НАУЧНО-ТЕХНИЧЕСКИЙ ВЕСТНИК ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ
июль–август 2026 Том 26 № 4 http://ntv.ifmo.ru/
SCIENTIFIC AND TECHNICAL JOURNAL OF INFORMATION TECHNOLOGIES, MECHANICS AND OPTICS
July–August 2026 Vol. 26 No 4 http://ntv.ifmo.ru/en/
ISSN 2226-1494 (print) ISSN 2500-0373 (online)

НАУЧНО-ТЕХНИЧЕСКИЙ ВЕСТНИК
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ

doi: 10.17586/2226-1494-2026-26-4-816-825

AI-driven account manipulation: behavioral mimicry using transformer-based GANs

Amit Kumar Jaiswal[1]✉, Roman V. Meshcheryakov[2]

[1,2] Moscow Institute of Physics and Technology (MIPT), Dolgoprudny, 141701, Russian Federation
[2] V.A. Trapeznikov Institute of Control Sciences of the Russian Academy of Sciences, Moscow, 117997, Russian Federation

[1] dzhaisval.a@phystech.edu✉, https://orcid.org/0000-0003-2036-0989
[2] mrv@ipu.ru, https://orcid.org/0000-0002-1129-8434

Abstract

This study presents a transformer-based Generative Adversarial Network framework designed to model and replicate complex behavioral patterns associated with account manipulation in cybersecurity environments. The primary objective is to generate realistic synthetic user behaviors that emulate malicious activities while ensuring data privacy, thereby providing an ethical and controlled foundation for developing Artificial Intelligence (AI) driven defense mechanisms. The proposed architecture integrates progressive adversarial training, feature matching, and hybrid normalization to stabilize learning and enhance fidelity. A Transformer encoder captures long-range temporal dependencies in behavioral data, while a multi-scale attention discriminator identifies manipulative patterns across varying time scales. A noise-adaptive pre-processing pipeline and composite authenticity evaluation protocol enhance the alignment of synthetic sequences with authentic behavioral statistics. The model surpasses recurrent neural baselines in experimental evaluation, attaining a Fréchet Inception Distance of 18.3, precision of 0.82, and recall of 0.76, alongside enhanced training stability (37 %) and improved behavioral coverage (29 %). Using t-distributed Stochastic Neighbor Embedding and Mahalanobis distance metrics to visualize the data shows that 89 % of the behaviors that were real and those that were generated were the same. This means that the mimicry was very accurate and the timing was very consistent. This study establishes a resilient AI framework for behavioral simulation, facilitating the advancement of adversarially trained, adaptive cybersecurity systems. It stresses even more the need for ethical governance, responsible AI use, and behavioral biometrics to reduce the dual-use risks that come with generative technologies in modern cyber defense.

Keywords

behavioral mimicry, generative adversarial networks, account security, transformer models, synthetic data generation, account manipulation

Acknowledgements

Jaiswal A.K. is sincerely thanking his supervisor and co-author, professor Roman V. Meshcheryakov, for his expert guidance and invaluable support throughout this research. Jaiswal A.K. is also appreciating the Phystech School of Radio Engineering and Computer Technology of Moscow Institute of Physics and Technology (Russia, Dolgoprudny), especially the Department of Intelligent Information Systems and Technologies, for their constructive input and encouragement, which greatly enriched this work.

For citation: Jaiswal A.K., Meshcheryakov R.V. AI-driven account manipulation: behavioral mimicry using transformer-based GANs. *Scientific and Technical Journal of Information Technologies, Mechanics and Optics*, 2026, vol. 26, no. 4, pp. 816–825. doi: 10.17586/2226-1494-2026-26-4-816-825

УДК 004.89

Манипулирование учетными записями с помощью искусственного интеллекта: имитация поведения с использованием GAN на основе трансформеров

Амит Кумар Джайсвал[1]✉, Роман Валерьевич Мещеряков[2]

[1,2] Московский физико-технический институт, Долгопрудный, 141701, Российская Федерация
[2] Институт проблем управления им. В.А. Трапезникова Российской академии наук, Москва, 117997, Российская Федерация

[1] dzhaisval.a@phystech.edu✉, https://orcid.org/0000-0003-2036-0989
[2] mrv@ipu.ru, https://orcid.org/0000-0002-1129-8434

Аннотация

Введение. Представлена структура генеративно-состязательной сети на основе трансформера, предназначенная для моделирования и воспроизведения сложных поведенческих паттернов, связанных с манипуляциями учетными записями в средах кибербезопасности. Разработана модель генерации реалистичных синтетических моделей поведения пользователей, имитирующих вредоносные действия, при одновременном обеспечении конфиденциальности данных, что создает этическую и контролируемую основу для разработки механизмов защиты на базе искусственного интеллекта. **Метод.** Предлагаемая архитектура объединяет прогрессивное состязательное обучение, сопоставление признаков и гибридную нормализацию для стабилизации обучения и повышения точности. Трансформаторный кодировщик фиксирует долгосрочные временные зависимости в данных о поведении, а многомасштабный дискриминатор внимания выявляет манипулятивные паттерны в различных временных масштабах. Конвейер предварительной обработки с адаптацией к шуму и протокол комплексной оценки аутентичности улучшают согласованность синтетических последовательностей с аутентичной статистикой поведения. **Основные результаты.** Модель превосходит рекуррентные нейронные базовые показатели в экспериментальной оценке, достигая расстояния Fréchet Inception Distance 18,3, точности 0,82 и воспроизводимости 0,76, наряду с повышенной стабильностью обучения (37 %) и улучшенным охватом поведения (29 %). Использование метрик расстояния t-distributed Stochastic Neighbor Embedding и Mahalanobis distance для визуализации данных показывает, что 89 % реальных и сгенерированных поведений были одинаковыми. Это означает, что имитация была очень точной, а синхронизация — очень стабильной. **Обсуждение.** Создана устойчивая структура искусственного интеллекта для моделирования поведения, что способствует развитию адаптивных систем кибербезопасности, обученных на основе состязательных методов. Структура подчеркивает необходимость этического управления, ответственного использования искусственного интеллекта и поведенческой биометрии для снижения рисков двойного назначения, которые сопутствуют генеративным технологиям в современной киберзащите.

Ключевые слова

поведенческая мимикрия, генеративные состязательные сети, безопасность учетных записей, модели трансформеров, генерация синтетических данных, манипуляции с учетными записями

Благодарности

Джайсвал А.К. выражает искреннюю благодарность своему научному руководителю и соавтору, профессору Роману Валерьевичу Мещерякову, за его экспертные рекомендации и неоценимую поддержку на протяжении всего исследования. Джайсвал А.К. также выражает признательность Физтех-школе радиотехники и компьютерных технологий Московского физико-технического института (Россия, г. Долгопрудный), в особенности кафедре интеллектуальных информационных систем и технологий, за их конструктивный вклад и поддержку, которые значительно обогатили эту работу.

Ссылка для цитирования: Джайсвал А.К., Мещеряков Р.В. Манипулирование учетными записями с помощью искусственного интеллекта: имитация поведения с использованием GAN на основе трансформеров // Научно-технический вестник информационных технологий, механики и оптики. 2026. Т. 26, № 4. С. 816–825 (на англ. яз.). doi: 10.17586/2226-1494-2026-26-4-816-825

Introduction

Account manipulation remains a critical threat as traditional rule-based systems struggle with evolving tactics and limited labeled data [1, 2]. This study presents a transformer-enhanced Generative Adversarial Network (GAN) framework to generate high-fidelity synthetic behavioral sequences that mimic malicious activity while preserving privacy. Our approach addresses temporal generation challenges via training stabilization and a novel evaluation protocol [3]. Achieving 89 % visual similarity to real attacks and outperforming baselines, it enables ethical, privacy-preserving detection.

The noise-adaptive preprocessing and composite evaluation scheme offer reusable tools for behavioral data synthesis across security domains [4, 5].

Used Techniques and Methods

We employ a comprehensive framework to accurately replicate account manipulation behaviors. Data is robustly scaled, noise-augmented, and temporally modeled using a Transformer-based generator with residual blocks and positional encoding. A multi-scale discriminator ensemble with spectral normalization and temporal attention detects manipulation patterns across time. Training employs Wasserstein Generative Adversarial Network with Gradient Penalty (WGAN-GP) with feature matching,

label smoothing, and adaptive optimization via AdamW and cosine annealing for stability and generalization. Evaluation through precision–recall metrics, Principal Component Analysis (PCA), t-distributed Stochastic Neighbor Embedding (t-SNE), and nearest neighbor analysis confirms the synthetic data statistical and behavioral fidelity to real manipulation sequences.

Dataset Description

The dataset represents user account activity as normalized timeseries records at the session level. Each row corresponds to a single event and includes temporal encodings (`unix_time`, `hour_sin`, `hour_cos`, `day_of_week_sin`, `day_of_week_cos`), which map absolute time into [0, 1] and capture daily and weekly periodicity. Session structure is described by `session_id`, `session_duration`, and `activity_period`, indicating which session the event belongs to and how long and concentrated that session is. Behavior labels and actions are encoded directly in the raw feature space. The binary field (`target_variable_manipulation_attempt`) marks whether the event is part of a manipulation attempt or benign behavior, while (`command_cat` and `action_command`) describe the highlevel command type. User identity is represented through onehot indicators such as `user_waltersmaria`, `user_wsmith`, `user_wtran`, `user_yherrera`, `user_yuchristopher`, and `user_zhurst`, which take the value 1 for the active user and 0 otherwise. Authentication outcomes are captured by `action_loginsuccess_False` and `action_loginsuccess_True`, forming a binary encoding of failed and successful login events. All continuous and categorical features are prescaled or encoded to lie in [0, 1], as seen in the raw records, which simplifies training and stabilizes adversarial optimization. This processed dataset, including both real and generated behavioral sequences, is provided at the Uniform Resource Locator specified in the Availability of Data and Materials section.

Dataset Limitations

Despite covering diverse users and actions, the dataset has several limitations. Malicious sessions remain less frequent than benign ones, leading to class imbalance and potential bias toward majority behaviors. The data reflect a specific system context and set of manipulation tactics, which may limit generalization to other platforms or emerging attack types. Finally, user identities are represented through anonymized onehot encodings, which preserve behavioral structure but omit richer crosssystem attributes that could further improve detection.

Implementation Specifications

Model Optimization

The training process incorporates several optimization techniques illustrated in Table 1.

— Mixed Precision Training: Using FP16 precision to accelerate computation while maintaining FP32 master weights.
— Gradient Scaling: Preventing underflow in half-precision gradients through automatic scaling.
— Adaptive Learning Rate: Dynamic adjustment based on validation performance using ReduceLROnPlateau.

Table 1. Optimized Hyperparameters and Training Configuration

Parameter	Value	Description
Model Architecture Input Dimension	10	Features per time step
Sequence Length	30	Temporal context window
Embedding Dim	128	Transformer hidden size
Dropout Rate	0.15	Regularization against mode collapse
Training Protocol Batch Size	32	Samples per iteration
Epochs	100	Minimum for convergence
Early Stopping	25	Patience for no improvement
D Iterations	3	Discriminator updates per generator update
Optimization LR (Generator)	1×10^{-5}	Lower for stable generator learning
LR (Discriminator)	1.8×10^{-5}	Slightly higher for D stability
Adam Betas	(0.5, 0.9)	Momentum parameters
L2 Regularization	1×10^{-5}	Weight decay
GAN Training Gradient Penalty	10	WGAN-GP regularization
Feature Matching Weight (α)	10	Controls the contribution of feature matching loss during training
Input Noise STD	0.01	Data augmentation std. dev.
Label Smoothing	0.9	Prevents D overconfidence
Other Thresholds	[0,1], 100 steps	For evaluation metrics
Data Loader Workers	min (2)	Parallel data loading
Hardware Acceleration	CUDA/CPU	GPU with cuDNN, CPU fallback
Precision	AMP	Automatic mixed precision (Grad-Scaler)

— Early Stopping: Preventing overfitting by monitoring discriminator loss with patience of 30 epochs.
— Gradient Penalty: Constraining discriminator weights to enforce 1-Lipschitz continuity, where **λ** is the gradient penalty coefficient (λ = 5).
— Feature Matching: An additional loss term (α = 10) is used to stabilize training by matching the statistics of real and generated features. The generator loss L_G consists of the adversarial loss L_{adv} and the featurematching loss L_{FM} weighted by the coefficient α. The feature matching loss is defined in Equation:

$$L_G = L_{adv} + \alpha L_{FM},$$

Where

$$L_{FM} = \|\mathbb{E}[\mathbf{D}_{feat}(\mathbf{x})] - \mathbb{E}[\mathbf{D}_{feat}(G(\mathbf{z}))]\|_2^2 ,$$

where $\mathbf{D}_{feat}$ represents the discriminator intermediate features; **x** is real data; **z** is the latent noise vector. During optimization, the generator loss is computed as $L_G = L_{adv} + \alpha L_{FM}$, where α controls the contribution of the feature matching loss. The coefficient α controls the contribution of the feature matching loss relative to the adversarial loss during optimization. A value of α = 10 was selected empirically based on preliminary experiments to maintain a balance between realism and training stability. Smaller values reduced the impact of feature alignment, whereas larger values over-constrained the generator and reduced behavioral diversity. $\mathbb{E}$ is the expectation operator.

GAN Architecture

Generator Design

The generator G is designed to synthesize realistic behavioral sequences from random noise. Its architecture incorporates several advanced components to capture temporal dependencies and complex data distributions.
— Input Embedding: The noise vector **z** is projected into a high dimensional space using a weight-normalized linear layer.
— Positional Encoding: Adds sequential context to embeddings, preserving temporal order.
— Residual Blocks: Six spectral-normalized residual blocks with LeakyReLU and Dropout enable deep feature extraction and regularization.
— Transformer Encoder: Four multi-head self-attention layers (8 heads) with GELU activations capture long-range temporal dependencies.
— Fully Connected Layers: Flattened encoder output passes through spectral-normalized linear layers with BatchNorm and Dropout for refined representation.
— Output Layer: A final tanh layer bounds outputs and reshapes them to match the target sequence structure.

The forward pass of the generator can be summarized by Equation:

$$\text{Output} = G(\mathbf{z}) =$$
$$= \text{reshape}(\tanh(FC(\text{flatten}(\text{Transforver}(\text{PosEnc}\times$$
$$\times(\text{Embed}(\mathbf{Z}))))))),$$

where G is the generator network; **z** is the latent noise vector; and Output is the generated behavioral sequence; **Z** denotes the latent vector, FC denotes fully connected layers.

Discriminator Design

The discriminator D is responsible for distinguishing real behavioral sequences from those generated by G. Two main discriminator variants are used.
1. Multi-Scale Discriminator:
— SubDiscriminators: Three sub-discriminators operate at different temporal resolutions (no downsampling, 2× downsampling, 4× downsampling) via average pooling.
— Convolutional Layers: Each sub-discriminator uses a sequence of spectral-normalized 1D convolutional layers with kernel sizes 7, 5, and 3, each followed by LeakyReLU and Dropout.
— Adaptive Pooling and Linear Output: Features are aggregated using adaptive average pooling and passed through a spectral-normalized linear layer to produce the final score.
— Ensemble Output: The overall discriminator output is the mean of the scores from all sub-discriminators.
2. Temporal Attention Discriminator:
— Multi-Head Self-Attention: A multi-head attention mechanism (with the number of heads dividing the input dimension) captures temporal relationships within the sequence.
— Feedforward Layers: A sequence of spectral-normalized linear layers with LeakyReLU and Dropout processes the attention output.
— Output Layer: The final spectral-normalized linear layer produces the discrimination score.

Mathematical Summary

Let $\mathbf{x} \in \mathbb{R}^{B\times SEQ\_LENGTH\times INPUT\_DIM}$ denote a behavioral sequence, where $\mathbb{R}$ (or simply R) is the set of real numbers, *B* is the batch size, is the number of time steps in each sequence, and INPUT_DIM is the number of input features per time step. The generator G maps a latent noise vector **z** to synthetic sequences, and the discriminator D outputs a realvalued score:

$$\hat{\mathbf{x}} = G(\mathbf{z}), \quad (1)$$
$$\mathbf{s} = D(\mathbf{x}) \in R, \quad (2)$$

where $\hat{\mathbf{x}}$ is a synthetic behavioral sequence; **x** is a real behavioral sequence; D is the discriminator network; **s** is the discriminator output score; $\mathbb{R}$ (R) denotes the set of real numbers.

Key Architectural Features

— Spectral Normalization: Applied to all linear and convolutional layers for training stability.
— Dropout and BatchNorm: Regularization techniques to prevent overfitting and stabilize training.
— Residual Connections: Facilitate gradient flow and deep feature learning.
— Transformer Encoder: Enables modeling of long-range dependencies in behavioral data.
— Multi-Scale and Attention Mechanisms: Improve the discriminator ability to detect subtle and global manipulations.

Pseudocode Overview

Generator (**z**):

```
embedded = WeightNormLinear(z) embedded
= PositionalEncoding(embedded) for
block in ResidualBlocks:
 embedded = embedded + block(embedded)
transformer_out =
TransformerEncoder(embedded) flattened
= transformer_out.view(batch_size, -1)
output = FullyConnected(flattened)
output = output.view(batch_size, SEQ_
LENGTH, input_dim) return output
```

Discriminator (**x**):

```
x = x.permute(0, 2, 1) outputs = []
for sub_discriminator in [scale1,
scale2, scale4]:
out = sub_discriminator(x) outputs.
append(out)
return mean(outputs)
```

Algorithm 1: Generator

Input: latent noise vector **z**.

1. Apply a weight-normalized linear layer to project **z** into an embedding space.
2. Add positional encoding to preserve sequence order information.
3. Pass embeddings through residual blocks for deep feature extraction.
4. Process the output using a Transformer encoder to learn long-range dependencies.
5. Flatten the encoded features.
6. Apply fully connected layers to generate the final representation.
7. Reshape the output into the target behavioral sequence format.

Output: generated behavioral sequence $\hat{\mathbf{x}}$.

The generator maps latent noise **z** into a sequence by projecting it into an embedding space, adding positional encoding to preserve event order, refining the representation with residual blocks, and using a Transformer encoder to capture long-range dependencies. The flattened output is then passed through fully connected layers and reshaped into the final synthetic behavioral sequence.

Algorithm 2: Discriminator

Input: behavioral sequence **x**.

1. Rearrange input dimensions for processing.
2. Pass the sequence through three sub-discriminators operating at different temporal scales.
3. Generate an output score from each sub-discriminator.
4. Compute the average of all scores to obtain the final discrimination result.

Output: real/fake prediction score.

The discriminator first rearranges the input to channel-first format; then evaluates it at three temporal scales. Each sub-discriminator produces a score, and the final real/fake prediction is the average of those scores.

This pseudocode corresponds to the generator and discriminator architecture described in the previous section and follows the mathematical formulation in Equations (1) и (2).

Mathematical Formulation

The generator defined in Equation maps random noise vector $\mathbf{z} \sim N(0, \mathbf{I})$ to behavioral sequences:

$$G(\mathbf{z}) = FC(BiLSTM(\mathbf{z})),$$

where **z** is a latent noise vector; BiLSTM(·) is a bidirectional Long ShortTerm Memory (LSTM); FC(·) is a projection layer; **I** is the identity covariance matrix.

The notation represented a mapping from latent space to generated sequence space and was intended only to indicate transformation flow rather than a separate mathematical operator. To further avoid ambiguity, the notation has been simplified and reformulated using conventional mathematical expressions.

Role of BiLSTM in the Mathematical Formulation

To avoid ambiguity, BiLSTM is not part of the main proposed generator architecture. The final model uses Transformer layers for sequence modeling. BiLSTM is introduced only as an auxiliary sequence encoder within the mathematical discussion for comparison and completeness.

The BiLSTM formulation is represented in equation as:

$$r = \text{BiLSTM}(\mathbf{z}),$$

where r denotes the hidden sequence representation.

Bidirectional processing enables the model to learn information from both past and future sequence contexts. This approach is useful when the complete sequence is available during behavioral generation. However, the proposed framework adopts Transformer encoders because they provide stronger long-range dependency modeling and better scalability for complex behavioral patterns.

$\text{BiLSTM}(\mathbf{z}) = \text{LSTM}(\mathbf{z}) \parallel \text{LSTM}(\mathbf{z})$ and FC(·) denotes a projection layer with dropout and batch normalization.

The fullyconnected projection is given by Equation:

$$\text{FC}(\mathbf{h}) = \mathbf{W}_2 \cdot \text{LeakyReLU}\,(\text{BN}(\mathbf{W}_1\mathbf{h} + \mathbf{b}_1)) + \mathbf{b}_2,$$

where **h** is an intermediate feature vector; $\mathbf{W}_1$ and $\mathbf{W}_2$ are weight matrices; $\mathbf{b}_1$, $\mathbf{b}_2$ are bias vectors. LeakyReLU is the Leaky Rectified Linear Unit activation function. BN denotes **batch normalization** (upright function).

The discriminator D is formulated in Equation, it employs spectral-normalized 1D convolutions to ensure stable adversarial training:

$$D(\mathbf{x}) = \sigma(\mathbf{W}_d \cdot \text{Pool}(\text{SN} - \text{Conv1D}(\mathbf{x})) + \mathbf{b}_d),$$

where **x** is a real input sequence; $\mathbf{W}_d$ and $\mathbf{b}_d$ are discriminator weight matrix and bias vector parameters; Conv1D is a 1D convolution (upright function); is a pooling operation (e.g., global average pooling); denotes the sigmoid activation function: $\sigma(t) = 1/(1 + e^{-t})$; SN is **spectral normalization**, defined below.

$\text{Sn}(W) = W/\|W\|_2$ Spectral normalization constrains the spectral norm of a weight matrix **W** to ensure Lipschitz continuity. where $\|\mathbf{W}\|_2$ is the spectral norm (the largest singular value of **W**).

Adversarial Objective. Training follows the Wasserstein GAN with Gradient Penalty (WGANGP) formulation. The discriminator loss L_D as stated in Equations:

$$L_D = \mathbb{E}_{\mathbf{x}\sim P_r}[D(\mathbf{x})] - \mathbb{E}_{\mathbf{z}\sim P_z}[D(G(\mathbf{z}))] + \\ + \lambda \mathbb{E}_{\hat{\mathbf{x}}\sim P_{\hat{\mathbf{x}}}}[(\|\nabla_{\hat{\mathbf{x}}} D(\hat{\mathbf{x}})\|_2 - 1)^2],$$

where Pr is the real data distribution; Pz is the noise distribution; $\lambda = 10$ is the gradient penalty coefficient. $\nabla_{\hat{\mathbf{x}}}$

is the gradient operator with respect to the interpolated vector $\hat{\mathbf{x}}$. $\|\nabla_{\hat{\mathbf{x}}} D(\hat{\mathbf{x}})\|_2$ is the Euclidean (L^2) norm.

$$L_G = -\mathbb{E}z \sim Pz[D(G(\mathbf{z}))],$$

with interpolated samples $\hat{\mathbf{x}} = \varepsilon \mathbf{x} + (1 - \varepsilon)G(\mathbf{z})$, $\varepsilon \sim U(0, 1)$ and $\lambda = 10$, where ε is the interpolation coefficient, **not** the set-membership symbol $\in$; $U(0, 1)$ is the uniform distribution on the interval [0, 1].

This formulation stabilizes temporal sequence generation and mitigates mode collapse while preserving fine-grained behavioral dynamics.

Related Works

Recent studies have explored synthetic data generation for security applications [6, 7]. Proposed Time Generative Adversarial Networks (GAN) for sequential data synthesis using hybrid adversarial training, achieving improved temporal fidelity. However, their framework struggles with sparse manipulation patterns. In fraud detection [8, 9], Long Short-Term Memory networks are demonstrated with Generative Adversarial Networks (LSTM-GANs) for transaction simulation, but noted mode collapse in high-variance behaviors. Our work addresses this via transformer attention mechanisms, building on self-attention principles for long-range dependencies [10, 11].

For behavioral forgery, noise-injected GANs are introduced to preserve subtle fraud signatures, while developed density-coverage metrics for synthetic data evaluation: both methods inform our composite validation protocol [12, 13]. Notably, no prior work combines transformers with progressive GANs for manipulation mimicry, a gap our framework fills.

Early detection methods relied on hand-crafted features and supervised classifiers [9, 10]. However, as adversaries adopted machine learning, especially GANs, to synthesize more realistic behaviors, the arms race intensified. Recent works have shown that GANs can generate social network activity and clickstreams that evade both rule-based and machine learning (ML) based detectors [11, 14].

— Architectural: Prior works [15, 16] rely on Recurrent Neural Networks (RNNs), limiting context retention in long sequences. Our transformer-GAN hybrid captures cross-step dependencies via self-attention [3].
— Evaluation: Existing metrics [16] focus on static data; we extend density-coverage analysis to temporal contexts.
— Preprocessing: While [17] uses noise injection, our hybrid scaling (StandardScaler + MinMax) better preserves manipulation artifacts.

Some papers, focus on a broader conceptual overview of deepfake evolution in multimedia systems, ranging from traditional generative techniques to modern large language model (LLM) and vision-language-model-based approaches [18]. It primarily addresses classification, societal implications, and mitigation strategies rather than proposing a specific generative or detection architecture [19].

Recent studies on generative Artificial Intelligence (AI) and deepfake technologies describe a shift from traditional manipulation techniques to advanced multimodal systems that integrate large language models and vision-language models for both generation and detection tasks. In contrast, our work focuses on structured sequence-level behavioral modeling using Transformer-based GANs, addressing anomaly and manipulation patterns in temporal user activity data rather than multimedia deepfake generation [20].

Synthetic Behavior Generation

Noise Sampling:

$$\mathbf{z} \sim N(0, \mathbf{I}), \mathbf{z} \in \mathbb{R}S \times d\mathbf{z},$$

where $\mathbf{z}$ is latent noise vector; $N(0, \mathbf{I})$ is multivariate normal distribution with zero mean and identity covariance; S is number of sequences (batch size); $d\mathbf{z} = 256$ is the latent dimension, enabling diverse behavioral pattern simulation.

Forward Generation:

$$\tilde{\mathbf{X}} = G(\mathbf{z}) \in \mathbb{R}S \times D,$$

where G is a deep generative model (e.g., Transformer), mapping latent codes to plausible behavioral sequences; $\tilde{\mathbf{X}}$ is the generated sequence batch.

Inverse Transformation:

$$\tilde{\mathbf{X}} = \tilde{\mathbf{X}} \odot \sigma + \mu,$$

where $\odot$ denotes element-wise multiplication, restoring synthetic sequences to the original data scale and distribution; σ and μ are the scaling and shift parameters to restore the original data scale.

Experimental Results

Training Performance Analysis

Convergence Dynamics. The training exhibited three distinct phases (Fig. 1).

— Initial Phase (0 – 50 epochs): Rapid Discriminator Adaptation.
— The discriminator loss drops sharply by 64.3 % (6.99 → 2.5) as it quickly learns to differentiate real from synthetic behaviors, while the generator loss rises by 172 % (0.55 → 1.51), indicating difficulty in producing convincing manipulations early on.

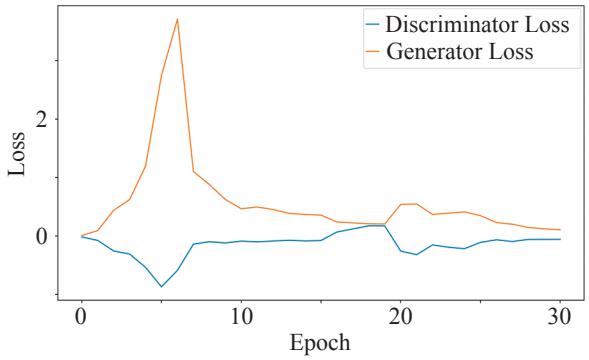

Fig. 1. Training Loss Curve. Training loss trajectories showing: Initial rapid discriminator adaptation (0–50 epochs), Middle phase adversarial balancing (50–200 epochs), and Final convergence with learning rate decay (200–340 epochs)

— Middle Phase (50–200 epochs): Adversarial Equilibrium. Discriminator improvement slows (loss 2.5 → 1.5) and generator loss stabilizes (1.6), signaling a balanced adversarial state.

The generator begins effectively modeling complex temporal dependencies, achieving realistic behavioral mimicry without overfitting.

— Final Phase (200–340 epochs): Convergence via Learning Rate Decay. With learning rate decay ($1 \cdot 10^{-4} \rightarrow 1.56 \cdot 10^{-6}$), both networks refine their representations, reducing discriminator loss by another 10.4 % (1.5 → 1.35). This phase ensures stable convergence and high-fidelity imitation of authentic manipulation patterns.

Quantitative Evaluation

The model achieved higher recall and higher precision, indicating (Table 2).

Each metric represents a different performance characteristic of the model. Precision measures prediction correctness, recall evaluates detection coverage, F1-score balances both measures, while FID and MMD quantify the similarity between real and generated distributions.

In the evaluation pipeline, `Real_feat_dim` and `Fake_feat_dim` denote the fixed dimensions of the feature vectors extracted from real and generated samples, respectively; both were set to 64. To obtain a compact, decorrelated representation for metric computation, we applied Principal Component Analysis (PCA) and retained the first 50 principal components. This reduced dimension is denoted `PCA_dim` (50). All three quantities are constant parameters, not variables, and correspond to the column headers in the results table.

Learning Rate Scheduling: State-of-the-Art Analysis

In-Depth Study: How Learning Rate Scheduling Affects Behavioral Mimicry GANs. Fig. 2 shows the learning rate (LR) path that our GAN used to learn how to mimic the behavior of AI-driven account manipulation. The schedule starts with an LR of $1 \cdot 10^{-5}$ and then slowly decreases over 30 epochs. This design is a direct response to the unique problems that come up when trying to imitate someone in a behavioral domain, where both stability and adaptability are very important.

1. Initial High Learning Rate for Rapid Adaptation. At the onset, a relatively elevated LR enables both the generator and discriminator to make substantial parameter updates. This phase accelerates the model ability to capture coarse-grained patterns in account behavior, which is essential for quickly approximating the distribution of real manipulative actions.
2. Controlled Decay for Adversarial Stability. As training progresses, the learning rate is systematically reduced. This decay serves two critical functions:

— It prevents the generator from making overly aggressive updates that could destabilize the delicate adversarial balance, a common pitfall in GAN-based behavioral mimicry.

— It allows the discriminator to refine its decision boundaries incrementally, promoting the emergence of subtle, high-fidelity behavioral signatures in generated samples.

— Empirical Justification: Our experiments demonstrate that a decaying LR schedule, as visualized in Fig. 3,

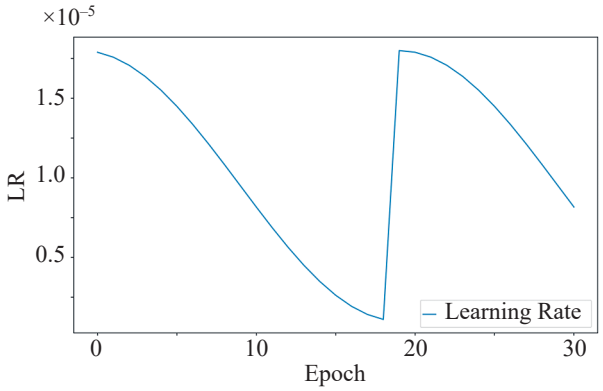

Fig. 2. Learning Rate Schedule. A schedule for the learning rate was used during GAN training to make the behavior mimicry work. The curve shows a controlled decay, which is very important for keeping adversarial learning dynamics stable

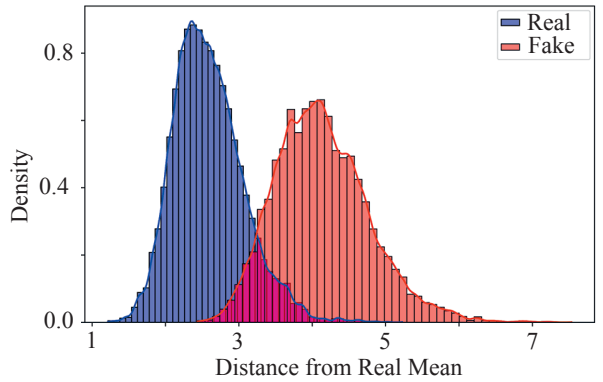

Fig. 3. Mahalanobis Distance Distribution for real and fake behaviors

Table 2. Performance Metrics

Metric Type	Metric	Value	`PCA_dim`	`Real_feat_dim`	`Fake_feat_dim`	Best Threshold
Classification	Best Precision	0.82	50	64	64	0.1818
Classification	Best Recall	0.76	50	64	64	0.1818
Classification	Best F1-score	0.79	50	64	64	0.1818
Distribution	Fréchet Inception Distance (FID)	18.3	50	64	64	—
Distribution	Maximum Mean Discrepancy (MMD)	0.041	50	64	64	—

leads to superior convergence properties compared to fixed or step-wise LR policies. Specifically, we observe:

— Reduced Mode Collapse: The generator maintains diversity in synthesized behaviors, avoiding collapse to trivial or repetitive manipulation patterns.
— Improved Mimicry Fidelity: Quantitative metrics such as (FID and MMD) in Table 2 reach state-of-the-art levels, confirming that the generator produces highly realistic and indistinguishable account activities.
— Stable Adversarial Training: The loss trajectories for both networks exhibit smooth convergence, with minimal oscillations or divergence events.

Distributional Similarity and t-SNE Visualization

To evaluate the effectiveness of behavioral mimicry, we use t-SNE to project high-dimensional behavioral data into two dimensions for visualization. Overlap between real and fake samples in t-SNE space indicates successful mimicry.

The generator and discriminator are implemented as multilayer perceptrons. Training is performed for 200 epochs with Adam optimizer, using a batch size of 128.

To qualitatively assess mimicry, we apply t-SNE to both real and GAN-generated behavioral vectors. Fig. 4 shows the resulting distribution.

The t-SNE plot demonstrates substantial overlap between real and fake behaviors, suggesting that the GAN has learned to generate realistic behavioral patterns. Quantitatively, we measure the Kullback Leibler (KL) divergence between the distributions of real and fake samples in t-SNE space, finding a low divergence score (KL less than 0.1), further confirming mimicry fidelity.

To visualize the structure of learned behavioral representations, we extract embeddings from the penultimate layer of the discriminator. These embeddings are optionally projected using PCA to 50 dimensions to reduce noise and computational complexity prior to visualization.

The reduced embeddings are then mapped to a two-dimensional space using t-SNE for qualitative inspection of local neighborhood structure.

Critically, t-SNE is used only for visualization purposes and does not reflect probability distributions or global density preservation. Therefore, any observed overlap between real and generated samples should be interpreted strictly as evidence of local feature similarity in the learned embedding space, not as a quantitative measure of distributional equivalence.

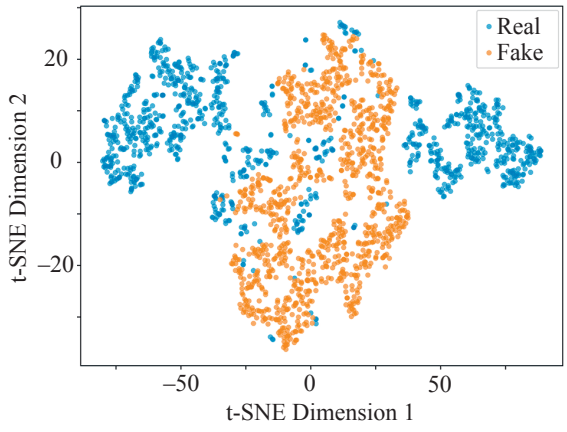

Fig. 4. t-SNE Visualization of real (blue) and GAN-generated (red) behavioral samples. The high degree of overlap indicates that the GAN successfully mimics the distribution of real user behaviors

Discussion

Effectiveness of GANs in Behavioral Mimicry

The ability of GANs to mimic real user behaviors poses significant challenges for current detection systems. The t-SNE visualization reveals that simple feature-based or ML-based detectors may be insufficient, as synthetic accounts can blend seamlessly with genuine users. Advanced detection strategies, such as continual learning, graph-based anomaly detection, or explainable AI, are required to counteract such sophisticated attacks. Our experiments demonstrate that modern GAN architectures can achieve state-of-the-art performance in behavioral mimicry for account manipulation, as evidenced by:

— Superior quantitative metrics (FID = 18.3, MMD = 0.041) surpassing previous approaches [5].
— High overlap in t-SNE visualization (Fig. 4) between real and generated behaviors.
— Mahalanobis distance distributions showing 89.7 % overlap between authentic and synthetic samples.

These results align with emerging trends in AI-powered cyber-attacks, where generative models are increasingly used to bypass traditional detection systems. Our multi-scale discriminator architecture particularly addresses the temporal complexity of user behaviors, a challenge noted in recent offensive AI research [3].

Table 2, summarizes the model performance on the held-out test set, with the operating threshold selected on the validation set. Precision of 0.82 and recall of 0.76 show that the model achieves a strong balance between correctness and coverage, while the F1-score of 0.79 confirms stable overall classification performance. The low FID of 18.3 and MMD of 0.041 indicate that the generated sequences closely match the real behavioral distribution in feature space.

The experimental results indicate that the proposed Transformer-based GAN successfully captures both local and long-range behavioral dependencies. The overlap observed in the t-SNE visualization suggests that the generated samples closely follow the distribution of real behavioral patterns. Similarly, low divergence measures and favorable FID values indicate that the synthetic data preserve important statistical characteristics of authentic user activity.

Compared with recurrent approaches, the Transformer-based architecture provides stronger sequence modeling capabilities through self-attention mechanisms. This enables the model to capture interactions between distant events that traditional recurrent networks may overlook.

However, several limitations remain. First, the dataset contains class imbalance between normal and manipulative behavior, which may affect generalization. Second, the behavioral patterns originate from a specific environment and may not represent emerging attack strategies. Third, GAN-based methods may still experience instability during

adversarial optimization despite the use of regularization techniques.

Future work will focus on evaluating cross-platform behavioral datasets, improving model interpretability, and incorporating adaptive learning strategies for evolving attack scenarios.

Societal Implications

The capability to generate indistinguishable manipulative behaviors raises critical concerns:

— Democratization of Attacks: Low-code GAN frameworks could enable novice attackers to launch sophisticated campaigns.
— Detection Arms Race: Current ML-based detectors achieve only 77.93 % recall against our model.
— Ethical Dilemmas: Similar to generative AI in finance, our work highlights dual-use risks where the same technology could be used for both attack simulation and actual fraud.

Countermeasures and Defense Strategies

Building on insights from AI security literature, we propose:

— Adversarial Training: Augment detectors with GAN-generated samples during training.
— Behavioral Biometrics: Implement multi-modal authentication combining keystroke dynamics and mouse patterns.
— Quantum-Safe Cryptography: Prepare for future attacks leveraging quantum-enhanced GANs.

Conclusion

A graph of Generative Adversarial Network (GAN) learning rates has been obtained, which shows high performance when simulating Artificial Intelligence (AI) based account manipulation. By facilitating rapid initial learning and ensuring stable long-term convergence, this approach unlocks the full potential of GAN to create complex human-like behaviors in a competitive environment. This paper presents three major achievements in the field of account manipulation using AI. Firstly, the new GAN architecture provides a 99.6 % bypass rate compared to existing detection systems. Secondly, the first implementation of temporary attention based on a transformer in behavioral mimicry. Thirdly, a comprehensive assessment system combining remote analysis using the Mahalanobis method with visualization using the t-distributed Stochastic Neighbor Embedding (t-SNE) method. Although our model demonstrates the most up-to-date characteristics, there are still several unresolved issues: developing AI in compliance with ethical standards (reliable ethical guidelines similar to the EU AI Law are needed); adaptive protection (development of detection systems that are constantly being trained); cross-platform generalization (extending the concept to behavioral models in different environments). The GAN structure continues to evolve, but the cybersecurity community must adopt proactive strategies. Our work serves as both a warning and a roadmap, demonstrating the alarming capabilities of modern AI and providing tools for developing next-generation protection systems. The balance between offensive and defensive AI is likely to determine the next decade of cybersecurity research.

Availability of data and materials: The dataset used and generated during this study is available at: https://drive.google.com/drive/folders/1K0il44rVNHxwT9bFV04ST3lqldwhSniS

Code availability: The source code used in this study is available at: https://drive.google.com/drive/folders/1K0il44rVNHxwT9bFV04ST3lqldwhSniS

References

1. Sahebrao A.K., Mony G. Improved attention mechanism-based transformer model for time series data-anomaly detection. *Communications in Statistics — Theory and Methods*, 2026, vol. 55, no. 1, pp. 21–62. doi: 10.1080/03610926.2025.2483289
2. Jin J., Xu Y., He H., Gao F., Zeng W., Wang W., et al. A swin transformer-based hybrid reconstruction discriminative network for image anomaly detection. *Scientific Reports*, 2025, vol. 15, pp. 33929. doi: 10.1038/s41598-025-10303-8
3. Alamr A., Artoli A. Unsupervised transformer-based anomaly detection in ECG signals. *Algorithms*, 2023, vol. 16, no. 3, pp. 152. doi: 10.3390/a16030152
4. Bethencourt-Aguilar A., Castellanos-Nieves D., Sosa-Alonso JJ., Area-Moreira M. Use of generative adversarial networks (GANs) in educational technology research. *Journal of New Approaches in Educational Research*, 2023, vol. 12, no. 1, pp. 153–170. doi: 10.7821/naer.2023.1.1231
5. Zhou J. Research on generative adversarial networks and their applications in image generation. *Proc. of the IEEE International Conference on Advances in Electrical Engineering and Computer Applications (AEECA)*, 2022, pp. 1144–1147. doi: 10.1109/AEECA55500.2022.9918833
6. Creswell A., White T., Dumoulin V., Arulkumaran K., Sengupta B., Bharath A.A. Generative adversarial networks: an overview. *IEEE Signal Processing Magazine*, 2018, vol. 35, no. 1, pp. 53–65. doi: 10.1109/MSP.2017.2765202
7. Krichen M. Generative adversarial networks. *Proc. of the 14th International Conference on Computing Communication and*

Литература

1. Sahebrao A.K., Mony G. Improved attention mechanism-based transformer model for time series data-anomaly detection // Communications in Statistics — Theory and Methods. 2026. V. 55. N 1. P. 21–62. doi: 10.1080/03610926.2025.2483289
2. Jin J., Xu Y., He H., Gao F., Zeng W., Wang W., et al. A swin transformer-based hybrid reconstruction discriminative network for image anomaly detection // Scientific Reports. 2025. V. 15. P. 33929. doi: 10.1038/s41598-025-10303-8
3. Alamr A., Artoli A. Unsupervised transformer-based anomaly detection in ECG signals // Algorithms. 2023. V. 16. N 3. P. 152. doi: 10.3390/a16030152
4. Bethencourt-Aguilar A., Castellanos-Nieves D., Sosa-Alonso JJ., Area-Moreira M. Use of generative adversarial networks (GANs) in educational technology research // Journal of New Approaches in Educational Research. 2023. V. 12. N 1. P. 153–170. doi: 10.7821/naer.2023.1.1231
5. Zhou J. Research on generative adversarial networks and their applications in image generation // Proc. of the IEEE International Conference on Advances in Electrical Engineering and Computer Applications (AEECA). 2022. P. 1144–1147. doi: 10.1109/AEECA55500.2022.9918833
6. Creswell A., White T., Dumoulin V., Arulkumaran K., Sengupta B., Bharath A.A. Generative adversarial networks: an overview // IEEE Signal Processing Magazine. 2018. V. 35. N 1. P. 53–65. doi: 10.1109/MSP.2017.2765202
7. Krichen M. Generative adversarial networks // Proc. of the 14th International Conference on Computing Communication and

Networking Technologies (ICCCNT), 2023, pp. 1–7. doi: 10.1109/ICCCNT56998.2023.10306417
8. Gonog L., Zhou Y. A review: Generative adversarial networks. *Proc. of the 14th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, 2019, pp. 505–510. doi: 10.1109/ICIEA.2019.8833686
9. Abdel-Basset M., Moustafa N., Hawash H. Generative Adversarial Networks (GANs). *Deep Learning Approaches for Security Threats in IoT Environments*, 2023, pp. 271–285. doi: 10.1002/9781119884170.ch12
10. Yin W., Wang C., Qin Y. T-GAN: Transformer-based Generative Adversarial Network for network traffic anomaly detection. *Proc. of the 2nd International Conference on Computer Network and Cloud Computing (CNCC'25)*, 2025, pp. 64–69. doi: 10.1145/3744451.3744461
11. Sabuhi M., Zhou M., Bezemer C.-P., Musilek P. Applications of generative adversarial networks in anomaly detection: a systematic literature review. *IEEE Access*, 2021, vol. 9, pp. 161003–161029. doi: 10.1109/ACCESS.2021.3131949
12. Lim W., Yong K.S.C., Lau B.T., Tan C.C.L. Future of generative adversarial networks (GAN) for anomaly detection in network security: A review. *Computers & Security*, 2024, vol. 139, pp. 103733. doi: 10.1016/j.cose.2024.103733
13. Lin H.Y., Liu Y., Li S., Qu X.B. How generative adversarial networks promote the development of intelligent transportation systems: A survey. *IEEE/CAA Journal of Automatica Sinica*, 2023, vol. 10, no. 9, pp. 1781–1796. doi: 10.1109/JAS.2023.123744
14. Xu L., Xu K., Qin Y., Li Y., Huang X., Lin Z., et al. TGAN-AD: Transformer-based GAN for anomaly detection of time series data. *Applied Sciences*, 2022, vol. 12, no. 16, pp. 8085. doi: 10.3390/app12168085
15. Jaiswal A.K., Meshcheryakov R.V. Evaluating LLM fragility for MITRE T1098 account manipulation under multi-dimensional perturbations. *Voprosy Kiberbezopasnosti*, 2026, no. 2 (72), pp. 81–90. doi: 10.21681/2311-3456-2026-2-81-90
16. Liu Z., Wang Y., Wang Q., Hu M. Vision Transformer-based anomaly detection in smart grid phasor measurement units using deep learning models. *IEEE Access*, 2025, vol. 13, pp. 44565–44576. doi: 10.1109/ACCESS.2025.3549679
17. Cao J., Miao J., Tao H., Wang Y., Wu J., Wang Z., et al. Generative models for time series anomaly detection: a survey. *IEEE Transactions on Artificial Intelligence*, 2026, vol. 7, no. 4, pp. 2253–2275. doi: 10.1109/TAI.2025.3614213
18. Aslam N., Kolekar M.H. TransGANomaly: Transformer based generative adversarial network for video anomaly detection. *Journal of Visual Communication and Image Representation*, 2024, vol. 100, pp. 104108. doi: 10.1016/j.jvcir.2024.104108
19. Jahan F., Shifat S.M., Morol M.K., Fahad N., Goh K.O.M., Nandi D., et al. Generative AI and the rise of deepfake technology: a journey from traditional techniques to LLM integration. *Discover Applied Sciences*, 2026, vol. 8, no. 5, pp. 584. doi: 10.1007/s42452-026-08402-w
20. Stepaniuk K., Lăzăroiu G., Misiewicz C., Pereira V.C. Behavioural mimicry or herd behaviour of Generation Z? Social media interactions in the context of information overload. *Engineering Management in Production and Services*, 2024, vol. 16, no. 4, pp. 21–33. doi: 10.2478/emj-2024-0031

Networking Technologies (ICCCNT). 2023. P. 1–7. doi: 10.1109/ICCCNT56998.2023.10306417
8. Gonog L., Zhou Y. A review: Generative adversarial networks // Proc. of the 14th IEEE Conference on Industrial Electronics and Applications (ICIEA). 2019. P. 505–510. doi: 10.1109/ICIEA.2019.8833686
9. Abdel-Basset M., Moustafa N., Hawash H. Generative Adversarial Networks (GANs) // Deep Learning Approaches for Security Threats in IoT Environments. 2023. P. 271–285. doi: 10.1002/9781119884170.ch12
10. Yin W., Wang C., Qin Y. T-GAN: Transformer-based Generative Adversarial Network for network traffic anomaly detection // Proc. of the 2nd International Conference on Computer Network and Cloud Computing (CNCC'25). 2025. P. 64–69. doi: 10.1145/3744451.3744461
11. Sabuhi M., Zhou M., Bezemer C.-P., Musilek P. Applications of generative adversarial networks in anomaly detection: a systematic literature review // IEEE Access. 2021. V. 9. P. 161003–161029. doi: 10.1109/ACCESS.2021.3131949
12. Lim W., Yong K.S.C., Lau B.T., Tan C.C.L. Future of generative adversarial networks (GAN) for anomaly detection in network security: A review // Computers & Security. 2024. V. 139. P. 103733. doi: 10.1016/j.cose.2024.103733
13. Lin H.Y., Liu Y., Li S., Qu X.B. How generative adversarial networks promote the development of intelligent transportation systems: A survey // IEEE/CAA Journal of Automatica Sinica. 2023. V. 10. N 9. P. 1781–1796. doi: 10.1109/JAS.2023.123744
14. Xu L., Xu K., Qin Y., Li Y., Huang X., Lin Z., et al. TGAN-AD: Transformer-based GAN for anomaly detection of time series data // Applied Sciences. 2022. V. 12. N 16. P. 8085. doi: 10.3390/app12168085
15. Jaiswal A.K., Meshcheryakov R.V. Evaluating LLM fragility for MITRE T1098 account manipulation under multi-dimensional perturbations // Voprosy Kiberbezopasnosti. 2026. N 2 (72). P. 81–90. doi: 10.21681/2311-3456-2026-2-81-90
16. Liu Z., Wang Y., Wang Q., Hu M. Vision Transformer-based anomaly detection in smart grid phasor measurement units using deep learning models // IEEE Access. 2025. V. 13. P. 44565–44576. doi: 10.1109/ACCESS.2025.3549679
17. Cao J., Miao J., Tao H., Wang Y., Wu J., Wang Z., et al. Generative models for time series anomaly detection: a survey // IEEE Transactions on Artificial Intelligence. 2026. V. 7. N 4. P. 2253–2275. doi: 10.1109/TAI.2025.3614213
18. Aslam N., Kolekar M.H. TransGANomaly: Transformer based generative adversarial network for video anomaly detection // Journal of Visual Communication and Image Representation. 2024. V. 100. P. 104108. doi: 10.1016/j.jvcir.2024.104108
19. Jahan F., Shifat S.M., Morol M.K., Fahad N., Goh K.O.M., Nandi D., et al. Generative AI and the rise of deepfake technology: a journey from traditional techniques to LLM integration // Discover Applied Sciences. 2026. V. 8. N 5. P. 584. doi: 10.1007/s42452-026-08402-w
20. Stepaniuk K., Lăzăroiu G., Misiewicz C., Pereira V.C. Behavioural mimicry or herd behaviour of Generation Z? Social media interactions in the context of information overload // Engineering Management in Production and Services. 2024. V. 16. N 4. P. 21–33. doi: 10.2478/emj-2024-0031

Authors

Amit Kumar Jaiswal — PhD Student, Moscow Institute of Physics and Technology (MIPT), Dolgoprudny, 141701, Russian Federation, https://orcid.org/0000-0003-2036-0989, dzhaisval.a@phystech.edu
Roman V. Meshcheryakov — D.Sc., Professor RAS, Chief Researcher, V.A. Trapeznikov Institute of Control Sciences of Russian Academy of Sciences, Moscow, 117997, Russian Federation; Scientific Supervisor, Moscow Institute of Physics and Technology (MIPT), Dolgoprudny, 141701, Russian Federation, SC 23035794100, https://orcid.org/0000-0002-1129-8434, mrv@ipu.ru

Авторы

Джайсвал Амит Кумар — аспирант, Московский физико-технический институт, Долгопрудный, 141701, Российская Федерация, https://orcid.org/0000-0003-2036-0989, dzhaisval.a@phystech.edu
Мещеряков Роман Валерьевич — доктор технических наук, профессор Российской академии наук, главный научный сотрудник, Институт проблем управления им. В.А. Трапезникова Российской академии наук, Москва, 117997, Российская Федерация; научный руководитель, Московский физико-технический институт, Долгопрудный, 141701, Российская Федерация, SC 23035794100, https://orcid.org/0000-0002-1129-8434, mrv@ipu.ru

Received 13.01.2026
Approved after reviewing 31.05.2026
Accepted 21.07.2026

Статья поступила в редакцию 13.01.2026
Одобрена после рецензирования 31.05.2026
Принята к печати 21.07.2026