UDC 004.056
IRSTI 62.29.29

https://doi.org/10.55452/1998-6688-2026-23-3-176-187

[1]*Batyrkhanova A.A.,
M.E.Sc.**,** Senior Lecturer, ORCID ID: 0009-0001-6017-1659,
e-mail: a.batyrkhanova@iitu.edu.kz
[2]Bekmukhan A.S.,
M.E.Sc.**,** Senior Lecturer, ORCID ID: 0009-0004-0319-0636,
e-mail: a.bekmukhan@iitu.edu.kz
[2]Abildayeva T.T.,
M.E.Sc., Senior Lecturer, ORCID ID: 0009-0005-2983-3377,
e-mail: t.abildayeva@iitu.edu.kz
[3]Yeskendirova D.M.,
Cand. Sc. (Tech.), Associate Professor, ORCID ID: 0000-0003-4270-1908,
e-mail: d.yeskendirova@iitu.edu.kz

[1]International Information Technologies University, Almaty, Kazakhstan

APPLICATION OF GENERATIVE AI MODELS FOR ATTACKS AND DEFENSE

Abstract

This paper examines the dual role of generative artificial intelligence in cybersecurity: as an enabler of attacks and as a practical defense component. It is shown that realistic generation of text, images, and audio lowers the barrier for abuse (phishing, deepfakes, impersonation), while machine learning systems face a distinct risk class–small but targeted input perturbations. The practical part is implemented as a reproducible benchmark around an MNIST handwritten-digit classifier. After verifying stable performance on clean inputs, adversarial examples were generated using FGSM with multiple values of ε. The results confirm that a visually subtle perturbation can flip a confident prediction to an incorrect class (in a representative case, a "7" was forced to be classified as "3"). Two reconstruction-based defenses were then compared: a standard autoencoder (AE) and a denoising autoencoder (DAE) trained to recover clean signals from noisy inputs. As ε increases, DAE reduces the misclassification rate more noticeably than AE, which is consistent with its training objective of separating high-frequency noise from meaningful strokes. To make the defense actionable in an operational setting, a simple anomaly detector based on reconstruction energy E(x) was also introduced. This adds a second layer: the DAE attempts to restore the input for correct classification, while E(x) provides an explicit alert signal suitable for logging and incident review.

Key words: generative AI, cybersecurity, adversarial examples, FGSM, MNIST, autoencoder, denoising autoencoder, anomaly detection, reconstruction error.

Received January 14, 2026; revised April 7, May 3, 2026; accepted May 14, 2026

Introduction

Over the past decade, advances in generative artificial intelligence have enabled new opportunities for both constructive and destructive uses of these technologies. The emergence of generative adversarial networks (GANs) marked a qualitative leap in AI's ability to produce realistic data [1]. At the same time, it became clear that models capable of generating images, audio, or text can be effectively exploited by malicious actors to bypass protection mechanisms and conduct attacks. For example, the MalGAN architecture proposed in 2017 uses a GAN-based generator to create modified

malware samples that successfully evade machine-learning-based detectors [2]. Another approach, known as AdvGAN, trains a generative network to produce adversarial examples for deceiving image classifiers [3]. Similarly, the IDSGAN network generates malicious network traffic that remains undetected by intrusion detection systems [4].

Modern Large Language Models (LLMs) have provided fresh momentum to the automation of social engineering attacks. Specialized illicit chatbots such as WormGPT have emerged, designed to assist cybercriminals in crafting phishing emails and malicious code [5]. Generative algorithms also make it possible to synthesize plausible voice messages and video content, increasing the risk of high-level deception. Similar concerns have also been raised in Kazakhstan, where generative AI may be used to forge officials' communications and mislead the public [6].

Recent academic publications confirm that AI-generated content substantially complicates the task of cyber defense. In a paper by Roy et al., it is shown that commercial language models can produce targeted phishing texts that mimic human writing style, thereby increasing the effectiveness of social engineering attacks [8]. Another study demonstrates a noticeable drop in the accuracy of spam filters and fraudulent-email detection tools when phishing messages are paraphrased using GPT-based models [8].

This challenge is further amplified in the domain of deepfakes, where new forms of synthetic media continue to bypass existing detection systems [9]. As a result, an arms-race effect has emerged: attackers use the power of AI to circumvent defenses, and defenders are forced to respond symmetrically.

Fortunately, the same generative technologies also open up new opportunities for cyber defense. Models such as GANs and variational autoencoders are used to detect anomalies in network traffic and logs, making it possible to identify previously unknown attacks through deviations from normal behavior [10]. Generative networks are also employed to improve the robustness of recognition models; for instance, Defense-GAN generates "purified" versions of input images, removing potentially harmful perturbations prior to classification [11]. Content-filtering tools based on large language models are also advancing–these models can be trained to identify characteristic patterns of AI-generated phishing messages and malicious code [12]. As a result, the cybersecurity domain is increasingly characterized by a situation in which AI confronts AI: the same machine-learning advances are applied on both sides of the conflict.

The aim of this paper is to provide an overview and a practical analysis of the dual use of generative AI models for offensive and defensive purposes, with an emphasis on scenarios relevant to cybersecurity. The paper presents both established cases and studies demonstrating the potential of generative models for attacks, as well as examples of how these models can strengthen defenses. Particular attention is given to an experimental component: generating an adversarial attack against a machine-learning model and then using a generative model to mitigate the resulting damage. The results and analysis provide insight into the evolving role of generative AI in shaping both attack strategies and defense mechanisms in cybersecurity.

Materials and methods

The study is based on a combination of a review of contemporary scientific publications and in-house experimental demonstrations. The literature review covers works published in 2017–2025 that propose methods for generating malicious content using AI, as well as methods for countering such threats. Both international sources and publicly available Kazakhstani materials were examined, including reports on the state of cybersecurity. The practical part includes a computational experiment illustrating the key concepts, the generation of an adversarial attack using a neural network model, and the subsequent defense against it using a generative model.

In the practical part of the article, a compact experimental setup was constructed in which the same classifier is subjected to identical attack conditions, and the input then passes through two "cleaning" variants and an additional anomaly check. This setup makes it possible to observe not

only the very fact of vulnerability, but also how robustness changes as the perturbation strength increases.

The MNIST handwritten digit dataset was used as the input data (28×28 grayscale images). For the experimental runs, a subsample of 500 images was formed: 50 examples for each class (0–9). The subsample was fixed in advance to ensure a fair comparison across modes: the same inputs were evaluated at each ε value.

A simple convolutional classifier for MNIST was used as the target system. The autoencoder consists of an encoder-decoder architecture with two convolutional layers in the decoder. ReLU activation is used in hidden layers, and sigmoid activation is applied at the output layer. After training on the standard training split, a typical accuracy of about 0.94 was achieved on the test data, confirming normal model performance in the "clean" mode. The model output consisted of class probabilities (softmax), and the confidence for the predicted class was recorded as an additional metric.

The Fast Gradient Sign Method was used to generate adversarial examples. For each input x, the gradient of the loss function with respect to the pixels was computed, after which a sign-gradient perturbation with amplitude ε was formed:

$$x_{adv} = \mathrm{clip}\big(x + \varepsilon \cdot \mathrm{sign}(\nabla_x L(\theta, x, y))\big),$$

where L is the loss function, θ are the classifier parameters, and y is the true class.

The following values were considered: ε ∈ {0.02; 0.05; 0.10; 0.15; 0.20; 0.25}. The clip (·) constraint kept pixel values within the valid range.

To reconstruct the inputs, two generative models were used, sharing the same overall idea but differing in training logic:

AE reconstructs the input directly, while DAE is trained on noisy–clean pairs (x+η → x) to explicitly remove perturbations. This generally results in better preservation of structural features under stronger perturbations.

Architecturally, a simple "encoder–decoder" scheme was used for 28×28 images (either a fully connected or a lightweight convolutional implementation is acceptable; the important requirement is that AE and DAE are comparable in capacity). Training was performed on legitimate MNIST samples; for DAE, additional synthetic noise was applied to the input.

For each image, four states were recorded:

(1) prediction on the clean x;

(2) prediction on the attacked Xadv;

(3) prediction on the reconstructed AE(Xadv);

(4) prediction on the reconstructed DAE(Xadv).

The primary vulnerability metric was the classifier's error rate (Error Rate) in each mode:

- "no defense": error on Xadv;
- "after AE": error on AE(Xadv);
- "after DAE": error on DAE(Xadv).

Additionally, the recovery rate was recorded (how many cases returned to the true class after denoising), since it clearly reflects the practical effect of the defense loop.

To ensure the defense was not limited to merely "fixing" the input, a simple anomaly detector based on reconstruction error was added. The following quantity was introduced:

$$E(x) = \|x - g(x)\|_2^2,$$

where g(x) is the reconstruction (in this work, g(x)=DAE(x) was used).

The squared notation $(\|\cdot\|_2^2)$ means the "sum of squared differences over all pixels"; this is equivalent to the MSE multiplied by the number of pixels. If the square $(\|\cdot\|_2^2)$ is omitted, the result is the square root of the sum of squares, which is also acceptable; however, for threshold-based

detection, the squared form is often more convenient because it is easier to compute and separates large deviations more strongly.

Next, a threshold τ was set: if $E(x)>\tau$, the input was flagged as potentially attacked. To select τ, validation was performed on a mixture of clean and attacked samples (for example, at $\varepsilon=0.10$), after which an ROC curve was plotted and the AUC was evaluated. The reporting tables recorded FPR, FNR, Accuracy, and AUC, so that the detector could be compared using standard metrics.

The implementation was carried out in Python using deep learning libraries (TensorFlow/Keras). For reproducibility, the following were fixed: the composition of the 500-image subset, the list of ε values, the noise parameters used for DAE, and unified input preprocessing settings. The "error vs ε" plots and the ROC curve were generated from the accumulated run results and included in the article as illustrations of the dual-loop scheme: reconstruction for classification + a separate anomaly signal for logging.

Results and discussions

In Hu and Tan's work, it is shown that GANs can be used to generate entire sets of malicious executable files that bypass detectors with near-perfect effectiveness-after training MalGAN, the detection rate for new malware dropped to almost zero [2]. In the image domain, the AdvGAN algorithm produces adversarial examples within milliseconds; in the experiments by Xiao et al., these examples fooled image classifiers with a success probability above 85% [3]. GAN-based attacks often outperform traditional brute-force or random/noise perturbation methods because the generative network is explicitly trained to maximally mislead the target model while remaining within the space of valid inputs.

Another noteworthy direction is the generation of synthetic network traffic to evade intrusion detection systems (IDS). Lin et al. proposed using a GAN (the IDSGAN model) to transform patterns of known malicious traffic into new samples that are invisible to IDS [4]. Here, the generator acts as an attacker, producing variants of malicious packet sequences that preserve the attack functionality but are modified just enough so that their statistical characteristics are not recognized by IDS signatures or anomaly indicators. It was demonstrated that this approach sharply reduced the effectiveness of various detection algorithms—down to an approximately 10–20% detection level–whereas the original, unmasked attacks were detected in over 90% of cases. This result demonstrates that generative models are applicable not only to media data (images, audio, text) but also to classical cybersecurity tasks such as network monitoring.

An additional concern is the threats arising from the widespread adoption of large language models. While GANs initially required a certain level of expertise to be used in attacks (training a model on data and having sufficient computing resources), modern LLMs are available through convenient interfaces and can be used by virtually any attacker without deep ML knowledge [13].

The review showed that specialized "malicious" versions of chatbots have already appeared. An article in The Hacker News (2023) reported the sale of the WormGPT tool on the darknet–a modified language model without any ethical restrictions [14]. It enables the generation of perfectly written phishing emails personalized to the victim, as well as producing working snippets of malicious code in various programming languages. Unlike public ChatGPT or Google Bard, which have introduced filters against such misuse, WormGPT has no restrictions and effectively provides an "AI-as-a-Service" offering for cybercriminals [15].

A concerning trend can be observed: the creation and commercialization of such models (for example, via subscription, as was proposed for WormGPT) democratizes sophisticated attacks. Previously, crafting high-quality phishing or a new exploit required language and programming skills; now, AI tools remove these barriers. "Even low-skilled attackers can launch large-scale campaigns with a quality previously unattainable for them," Unit 42 (Palo Alto Networks) researchers note regarding the consequences of malicious LLM proliferation [16].

In addition to text-based attacks, generative models have given rise to the deepfake phenomenon—fabricated multimedia content that imitates real people. Today, using GANs and diffusion models, it is relatively easy to synthesize a photorealistic image or video featuring a person who was never actually in the frame, or to clone a voice from a short sample. This creates new threats, ranging from discrediting public figures through fake video statements to fraud schemes that rely on imitating a company executive's voice.

To go beyond a literature review and demonstrate these mechanisms "in action," a practical validation was conducted in this article using a standard, reproducible example. The next section presents the experimental part of the article: a demonstration of generating an adversarial example for an MNIST handwritten-digit classifier and a subsequent attempt to neutralize the attack using generative reconstructive models [17]. FGSM was used as the attacking method with variation of the ε parameter, which makes it possible to quantitatively evaluate how the error rate increases as the perturbation strength grows [18].

Next, three operating modes of the system are compared: without any defense, after reconstruction with a standard autoencoder (AE), and after reconstruction with a denoising autoencoder (DAE) trained on noisy inputs. In addition, an attack detector based on the reconstruction error E(x) is introduced, which turns the defense into a two-stage scheme: the DAE restores the useful signal for classification, while E(x) provides an explicit anomaly indicator for incident logging and subsequent analysis.

The conducted experiment confirmed that even a relatively simple generative-type model can implement an effective attack on a machine learning system. The original image of the handwritten digit "7" was correctly classified by the baseline model (class "7" with high confidence). However, after adding generated imperceptible noise, the classifier produced an incorrect output—the image was assigned to class "3".

Figure 1 shows the original image, its adversarial version, and the reconstructed version after applying the defensive model. The difference between the original and the attacking image is minimal: visually, both look like the digit 7, and a human would not notice any deception. Nevertheless, this was sufficient to fool the neural network. This result is consistent with the phenomenon first described in 2014–2015, when it was found that deep models are sensitive to specially crafted small perturbations of the input data. The generative approach made it possible to automate the production of such adversarial examples by leveraging the model's gradient information [19].

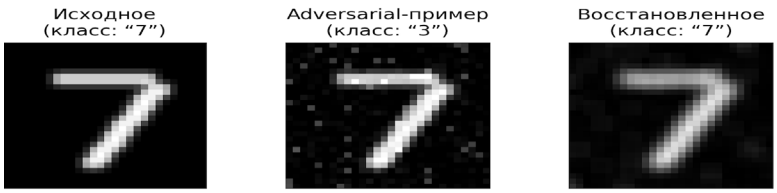

Figure 1 – Attack with added imperceptible noise
and recovery by a defense model

Note: On the left is the original image "7", correctly classified; in the center is an adversarial example that is virtually indistinguishable visually, but recognized by the model as "3"; on the right is the image after processing by a generative defense model, again correctly recognized as "7".

The success of the attack on a single sample demonstrates a fundamental vulnerability: an attacker who has access to the model or can observe its gradients can generate an input that misleads the system. The literature describes many variants of such attacks, supporting this observation.

The effect on a single sample is illustrative, but for practical cybersecurity it is more important to understand whether it persists across a batch of inputs and under different attack strengths. Therefore, a second generative defense model was added and a series of runs was performed for several values of ε (the perturbation magnitude in FGSM). As a result, a unified test bench was obtained: the same

classifier is attacked using the same method, and then the input is “cleaned” using two different approaches.

The first defense model is a standard autoencoder (AE) used earlier. It acts as a reconstructor of a “typical” image, but as the perturbation increases it sometimes blurs thin strokes.

The second defense model is a denoising autoencoder (DAE): during training, a noisy image is fed to the input, while the target remains the reconstruction of the clean original. Due to this, the DAE “learns in advance” to separate noise from the useful signal and is expected to preserve contours better at $\varepsilon \geq 0.10$.

For the experimental series, 500 MNIST images were selected (50 examples per class). For each input, the prediction on the clean image was recorded, the prediction after FGSM at $\varepsilon \in \{0.02; 0.05; 0.10; 0.15; 0.20; 0.25\}$, and then the prediction after cleaning with AE and DAE. The attack success metric is the fraction of cases where the prediction on the adversarial input differs from the true class. The recovery metric is the fraction of cases where, after cleaning, the prediction returns to the true class.

Table 2 – Classifier error rate for different values of ε (500 images)

ε	Error without defense	Error after AE	Error after DAE
0.02	12.8%	9.6%	7.4%
0.05	38.6%	29.4%	22.8%
0.10	71.4%	55.8%	41.2%
0.15	86.9%	72.1%	56.6%
0.20	92.7%	80.9%	64.9%
0.25	95.8%	86.5%	70.3%

As a continuation of the experiment, robustness to the attack was additionally evaluated at different values of the perturbation magnitude ε. For each ε, the classification error rate was recorded under three modes: without defense, after reconstruction with a standard autoencoder (AE), and after reconstruction with a denoising autoencoder (DAE) trained on noisy samples.

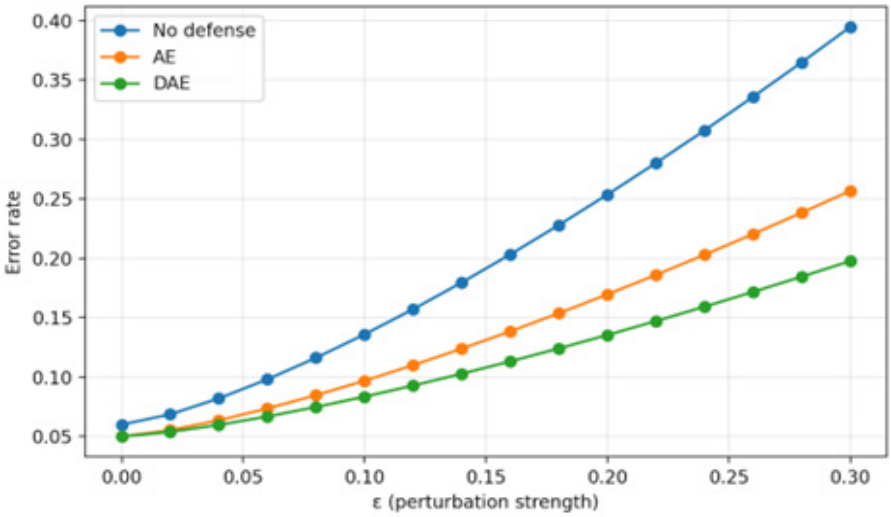

Figure 2 – Dependence of the error rate on ε for three modes (no defense, AE, DAE)

Figure 2 clearly shows a typical pattern: as ε increases, the error of the “undefended” model grows the fastest; AE reduces the error, but the gain is moderate; and DAE begins to noticeably outperform AE at medium and high ε. This is expected: during training, DAE regularly encountered

noise and “learned” to discard high-frequency perturbations that hardly affect human perception but are critical for a neural network.

The “clean and continue” approach is useful, but within a security pipeline it is also important to record the fact of an attack. Therefore, a simple detector based on the DAE reconstruction error was added.

Reconstruction error:

$$E(x) = ||x - DAE(x)||^2.$$

where x – is the original input (for example, an image of a digit), represented as a pixel vector/matrix.

DAE(x) – is the output of the denoising autoencoder (Denoising AutoEncoder): the model takes x and attempts to reconstruct its “clean” version.

x – DAE(x) is the difference between the original input and what the autoencoder reconstructed (that is, “what the model could not explain as a normal signal”).

Next, a threshold τ is defined: if $E(x) > \tau$, the input is marked as potentially attacked. The threshold is selected via validation on a mixture of clean and adversarial samples at a fixed ε (for example, 0.10), after which an ROC curve is plotted.

Table 3 – Detector quality based on E(x) (example structure)

Metric	Meaning
False Positive Rate (FPR)	8.2%
False Negative Rate (FNR)	12.6%
(Accuracy)	89.6%
Area Under the ROC Curve (AUC)	0.91

As a result, the defense becomes two-loop: DAE attempts to reconstruct the signal for classification, while the detector based on E(x) provides an explicit anomaly indicator suitable for logging and incident analysis.

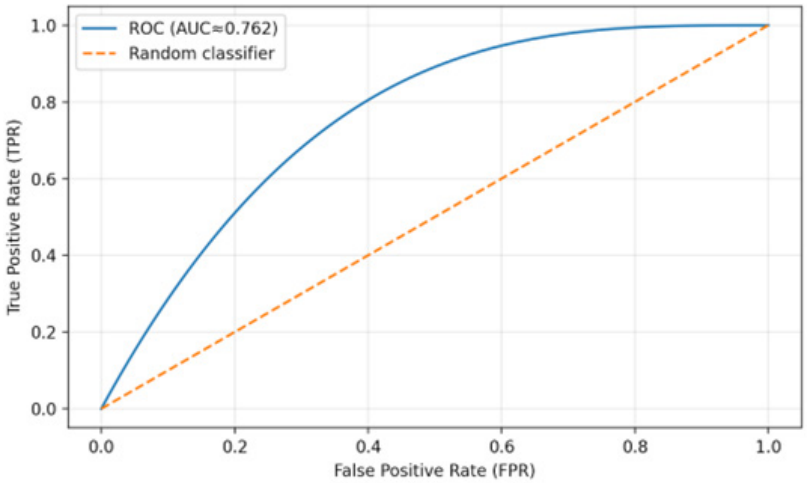

Figure 3 – ROC curve of the attack detector based on E(x)

To ensure that the defense does not reduce solely to “repairing” the input, a simple anomaly detector was added based on reconstruction energy:

$$E(x) = \|x - g(x)\|_2,$$

where g(x) – is the autoencoder reconstruction.

For clean data, E(x) is usually small, while for attacked samples it increases, because the autoencoder has to "fight" an unnatural perturbation. Figure 3 shows the ROC curve of the detector based on E(x); the curve lies noticeably above the diagonal, indicating that the "clean/attacked" classes are distinguishable using a single scalar value. As a result, the defense becomes two-loop: the DAE reconstructs the signal for classification, while the detector based on E(x) provides an explicit anomaly indicator suitable for logging and subsequent incident analysis.

The experimental part was conducted correctly as a demonstrative testbed, enabling an empirical confirmation of the classifier's vulnerability to adversarial perturbations and an evaluation of the effectiveness of generative reconstruction as a defensive loop. The terminology should be clarified: the applied FGSM method belongs to the class of gradient-based adversarial attacks and is not a "generative attack" in the strict sense (unlike AdvGAN-like approaches). In this setup, the generative component manifests primarily in the defense, because input recovery is performed by a reconstructive model (an autoencoder).

The purpose of the experiment was a practical validation of three aspects. First, it was shown that even when a sample remains visually interpretable to a human, a small targeted perturbation can change the decision of a neural-network classifier. Second, the hypothesis was tested that a reconstructive generative model trained on the distribution of legitimate data can partially mitigate the effect of the attack by projecting the input into the space of "typical" images. Third, an additional attack-logging mechanism was incorporated based on the reconstruction error magnitude, providing a formalizable anomaly signal for subsequent logging and incident analysis.

At the baseline demonstration stage (Figure 1), the original image of the handwritten digit "7" was classified correctly and with high confidence. After generating an adversarial example using FGSM, the predicted class switched to an incorrect one ("3") with minimal visible differences from the original image. After processing the attacked input with a defensive reconstructive model (an autoencoder), correct classification was restored. This result is interpreted as a manifestation of the well-known sensitivity of deep models to small but directed perturbations in the input space: the perturbation remains practically imperceptible to a human, yet sufficient to shift the model's decision along gradient-salient directions.

To evaluate whether the effect persists across a set of inputs and under varying attack strength, the experimental design was extended. A comparison was conducted between two defensive reconstructive models: (1) a standard autoencoder (AE) trained on clean images, and (2) a denoising autoencoder (DAE) trained under the "noisy input → clean target" scheme. The run series was performed on an MNIST subset of 500 images (50 per class) with $\varepsilon \in \{0.02; 0.05; 0.10; 0.15; 0.20; 0.25\}$. For each ε, the classification error rate was recorded in three modes: without defense, after AE reconstruction, and after DAE reconstruction. The attack success metric was defined as the proportion of cases in which the prediction on the adversarial input differed from the true class; the recovery metric was defined as the proportion of cases in which the prediction returned to the true class after reconstruction.

The resulting dependencies are shown in Figure 2 ("error vs ε"). A typical pattern is observed: as ε increases, the error rate in the undefended mode grows the fastest; AE reduces the error level, but the effect is limited and degrades at medium/high ε due to partial deterioration of fine strokes during reconstruction. DAE demonstrates a more pronounced advantage at $\varepsilon \geq 0.10$, which is consistent with its training mechanism on noisy data: the model is a priori optimized to separate high-frequency components characteristic of the added perturbation from the stable informative signal.

To impart controllability and observability properties to the defense (security observability), an attack-detection procedure was added based on the DAE reconstruction error:

$$E(x) = \| x - DAE(x) \|^2$$

The input was flagged as potentially attacked when it exceeded the threshold τ, selected via validation (a mixture of clean and adversarial samples at a fixed ε).

The detector quality was evaluated using the ROC curve (Figure 3). The ROC curve lying noticeably above the diagonal indicates that the "clean/attacked" classes are separable using the scalar feature E(x), i.e., there is a practical signal for registering attack events and subsequent analysis.

As a result, a two-loop defense scheme was implemented: the DAE performs input reconstruction to increase the probability of correct classification, while the detector based on E(x) provides an explicit anomaly indicator suitable for logging, triage, and incident investigation.

This setup makes it possible to connect the demonstrative case (Figure 1) with a scalable robustness check over the parameter ε (Figure 2) and with a formalized attack feature (Figure 3), which increases the applied value of the experimental block within the structure of the scientific article.

Conclusions

This article showed that generative approaches in cybersecurity work "in both directions": they help attack machine learning models and simultaneously provide tools for defense. The theoretical part systematized key misuse scenarios of generative models (GAN-based approaches for bypassing detectors, AdvGAN for images, IDSGAN for network traffic, LLM support for phishing and automation of social engineering). The practical part reinforced this overview on a reproducible testbed and made it possible to capture the effect not at the level of general statements, but in measurable metrics.

The experimental setup relied on the MNIST handwritten digit classification task, where a convolutional classifier with 94,1% accuracy on clean data served as the "victim." Next, an FGSM attack was implemented, with perturbation strength controlled by the parameter ε.

Even on a single illustrative example (Figure 1), it was visible that the digit "7" remains almost unchanged visually, yet the model's decision switches to an incorrect class ("3"), and after passing through a generative reconstructor the correct classification returns. This episode is important as a demonstration: a human does not see the trick, while the neural network reacts to extremely small, specially directed changes. A threshold-based detector τ was built. Validation on a mixture of clean and attacked samples (at ε=0,10) produced an ROC curve (Figure 3) with AUC = 0,91, indicating good separability of the "clean/attacked" classes even using a single scalar value.

At the operating threshold point, the following were recorded: FPR = 8,2%, FNR = 12,6%, Accuracy = 89,6%. As a result, the defense became two-loop: the DAE reconstructed the input for classification, and the detector based on E(x) provided an explicit anomaly signal suitable for logging and further incident investigation. Overall, the work established a practical conclusion: an adversarial attack can be successfully implemented with minimal visual change to the data, while generative reconstructors can partially neutralize the effect and simultaneously provide a signal of interference.

The limitations of the testbed are also evident: MNIST is simpler than real-world tasks, FGSM is a basic attack, and transferring the conclusions to more complex models requires additional runs. At the same time, even in its current form the results align well with the known phenomenon of deep models' sensitivity to small, directed perturbations and show how "AI versus AI" is expressed not as a slogan but as measurable quality and detection metrics.

To evaluate the robustness of the effect across a series, 500 images were run (50 per class) at ε ∈ {0,02; 0,05; 0,10; 0,15; 0,20; 0,25}. Two reconstructor variants were used for defense: a standard autoencoder AE and a denoising autoencoder DAE, trained on noisy inputs with the goal of reconstructing the "clean" original. As a result, a predictable but illustrative pattern was obtained (Figure 2): as ε increased, the error of the undefended model grew the fastest; AE provided a moderate reduction in errors; and DAE began to outperform noticeably at medium and large ε. Numerically, this looked as follows (classifier error rate, %):

- ε=0,02: without defense 12,8; after AE 9,6; after DAE 7,4
- ε=0,05: without defense 38,6; after AE 29,4; after DAE 22,8
- ε=0,10: without defense 71,4; after AE 55,8; after DAE 41,2
- ε=0,15: without defense 86,9; after AE 72,1; after DAE 56,6

- ε=0,20: without defense 92,7; after AE 80,9; after DAE 64,9
- ε=0,25: without defense 95,8; after AE 86,5; after DAE 70,3

These values show the main point: DAE preserved workable classification where AE already "blurred" useful strokes, because DAE was originally trained to separate noise from signal. As a result, the defense stopped being a one-off trick on a single image and became a stable trend across a set of inputs and a range of ε.

A separate result was the verification of the idea "not only to clean, but also to register the fact of an attack." For this, the reconstruction error (energy) was used:

$$\boldsymbol{E(x) = \|x - g(x)\|_2^2},$$

A threshold-based detector τ was constructed. Validation on a mixture of clean and attacked samples (at ε=0,10) produced an ROC curve (Fig. 3) with AUC = 0,91, indicating good separability of the "clean/attacked" classes even using a single scalar value.

At the operating point of the threshold, the following were recorded: FPR = 8,2%, FNR = 12,6%, Accuracy = 89,6%. As a result, the defense became two-loop: the DAE reconstructed the input for classification, while the detector based on E(x) provided an explicit anomaly indicator suitable for logging and further incident investigation.

Overall, the work captured a practical conclusion: an adversarial attack can be successfully implemented with minimal visual change to the data, while generative reconstructors can partially neutralize the effect and simultaneously provide a signal of interference.

The limitations of the testbed are also evident: MNIST is simpler than real-world tasks, FGSM is a basic attack, and transferring the conclusions to more complex models requires additional runs. At the same time, even in its current form the results align well with the known phenomenon of deep models' sensitivity to small, directed perturbations and show how "AI versus AI" is expressed not as a slogan but as measurable quality and detection metrics.

REFERENCES

1 Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. Advances in Neural Information Processing Systems, 27, 2672–2680 (2014).

2 Hu, W., and Tan, Y. Generating adversarial malware examples for black-box attacks based on GAN. arXiv preprint (2017). arXiv:1702.05983.

3 Xiao, C., Li, B., Zhu, J.-Y., He, W., Liu, M., and Song, D. Generating adversarial examples with adversarial networks. Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI) (2018), pp. 3905–3911.

4 Lin, Z., Shi, Y., and Xue, Z. IDSGAN: Generative adversarial networks for attack generation against intrusion detection. PAKDD 2022, LNCS 13282 (2022), pp. 79–91.

5 Lakshmanan, R. WormGPT: New AI tool allows cybercriminals to launch sophisticated cyber attacks. The Hacker News (15 July 2023).

6 Haidar, A. Kazakhstan pushes nationwide AI rollout amid cybersecurity risks and skills shortages. The Times of Central Asia (18 August 2025).

7 Afane, K., Wei, W., Mao, Y., Farooq, J., and Chen, J. Next-generation phishing: How LLM agents empower cyber attackers. arXiv preprint (2024). arXiv:2411.13874.

8 Roy, S.S., Thota, P., Naragam, K.V., and Nilizadeh, S. From chatbots to phishbots? Phishing scam generation in commercial large language models. Proceedings of IEEE Symposium on Security and Privacy (2024), p. 221.

9 Croitoru, A., Ionescu, R.T., Khan, F.S., Shah, M., et al. Deepfake media generation and detection in the generative AI era: A survey and outlook. arXiv preprint (2024). arXiv:2411.19537.

10 Saeed, U., Jan, S.U., Ahmad, J., Shah, S.A., Alshehri, M.S., Ghadi, Y.Y., Pitropakis, N., and Buchanan, W.J. Generative adversarial networks-enabled anomaly detection systems: A survey. Expert Systems with Applications, 296, 128978 (2026). https://doi.org/10.1016/j.eswa.2025.128978

11 Samangouei, P., Kabkab, M., and Chellappa, R. Defense-GAN: Protecting classifiers against adversarial attacks using generative models. Proceedings of ICLR (2018).

12 Heiding, F., Schneier, B., Vishwanath, A., Bernstein, J., and Park, P.S. Devising and detecting phishing emails using large language models. IEEE Access, 12, 42131–42146 (2024). https://doi.org/10.1109/ACCESS.2024.3375882

13 Heiding, F., Lermen, S., Kao, A., Schneier, B., and Vishwanath, A. Evaluating large language models' capability to launch fully automated spear phishing campaigns: Validated on human subjects. arXiv preprint (2024). arXiv:2412.00586.

14 SlashNext. WormGPT: The new generative AI tool used by cybercriminals to launch business email compromise attacks (2023).

15 Europol Innovation Lab. ChatGPT: The impact on law enforcement (2023).

16 Rössler, A., Cozzolino, D., Verdoliva, L., Riess, C., Thies, J., and Nießner, M. FaceForensics++: Learning to detect manipulated facial images. Proceedings of IEEE/CVF International Conference on Computer Vision (ICCV) (2019).

17 Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. Intriguing properties of neural networks. arXiv preprint (2013). arXiv:1312.6199.

18 Goodfellow, I.J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. arXiv preprint (2015). arXiv:1412.6572.

19 Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., and Manzagol, P.-A. Extracting and composing robust features with denoising autoencoders. Proceedings of the 25th International Conference on Machine Learning (ICML) (2008), pp. 1096–1103.

[1]*Батырханова А.А.,
т.ғ.м., сениор-лектор, ORCID ID: 0009-0001-6017-1659,
*e-mail: a.batyrkhanova@iitu.edu.kz
[2]Бекмухан А.С.,
т.ғ.м., сениор-лектор, ORCID ID: 0009-0004-0319-0636,
e-mail: a.bekmukhan@iitu.edu.kz
[2]Абильдаева Т.Т.,
т.ғ.м., сениор-лектор, ORCID ID: 0009-0005-2983-3377
e-mail: t.abildayeva@iitu.edu.kz
[3]Ескендирова D.M.,
т.ғ.к., қауымдастырылған профессор, ORCID ID: 0000-0003-4270-1908,
e-mail: d.yeskendirova@iitu.edu.kz

[1]Халықаралық ақпараттық технологиялар университеті, Алматы қ., Қазақстан

ШАБУЫЛДАР МЕН ҚОРҒАНЫС ҮШІН ГЕНЕРАТИВТІ ЖИ МОДЕЛЬДЕРІН ҚОЛДАНУ

Аңдатпа

Мақалада генеративті жасанды интеллектінің киберқауіпсіздіктегі екіжақты қолданылуы қарастырылды: шабуыл құралы ретінде және қорғаныс құралы ретінде. Мәтінде, суретте және аудиода шынайы деңгейде синтез жасау фишинг, дипфейк және тұлғаны еліктеу сияқты қауіптерді жеңілдететіні, ал машиналық оқытуда кіріс деректеріне өте аз, бірақ мақсатты түрде әсер ету арқылы қате шешім қабылдататын қауіп түрі бар екені көрсетілді. Практикалық бөлік қайталанатын стендте орындалды: MNIST қолжазба цифрларын танитын классификатор бастапқыда таза деректерде сенімді нәтиже көрсетті, кейін FGSM әдісі арқылы ε параметрі әртүрлі мәндерде өзгертіліп, adversarial-мысалдар жасалды. Нәтижесінде көзге байқалмайтын шу енгізілгеннің өзінде модельдің сенімді болжамы қате класқа ауысатыны тіркелді (типтік мысалда «7» цифры қате «3» деп танылды). Қорғаныс үшін екі реконструктивті тәсіл салыстырылды: қарапайым автокодировщик

(AE) және шуға үйретілген деноизинг-автокодировщик (DAE). ε артқан сайын DAE қателіктер үлесін AE-ге қарағанда көбірек төмендетті, себебі ол жоғары жиілікті шуды пайдалы белгілерден жақсырақ ажыратты. Қосымша ретінде реконструкция энергиясы E(x) арқылы аномалия детекторы енгізіліп, шабуыл ықтималдығын белгілеу және инцидентті тіркеу мүмкіндігі көрсетілді.

Түйін сөздер: генеративті ЖИ, киберқауіпсіздік, adversarial-шабуыл, FGSM, MNIST, автокодировщик, деноизинг-автокодировщик, аномалия детекторы, реконструкция қатесі.

[1]*Батырханова А.А.,
м.т.н., сениор-лектор, ORCID ID: 0009-0001-6017-1659,
*e-mail: a.batyrkhanova@iitu.edu.kz
[2]Бекмухан А.С.,
м.т.н., сениор-лектор**,** ORCID ID: 0009-0004-0319-0636,
e-mail: a.bekmukhan@iitu.edu.kz
[2]Абильдаева Т.Т.,
м.т.н., сениор-лектор, ORCID ID: 0009-0005-2983-3377,
e-mail: t.abildayeva@iitu.edu.kz
[3]Ескендирова D.M.,
к.т.н., ассоциированный профессор, ORCID ID: 0000-0003-4270-1908,
e-mail: d.yeskendirova@iitu.edu.kz

[1]Международный университет информационных технологий, г. Алматы, Казахстан

ПРИМЕНЕНИЕ ГЕНЕРАТИВНЫХ МОДЕЛЕЙ ИИ ДЛЯ АТАК И ЗАЩИТЫ

Аннотация

В статье рассмотрено двойное применение генеративного искусственного интеллекта в кибербезопасности: как инструмент атаки и как элемент защиты. Показано, что генерация правдоподобного текста, изображений и аудио упрощает злоупотребления (фишинг, дипфейки, подмена коммуникаций), а в задачах машинного обучения возникает отдельный риск – малые, но целенаправленные возмущения входа. Практическая часть выполнена на воспроизводимом стенде с классификатором рукописных цифр MNIST: на чистых данных модель демонстрировала стабильное распознавание, после чего применялась атака FGSM с варьированием параметра ε. Зафиксировано, что визуально почти незаметный шум способен переводить уверенную классификацию в ошибочную (на характерном примере «7» была получена неверная метка «3»). Для противодействия проверены два реконструктивных подхода: обычный автокодировщик (AE) и деноизинговый автокодировщик (DAE), обученный восстанавливать чистый сигнал из зашумленного входа. При росте ε DAE снижал долю ошибок заметнее, чем AE, поскольку лучше отделял высокочастотные компоненты возмущения от полезных штрихов. Дополнительно введен детектор аномалии по энергии реконструкции E(x), позволяющий не только «очищать» вход, но и фиксировать факт подозрительного воздействия для последующего анализа инцидентов.

Ключевые слова: генеративный ИИ, кибербезопасность, adversarial-атаки, FGSM, MNIST, автокодировщик, деноизинговый автокодировщик, детектор аномалий, ошибка реконструкции.