Crimson Publishers
Wings to the Research

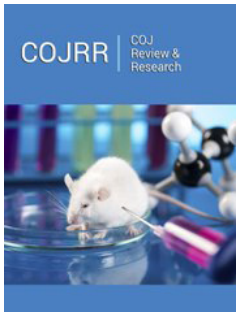
Opinion

Cybersecurity of Artificial Neural Networks

Akmoldina Anara Inshibaevna*, Mukhiyadin Ainur Ulykpanovna and Abdrakhmanov Darkhan Maratovich

Department of Information Systems and Technologies, ESIL University, Kazakhstan

ISSN: 2639-0590



***Corresponding author:** Akmoldina Anara Inshibaevna, Department of Information Systems and Technologies, ESIL University, Astana, Kazakhstan

Submission: March 05, 2026
Published: March 25, 2026

Volume 5 - Issue 1

How to cite this article: Akmoldina Anara Inshibaevna*, Mukhiyadin Ainur Ulykpanovna and Abdrakhmanov Darkhan Maratovich. Cybersecurity of Artificial Neural Networks. COJ Rev & Res. 5(1). COJRR. 000603. 2026.
DOI: 10.31031/COJRR.2026.05.000603

Opinion

Artificial intelligence technologies are developing rapidly in 2020-2024 and are widely used in financial, medical, security, manufacturing, and transportation systems. This has made neural networks the main target for cyber-attacks [1]. According to statistics from 2024, the number of attacks on artificial intelligence systems increased by 340%, which can also increase financial costs. In medicine, diagnostic systems run the risk of making mistakes and are more likely to disrupt the management of autonomous vehicles and pose a direct threat to national security and weakens the privacy of personal data. Therefore, ensuring the cybersecurity of neural networks is a very important, modern and relevant issue. Improving the security of intelligent systems by studying attacks on artificial neural networks, modeling their effects, and developing effective defense mechanisms.

The program "Cybersecurity of neural systems" is designed as an interactive educational simulator for visual display of vulnerabilities and defense mechanisms in machine learning models. Goal: to make the abstract concepts of attacks (for example, Adversarial Attacks) and protection (Adversarial Training) understandable to a wide range of users. Architectural approach: the project is implemented as a single-page web application (HTML/CSS/JavaScript), where all the modeling logic (network status, damage/recovery calculation) is on the client (browser) side [2].

Main blocks (reuse from the flowchart):

i. Presentation layer (Presentation): Provides a user interface, including control fields, buttons, and a theme.

ii. Simulation Logic: Implemented in JavaScript, it manages the state of the neural network (infected/protected), processes button clicks (attack/protection) and updates the event log.

iii. Visualization: CSS and JavaScript are used to dynamically display neurons and connections and change their color (blue -> red -> green) depending on their state.

The program interface is shown in the following figure (Figure 1):

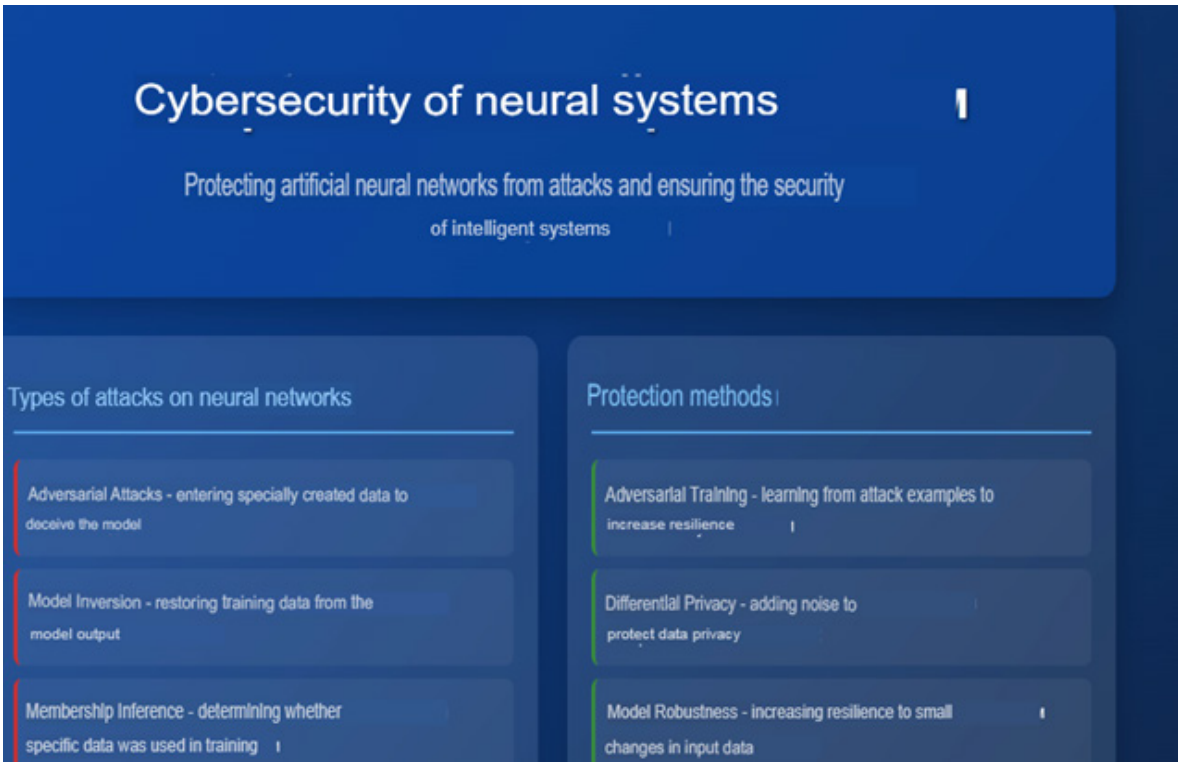


Figure 1: The program interface.

This section on the topic "cybersecurity of neural systems" is divided into two main categories:

i. Types of attacks on neural networks (same block)

This list defines the risks that can be modelled to verify the stability of the model.:

a. Adversarial Attacks: Data entry specifically designed to deceive the model.

b. Model Inversion: Restoring the trained data on the output data.

c. Membership Inference: Determining the use of evidence during training.

d. Model Stealing: copying the architecture and parameters of the model.

e. Data Poisoning: entering malicious data into a training set.

f. Backdoor Attacks: hidden insertion of weaknesses into the model.

ii. Protection methods (positive block)

This list includes countermeasures that can be launched to increase the stability of the neural network and reduce damage from attacks:

a. Adversarial Training: teaching an example of an attack to increase resilience.

b. Differential Privacy: Adding noise to protect data privacy.

c. Model Robustness: increasing resilience to small changes in input data.

d. Federated Learning: Distributed data learning without centralization.

e. Model Watermarking: the introduction of hidden markers to identify the model.

f. Anomaly Detection: detection of suspicious requests and rejections (Figure 2).

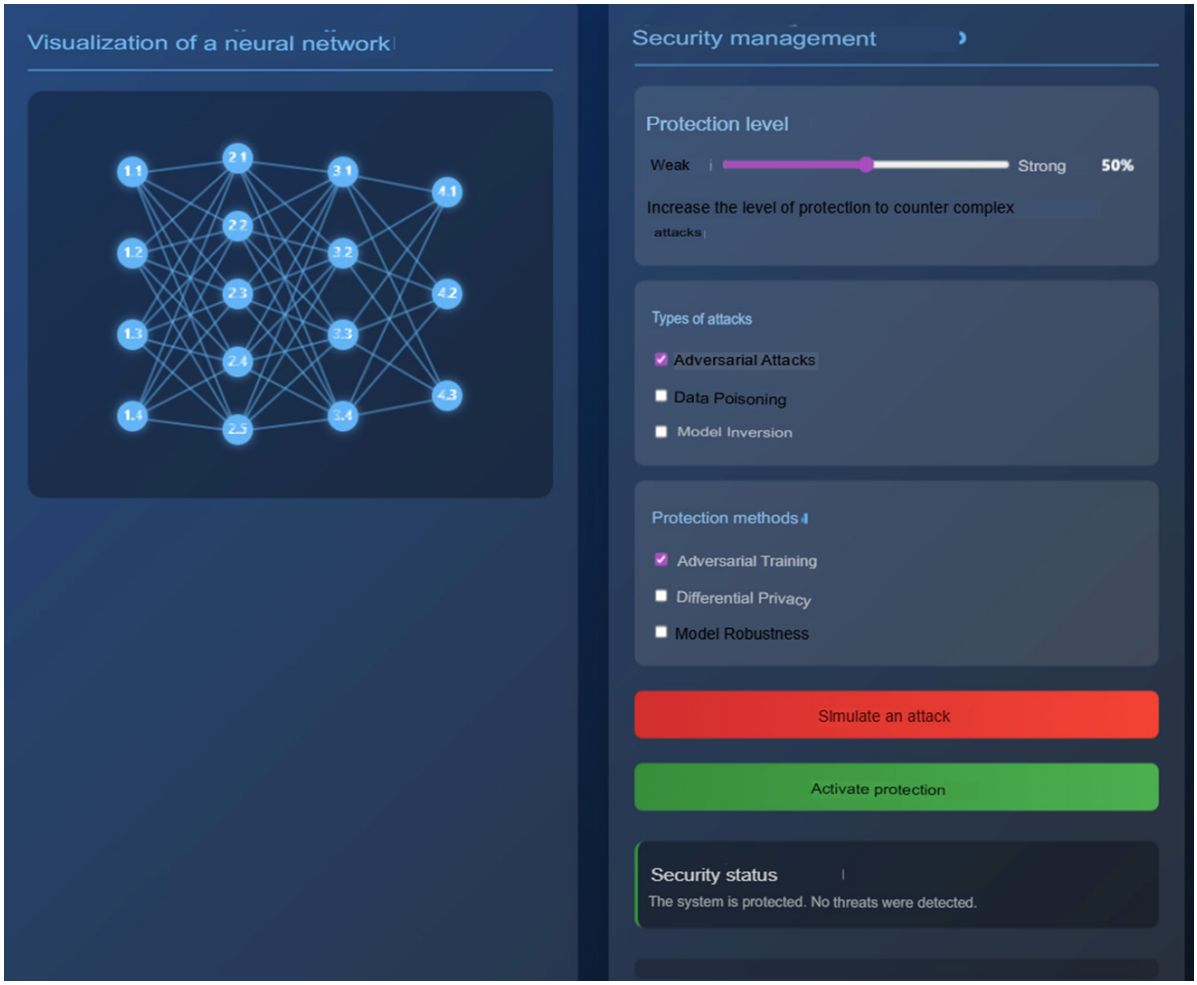


Figure 2: The general interface of the simulator.

This image shows the initial operating state of the simulator at the time of opening. The interface is divided into two main blocks, each of which plays an important role in the modeling process.

i. The left block: Visualization of a neural network

This is the simulation visualization center.

Structure: The figure shows a diagram of an artificial neural network consisting of four layers. Nodes (neurons) and connections (synapses) in each layer are marked in blue.

Initial state: since all elements are blue, this means that the network has not been attacked and is in a "normal state".

ii. Right block: Security management (security management)

This is a simulation control panel where the user sets parameters and launches actions.:

Protection Level (Protection level): The slider is set to 50% (medium). This indicator determines the effectiveness of protection methods.

Attack/Defense Options:

The type of attacks: Adversarial Attacks is selected.

Protection methods: Adversarial Training is selected.

Action buttons: The buttons "Simulate attack" in red and "Activate defense" in green are ready to start the simulation.

Security status: in the initial state "Protection system. Threats are not being handled."

iii. Conclusion (Preparation for action)

The figure shows the initial configuration for creating an Adversarial Attacks attack on a neural network and protecting it using the Adversarial Training method. When the user clicks "simulate attack", the process of visual compromise of the network begins (Figure 3).

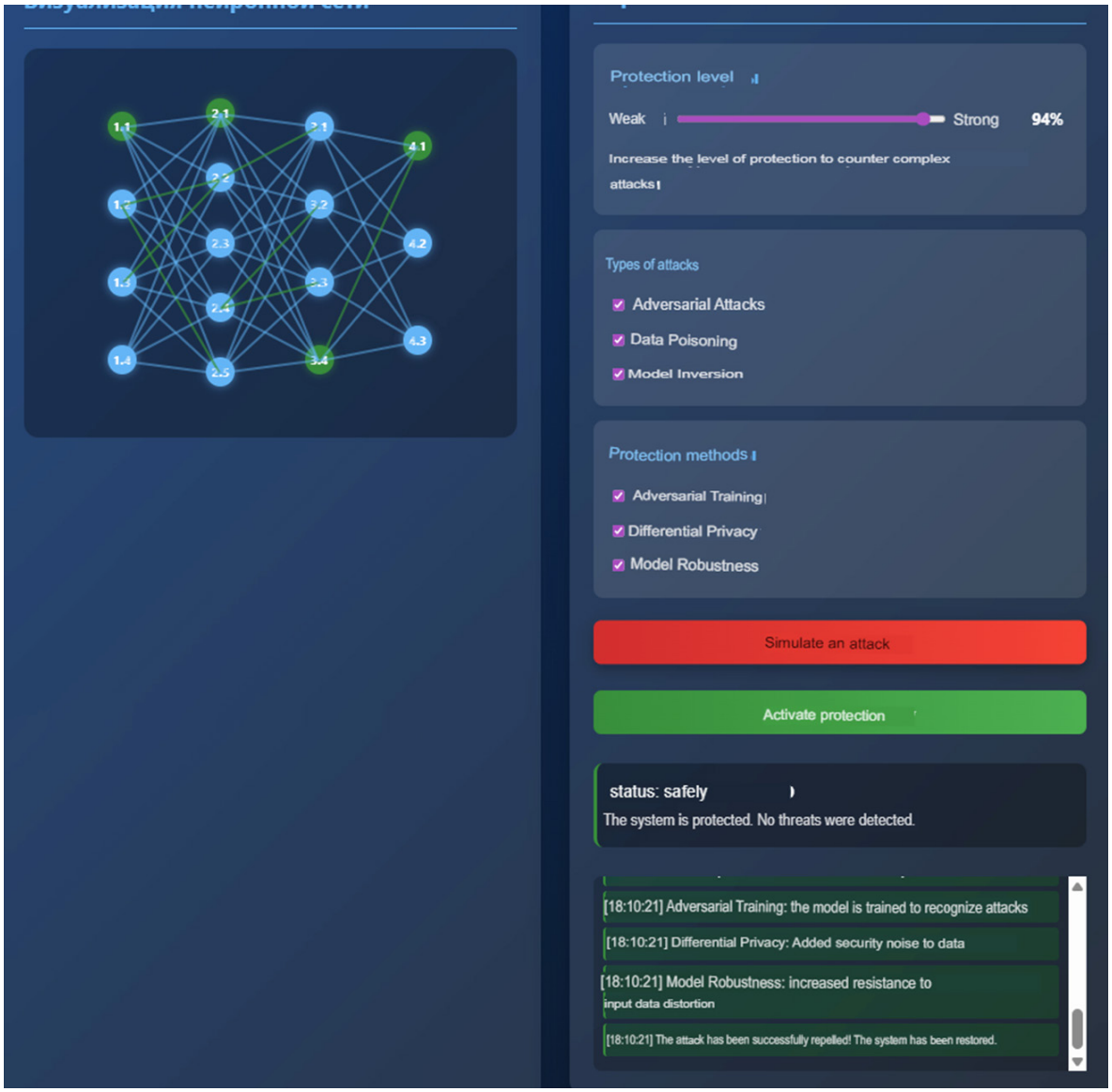


Figure 3: Successful protection and recovery.

The figure clearly shows that high-performance protection is used:

Protection level (Protection level): closer to the maximum-94%. This high level is the basis for the successful functioning of protection methods.

Methods of Attack/Defense:

Attacks: Three complex threats have been selected, such as Adversarial Attacks, Data Poisoning, and Model Inversion.

Protection: Three countermeasures have been activated-Adversarial Training, Differential Privacy, and Model Robustness.

Status " " STATUS: safely. There was no threat," which means that the attacks were completely repelled.

The visualization block clearly confirms that the attack was repelled:

Node status: many nodes are colored green. The green color means that the neurons have been successfully protected and restored.

Connections: Connections between nodes have also been restored.

Logic: The elements that were red during the attack were "repaired" (the red color disappeared), and some of them were protected (green).

The event log at the bottom recorded the activation and the result of the protection:

Registered actions: the log contains records of the contribution of each of the methods of Adversarial Training, Differential Privacy, Model Robustness.

Final message: Finally, "the attack has been debugged! The system has been restored."

Instructions for using the program: three-stage simulation

Using the simulator consists of three main steps: Preparation, Attack and Defense.

Stage 1: Setup and preparation of the system.

At this stage, we identify the main parameters needed for modeling.

Protection Level selection: The first thing you need to do is adjust the protection level slider. If you want to find system vulnerabilities-20%, and to check maximum security, select the interval of 90-100%. (For example, we got a score of 94%).

Identify the types of threats: In the "type of attacks" section, select which threats you want to try. These may include Adversarial Attacks, Data Poisoning or Model Inversion.

Securing protection methods: from the "Protection Methods" block, select the tools that can withstand the selected attacks (for example, Differential Privacy or Adversarial Training).

Stage 2: Simulate an attack and control the consequences.

Here we conduct a direct attack and check the system's response to external influences.

Activate: Press the red "simulate attack" button.

Visual reaction: Pay attention to the network diagram: connections and nodes instantly turn red. This is a sign that the system is damaged.

Security status: The status on the screen will change to "CRITICAL".

Event Log: All actions are logged in the "Event Log" section and information is displayed about which type of attack was activated.

Stage 3: System recovery and conclusion.

At the last stage, we activate the previously selected protection mechanisms and evaluate the result.

Activate Protection: Press the green "Activate protection" button.

Recovery process: The system will start automatically repairing damaged nodes based on the selected algorithms.

Result check: If everything is successful, the red color will disappear and the nodes will turn green.

Final status: If the protection level is high (for example, 94%), the status will return to the "SAFE" state again. And if the protection is weak, the system may show a sign of "bias".

Analysis: in the "attack successfully debugged! The system has been built" [3-7].

Conclusion

"As a result of this work, I gained a deeper understanding of the security of neural networks, not only theoretically, but also practically, which is very important in the development and protection of modern artificial intelligence systems." The architecture of the Web system reflects the security principles of artificial neural networks by modeling attacks, protection methods, and visual changes to the state of the model. The solution is based on a multi-level structure, including an interface level, an attack modeling level, a visualization level, as well as a logical system and a mechanism for making defensive decisions.

Practical advantages of the program:

And to increase the level of protection of neural networks.

Allows you to detect attacks at an early stage.

It provides safe work in areas such as medicine, finance, and autonomous transportation.

A ready-made learning model for students and researchers.

Visual interpretation of offensive and defensive actions using an interactive simulator

A methodological framework for the safe implementation of AI systems.

This program is a comprehensive study aimed at the security of artificial intelligence systems. The research results deserve to be applied both in science, industry, and education.

a. Classification and characterization of the main types of attacks on neural networks.

b. Analysis of the operational logic of Adversarial, Model Inversion, Data Poisoning and Backdoor attacks.

Comparison of the effectiveness of protection methods:

i. Adversarial Training

ii. Differential Privacy

iii. Federated Learning

iv. Model Robustness

v. And identifying vulnerabilities of neural networks.

vi. Research on anomaly tracking algorithms to detect attacks.

vii. And simulate attacks and defensive actions by creating interactive visualizations.

Expected program results:

A complete classification of attacks against neural networks will be developed.

And the risk level of each type of attack is determined.

Protection methods are relatively appreciated

The interactive model shows the dynamics of attack and defense.

A new hybrid protection model will be developed.

The security level of neural networks increases within 30-60

A collection of practical recommendations has been prepared.

References

1. Ian JG, Jonathon S, Christian S (2014) Explaining and harnessing adversarial examples. Machine Learning pp. 1-11.
2. Fredrikson M, Jha S, Ristenpart T (2015) Model inversion attacks that exploit confidence information and basic countermeasures. ACM pp. 1322-1333.
3. Florian T, Fan Z, Ari J, Michael KR, Thomas R (2016) Stealing Machine Learning Models via Prediction APIs. Cryptography and Security arXiv: 1609.02943v2.
4. Biggio B, Nelson B, Laskov P (2012) Poisoning attacks against support vector machines. Machine Learning arXiv: 1206.6389v3.
5. Tianyu G, Brendan DG, Siddharth G (2019) BadNets: Identifying vulnerabilities in the machine learning model supply chain. Cryptography and Security arXiv: 1708.06733v2.
6. Andy Z, Zifan W, Nicholas C, Milad N, Zico KJ, et al. (2023) Universal and transferable adversarial attacks on aligned language models. Computation and Language arXiv: 2307.15043.
7. Nicholas C, Matthew J, Christopher ACC, Daniel P, Will P, et al. (2023) Poisoning web-scale training datasets is practical. Cryptography and Security arXiv:2302.10149.