [21], [22].

Even with progress, gaps remain in what we know. Some honeynets watch but do not protect. Others apply rules separately, lacking shared awareness across components. Rare are the efforts that tackle live abuse spotting alongside traffic control, smart response shifts, and linked threat data - all woven into one system design [23],[25].

AISafeHoneynet responds to growing demands for advanced honeynet systems. Built around artificial intelligence, it analyzes behavior patterns without relying on outdated methods. Security mechanisms operate at the system level, reducing risks from external takeovers. Instead of isolated operation, information flows between nodes, enabling shared insights across regions. Automation allows quick adaptation to emerging threats. Protection against botnets and DDoS attacks forms one core objective. At the same time, visibility into global attack trends improves significantly. Design choices prioritize resilience, learning capacity, and cooperation among participants.

IV. METHODOLOGY

Beginning with close tracking, AISafeHonteynet spreads its work across several connected stages. One stage passes findings to the next - monitoring leads to examination, then judgment, followed by intervention - not everything happens at once. Layers operate together, yet each holds a distinct role, preventing overload. Because functions are split this way, responses stay grounded in observed behavior. Realism for intruders is preserved, even as safeguards tighten. Insights build gradually, not all at once. Timing stays sharp due to division of labor among parts [6], [14].

Right from the start, honeypots serve as alert mechanisms by tracking how attackers behave once inside. Instead of mimicking basic setups, these interactive traps mirror actual operating environments so intrusions unfold without interruption. As probing begins, system activity like command runs, new processes, shifts in access rights, or altered files gets recorded - alongside network behaviors such as external links, spikes in data flow, and repeated destinations. Because information flows nonstop, minor shifts in malicious goals show up sooner rather than later, helping spot misuse before harm spreads [1], [4].

Information moves through layers before reaching secure zones meant for study. At each step, clutter fades - useless entries drop away, patterns grow clear. Instead of rushing straight to response mode, details take time to settle into useful forms. Noise gives way to order as behaviors are reshaped into measurable signs. The delay supports ongoing observation without tipping off intruders. What looks like scanning today might signal danger tomorrow - context decides. Simple signals evolve into rich descriptions of intent. Meaning emerges slowly, built from repetition, timing, structure. Each stage sharpens insight without breaking stride [11], [15].

Over time, machine learning helps examine how attackers change their actions. Rather than using fixed identifiers or hand-written guidelines, this approach tracks shifts in behavior - like abrupt surges in network scans, persistent tries to reach command servers, or data flows resembling early DDoS stages. As attack methods shift, ongoing adaptation enables the system to stay effective against novel or stealthy maneuvers [12], [14].

When risk estimates exceed a set confidence level, responses unfold step by step. Instead of cutting off operations at once, outgoing traffic might face delays, filtering, reduced speed, or partial rerouting. Engagement with the intruder continues under control, allowing observation without endangering outside networks. Judgments rely on gathered patterns of behavior, not guesses - this cut down incorrect alerts and prevents hasty moves that might end useful monitoring too soon [10], [18].

From every encounter with an attacker, new knowledge feeds into a common learning process. These local observations get condensed, then readied for safe exchange among separate AISafeHoneynet instances. Sharing distilled insights instead of unprocessed information enhances group protection without compromising privacy, size adaptability, or independent operation. Over time, each node gains not just localized shielding but also helps refine threat recognition and reaction across far-flung networks [20], [21].

V. PROPOSED SYSTEM

A single breach does not cripple the entire setup when layers handle distinct tasks like monitoring, handling information, examining threats, applying rules, or exchanging insights. Separating duties this way boosts adaptability, simplifies understanding, reduces cascading failures - components update on their own timeline, sidestepping full rewrites [16], [17]. When one section shifts function, ripple effects stay contained; longevity emerges naturally from such isolation.

Hidden within the environment, high-interaction honeypots imitate real operating systems, services, and applications in a way that feels authentic. Since these systems appear legitimate, hackers explore them openly - probing ports, exploiting weaknesses, then navigating once inside. Each action taken on the machine is captured without notice: accessing logs, escalating privileges, executing commands, modifying configurations. Monitoring runs silently beneath the surface, leaving attackers unaware of the trap; this preserves realistic conduct and yields valuable insights for study [1], [6].

A signal flows directly from honeypots into a monitoring unit, capturing live behavioral traces. Right there, computation takes place - similar to how smart sensors operate in everyday networked gadgets - removing noise, aligning event formats, isolating meaningful sequences. Handling information nearby reduces traffic across networks, speeds up responses too. Early warnings surface ahead of attacks, sitting right at the front edge of malicious chains [11], [18].

Once processed, data travels through secure channels - reaching centralized hubs or decentralized analysis units. For smooth movement without lag, small encrypted packages carry signals about actions. When it arrives, cloud environments organize and keep vast amounts of information efficiently. Such structures help identify long-term shifts, examine earlier developments, align sequences, connect incidents from different intrusion trials and software updates [21], [23].

Above data handling sits a system powered by artificial intelligence, watching attacker moves as they unfold. By connecting events within single devices to wider network trends, it decides whether actions seem safe, probing, hostile, or part of post-breach harm. When new behavior samples appear, the core models update - learning evolves, not fixed. That shift maintains precision in spotting threats, even if tactics morph or arrive in unfamiliar shapes [12], [15].

If a more serious threat appears, built-in safeguards at the system level activate on their own. Data leaving the network becomes limited, while endpoints and transmission methods are checked carefully; at the same time, compromised fake networks are isolated through virtual segmentation. Actions proceed step by step - calm enough to avoid alerting attackers, firm enough to prevent damage to external systems such as participation in mass data surges. Through steady control and ongoing operation, threats remain contained under observation, avoiding wider compromise [9], [18].

From the top down, a single intelligent framework connects multiple AISafeHoneynet instances through secure channels. Rather than sending raw attack logs, they transmit condensed alerts, behavioral trends, or modified algorithms. This approach enhances collective detection while expanding situational awareness - each unit still manages its functions independently, maintains confidentiality, and grows efficiently without overload [20], [22].

From the start, information moves without pause across the AISafeHoneynet setup - tracked within the honeypot level, interpreted using artificial intelligence tools, managed via responsive security rules, then improved by collective insights. Because of this flow, the structure forms a network trap that draws in intruders while examining their actions, yet simultaneously defends its own integrity, shifts with new dangers, and feeds into wider joint protection efforts. As time passes, operations grow more secure, threat details become clearer, and chances of harmful spillover drop noticeably.

VI. DISCUSSION

AISafeHoneynet signals a change in both structure and assessment of honeynet setups within real-world defense contexts. Instead of standalone monitoring, its framework prioritizes flexibility, protection, and shared awareness across components. Rather than just gathering logs passively, it counters persistent flaws seen in older models by responding to fluid, network-wide, self-driven assaults. Effectiveness now includes reducing field vulnerabilities without weakening the insights gained during deployment.

One key difference lies in traditional honeynets - they mostly wait instead of taking initiative. While such systems are effective at attracting attackers and collecting data, they often lack self-driven response once compromised. What sets AISafeHoneynet apart is continuous behavioral tracking combined with adaptive intelligence. Instead of jumping to highest alert instantly, it adjusts its stance gradually based on evolving user activity. This means responses fit the situation: mild when appropriate, firm when needed - blocking serious damage without unnecessary escalation [1], [4].

Learning systems improve how quickly threats are spotted once a network has been breached. Instead of relying on fixed rules that need constant human adjustments, these models change by themselves when attack methods shift. Spotting bots joining networks or preparing large-scale disruptions becomes easier this way. These actions usually appear later, after access is gained, escaping older tools focused on known patterns [12], [15]. Watching sequences of behavior - not single moments - helps AISafeHoneynet give alerts sooner, with better accuracy and awareness of setting.

What sets the system apart is how it handles threats - not by cutting them off fast, but by slowing interactions step by step. Rather than halting attacks outright or disabling fake targets instantly, it eases restrictions in stages: limiting data flow, dropping certain requests, or partially disconnecting suspicious nodes. By doing so, it keeps malicious behavior looking natural - important for gathering useful details about real-world risks [9],[10]. At the same time, outside networks stay shielded from misuse. Most older setups lean too hard toward either security or deception; this method manages both without sacrificing one for the other.

Federated threat intelligence boosts both performance and reach within AISafeHoneynet. Most existing honeynets function alone, so critical details about attacks rarely move beyond their initial setup. Instead of exchanging full logs, this system spreads summarized knowledge - allowing different networks to learn from one another without exposing private information or losing control over local configurations [20], [22]. Privacy stays intact even as cooperation grows. In today’s fragmented digital landscape, many groups hesitate - or legally can’t - release internal security records; here, that limitation becomes manageable.

One hurdle remains even when advantages are clear: systems using artificial intelligence need precise adjustments to avoid missed threats or slow alerts, particularly where attackers act unpredictably. Where models learn across separate networks, results rely heavily on how varied, large, and accurate each local dataset is. Nodes with little data or weak setup might add minimal value, dragging down group progress over time. Success often comes not just from design but from steady testing, fine-tuning settings, and watching performance unfold across real-world uses.

Looking ahead, this work shows how AISafeHoneynet shifts honeynet development from static monitoring into proactive, accountable protection shaped by insight. Instead of just watching attacks, it uses behavior tracking, dynamic response adjustments, together with shared model improvements to build stronger defenses than older designs allow. What sets it apart is not only better data on threats but fewer chances for abuse in practice - leading to systems that last longer, earn trust, fit securely within current cyber defense frameworks.

VII. CONCLUSION AND FUTURE SCOPE

A different kind of honeynet begins not with capture, but with control - shaping how fake systems watch attackers without being exploited. Instead of sitting idle, these decoys talk to smart modules that interpret threats in real time, feeding insights to secure zones behind the scenes. Teams tasked with defense see more clearly, their workload lightened by automated streams of meaningful context. Behavior unfolds through sequences: movement across nodes, choices in command usage, rhythms in communication - all stitched into a timeline that reveals purpose. What emerges is not just data, but awareness shaped by timing and interaction.

Over time, as attack records build up, distinct behavior trends begin to show - making it simpler to tell apart routine scans from serious threats after a breach. Stored insights move smoothly into systems designed for sorting and cross-checking, helping structure what once was scattered input. Because decisions rely on actual evidence, response actions happen only when needed, keeping the way attackers engage realistic without enabling abuse like spam networks or traffic floods. The setup grows easily with demand, so whether used in compact studies or broad monitoring setups, core ideas stay effective across different sizes.

When measurements happen constantly, security workflows get noticeably better compared to relying on hunches or fixed beliefs. The moment unusual activity shows up, automated warnings activate - this allows defenses to react early, well ahead of serious harm. Instead of leaning heavily on human oversight or unchanging rules, AISafeHoneynet supports quick reactions without breaking data gathering flow. Over time, this makes honeynets sturdier tools, dependable enough to stay active within current cyber protection plans.

One path forward involves refining how AISafeHoneynet anticipates threats - by adopting predictive algorithms that map out likely adversary moves. Instead of just reacting, systems might recognize patterns across multiple targets using extended behavioral tracking. Progress could come from analyzing longer sequences of network activity, revealing hidden connections over time. Encrypted traffic, often a blind spot, may yield clues when examined with enhanced pattern recognition techniques. Another direction includes safeguarding logs so they resist alteration, possibly through decentralized recording methods. Trust in collected data improves if verification is built into every stage. Evidence remains credible only if its origin and changes are clearly documented. Moving beyond basic detection means weaving together context, history, and resilience at each layer.

One path worth exploring is adapting the framework for large-scale IoT and cloud-native honeynets, as digital exposure grows faster each year. Beyond current limits, extra behavioral monitors across system and network levels may reveal deeper patterns in how attackers shift laterally, move through systems, or maintain access. When visibility improves - paired with responsive analysis and shared knowledge - honeynet functions might transform slowly from standalone traps into interconnected defenses, capable of meeting emerging automated attacks driven by artificial intelligence.

REFERENCES

[1] L. Spitzner, “Honeynets: Catching the Insider Threat,” IEEE Security & Privacy, vol. 1, no. 2, pp. 68–70, 2003.

[2] T. Holz, F. Freiling, and M. Dornseif, “Capture–HPC: High-Interaction Malware Collection,” Proceedings of the DIMVA Conference, pp. 11–30, 2006.

[3] P. Baecher, M. Koetter, T. Holz, M. Dornseif, and F. Freiling, “The Nepenthes Platform: An Efficient Approach to Collect Malware,” Proceedings of RAID, pp. 165–184, 2006.

[4] N. Provos, “A Virtual Honeypot Framework,” Proceedings of the 13th USENIX Security Symposium, San Diego, CA, USA, pp. 1–14, 2004.

[5] D. Dagon et al., “Honeystat: Local Worm Detection Using Honeypots,” Proceedings of DIMVA, pp. 39–58, 2006.

[6] Honeynet Project, “Know Your Enemy: Honeynets,” Technical Report, 2003.

[7] E. Cooke et al., “The Zombie Roundup: Understanding, Detecting, and Disrupting Botnets,” Proceedings of the USENIX Security Symposium, pp. 39–54, 2005.

[8] P. Wang, S. Sparks, and C. Zou, “An Advanced Hybrid Peer-to-Peer Botnet,” IEEE Transactions on Dependable and Secure Computing, vol. 7, no. 2, pp. 113–127, 2010.

[9] C. Rossow et al., “Amplification Hell: Revisiting Network Protocols for DDoS Abuse,” Proceedings of NDSS, 2014.

[10] Honeynet Project, “Know Your Enemy: Honeywall,” Technical Report, 2005.

[11] T. Garfinkel and M. Rosenblum, “A Virtual Machine Introspection Based Architecture for Intrusion Detection,” Proceedings of NDSS, 2003.

[12] W. Lee and S. J. Stolfo, “Data Mining Approaches for Intrusion Detection,” Proceedings of the 7th USENIX Security Symposium, pp. 79–93, 1998.

[13] Y. Liao and V. R. Vemuri, "Use of K-Nearest Neighbor Classifier for Intrusion Detection," Computers & Security, vol. 21, no. 5, pp. 439–448, 2002.

[14] N. M. Ahmed et al., "A Survey of Network Anomaly Detection Techniques," Journal of Network and Computer Applications, vol. 60, pp. 19–31, 2016.

[15] R. Sommer and V. Paxson, "Outside the Closed World: On Using Machine Learning for Network Intrusion Detection," IEEE Symposium on Security and Privacy, pp. 305–316, 2010.

[16] K. Benzekki et al., "Software-Defined Networking (SDN): A Survey," Security and Communication Networks, vol. 9, no. 18, pp. 5803–5833, 2016.

[17] S. Scott-Hayward et al., "SDN Security: A Survey," IEEE Communications Surveys & Tutorials, vol. 18, no. 1, pp. 623–654, 2016.

[18] H. Wang et al., "FloodGuard: A DoS Attack Prevention Extension in Software-Defined Networks," IEEE/IFIP NOMS, pp. 1–9, 2015.

[19] A. Vasilomanolakis et al., "Taxonomy and Survey of Collaborative Intrusion Detection," ACM Computing Surveys, vol. 47, no. 4, 2015.

[20] Q. Yang et al., "Federated Machine Learning: Concept and Applications,"

ACM TIST, vol. 10, no. 2, pp. 1–19, 2019.

[21] L. Zhao et al., "Intrusion Detection Using Federated Learning,"

IEEE Access, vol. 8, pp. 186364–186378, 2020.

[22] Y. Li et al., "Privacy-Preserving Malware Detection Using Federated Learning," Computers & Security, vol. 102, 2021.

[23] M. Husák et al., "Survey of Attack Projection, Prediction, and Forecasting in Cyber Security," Computers & Security, vol. 82, pp. 1–21, 2019.

[24] S. Marchal et al., "DNSScope: Detecting Botnet Command and Control over DNS," Proceedings of NDSS, 2016.

[25] M. Ring et al., "Detection of Slow Port Scans in Flow-Based Network Traffic," IEEE Access, vol. 7, pp. 21746–21757, 2019.