Section 4.1. Multi-Agent System for Detecting and Counteracting Attacks on the Enterprise Information System

Yuliia Kostiuk[1]

[1]*Ph.D. (Computer Science), Associate Professor at the Department of Information and Cyber Security named after professor Volodymyr Buriachok, Borys Grinchenko Kyiv Metropolitan University, Kyiv, Ukraine, ORCID: https://orcid.org/0000-0001-5423-0985*

Citation:
Kostiuk, Yu. (2025). Multi-Agent System for Detecting and Counteracting Attacks on the Enterprise Information System. In P. Kolisnichenko (Ed.), *Insider threats and security in corporations.* 274 p. (pp. 205–232). Scientific Center of Innovative Research. https://doi.org/10.36690/ITSC-205-232

This monograph`s chapter is an open access monograph distributed under the terms and conditions of the Creative Commons Attribution (CC BY-NC 4.0) license

OPEN ACCESS

Abstract. *Modern enterprises face growing cyber incident frequency and increasingly diverse vectors, including AI-driven and multi-vector attacks, while cloud services, IoT, and decentralised architectures strain conventional security controls. Multi-agent attack-detection-and-prevention systems (ADPSs) are proposed as a distributed defence paradigm in which autonomous components monitor and interpret heterogeneous telemetry across network, server, and workstation layers. This study aims to design a scalable and resilient multi-agent system that detects and counteracts attacks on an enterprise information system through coordinated, context-aware decision making and continuous adaptation to evolving threats. The approach specifies an agent-based architecture and formal models for agent behaviour, cooperation, and belief updating. Threat assessment integrates neural networks with fuzzy logic and Bayesian inference, enabling dynamic updating of threat models using real-time observations and historical data. System performance is assessed through operational metrics including false positive rate, belief stability, and response effectiveness. The proposed architecture supports modular deployment of specialised agents that collect and analyse distributed security signals and coordinate responses. By combining deep learning with probabilistic modelling and adaptive learning, the system is positioned to improve detection precision and mitigate limitations of traditional ADPSs, while maintaining rapid adaptability and resilience under modern enterprise conditions. A multi-agent cyber-defence platform can strengthen enterprise security by enabling distributed monitoring, cooperative analytics, and policy-aligned response selection under uncertainty. Future work should validate the approach in real enterprise deployments, benchmark against established ADPS tools, and advance explainability, adversarial robustness, and privacy-preserving learning for sensitive logs and threat-intelligence integration.*

Keywords: *enterprise information system; multi-agent system; information security; attack detection; incident response; cyber threats; neural networks; fuzzy logic; Bayesian inference; adaptive learning; SIEM integration; IT security.*

ISBN 978-9916-9320-4-9 (pdf)
© Scientific Center of Innovative Research, 2025

1. Problem Statement for Multi-Agent ADPSs in Enterprise Information Systems. Modern enterprises confront an ever-expanding spectrum of cyber-threats, ranging from automated vulnerability scans to sophisticated multi-vector advanced persistent threats (APTs). Attack-detection-and-prevention systems (ADPSs) (Shameli-Sendi et al., 2018; Shulika et al., 2024; Vigna et al., 2003) are pivotal in safeguarding corporate information assets through active, real-time monitoring, analysis, and mitigation (Vigna & Valeur et al., 2003).

Current cybersecurity research (Assante & Lee, 2015; Hughes et al., 2020) reveals a persistent rise in security incidents, with escalating diversity of attack vectors and increasing sophistication, including AI-driven attacks and obfuscation techniques. The proliferation of distributed technologies, cloud services, and IoT ecosystems challenges conventional security architectures. As enterprise information systems (ISs) become progressively decentralised, multi-agent ADPSs capable of gathering and analysing data from diverse, distributed sources are essential.

State-of-the-art detection strategies (Almgren et al., 2000; Kostiuk et al., 2025; Kostiuk & Samoilenko et al., 2025) leverage behavioural analytics, machine learning, and big-data techniques, yet remain susceptible to false positives and false negatives, necessitating correlation analysis and adaptive self-learning mechanisms. Implementing a multi-agent ADPS involves deploying autonomous agents for collecting data on network traffic, user activity, file-system events, and threat indicators, with coordinated analysis to uncover complex attack patterns. The integration of intelligent agents employing self-learning and predictive-analysis techniques based on neural networks improves adaptability to novel attack types. The proposed multi-agent system must exhibit a modular architecture facilitating interaction with SIEM platforms (Kriuchkova et al., 2024; Kostiuk & Korshun et al., 2024; Bhardwaj et al., 2022), cloud security services (Kostiuk & Zhyltsov et al., 2025; Logesh et al., 2023; Samoilenko et al., 2024), and threat intelligence feeds (Taher et al., 2019).

Consequently, developing a multi-agent system to detect and counterattack attacks on enterprise ISs is a pressing research direction, aiming to create adaptive, efficient, and distributed defenses against modern cyber-threats.

2. Existing Multi-Agent Approaches to Attack Detection in Enterprises. Giovanni Vigna's work (Vigna et al., 2003; Vigna & Valeur et al., 2003; Assante & Lee, 2015) focuses on developing techniques for identifying complex cyber threats (Almgren et al., 2000), including models for analysing malware and behavioural anomalies. His research (Vigna et al.,

https://doi.org/10.36690/ITSC ISBN 978-9916-9320-4-9 (pdf)
© Scientific Center of Innovative Research, 2025

2003; Vigna & Valeur et al., 2003; Assante & Lee, 2015) underpins flexible, modular intrusion-detection systems capable of rapidly adapting to evolving attack vectors, emphasising event correlation, behavioural analytics, and threat modelling.

Robert Lee (Assante & Lee, 2015) focuses on safeguarding industrial control systems (ICSs), deploying multi-agent architectures for monitoring and anomaly detection in complex industrial networks, with emphasis on early incident identification and rapid response strategies critical for maintaining production continuity.

A review of the literature (Vigna et al., 2003; Vigna & Valeur et al., 2003; Assante & Lee, 2015) reveals that both researchers emphasize integrating multi-agent solutions (Almgren et al., 2000) into existing enterprise security architectures. Contemporary threats demand adaptive, self-learning, and cooperative systems capable of actively mitigating intrusions in real time, evolving in tandem with advances in attack techniques. These contributions provide a robust foundation for multi-agent attack-detection and counteraction systems meeting modern information security requirements.

3. Analysis of Enterprise Information Systems. Contemporary enterprise information systems (Shameli-Sendi et al., 2018; Shulika et al., 2024; Roshan et al., 2023) comprise multiple server classes (web, database, application servers) (Kostiuk et al., 2025), routers, network devices, and end-user workstations, with integration of cloud computing platforms and mobile devices.

Investigation of information-security incidents identified diverse threats, including DDoS assaults, SQL injection exploits, phishing campaigns, and sophisticated AI-driven techniques. The most vulnerable components (Skladannyi et al., 2025; Callegari et al., 2017; Kostiuk & Vorokhob et al., 9) include network devices, critical data servers (Kriuchkova et al., 2024; Kostiuk &Korshun et al., 2024; Bhardwaj et al., 2022), and end-user devices, particularly with weak passwords or inadequately protected protocols.

Attack phases include initial penetration, privilege escalation, access persistence, and execution of malicious actions. Detection systems must integrate heterogeneous data sources: server event logs, network traffic traces, and endpoint process information.

Evaluation of prevalent ADPSs (Snort, Suricata, OSSEC, Zeek, Prelude) assessed capabilities for multilayer monitoring, adaptability, proactive response, and extensibility. None fully satisfies all requirements, especially for emergent attack types. Traditional methodologies have limited

https://doi.org/10.36690/ITSC ISBN 978-9916-9320-4-9 (pdf)
© Scientific Center of Innovative Research, 2025

capacity for large-scale data processing and complex anomaly identification. Neural network models (Liu et al., 2024; Skladannyi et al., 2025; Taher et al., 2019) effectively recognise attack signatures (Javid et al., 2016) but require extensive training datasets. A hybrid methodology (Samoilenko et al., 2024) combining neural networks with statistical traffic analysis and heuristic risk assessment algorithms enhances detection efficacy.

4. Architecture of a multi-agent system for detecting and countering attacks. Multi-agent systems provide dynamically configurable, flexible defence mechanisms within distributed information environments. The architecture facilitates coordinated operation of heterogeneous agents, optimises attack-detection through functional specialisation, and enables real-time information exchange.

The architecture comprises numerous interacting intelligent agents (Shameli-Sendi et al., 2018), each tasked with specific real-time functions for monitoring, analysing, and responding to attacks. Agents communicate via machine-learning and deep-learning models (Shulika et al., 2024), adapting continuously to novel threats (Figure 4.1).

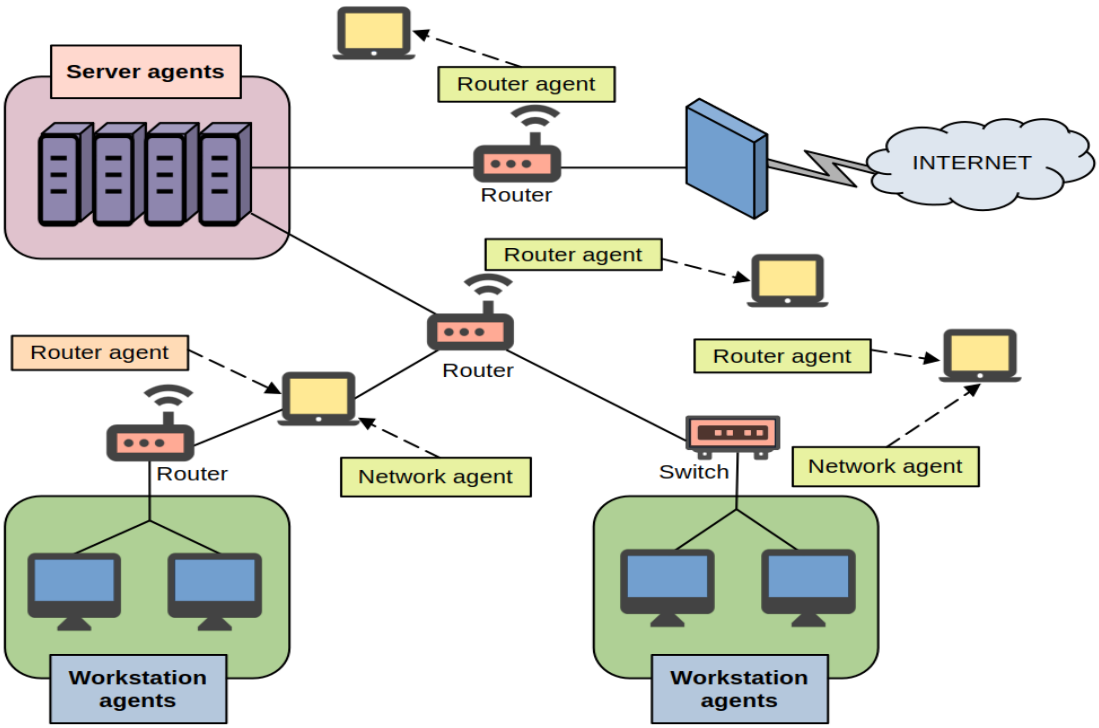

Figure 4.1. Architecture of a Multi-Agent System for Detecting and Countering Attacks

Source: systematized by the author

https://doi.org/10.36690/ITSC
ISBN 978-9916-9320-4-9 (pdf)
© Scientific Center of Innovative Research, 2025

Key components include modules for acquiring and analysing data from heterogeneous sources (Vigna et al., 2003): traditional repositories (server and network-device event logs) and contemporary feeds (cloud platforms, IoT devices, mobile terminals). Integration of blockchain (Vigna & Valeur et al., 2003) ensures transparency and integrity of security-event storage. Analysis enables identification of potential threats, detection of active attacks, and enterprise-level risk assessment (Assante & Lee, 2015).

Attack detection operates across network, server, and endpoint tiers, employing diverse algorithms (Hughes et al., 2020; Liu et al., 2024; Skladannyi et al., 2025) from statistical and heuristic techniques to machine-learning models. Automated-response components (Shameli-Sendi et al., 2018) make local decisions and initiate mitigation actions. Agents transmit alerts to centralised monitoring platforms for strategic decision-making. Architectural flexibility (Vigna et al., 2003; Vigna & Valeur et al., 2003; Assante & Lee, 2015) permits seamless integration of new data sources and detection techniques.

Figure 4.2 presents the component architecture. Modules collect, analyse, and process data from server logs, network appliances, cloud platforms, IoT devices, and mobile terminals. Data flows to the acquisition module, then to the threat-analysis module, collaborating with agents at network, server, and endpoint layers. Detected anomalies are relayed to the response module, which initiates protective measures or transmits incident information to the centralized monitoring and decision-support subsystem (DSS). Events are recorded via a logging module, with optional blockchain-based storage. A feedback mechanism linking the analytical core to a machine-learning module ensures ongoing adaptability. The architecture provides multilayer governance, elastic scalability, and rapid adaptation to evolving cyber threats.

5. Formal Mathematical Model of Agents and Their Behaviour. The system distributes router agents according to clearly defined areas of responsibility, optimising data processing and promoting specialisation. The formalisation of the multi-agent system's structure is:

$$MAS = \{A_R, A_N, A_S, A_W\}, \quad (4.1)$$

where A_R – is a set of router agents, A_N – is a set of agents operating on network nodes, A_S – is a set of server-level agents, A_W – is a set of workstation agents.

https://doi.org/10.36690/ITSC ISBN 978-9916-9320-4-9 (pdf)
© Scientific Center of Innovative Research, 2025

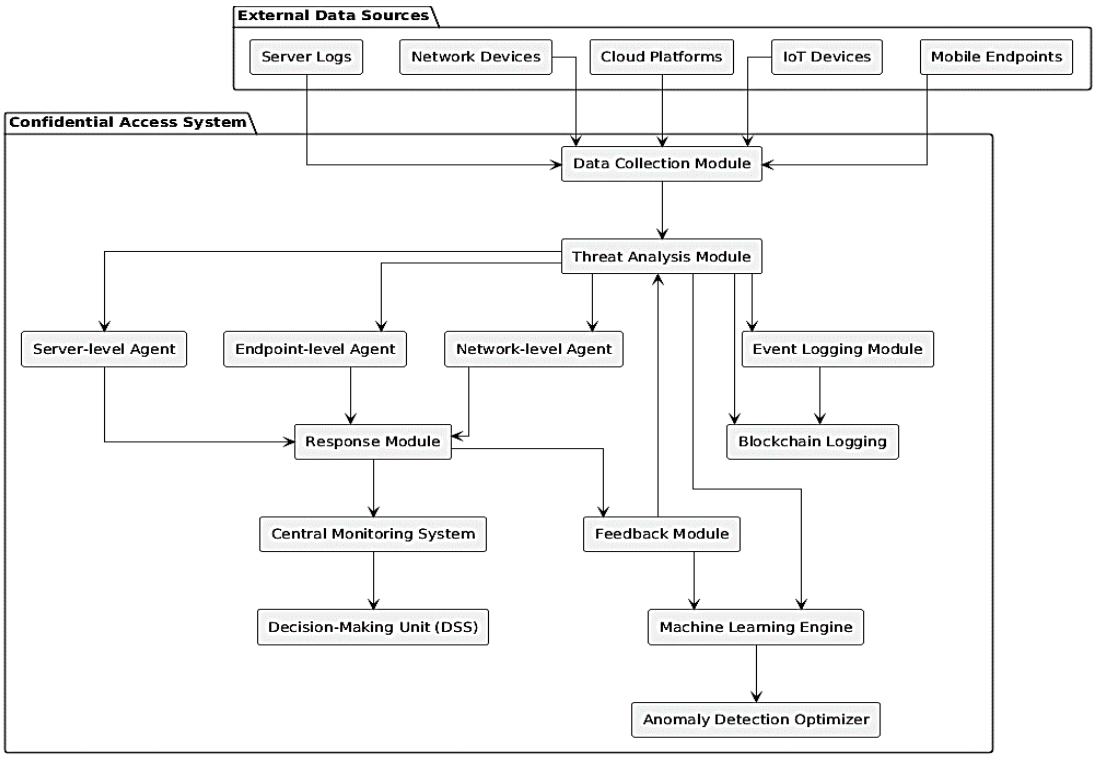

Figure 4.2. Diagram of Components of a Multi-Agent System for Detecting and Counteracting Attacks on an Enterprise Information System

Source: developed by the author

Given the critical role of routers, agents are organised considering external and internal transmission environments. Network traffic is categorised into external and internal flows. The formalisation of the router-agent set is:

$$A_R = \{A_R^v, A_R^0\}, \qquad (4.2)$$

where A_R – is the set of router agents divided into two subsets: external router agents A_R^0 monitoring traffic from outside the enterprise, and internal router agents A_R^v monitoring traffic within the enterprise.

The functional segregation is formalised as:

$$A_R^0 = \{a_R^0 | a_R^0 \in A_R, \forall a_R^0 \; performs \; external \; traffic \; analysis\}, \qquad (4.3)$$

where A_R^0 – a set of external router agents that monitor and analyze traffic entering the enterprise network from the external environment.

$$A_R^v = \{a_R^v | a_R^v \in A_R, \forall a_R^v \; monitors \; internal \; traffic\}, \qquad (4.4)$$

where A_R^v – a set of internal router agents. This distribution increases attack detection effectiveness by specializing agents in analyzing different traffic types.

https://doi.org/10.36690/ITSC
ISBN 978-9916-9320-4-9 (pdf)
© Scientific Center of Innovative Research, 2025

Router agents undertake three principal tasks: detecting anomalies in data flows, scrutinising system event logs, and identifying deviations from baseline network behaviour. The analysis model for network-traffic inspection by router agents is:

$$D_R = \bigcup_{a_R \in A_R} f(a_R, R), \qquad (4.5)$$

where D_R – is a set of data obtained from router event logs, R - is a set of entries in router event logs, $f(a_R, R)$ – is a function that determines the process of data analysis by the router agent a_R.

Each router agent performs in-depth analysis of data stored in event logs, particularly information about network traffic routing, to identify anomalous patterns or potential threats. This process enables timely identification of suspicious deviations in network-device behaviour and facilitates rapid countermeasures.

The system analyses both live network traffic and historical data in router event logs, enabling discovery of covert or long-duration attacks that may not be immediately discernible in real-time flow. The set of anomalous log records is formalised as:

$$A_{nom}(R) = \{r_i \in R | P(r_i) > \theta\}, \qquad (4.6)$$

where $A_{nom}(R)$ – a set of abnormal entries in the router event logs, r_i – an individual entry in the event log, $P(r_i)$ – the probability that the entry r_i is abnormal, θ – the threshold value determining abnormal events.

The analysis of accumulated data combined with probability assessment allows timely detection of both overt and covert threats.

The set of network agents is formalised as:

$$A_N = \{a_N^i | \, i = 1, \dots, n\}, \qquad (4.7)$$

where A_N – set of network agents that are part of a multi-agent system for detecting and countering attacks on an enterprise information system analyzes information about packets transmitted by the N network, detecting anomalies and potential threats in real time.

The data-processing workflow by network agents is:

$$D_N = \bigcup_{a_N \in A_N} f(a_N, P), \qquad (4.8)$$

where D_N – a set of data received by network agents, P – a set of network packets transmitted by the network, $f(a_N, P)$ – a packet processing function of agent a_N, including analysis, classification, and anomaly detection.

https://doi.org/10.36690/ITSC ISBN 978-9916-9320-4-9 (pdf)
© Scientific Center of Innovative Research, 2025

Network agents identify unauthorized access attempts and detect deviations from standard traffic patterns signaling malicious activities (DDoS attacks, port scans, protocol vulnerabilities).

The set of anomalous packets is:

$$A_{nom}(P) = \{p_i \in P | S(p_i) > \theta\}, \quad (4.9)$$

where $A_{nom}(P)$ – is a set of abnormal network packets, p_i – is an individual network packet, $S(p_i)$ – is a function for assessing the degree of abnormality of packet p_i, θ – is a threshold value defining abnormal events.

Traffic patterns are analysed and compared against historical datasets of known attack signatures using a correlation function:

$$Attack(P) = \{p_j \in A_{nom}(P) | C(p_j, H) > \lambda\}, \quad (4.10)$$

where $Attack(P)$ – is a set of network packets identified as potentially malicious, $C(p_j, H)$ – is a correlation function of packet p_j with historical data on attacks H, λ – is a threshold value of similarity determining whether a packet is part of an attack.

Server agents specialise in monitoring system-level events and analysing potential threats. The set of server agents is:

$$A_S = \{A_S^i | \ i = 1, \dots, n\}, \quad (4.11)$$

where $A_S = \{A_S^i, \dots A_S^n\}$ – a set of server agents within a multi-agent system for detecting and counteracting attacks on an enterprise information system consists of several agents of different types A_S^i, where $i = 1 \dots n$ and depends on the functional purpose of a particular server, A_S^i – agent running on server S_i, n – total number of servers in the system.

Each server hosts multiple agents dedicated to analysing specific critical event types.

The critical event analysis process is:

$$E_S = \bigcup_{A_S^i \in A_S} g(A_S^i, L_S^i), \quad (4.12)$$

where E_S – a set of critical security events detected by server agents, L_S^i – server event log S_i, $g(A_S^i, L_S^i)$ – unction of processing and analyzing security events by agent A_S^i.

Server agents (Assante & Lee, 2015; Hughes et al., 2020; Liu et al., 2024) monitor system logs, analysing incidents indicating intrusion attempts, unauthorised access, or malicious activity.

Threat detection within event logs is formalised as:

https://doi.org/10.36690/ITSC ISBN 978-9916-9320-4-9 (pdf)
© Scientific Center of Innovative Research, 2025

$$Threat_S = \{e_j \in E_S \mid R(e_j) > \tau\}, \qquad (4.13)$$

where $Threat_S$ – is a set of events classified as threats, e_j – is a single event from the security log, $R(e_j)$ – is a risk assessment function for event e_j, τ – is a threshold value defining critical threats.

Agents are optimised to detect threats characteristic of their assigned roles. The adaptive response process is:

$$Response_S = \bigcup_{Threat_S} h(e_j, P_R), \qquad (4.14)$$

where $Response_S$ – a set of server agent responses to threats, P_R – a set of rules for responding to threats, $h(e_j, P_R)$ – a function for selecting an appropriate response to a threat e_j in accordance with the security policy.

Each workstation is equipped with agents responsible for local incident detection and collective decision-making. The set of workstation agents is:

$$A_W = \{A_W^j \mid j = 1, \dots, m\}, \qquad (4.15)$$

where $A_W = \{A_W^j, \dots A_W^m\}$ – a set of workstation agents in the context of a multi-agent system for detecting and counteracting attacks on an enterprise information system consists of several agents of different types A_W^j, where $j = 1 \dots m$, depending on the functional purpose of a particular workstation, A_W^j - is an agent working on workstation W_j, m – is the total number of workstations in the system.

The formula for detecting anomalous activity on workstations is:

$$Anom_W = \bigcup_{A_W^j \in A_W} f(A_W^j, L_W^j), \qquad (4.16)$$

where $Anom_W$ – a set of abnormal events on workstations, L_W^j – event log of workstation W_j, $f(A_W^j, L_W^j)$ – function of event analysis by agent A_W^j to identify potential threats.

Threat detection based on risk levels is:

$$Threat_W = \{a_k \in Anom_W \mid R(a_k) > \tau_W\}, \qquad (4.17)$$

where $Threat_W$ – set of threats detected on workstations, a_k – individual anomalous activity, $R(a_k)$ – risk assessment function for event a_k, τ_W – threshold value for recognizing an event as a threat.

Coordination between agents at different infrastructure layers is:

$$C_W = \bigcup_{a_k \in Threat_W} g(A_R, A_S), \qquad (4.18)$$

where C_W – is a set of actions of workstation agents in response to threats, P_R – is a set of rules for responding to threats, $g(A_R, A_S)$ – is a function of coordinating workstation agents A_W with router agents A_R and server agents A_S to neutralize threat a_k.

https://doi.org/10.36690/ITSC ISBN 978-9916-9320-4-9 (pdf)
© Scientific Center of Innovative Research, 2025

This ensures coordinated threat neutralisation (Hughes et al., 2020; Liu et al.,, 2024; Skladannyi et al., 2025) across network segments.

All agents within a multi-agent system designed for detecting and countering attacks on an enterprise information system share a unified structural framework and are defined by a common set of components. These components enable each agent to efficiently perform its designated functions related to the detection, analysis, and response to cyber threats. Each agent is defined by a state A, which includes five main components: $A = (P, B, S, G, I)$. The component P вis responsible for the agent's senses, which is a set of inputs that the agent receives from the environment, including security systems, sensors, network devices, or other infrastructure elements, that helps it perceive external influences. B − is the agent's beliefs, represented by a neural network that integrates information and knowledge about the environment, allowing the agent to adapt its reactions to changing conditions and effectively classify the information received, for example, to detect anomalies or attacks. The component S describes a situation characterized by specific values of input data received from external sources, as well as the results of their classification by the neural network, which determine whether this information is critical for further action. G − are the agent's goals, which define the desired state of the environment that it seeks to achieve, for example, preventing certain types of attacks or restoring the security of an information system. Finally, I − are the agent's intentions, consisting of a set of possible action plans that the agent can implement to achieve its goals, depending on the current situation and the assessment of the results of previous actions (Almgren et al., 2000; Kostiuk et al., 2025; Kostiuk & Skladannyi et al., 2025; Callegari et al., 2017; Kostiuk & Vorokhob et al., 2025; Kriuchkova et al., 2024; Kostiuk & Skladannyi et al., 2024). This structure allows each agent to respond independently and flexibly to changes in the system and work in cooperation with other agents to ensure effective counteraction to cyber threats at different levels of enterprise information security.

Formal representation of the agent state:

$$A_i = (P_i, B_i, S_i, G_i, I_i), \forall i \in A, \qquad (4.19)$$

where A_i − agent, P_i − agent's perceptions, B_i – beliefs, S_i − agent, G_i − agent's perceptions, I_i – intentions.

The situation is constructed as:

https://doi.org/10.36690/ITSC ISBN 978-9916-9320-4-9 (pdf)
© Scientific Center of Innovative Research, 2025

$$S_i = f(P_i, B_i) = \sum_{k=1}^{n} w_k \cdot p_k + \sigma(B_i), \quad (4.20)$$

where w_k – the weighting coefficients of the input signals, p_k – sensory data, $\sigma(B_i)$ – is the activation function of the neural network that takes into account the agent's beliefs.

The criticality function determines whether a situation poses a critical threat. It is defined by a sigmoidal dependence on the assessed state of the system, enabling a smooth and quantitative evaluation of the danger level:

$$Crit(S_i) = \frac{1}{1+e^{-\lambda S_i}}, \quad (4.21)$$

where $Crit(S_i)$ – is the criticality function, λ – is the sensitivity parameter of the criticality assessment.

The output value serves as the foundational criterion for determining the appropriate course of action—whether to initiate an active response, continue monitoring, or temporarily suspend action.

Based on the assessed criticality, the agent determines the most appropriate course of action: initiating an active response, engaging in continuous monitoring without immediate intervention, or entering a passive waiting state. The transition process is:

$$I_i = \{I_{react}, if\ Crit(S_i) > \theta_1\ I_{monitor}, if\ \theta_2 \leq Crit(S_i) \leq \leq_{idle}, \\ if\ Crit(S_i) < \theta_2\ \theta_1, \quad (4.22)$$

where I_{react} – active response to the threat, $I_{monitor}$ – monitoring without immediate intervention, I_{idle} – inactivity, θ_1, θ_2 – threshold values.

This approach enables adaptive selection of behavioral strategy based on threat level, ensuring optimal balance between response timeliness and efficient resource utilisation.

For an agent to effectively adapt to evolving conditions and refine its internal beliefs based on accumulated experience, a mechanism for continuous updating through learning is essential. The process of belief adaptation is:

$$B_i^{t+1} = B_i^t + \alpha \sum_{j=1}^{m} \delta_j \cdot \nabla B_i, \quad (4.23)$$

where B_i^{t+1} – updated beliefs, α – the learning rate, δ_j – the corrective signal, ∇B_i – the gradient of the error function.

This process ensures continuous refinement of the agent's perception models, enabling adaptation to evolving cyber threat dynamics and enhancing accuracy of real-time decision-making.

https://doi.org/10.36690/ITSC ISBN 978-9916-9320-4-9 (pdf)
© Scientific Center of Innovative Research, 2025

The process of selecting the optimal action is:

$$I_i = argU\ (I_k, S_i, G_i), \qquad (4.24)$$

where $U(I_k, S_i, G_i)$ – the action utility function I_k, which takes into account the current situation and the agent's goals.

In a multi-agent system, the effectiveness of protection depends critically on the coherence and coordination of all agent actions. The joint response of agents to a threat is represented by the change in the predicted state resulting from their coordinated actions. Agent cooperation is:

$$S_i^{t+1} = S_i^t + \gamma \sum_{j=1}^{m} \phi I_k, \qquad (4.25)$$

where S_i^{t+1} – the predicted state, γ – the coefficient of influence of actions, ϕ_k – the effectiveness of action I_k.

This cooperation mechanism enables agents to adapt to threat environment fluctuations collectively, optimise allocation of computational and security resources, and enhance overall effectiveness in responding to complex, multi-vector cyberattacks.

6. Probabilistic, Fuzzy and Optimisation Models for Threat Assessment. The multi-agent system employs a Bayesian probability update mechanism integrating current observations with prior knowledge. The threat risk assessment using Bayesian updating is:

$$P(P_i, B_i) = \frac{P(T)P(T)}{P(P_i)}, \qquad (4.26)$$

where $P(P_i, B_i)$ – the probability of a threat under the conditions of the received data, $P(T)$ – the probability of receiving current data in the presence of a threat, $P(T)$ – the a priori probability of a threat, $P(P_i)$ – the normalization factor.

The overall security assessment based on collective decisions of agents is:

$$Sec = \frac{1}{N} \sum_{i=1}^{N} \omega_i U(I_i, S_i, G_i), \qquad (4.27)$$

where Sec – the generalized security level, ω_i – the significance of each agent's contribution.

Attack detection agents are equipped with foundational functions enabling efficient data interaction, continuous adaptation, and automated responses (Figure 4.3).

https://doi.org/10.36690/ITSC ISBN 978-9916-9320-4-9 (pdf)
© Scientific Center of Innovative Research, 2025

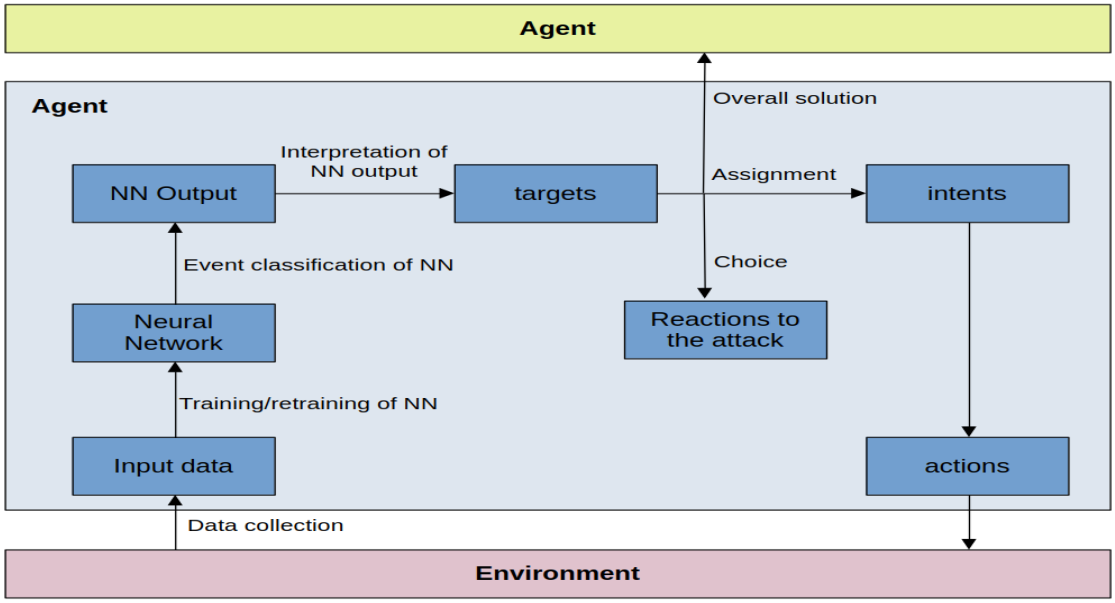

Figure 4.3. Management Architecture of an Attack Detection Center
Source: developed by author

Data collection (Assante & Lee, 2015; Hughes et al., 2020; Liu et al., 2024) supports initial training and periodic retraining of neural networks and provides real-time input for threat detection. Based on accumulated data, neural networks are either newly constructed or adaptively updated to reflect changes in the operational environment and emergence of novel attack vectors. During data analysis, the neural network processes information and provides an assessment of the system's current state. The agent interprets the output, identifying a set of elementary actions tailored to the detected threat type and severity. The agent then engages in local planning to specify concrete actions necessary to neutralise or mitigate the threat. In complex scenarios requiring coordination across multiple system components, the agent escalates to global planning level, collaborating with other agents to establish a unified action plan. Finally, the execution stage is initiated, during which the agent selects and executes required elementary actions (Skladannyi et al., 2025; Kostiuk & Korshun et al., 2024; Bhardwaj et al., 2022) to neutralise the attack or minimise its impact.

Threat risk assessment based on a Bayesian model is:

$$P(T_i|D) = \frac{P(D|T_i)P(T_i)}{\sum_j P(D|T_j)P(T_j)}, \qquad (4.28)$$

where $P(T_i|D)$ – the probability of an attack T_i in the presence of data D, $P(D|T_i)$ – the probability of obtaining such data during an attack, $P(T_i)$ – a posteriori probability of an attack,

https://doi.org/10.36690/ITSC
ISBN 978-9916-9320-4-9 (pdf)
© Scientific Center of Innovative Research, 2025

as derived from the Bayesian model, serves as a dynamic indicator of threat likelihood based on the integration of prior knowledge and real-time observational data.

Neural network training based on backpropagation enables incremental adjustment of network weights:

$$w_{ij}^{(t+1)} = w_{ij}^{(t)} + \eta \cdot \delta_j \cdot o_i, \qquad (4.29)$$

where w_{ij} – the weight of the connection between neurons i and j, η – the learning rate, δ_j – the local error of the neuron, o_i – the output signal of neuron i.

To detect behavioural changes that may signal the presence of an attack, a method evaluates similarity or divergence between system state vectors captured at different time intervals. The anomaly evaluation function based on Euclidean distance in a multidimensional feature space is:

$$d(A, B) = \sqrt{\sum_{i=1}^{n}(A_i - B_i)^2}, \qquad (4.30)$$

where A, B – state vectors of the system at two different points in time.

The greater the Euclidean distance between state vectors, the higher the likelihood that the system has experienced significant changes potentially indicative of abnormal or malicious activity.

To forecast evolution of threats over time and estimate likelihood of system transitioning between different security states, models based on Markov processes are employed. This approach captures the probabilistic nature of behavioural changes in the system under attack. Threat prediction using a Markov process is:

$$P(X_t = s_i) = p_{ij}, \qquad (4.31)$$

where p_{ij} – probability of transition from one security state to another.

This enables not only continuous monitoring of the current state but also prediction of potential threat trajectories, enhancing effectiveness of proactive defence measures.

Since information perceived by agents in real-world environments is subject to continuous change, a mechanism for adaptive updating of beliefs is essential. A model of gradual belief adjustment integrates new information while retaining influence of prior experience, ensuring both responsiveness to recent observations and stability of long-term knowledge. The model for updating an agent's beliefs is:

https://doi.org/10.36690/ITSC ISBN 978-9916-9320-4-9 (pdf)
© Scientific Center of Innovative Research, 2025

$$B_t = \alpha B_{t-1} + (1 - \alpha) I_t, \qquad (4.32)$$

where B_t – the agent's updated belief, I_t – the new information received, α – the coefficient of confidence in previous knowledge.

This enables agents to maintain balanced integration of accumulated experience and new observations, ensuring both stability and adaptability of behaviour.

Fuzzy logic provides a robust framework for handling data ambiguity, uncertainty, and incomplete information inherent in real-world information system environments. Unlike binary logic, fuzzy logic allows representation of threat levels along a continuum, enabling more nuanced and context-sensitive evaluations. The threat level assessment based on fuzzy logic is:

$$P = (min(A_i, B_i)), \qquad (4.33)$$

where A_i – the degree to which an event corresponds to a certain type of attack, B_i – the impact of the threat on the system.

This approach provides greater flexibility in accounting for a wide range of risk factors and supports informed decision-making under conditions of uncertainty.

An effective response to identified threats necessitates development of an optimal action plan that maximises expected benefit of selected countermeasures. An optimisation model evaluates available response options by considering both probability of success of each action and its utility given the current system state. The formal model for forming a threat counteraction plan is:

$$arg\ arg \sum_i U(A_i, S) P(A_i|S), \qquad (4.34)$$

where A – the set of possible actions, $U(A_i, S)$ – the utility of an action in a certain state. The system selects actions offering optimal balance between response effectiveness and resource expenditure, essential for prompt neutralisation of threats and sustained system stability.

The False Positive Rate (FPR) quantifies the proportion of benign events incorrectly classified as threats. This metric is critical for assessing the system's tendency to generate false alarms, which can lead to unnecessary interventions and reduced operational efficiency. The FPR is:

$$P_{fp} = \frac{FP}{FP+TN}, \qquad (4.35)$$

where FP – the number of false positives, TN – the number of correct negative decisions.

https://doi.org/10.36690/ITSC ISBN 978-9916-9320-4-9 (pdf)
© Scientific Center of Innovative Research, 2025

A low false positive rate is critical for maintaining operational efficiency, preventing security personnel from being overwhelmed by unnecessary alerts, and ensuring attention and resources are focused on genuine threats.

In cooperative decision-making among agents, it is essential to consider not only immediate outcomes of individual actions but also cumulative reward associated with achieving the overarching objective of protecting the information system. A reward-based model incorporates a discount factor, reflecting decreasing value of future rewards over time. The cooperative decision-making model based on the reward function is:

$$R_t = \sum_{i=1}^{n} \gamma^t r_i, \qquad (4.36)$$

where R_t – the agent's accumulated reward, r_i – the instantaneous reward, γ – the discount factor.

This approach enables agents to prioritise both short-term gains and long-term effectiveness of protective actions, maintaining coordinated and strategically aligned response to threats.

The effectiveness criterion balances two key aspects: the accuracy of threat detection and the minimisation of false positives. This provides an integrated measure of the system's overall effectiveness. The effectiveness criterion is:

$$E = \frac{D-F}{D+F}, \qquad (4.37)$$

where D – the number of successfully detected attacks, F – the number of false alarms.

This criterion enables an objective assessment of the balance between the system's sensitivity to detecting attacks and its ability to minimise false alarms, which is essential for ensuring stable, efficient, and reliable operation.

7. Neural-Network Belief Model and Adaptive Learning of Agents. Information sources reflecting the state of IS components were analysed, with relevant data selected for agent processing. Router agents process parameters such as subject, importance, timestamp, and source name. Network agents analyse source/destination IP addresses, ports, packet ID, protocol, TCP flags, and ICMP type. Server agents (Kostiuk & Vorokhob et al., 2025; Kriuchkova et al., 2024; Kostiuk & Korshun et al., 2024) handle event codes, severity levels, user identities, and event timestamps. Workstation agents (Bhardwaj et al., 2022) examine event types, descriptors, timestamps, and user context.

https://doi.org/10.36690/ITSC ISBN 978-9916-9320-4-9 (pdf)
© Scientific Center of Innovative Research, 2025

Collected data are structured into feature vectors for situational assessment. A belief model for agents was developed using a multilayer perceptron (MLP). A four-layer perceptron architecture (Kostiuk & Zhyltsov et al., 2025) was implemented. The input layer corresponds to the number of features from the agent's perception vector; the output layer (Logesh et al., 2023) consists of two neurons: one indicating confidence that the event is normal, the other representing the probability of an attack. Activation functions (sigmoid, ReLU) are employed in hidden layers.

Input parameters to the neural network (Samoilenko et al., 2024) are:

$$h_j = f\left(\sum_{i=1}^{n} w_{ij} x_i + b_j\right), \qquad (4.38)$$

where h_j – the value of the hidden layer neuron, $f(\cdot)$ – the activation function (for example, sigmoid or ReLU), w_{ij} – the weight between the input and hidden layer, x_i – the input data (agent's sensations P), b_j – the bias.

This mathematical relationship encapsulates the fundamental mechanism of neuron operation within the hidden layers of an artificial neural network. Specifically, it describes how input parameters - derived from an agent's sensory data - are weighted and then passed through a nonlinear activation function to produce an output signal. This output is subsequently propagated through the subsequent layers of the network, contributing to the final decision output. Such a mechanism is critical for accurately modelling complex patterns of system behaviour and for effectively distinguishing between normal and malicious activity (Kostiuk & Vorokhob et al., 2025). As such, the neurons in the hidden layers play a central role in the system's capacity to recognise sophisticated threat patterns and to support high-quality, context-aware decision-making in dynamic cyber environments.

An essential step in training the neural network for event classification in an attack detection system is the adjustment of connection weights between neurons, which determines the accuracy and generalisation capabilities of the model. This is accomplished through the backpropagation algorithm, which iteratively minimises the error function by calculating gradients and adjusting the network's weights accordingly.

The process of updating neural network weights using the backpropagation method is formally described as:

https://doi.org/10.36690/ITSC ISBN 978-9916-9320-4-9 (pdf)
© Scientific Center of Innovative Research, 2025

$$w_{ij}^{(t+1)} = w_{ij}^{(t)} - \eta \frac{\partial E}{\partial w_{ij}}, \qquad (4.39)$$

where $w_{ij}^{(t+1)}$ – the updated weight value, η – the learning rate, E – the error function, $\frac{\partial E}{\partial w_{ij}}$ – the error gradient.

The presented formula illustrates the foundational principle of neural network training, whereby the weights connecting neurons are incrementally adjusted to minimise the overall error function. This process enables the network to adapt to incoming data by reducing prediction errors, thereby significantly improving classification accuracy and enhancing the reliability of decisions during the identification of both threats and normal system events (Kriuchkova et al., 2024; Kostiuk & Sokolov et al., 2025). As a result of this learning process, the neural network gradually refines its ability to distinguish between benign and malicious behaviours, which is essential for the timely detection of cyber threats and the implementation of effective countermeasures in a dynamic operational environment.

To classify events in an attack detection system, it is necessary to compute probabilistic estimates of the likelihood that each event is associated with either “normal activity” or “attacking influence.” This probabilistic approach extends beyond binary classification by providing a nuanced confidence level for each prediction, thereby supporting the development of flexible and context-aware response strategies.

The most widely used technique for deriving such probabilistic estimates is the sigmoid function, which maps the output of a linear combination of network signals to a continuous range from 0 to 1. This function is defined as:

$$P(Normal) = \frac{1}{1+e^{-Z_1}},\ P(Attack) = \frac{1}{1+e^{-Z_2}}, \qquad (4.40)$$

where $P(Normal)$ – the probability that the event is normal, $P(Attack)$ – the probability of an attack, Z_1, Z_2 – the corresponding linear combinations of weighting coefficients and outputs of the hidden layer.

The above equation describes the process of computing the probability that a given event belongs to one of two categories - *normal* or *attack* - by applying a sigmoidal activation function to a linear transformation of the neural network's input parameters. This transformation maps the result to the interval [0,1][0, 1][0,1], thereby enabling a probabilistic interpretation of class membership (Kostiuk et al., 2025; Kostiuk & Samoilenko et al., 2025; Kostiuk & Skladannyi et al., 2024). These probabilistic estimates form the foundation for informed decision-making by system agents, allowing not

https://doi.org/10.36690/ITSC ISBN 978-9916-9320-4-9 (pdf)
© Scientific Center of Innovative Research, 2025

only the detection of an attack but also the evaluation of its potential impact on the enterprise's information infrastructure.

One of the fundamental stages in training agents in a multi-agent system is optimising the classification process, specifically the assignment of events to the *normal* or *attacking* categories. To accomplish this, an appropriate loss (error) function must be selected - one that guides the training of neural networks modelling agent beliefs by quantifying the difference between predicted and actual outcomes.

In binary classification tasks, the most widely used and effective error function is cross-entropy, which provides a precise measure of the divergence between predicted probabilities and true labels.

The cross-entropy loss function for attack classification is defined as:

$$E = -\sum_i (y_i log(\hat{y}_i) + (1 - y_i) log(1 - \hat{y}_i)), \quad (4.41)$$

where E – error function, y_i – real class, $\hat{y}_i$ – predicted value.

This function, commonly applied in binary classification tasks, quantifies the divergence between the actual class labels and the probability estimates generated by the model. It serves as a foundation for adjusting the neural network's parameters such that, over the course of training, the value of the loss function is progressively minimised. This process leads to improved prediction accuracy and reduced uncertainty in the recognition of attack patterns. Therefore, the use of the cross-entropy function during agent training enhances the system's sensitivity to real threats while minimising the likelihood of false classifications, which is crucial for reliable decision-making in a dynamic information environment.

8. Adaptive Belief Updating and Cooperative Multi-Agent Learning. A multi-agent system incorporates mechanisms for dynamically updating agent knowledge. The belief update based on new observations is:

$$B_t = (1 - \alpha)B_{t-1} + \alpha P, \quad (4.42)$$

where B_t – the agent's updated belief, α – the learning coefficient, P – the new information received (agent's perception).

This equation describes a mechanism for dynamically updating an agent's beliefs based on newly acquired data. According to this formulation, the updated belief is a weighted combination of prior beliefs and current observations. This approach allows agents to integrate accumulated experience with real-time information, thereby enabling a more adaptive and context-aware response to evolving environmental conditions and emerging

https://doi.org/10.36690/ITSC ISBN 978-9916-9320-4-9 (pdf)
© Scientific Center of Innovative Research, 2025

threats. Such a mechanism ensures the flexibility, adaptability, and resilience of agents in the dynamic and unpredictable landscape of cyberspace, allowing them to continuously refine their behavioural models in line with the current state of the information environment.

In the operational process of agents within a multi-agent attack detection system, the ability to rapidly and accurately assess risk associated with each observed event is critical. To support this, a probability normalization method is employed, allowing agents to quantitatively evaluate and compare the relative likelihood of an event being malicious versus benign. This supports informed and timely decision-making based on an interpretable and balanced threat scale.

The normalized risk for agent decision-making can be formalized as:

$$R = \frac{P(Attack) - P(Normal)}{P(Attack) + P(Normal)}, \qquad (4.43)$$

The use of this expression enables the agent to evaluate the risk level associated with a specific event by calculating the normalized ratio between the probabilities of the event being classified as an attack versus normal activity. This facilitates accurate, data-informed decision-making within information security systems [9–13, 15, 19–21]. As such, risk normalization is a vital component of intelligent agent behaviour, as it supports a balanced interpretation of potential threats in contrast to benign operations, thereby ensuring a rational and context-sensitive response to security incidents.

Assessing the stability of agent beliefs:

$$S = \frac{1}{T}\sum_{i=1}^{T}|B_t - B_{t-1}|, \qquad (4.44)$$

The given formula is used to evaluate fluctuations in an agent's beliefs over time, providing insight into the stability of the agent's situational awareness model and its adaptability in the presence of emerging threats. Assessing belief stability is a critical component in evaluating the effectiveness of agents operating within a multi-agent threat detection and response system. High belief stability suggests that agents are successfully adapting to environmental changes without overreacting to minor or transient data anomalies. In contrast, frequent or significant fluctuations may indicate model degradation, requiring adjustments or retraining.

Therefore, regular evaluation of belief stability not only sustains the reliability of the attack detection system but also enables the timely detection of classification performance decline. This provides a mechanism for the

https://doi.org/10.36690/ITSC ISBN 978-9916-9320-4-9 (pdf)
© Scientific Center of Innovative Research, 2025

dynamic retraining and behavioural optimisation of agents, ensuring their continued effectiveness in a rapidly evolving cyber-threat landscape. To maintain the relevance and accuracy of threat detection models in a multi-agent system, it is essential to periodically assess the alignment between current agent beliefs and reference threat characteristics. This comparison enables the identification of drift in risk perception - i.e., the accumulation of discrepancies between the model's outputs and updated threat realities. When such deviations exceed a defined threshold, retraining of the agent's neural network is triggered to restore model accuracy.

The need for retraining can be formally expressed as:

$$D = \sum_i |B_i - T_i|, \qquad (4.45)$$

where T_i – confidence level for a specific type of threat.

This expression quantifies the degree of discrepancy between the agent's current beliefs and the established reference confidence values associated with various types of threats. It enables the detection of significant shifts in the agent's perception of events, serving as a diagnostic indicator for when retraining of the neural network is required.

Thus, the belief conformity control mechanism plays a crucial role in maintaining the relevance, accuracy, and reliability of agents in a dynamic threat environment. By continuously monitoring alignment with reference models, it allows for the timely identification of classification quality degradation and supports the ongoing adaptation of the multi-agent system in response to evolving attack behaviours. This ensures that the system remains robust, responsive, and effective in real-time security contexts.

To ensure consistent and objective decision-making within a multi-agent system for detecting and countering attacks, it is essential to integrate threat assessments provided by different agents. This integration enables the system to consider diverse data sources and analytical perspectives, thereby enhancing the overall accuracy and robustness of threat evaluation.

For this purpose, a weighted threat assessment function is employed, which aggregates individual assessments from multiple agents while accounting for their relative importance in the collective decision-making process. The significance of each agent may be determined based on factors such as its reliability, domain of responsibility, historical accuracy, or relevance to the specific context.

The weighted threat assessment function is formally defined as:

https://doi.org/10.36690/ITSC ISBN 978-9916-9320-4-9 (pdf)
© Scientific Center of Innovative Research, 2025

$$W = \sum_j \lambda_j P_j(Attack), \qquad (4.46)$$

The formula allows for a generalized assessment of the threat level based on the collective assessments of several agents operating in the system by weighting the probabilities of attacking influences determined by each agent separately, which helps to increase the accuracy of the situation assessment. Taking into account the weighting coefficients λ_j for each agent allows you to adaptively adjust the contribution of different data sources depending on their reliability, specialization, or context of operation, which significantly increases the efficiency of collective decision-making in a dynamic cyber threat environment.

After the agent completes the analysis of sensor data, evaluates the criticality of the current situation, and processes threat prediction outcomes, it must arrive at a final decision regarding the classification of the event - as either an *attack* or a *benign anomaly*. To formalise this decision-making process, the agent compares the calculated risk weight against a predefined threshold value. This comparison enables a consistent transition from abstract risk assessment to concrete response actions.

The formal expression for forming the final decision on an attack is defined as:

$$D = \{1,\ W > \theta\ 0,\ W \leq \theta\ , \qquad (4.47)$$

If the risk weighting exceeds the threshold value θ, the agent decides whether to respond to the attack. This approach strikes a balance between the system's sensitivity to real attacks and minimizing the number of false positives, which is critical for the effective functioning of a multi-agent architecture in a highly dynamic cyber threat environment.

The functioning of this multilayer neural network is described by a system that provides automatic detection and classification of anomalies based on data received from agents, making the process of detecting attacks more efficient and adaptive to various cyber threat scenarios:

$$Net_{ij} = \{\sum_k w_{ijk} In_{ijk}\ Out_{ij} = f(Net_{ij} - \theta_{ij})\ In_{ijk} = Out_{i-1k}\ In_{ojk} = x_k\ , \qquad (4.48)$$

where x – the set of input values of the perceptron, In – the set of input values of the neuron, Out – the set of output values of the neuron, i – the number of the perceptron layer, j – he number of the neuron in the perceptron layer, k – he number of the neuron's input, f – the neuron's activation function, w – the weight of the neuron's input, θ – level of neuron activation.

https://doi.org/10.36690/ITSC ISBN 978-9916-9320-4-9 (pdf)
© Scientific Center of Innovative Research, 2025

To form agents' beliefs in the system, neural networks use a standard back-propagation learning algorithm, which is necessary to ensure that agents adapt to environmental changes and respond correctly to attacks (Logesh et al., 2021). An important aspect when using neural networks in multiagent systems is the availability of feedback, which allows correcting the actions of agents in case of incorrect decisions (Samoilenko et al., 2024). To do this, each agent is assigned a belief quality indicator that reflects the accuracy of the agent's assessment of the state of the information system (IS). In case of errors during the analysis of the IS state, the quality indicator decreases, and if it reaches a threshold, the neural network is retrained to improve its performance (Kostiuk & Vorokhob et al., 2025). Such a dynamic self-learning mechanism not only keeps agents' beliefs up to date with changes in the cyber environment but also ensures high resistance of a multi-agent system to new and unknown attacks, thereby increasing the overall security level of the enterprise information system.

To evaluate the quality of agents' beliefs, we introduce the quality function Q_i, which is defined as the weighted average of the accuracy of the agent's predictions for a certain time interval:

$$Q_i = \frac{1}{T}\sum_{i=1}^{T} w_t \cdot A_i(t), \qquad (4.49)$$

where w_t – the weighting factor for each time point t, which determines the importance of the current state for updating beliefs, a $A_i(t)$ – the accuracy of the assessment of the state of the information system by agent i at the time t.

Thus, if the agent ineffectively assesses the state of the IS, the function Q_i decreases, which is a trigger for changing its behavior or training. This approach allows for timely detection of a decrease in the efficiency of agents, automatically initiating the process of retraining or adapting models, which is critical to maintaining a high level of reliability and adaptability of a multi-agent threat detection and counteraction system.

When the quality indicator reaches the threshold value Q_{min}, the process of retraining the neural network is started, which is formalized by the equation for updating the network weights using the gradient descent algorithm:

$$W^{(k+1)} = W^k - \eta \frac{\partial E}{\partial W^{(k)}}, \qquad (4.50)$$

where W^k – the current set of network weights at the k- th iteration, η – learning rate, E – the error function that depends on the difference between the predicted and actual values.

https://doi.org/10.36690/ITSC ISBN 978-9916-9320-4-9 (pdf)
© Scientific Center of Innovative Research, 2025

The equation guarantees that the weights are adjusted in such a way as to minimize the network error, which directly affects the accuracy of the agents' assessment of the IS state. Thus, regular retraining of agent neural networks in the face of a decrease in the quality indicator allows maintaining high accuracy of event classification and ensures continuous adaptation of the system to new types of threats and changes in the behavior of the information environment.

In addition to updating the weights, an important mechanism for correcting agent decisions is the use of a penalty function for agents that make incorrect decisions. Formally, the penalty function P_i is defined as:

$$P_i = \alpha(1 - Q_i)^2, \qquad (4.51)$$

where α – a penalty coefficient that regulates the degree of influence on the agent.

This function provides flexible customization of the process of updating beliefs, since agents that often make wrong decisions receive a larger penalty, which stimulates their adaptation to new environmental conditions. The introduction of the penalty function allows the multi-agent system to form a mechanism for the natural selection of agents based on their efficiency, contributing to the improvement of the overall quality of decisions and increasing the system's stability in a dynamic cyber environment.

To model the feedback between agents, we use the equation for adjusting beliefs through weighted average interaction with other agents:

$$B_i^{(k+1)} = B_i^{(k)} + \gamma \sum_{j \in N_i} w_{ij}(B_j^{(k)} - B_i^{(k)}), \qquad (4.52)$$

where $B_i^{(k)}$ – the current level of belief of agent i at the k-th step, γ – the learning coefficient, N_i – the set of neighboring agents, w_{ij} – the weighting factor of agent j influence on agent i.

The formula shows that the agent updates its beliefs based on the difference between its own assessment and its neighbors' assessments, thereby contributing to knowledge consolidation in a multi-agent system (Logesh et al., 2023). This approach ensures cooperative learning of agents, which increases the consistency of their actions, contributes to the formation of a unified threat assessment in the system, and generally improves the stability and adaptability of a multi-agent architecture in a changing cyber environment.

Finally, to assess the stability of the entire multiagent system, we define the global indicator of belief consistency C as the standard deviation of individual agent beliefs:

https://doi.org/10.36690/ITSC ISBN 978-9916-9320-4-9 (pdf)
© Scientific Center of Innovative Research, 2025

$$C = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(B_i - \underline{B})^2}, \qquad (4.53)$$

where $\underline{B}$ – the average value of the beliefs of all agents in the system. Minimising the global consistency indicator C reflects a high level of coherence among agents, which is essential for the proper functioning of the multi-agent system as a whole.

Thus, the global consistency measure not only quantifies the alignment of beliefs and assessments across agents, but also serves as a diagnostic tool for detecting inconsistencies that may indicate the need for retraining or reconfiguring specific agents to maintain an effective collective threat response.

Conclusions. The development of modern information and intelligent enterprise systems is accompanied by escalating cyber threats, necessitating effective detection and mitigation mechanisms. A multi-agent system for detecting and countering attacks serves as a comprehensive tool for monitoring, analyzing, and responding to threats, offering enhanced adaptability to evolving cyberspace conditions. The deployment of autonomous agents equipped with machine learning, behavioral analysis, and predictive capabilities enables rapid identification of anomalous activities and formulation of effective response strategies grounded in risk assessment and collective decision-making. This approach ensures flexibility, scalability, and resilience against complex multi-vector attacks, critical amid increasing decentralization of corporate networks and proliferation of cloud technologies.

The rising sophistication of attacks incorporating artificial intelligence, obfuscation techniques, and multi-vector strategies renders traditional detection methods inadequate. The multi-agent architecture addresses this by distributing analytical and detection functions among autonomous agents, each specializing in particular system facets (network traffic, user behavior, file system modifications). This design facilitates enhanced scalability and improved detection efficacy through parallelized data processing. The distributed structure bolsters system resilience; should one agent fail, others maintain operational continuity. Through cooperative mechanisms and adaptive learning, agents respond swiftly to emerging threats and refine their behavioral models in alignment with changes in the cyber environment.

In summary, multi-agent systems for attack detection and mitigation constitute a promising research frontier in information security, aimed at creating adaptive and effective cyber defense solutions. The integration of artificial intelligence, machine learning, and distributed data processing methods significantly elevates protection levels, equipping enterprises with

https://doi.org/10.36690/ITSC ISBN 978-9916-9320-4-9 (pdf)
© Scientific Center of Innovative Research, 2025

resilience against emerging threats and enabling continuous surveillance of their information systems.

Funding. The author declare that no financial support was received for the research, authorship, and/or publication of this article.

Conflict of interest. The author declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Generative AI statement. The author declare that no Generative AI was used in the creation of this manuscript.

Publisher's note. All claims expressed in this article are solely those of the author and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References:

1. Almgren, M., Debar, H., & Dacier, M. (2000). A lightweight tool for detecting web server attacks. In *Proceedings of the ISOC Symposium on Network and Distributed Systems Security*, San Diego, CA. https://www.ndss-symposium.org/wp-content/uploads/2017/09/A-Lightweight-Tool-for-Detecting-Web-Server-Attacks-paper-Magnus-Almgren.pdf
2. Assante, M. J., & Lee, R. M. (2015). The industrial control system cyber kill chain. *SANS Institute*. https://doi.org/10.20935/AcadQuant7690
3. Bhardwaj, A., Chandok, S. S., Bagnawar, A., Mishra, S., & Uplaonkar, D. (2022). Detection of cyber attacks: XSS, SQLI, phishing attacks and detecting intrusion using machine learning algorithms. *IEEE Global*. http://dx.doi.org/10.1109/GlobConPT57482.2022.9938367
4. Callegari, C., Giordano, S., & Pagano, M. (2017). Entropy-based network anomaly detection. In *International Conference on Computing, Networking and Communications (ICNC)* (pp. 334–340). doi: 10.1109/ICCNC.2017.7876150.
5. Hughes, K., McLaughlin, K., & Sezer, S. (2020). Dynamic countermeasure knowledge for intrusion response systems. In *2020 31st Irish Signals and Systems Conference (ISSC)* (pp. 1–6). https://doi.org/10.1109/ISSC49989.2020.9180198
6. Javaid, A., Niyaz, Q., Sun, W., & Alam, M. (2016). A deep learning approach for network intrusion detection system. *Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies*, 21–26. DOI 10.4108/eai.3-12-2015.2262516
7. Kostiuk, Y., Skladannyi, P., Sokolov, V., Hulak, H., & Korshun, N. (2025). Models and algorithms for analyzing information risks during the security audit of personal data information system. *Cyber Hygiene & Conflict Management in Global Information Networks 2024, 3925*, 155–17.1 https://ceur-ws.org/Vol-3925/paper13.pdf
8. Kostiuk, Y., Skladannyi, P., Samoilenko, Y., Khorolska, K., Bebeshko, B., & Sokolov, V. (2025). A system for assessing the interdependencies of information system agents in information security risk management using cognitive maps. *Cyber Hygiene &*

ISBN 978-9916-9320-4-9 (pdf)
© Scientific Center of Innovative Research, 2025

Conflict Management in Global Information Networks 2024, 3925, 249–264. https://ceur-ws.org/Vol-3925/paper21.pdf

9. Kostiuk, Y., Skladannyi, P., Sokolov, V., Vorokhob, M. Models and technologies of cognitive agents for decision-making with integration of Artificial Intelligence. Proceedings of the Modern Data Science Technologies Doctoral Consortium (MoDaST 2025). Aachen: CEUR, 2025. Vol. 4005. P. 82–96. ISSN 1613-0073.
10. Kostiuk, Y., Skladannyi, P., Korshun, N., Bebeshko, B., & Khorolska, K. (2024). Integrated protection strategies and adaptive resource distribution for secure video streaming over a Bluetooth network. *Cybersecurity Providing in Information and Telecommunication Systems II 2024, 3826. pp. 129-138. ISSN 1613-0073* . https://ceur-ws.org/Vol-3826/paper12.pdf
11. Kostiuk, Y., Skladannyi, P., Sokolov, V., Zhyltsov, O., Ivanichenko, Y. Effectiveness of Information Security Control using Audit Logs. Proceedings of the Workshop on Cybersecurity Providing in Information and Telecommunication Systems (CPITS 2025), 2025. Aachen: CEUR, 2025. Vol. 3991. P. 524–538. ISSN 1613-0073.
12. Kriuchkova, L., Sokolov, V., & Skladannyi, P. (2024). Determining the zone of successful interaction in RFID technologies. In *IEEE 29th International Seminar/Workshop on Direct and Inverse Problems of Electromagnetic and Acoustic Wave Theory*, 168–171. https://doi.org/10.1109/diped63529.2024.10706153
13. Liu, N., Wu, H., Zhang, Y., & Wang, C. (2024). Adversarial attacks against black-box network intrusion detection based on heuristic algorithm. In *2024 10th International Conference on Computer and Communications (ICCC)* (pp. 1954–1958). https://doi.org/10.1109/ICCC62609.2024.10941941
14. Logesh, B., Perasani, B., Kumar, A. J., Genji, Y., Kiran, C. U., & Godara, J. (2023). Web attack detection using deep learning. In *Proc. KILBY 100 7th Int. Conf. Comput. Sci. (ICCS 2023)*. http://dx.doi.org/10.2139/ssrn.4483837
15. Roshan, K., Zafar, A., & Ul Haque, S. B. (2023). A novel deep learning based model to defend network intrusion detection system against adversarial attacks. In *2023 10th International Conference on Computing for Sustainable Global Development (INDIACom)* (pp. 386–391). https://doi.org/10.48550/arXiv.2308.00077
16. Samoilenko, Y., Smitiukh, Y., Kostiuk, Y., Stepashkina, K., & Hnatchenko, D., Yaremych, V. (2024). Using Node-Red to visualize dairy production data via Modbus. In R. Szewczyk, C. Zieliński, M. Kaliczyńska & V. Bučinskas (Eds.), *Automation 2024: Advances in Automation, Robotics and Measurement Techniques (Lecture Notes in Networks and Systems, Vol. 1219)* (pp. 81-90). Springer. https://doi.org/10.1007/978-3-031-78266-4_8
17. Shameli-Sendi, A., Louafi, H., He, W., & Cheriet, M. (2018). Dynamic optimal countermeasure selection for intrusion response system. *IEEE Transactions on Dependable and Secure Computing, 15*(5), 755–770. https://doi.org/10.1109/TDSC.2016.2615622
18. Shulika, K., Balagura, D., Smirnov, A., Nepokritov, D., & Lytvyn, A. (2024). A method of using modern endpoint detection and response (EDR) systems to protect against complex attacks. *Modern State of Scientific Research and Technologies in Industry, 2*(28), 182–195. http://dx.doi.org/10.30837/2522-9818.2024.2.182
19. Skladannyi, P., Kostiuk, Y., Rzaeva, S., Samoilenko, Y., & Savchenko, T. (2025). Development of modular neural networks for detecting different classes of network

ISBN 978-9916-9320-4-9 (pdf)
© Scientific Center of Innovative Research, 2025

attacks. *Cybersecurity: Education, Science, Technology, 3*(27), 534–548. https://doi.org/10.28925/2663-4023.2025.27.772

20. Taher, K. A., Jisan, B. M. Y., & Rahman, M. M. (2019). Network intrusion detection using supervised machine learning technique with feature selection. *International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST)*, 643–646. http://dx.doi.org/10.1109/ICREST.2019.8644161
21. Vigna, G., Robertson, W., Kher, V., & Kemmerer, R. A. (2003). A stateful intrusion detection system for World-Wide Web servers. In *Proceedings of the 19th Annual Computer Security Applications Conference (ACSAC)* (pp. 34–43). http://dx.doi.org/10.1109/CSAC.2003.1254308
22. Vigna, G., Valeur, F., & Kemmerer, R. A. (2003). Designing and implementing a family of intrusion detection systems. In *Proceedings of the 9th European Software Engineering Conference held jointly with 11th ACM SIGSOFT International Symposium on Foundations of Software Engineering (ESEC/FSE-11)* (pp. 88–97). https://doi.org/10.1145/949952.940084

https://doi.org/10.36690/ITSC ISBN 978-9916-9320-4-9 (pdf)
© Scientific Center of Innovative Research, 2025