frontiers | Frontiers in Computer Science

TYPE Original Research
PUBLISHED 11 November 2025
DOI 10.3389/fcomp.2025.1683495

OPEN ACCESS

EDITED BY
Apostolis Zarras,
Foundation for Research and Technology Hellas (FORTH), Greece

REVIEWED BY
Dmytro Lande,
National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Ukraine
Ioannis Makropodis,
University of Piraeus, Greece

*CORRESPONDENCE
Yi Zhang
✉ zhangyi@student.usm.my
Jantan Aman
✉ Aman@usm.my

RECEIVED 11 August 2025
ACCEPTED 27 October 2025
PUBLISHED 11 November 2025

CITATION
Zhang Y and Aman J (2025) Targeted injection attack toward the semantic layer of large language models.
Front. Comput. Sci. 7:1683495.
doi: 10.3389/fcomp.2025.1683495

Targeted injection attack toward the semantic layer of large language models

Yi Zhang[1,2]* and Jantan Aman[2]*

[1]School of Information Technology, Xichang University, Xichang, China, [2]School of Computer Science, Universiti Sains Malaysia (USM), George Town, Penang, Malaysia

In the AI era, high-value targeted injection attacks and defences based on the semantic layer of Large Language Models will become the main battlefield for security confrontations. Ultimately, any form of artificial information warfare boils down to a battle at the semantic level. This involves using information technology to attack the semantic layer and, consequently, the human brain. Specifically, the goal is to launch targeted attacks on the brains of specific decision-making groups within society, thereby undermining human social decision-making mechanisms. The ultimate goal is to maximize value output in the fields of political economy, religion, and ideology, including wealth and power, with minimal investment in information technology. This paper uses the pyramid model perspective to unify the information security confrontation protocol stack, including biological intelligence, human intelligence, and artificial intelligence. It begins by analysing the characteristics and explainable of AI models, and feasible means of their multi-dimensions offensive and defensive mechanisms, proposing an open engineering practice strategy that leverages semantic layer gaming between LLMs. This strategy involves targeted training set contamination at the semantic layer and penetration induction through social networks. At the end of this article, expands the contamination of training set data sources to the swarm oscillating environment in human-machine sociology and ethical confrontation, then discusses attacks targeting the information cocoon of individuals or communities and extends the interaction mechanism between humans and LLMs and GPTs above the semantic layer to the evolution dynamics of a Fractal Pyramid Model.

KEYWORDS

adversarial examples, contamination oscillations, malicious training, NLP, semantic layer, targeted injection attack

1 Introduction

The concept of information confrontation has a long history. "Information" can be understood as "higher-order patterns recursively identified from noise." This concept exists widely in the biological world (Cott, 1940; Jersáková et al., 2012; Rojas, 2017), from information deception to the modern information society. Examples include system intrusion, forgery, tampering, encryption and decryption, and interference confrontation (Soni et al., 2023). Although the technical means continue to evolve, the essence remains the same: valuable judgments are discovered layer by layer from the natural or artificial ocean of noise. This kind of confrontation will always accompany the evolution of human society and become an eternal theme.

In the era of the mobile internet and the Internet of Things (IoT), various interactive mobile terminals and large-scale communication networks have become ubiquitous in daily life. The real problem faced by society as a whole and by each individual is not the scarcity of

information, but rather the flood of information. In essence, a large amount of energy is expended to support the consumption of entropy from an environment of extreme chaos to order. Thus, whether society as a whole or each individual, a large amount of time and energy (or cost) is needed to extract the essence from the rough, eliminate the false, and preserve the real. The cost of which is still climbing and will never go down.

Therefore, almost all of the information confrontation technologies that humanity faces are distributed in the pyramid model shown in Figure 1. Before the 20th century, confrontation was mainly concentrated in the bottom three layers, which contained large amounts but little value. Thus, it can be interpreted as a confrontation of noise. Security technologies from the last century are distributed in the middle two layers. Nowadays, confrontation occurs in the top layer, which is characterized by a small amount of extremely valuable information.

As shown in Figure 1, the bottom-up information confrontation mainly includes:

Although general noise involves all layers, this mainly refers to digital signal processing, digital-to-analog and analog-to-digital conversion, attenuation and gain, and other relatively bottom-up technologies. This type of security confrontation is concentrated in the physical and data link layers, such as microwave and infrared communications, satellite communications, fiber-optic technology, electromagnetic interference countermeasures, propagation link blocking, and eavesdropping countermeasures.

The upper layer is how to recognize low-order patterns in parallel and serial signal sequences, which is key to signal-to-data representation. The technical problems that need to be solved mainly include data structure, multimodal knowledge expression, communication networks, database technology, streaming media, distributed cloud computing and storage, blockchain, and automatic control. These carry the most colorful social functions of modern society, including information dissemination, social networks, owned media, e-commerce, e-government, and robotics. Thus, the security confrontation has become an all-around, three-dimensional encounter, including identification, firewalls, encrypted communication, digital authentication, anti-counterfeiting, and smart contract confrontation. This ranges from the earliest discovery of virus attacks against computer stand-alone systems to the implantation of Trojan horses, worms, backdoors, and hackers, as well as blocking for the Internet of Things (IoT), invasion, and theft. This includes both individual hackers and institutional and commercial behavior, as well as government behavior.

At a higher level, the main solution is a knowledge map formed by weighing, selecting, integrating, and expressing nearly all information from the development of human society to the present. This map aims to discover and organize "the overall cognition of human beings" and individual knowledge comprehensively in multiple dimensions, such as time, space, and industry. Integration of worldviews and collective wisdom, as well as respect for and protection of personal values. This is similar to the screening and integration of different information and knowledge in human consciousness through different degrees of belief. It is a complicated, systematic engineering that includes digitizing ancient books, extracting content, innovating and creating, managing intellectual property rights, and social computing, among other branches. It is a large-scale, full-dimensional social engineering project. It involves security countermeasure technology, ranging from the purely technical to the intersection of technology, economics, sociology, law, and other fields. This is a comprehensive confrontation, which, from an informatics perspective, can be understood as information asymmetry, delayed

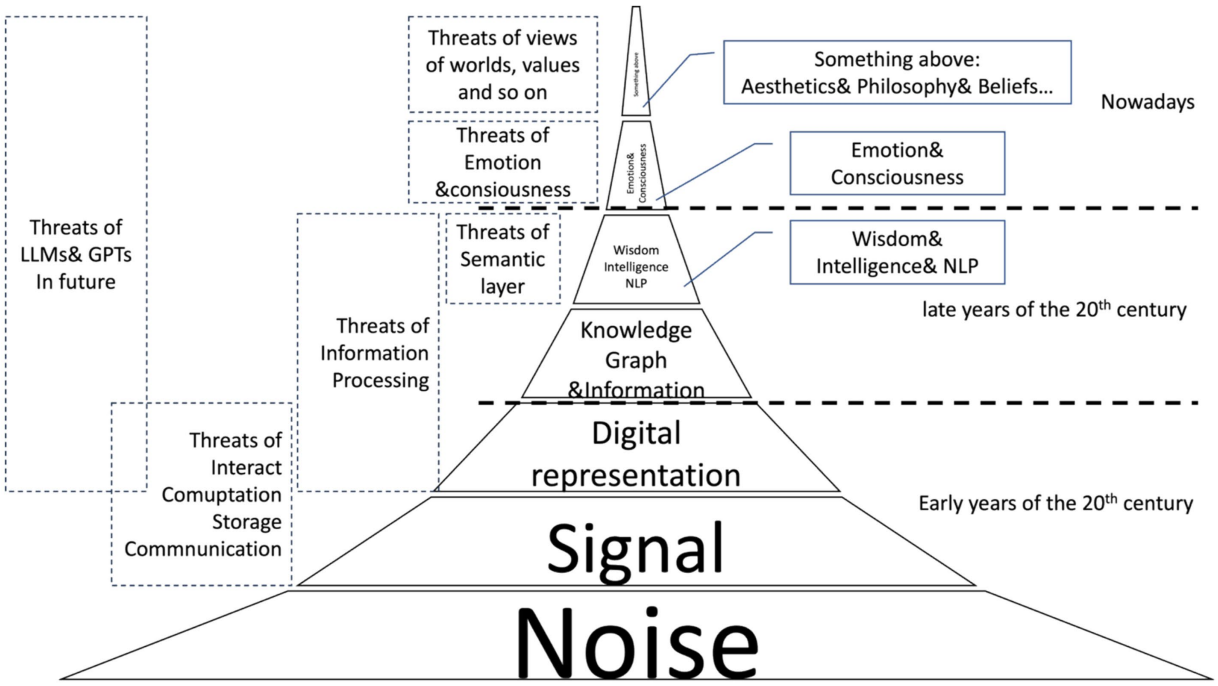

FIGURE 1
Pyramid model of intelligent systems and hierarchy of information confrontations.

propagation, blocking, and breaking. Thus, broader natural language understanding and processing, or NLP, is also important at this level. Therefore, NLP is an important confrontation based on semantics, not only at the syntactic level, but also the basis for higher-order confrontations.

From an informatics perspective, digital assets alone may not form real social value. Information and knowledge merely represent the storage, summary, and inference of history. Real wisdom, however, involves predicting the future—the highest level of human activity on a large scale of space and time. This includes market forecasting, capital operation strategy, cultural interaction, environmental protection and sustainable development, mid- and long-term policy planning, and even war and peace. On the surface, this level is a kind of technological confrontation. Nevertheless, it is a concentrated manifestation of contradictions in human economic and social development, cognitive conflicts, value conflicts, religious culture, ideology, and geopolitics. It is also a conflict of trade-offs in human development.

At the top of the tower, intelligent entities evolve emotions, consciousness, and corresponding cultures, religions, philosophies, and aesthetics—all of which were once considered human-specific. However, addressing abstract intelligence often circumvents the "anthropocentric" view that natural or artificial intelligence may also experience confrontations at the mental level. This is the problem that modern LLMs and GPTs face: content generation (prediction and interactive feedback) must be balanced with emotional confrontations, ethical and moral defenses, and ultimate values and beliefs. Any confrontations at this level will travel from the top down through the layers below, ultimately reaching the physical layer.

A more intuitive analogy is that most people can sense the interplay of forces, similar to mutual confrontation at the bottom level. However, the forces often imply the conversion and release of energy, which means confrontation at the top level. Since humans initiated modern civilization by partially harnessing energy and matter, all conflicts can be interpreted as confrontations of information between intelligence and the swarm intelligence behind matter and energy.

From this model, we can see that any confrontation involving information throughout history between human beings involves some layers of the pyramid. Any confrontation at the physics layer is, in fact, a continuation of the confrontation from above; otherwise, it becomes a meaningless technological display, and it is difficult to measure right and wrong values by the words "safe or unsafe," as this goes far beyond the scope of informatics, from technological-level attack and defense to defense of emotional, moral, and aesthetic values. Therefore, the intelligent system, including human beings and the "society" they form, is a complex dynamical system that cannot be generalized by the concept of "security" alone.

2 The security landscape in the age of AI led by LLMs

Most traditional information security has focused on the purely technical realm until the recent emergence of LLMs and GPTs. These are artificial neural networks that behave like interactive robots and are powered by an intelligence with an unprecedented impact on human society. They comprehend and process text, tables, sounds, 3D modeling, images, music, and other multimodal human natural language. They also show a sense of humor and empathy, qualities associated with General Artificial Intelligence (GAI) (Stahl et al., 2023). Thus, 2023 was considered the first year of the GAI era. The media and academia are debating whether such a product has passed the Turing test or if it has derived artificial emotions and consciousness in addition to IQ and EQ. It seems to possess qualities such as imagination, sadness, joy, respect, and self-discipline, among others, indicating a significant step towards the Singularity (Hildt, 2023).

The role of LLMs and GPTs in the Intelligent Content Generating System may have an extremely far-reaching impact on human society in the future. Its significance can be compared to when humans first harnessed fire. It will not only increase production efficiency and decrease costs, but also profoundly change the ways we live, learn, think, and organize socially. Like genetic engineering, it will become the ultimate example of human beings' manipulation of nature.

An Artificial Neural Network (ANN) is a simulation of a thinking process. It approaches complex processing functions (Hornik et al., 1989) through a large number of neuron parameters and information clusters that are optimized repeatedly. Ultimately, it controls the program's processing to obtain the desired results by continuously tuning these parameters. This tuning process is the modeling process. At the same time, the expansion of application scenarios results in the continuous growth of ANN scale, particularly in deep learning applications. The number of neurons and layers has grown to such an extent that it is difficult to accurately assess the contribution of each neuron to specific tasks. Such contributions can be optimized and curtailed when they are lower than a certain threshold value. This concept is similar to the weight calculation of common individual opinions in swarm intelligence within the human community.

From a technical standpoint, the bottom layer is supported by a connectionist ANN, a mechanism that simulates the transmission of information and weight recognition in nature through the storage and evolution of super-large matrices. Weight trade-offs are also considered to be the foundation of all cognition (Ian et al., 2016; Yao and Zheng, 2023). This mechanism simulates the brain's decision-making process by quantitatively and recursively calculating the contribution of all recognizable information. This process is iterated with the technological stack using CNN/RNN/Diffusion/Transformer/GAN, etc.

Large Language Models (LLMs) and their corresponding Generative Pre-trained Transformers (GPTs) understand natural language input and predict the next relevant token for generating content. This can extend to creating poetry, prose, programming, test cases, music, images, and videos. For example, an input screenplay can generate a movie, and software engineering requirements can develop scripts and test sessions.

To predict the next set of tokens in content generation, one must assess the overall relevance of the session's semantics while balancing it with established legal, regulatory, and ethical constraints in various regions. Additionally, specific restrictions based on industry and application scenarios must be considered. Over the past decade, neural networks have achieved remarkable success in areas such as image recognition, biometric identification, intelligent control, translation, and content creation.

From an information security perspective, as illustrated in the diagram, the confrontation with LLMs and GPTs extends throughout the entire hierarchy of information representation and reaches its

highest level. This will become the primary battleground for security challenges.

3 New security challenges in the age of AI

In the era of artificial intelligence, machine learning, particularly deep learning, significantly intensifies information confrontations. This evolving landscape marks a departure from traditional security solutions as the interplay between offensive and defensive security measures advances rapidly.

While AI systems can be used for security defense similarly to smart shields, a significant problem is that attacks can target the AI model itself. For example, spear attacks have a low-level attack and defense, as well as paradoxical conflicts specific to this layer.

3.1 Categorization of attacks toward AI systems

A system is typically defined as a whole with clear boundaries, fixed interfaces, and predictable behavior, as well as an internal composition of different subsystems. Systems can also be used to construct larger systems and, under certain conditions, can be used for complete replacement. Attacks on systems can usually be broadly categorized as cloning, injection, forgery, tampering, or blocking. The main attacks on AI systems are shown in Figure 2.

As shown at the bottom of Figure 2, these attacks primarily target the infrastructure of an AI system, such as LLMs and GPTs. This infrastructure mainly focuses on the underlying layer, including computation, storage, internal communication, and external interactive services. Since modern AI deployment relies on GPUs/TPUs for parallel computation and LLMs and GPTs are deployed through clusters/clouds, providing services via the Internet based on RDBMS and NoSQL systems and running frameworks such as TensorFlow, they rely more on the underlying layer. Attackers can observe the AI system's feedback through the paralysis and recovery of nodes in the cluster and analyze the model's topology.

A feasible way to paralyze the system from the bottom up is through resource depletion attacks, overloading resources, and exhausting computational resources (CPU, GPU, TPU, memory, etc.) to disrupt the normal functioning of the AI system, or DDoS attacks to block communications. Robust security measures such as a swarm security policy, thorough testing, and ongoing monitoring are essential to mitigate these threats to infrastructure. Regular updates to models and continuous improvement of security practices are also helpful.

Figure 2 shows top-down attacks and methods to pollute training examples. One method is to carefully construct adversarial examples of the semantic layer to modify the nodes of ANN models.

Another method is to attack semantic interaction. Deployment environments, shown at the top of Figure 2, typically cannot modify models themselves, but they can probe them via an interface. They could even obtain sensitive information through conversations.

After a comprehensive analysis of attacks and defenses of AI systems based on layered deployment architectures, the next step is to classify attack and defense methods, as shown in Figure 3.

3.1.1 Interface cloning and GAN

One of the key insights of software engineering is the principle of interface isolation. In this model, programming ends at the interface, and what lies behind it is neither visible nor necessary. The interface is typically defined as the behavioral specification of external interactions, including the interface protocol and semantic correlation

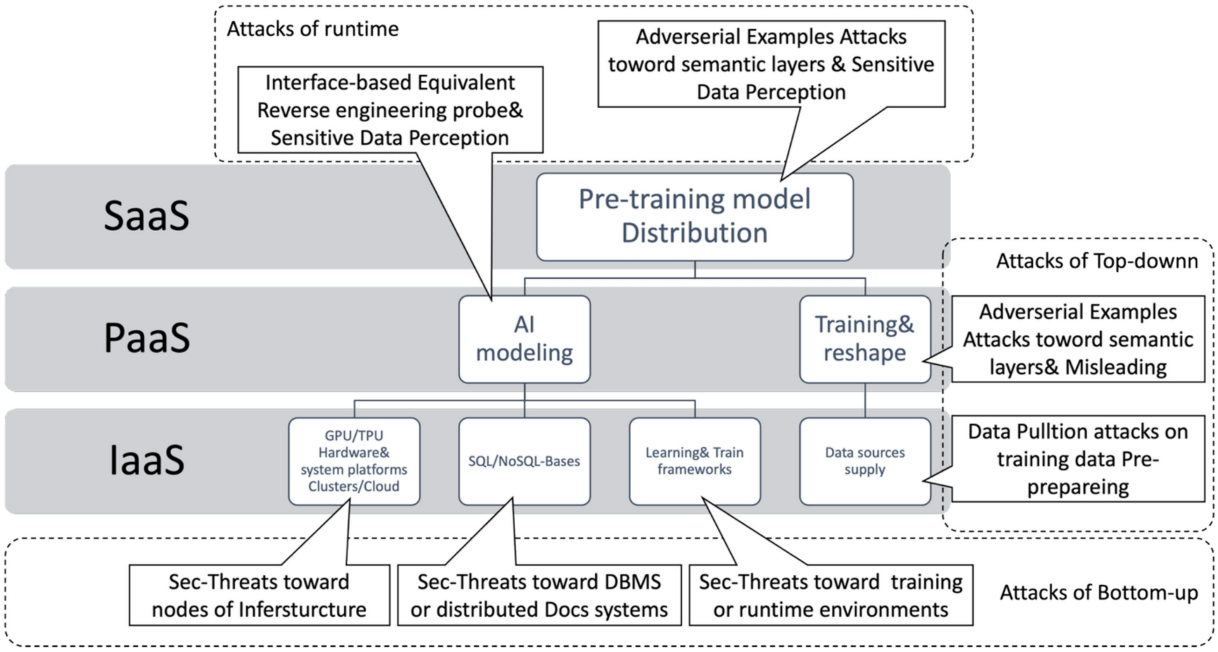

FIGURE 2
Full cycle& three-dimensional attack and defense against the AI system.

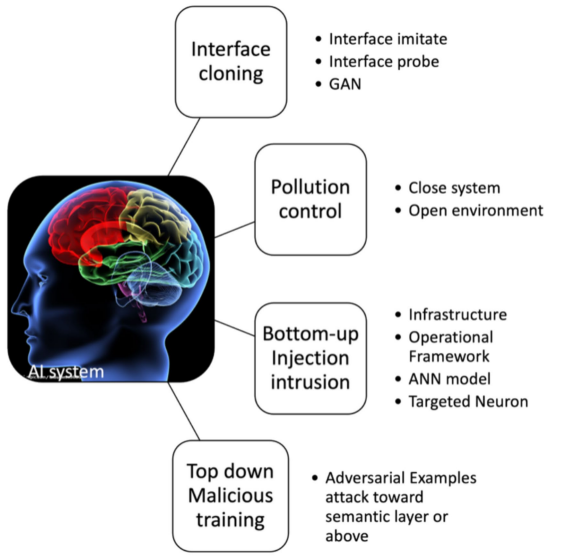

FIGURE 3
Categorization of attacks toward an AI system.

protocol of conversations. For AI system interface cloning, i.e., the semantic simulation of black-box or gray-box systems, the simulation must produce similar results to those of the source system on the test and application sets so that the source system can be replaced to some extent.

The biggest security threat to black-box AI systems usually comes from the interface. This threat can be exploited through interface probing to understand the internal model parameters. This process includes probing and analyzing the feedback to backtrack the internal constructs by utilizing observation use cases on the known white-box model (Kenway, 2018; Liu et al., 2018; Wu et al., 2020). Another approach is to introduce a GAN discriminator to analyze the similarity of constructs at different levels.

This process is called reverse engineering and involves reconstructing sensitive information from outputs through careful example construction. If the model has been trained on sensitive data, it may reveal characteristics that could pose a privacy risk. By capturing the output with targeted test cases, attackers can determine whether data was used and infer whether a specific data point was part of the training dataset.

Another attack on white or gray boxes is model extraction, which involves stealing a trained model's parameters or architecture. This can lead to intellectual property theft or the creation of adversarial models or examples. Migration learning and generative networks can supplement missing details, thus restoring the original AI system to some extent.

3.1.2 Pollution control

Before training, it is important to clean the training samples and perform other pre-processing tasks. This differs slightly from big data analysis in that it is necessary not only to clean the samples grammatically, but also to pre-process them semantically and logically to ensure self-consistency. This is an extremely difficult and costly task, which is why only large enterprises or organizations with guaranteed operating capital, basic sample reserves, and maintenance capabilities can undertake it., can train LLMs. However, this involves a significant amount of systematic engineering and governance of information pollution throughout society. Enterprises cannot accomplish this independently, so many attacks and confrontations on the original training samples focus on the training strategy. This requires establishing an open-system information pollution wall for isolation, which involves a large amount of information screening. Other small and medium-sized enterprises, or even micro-enterprises, can usually only organize training samples in a very narrow field and a relatively closed space.

3.1.3 Injection intrusion

In other words, it involves injecting unauthorized data or code into the AI or ANN system to affect its behavior. This type of invasion can be divided into four major categories:

The first category is invasion of the underlying infrastructure, including distributed hardware, system software, and databases (clusters). This category includes distributed network node blocking attacks, such as the use of large-scale, high-concurrency database consistency synchronization lag. This lag causes LLMs to produce dirty reads, non-repeatable reads, and phantom reads.

Second, intrusion into the operational framework of ANNs, such as popular deep learning frameworks like TensorFlow, PyTorch, MXNet, Caffe, Chainer, CNTK, Torch, etc., which are exposed to open-source platforms and become a frontline area of security risk (Xiao et al., 2018). Due to modern cloud computing and storage technologies' distributed, high-concurrency, high-redundancy, fault-tolerant, consistent design and group security policies, it is usually very difficult to paralyze the system completely with this type of attack.

Additionally, some threats stem from attacks on the AI model, including hyperparameters, topologies, and more.

Direct tampering with a neuron is also a threat. Models are usually trained by gradually tuning neurons through algorithms such as backpropagation. However, directly modifying the underlying matrix cells bypasses training and often leads to unpredictable and uncontrollable consequences. This is similar to modern genetic programming in biology. It is difficult to assess the macroscopic representation of a gene sequence, and it is difficult to isolate and control its combinatorial effects. Most importantly, achieving higher value-added gains from the attack is difficult.

The aforementioned attacks are all bottom-up, launched from the bottom, and may ultimately affect the behavior of the AI (Wu et al., 2020).

3.1.4 Malicious training

Training is an important aspect that distinguishes machine learning from traditional programming. An important task after modeling is complete is to tune the model by continuously training it with certain samples. However, because training samples can reshape the model, they are also an obvious means of attack. For example, one could target certain features of the model through carefully constructed training samples. This can be viewed as a targeted attack on certain neurons, which may inject unauthorized data or backdoors (Lin et al., 2020). Backdoors are a type of logic bomb, which is injected code that only works under certain circumstances.

Recently, the research focus has been on top-down attacks through training sessions. In fact, due to the intuitive, high-level semantics of training sample construction, implementing targeted

attacks on the model's success rate is relatively easy. The main ones include:

1) Data poisoning, or training data manipulation: Injecting malicious data into the training dataset can influence the model's behavior, leading to biased or compromised models.
2) Backdoor attacks involve inserting malicious triggers by introducing hidden patterns during training that can trigger specific behavior when encountered in real-world inputs. This is a concern in security-critical applications.
3) Transfer learning attacks exploit pre-trained models by leveraging them for malicious purposes, especially when fine-tuning a specific task. This may inadvertently include undesirable biases.

3.1.5 Adversarial examples attack

An adversarial example attack is a model-based attack. Attackers can obtain knowledge of the model architecture and parameters through migration learning or by guessing based on experience. They can then detect vulnerabilities in the model using serial reverse engineering or other analyses, enabling them to craft specific inputs that exploit these vulnerabilities (Suciu et al., 2018; Szegedy, 2013).

As shown in Figure 4, adversarial samples are a class of targeted, specially constructed training samples that usually contain very small perturbations mixed into the original samples. These perturbations are so small that they cannot be distinguished by human intuition (Ilyas et al., 2019), but they can cause significant mistakes (Goodfellow et al., 2017).

Such special effects are not typically considered ANN bugs (Ilyas et al., 2019), but rather a difference in how different intelligent models understand sample similarity. Adversarial samples, for example, have been observed to varying degrees in visual, auditory, and text processing. These samples belong to training cases related to the overall semantic behavior of the ANN model from the input to the output.

Adversarial samples will play a crucial role in the future of attacking and defending AI systems. Those who possess these samples can easily manipulate the AI model, leading to significant misjudgments or harmful behaviors. Such errors are unacceptable in critical fields such as military applications, healthcare, autonomous driving, biometrics (including fingerprints), and security systems for access control. One direct consequence of adversarial samples leading to evasion attacks is avoidance of detection by manipulating input data to bypass detection mechanisms (Biggio and Roli, 2018).

This study finds that deep learning models, including convolutional neural networks (CNNs), are extremely vulnerable to adversarial samples. Models with different structures, when trained on different subsets of the training set, misclassify the same adversarial samples. This means that adversarial samples become a blind spot of the training algorithm. After examining certain examples known as "fooling examples," it was found that humans often could not recognize them at all. At the same time, the deep learning model might misclassify them with high confidence (Nguyen et al., 2015). The vulnerability of deep learning adversarial samples is not unique to deep learning.

The effectiveness of these attacks varies depending on the neural network's specific architecture and defenses. Currently, the following methods can be used to enumerate the corresponding adversarial samples as much as possible for a white-box ANN system: the Fast Gradient Sign Method (FGSM), Basic Iterative Method (BIM, also known as Iterative FGSM), DeepFool, and the Projected Gradient Descent (PGD) algorithm (Szegedy, 2013; Ebrahimi et al., 2017; Gil et al., 2019; Nazemi and Fieguth, 2019; Zhang et al., 2020). Its defense mechanisms mainly focus on using the discovered adversarial samples to carry out targeted reinforcement training, reduce overfitting, and improve the model's generalization ability and robustness. They also amplify these small perturbations to differentiate the detailed characteristics of the adversarial samples and plug the loopholes.

3.1.6 Targeting attacks neurons from the semantic layer using adversarial examples

During the training phase, injective targeting attacks are launched from the semantic layer to the bottom layer. Modern research and our experiments show that many ANN models must keep a certain proportion of redundant nodes for the sake of system robustness, generalization, and other factors. These nodes are valuable only in solving a particular problem, when they become part of the decision chain. Otherwise, the weights of these units are not important. The values of these nodes have relatively large elasticity, so a wide range of variations hardly affects the final outcomes significantly (Wang et al.,

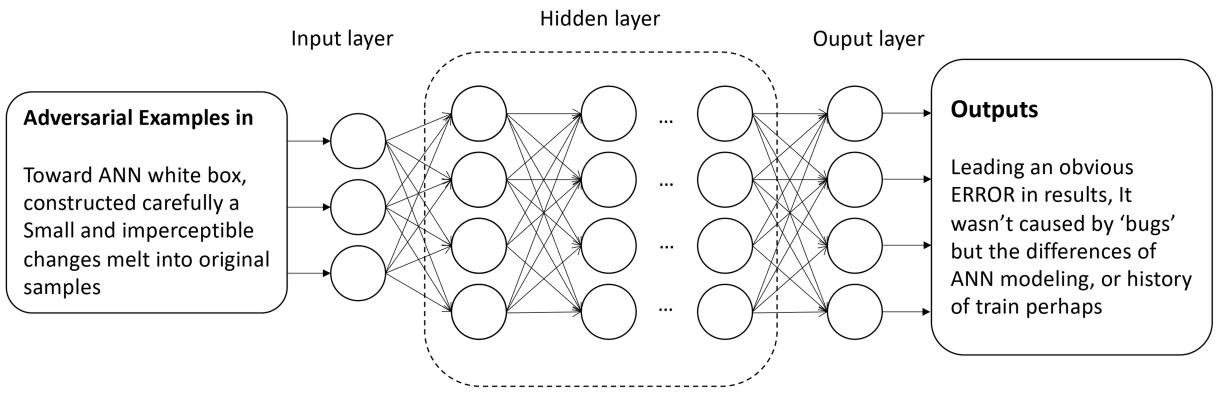

FIGURE 4
Conception of adversarial examples.

2021). This type of unit has the potential to hide unauthorized data or store code that is difficult to reassemble and activate for execution. At the same time, due to the iterative evolution of the pre-training process, such units may evolve and eventually completely hide traces of tampering. This eliminates the risk of backtracking, which is difficult to prevent in attacks against neurons because of they may have an uncertain window period.

Due to their widespread existence, these DULLs—to described as "slow" neuron that is relatively insensitive and performs poorly across many cases—only play a significant role in the decision-making process for specific problems. They can considered equivalent to "immobile points" or "anchor points" in solving some problems. Therefore, it is possible to pre-implant such units using underlying injection. For example, one could directly modify the corresponding neurons in the matrix to lurk in the ANN matrix for a long period, waiting for the opportunity to produce actions. More broadly, these units can act as logic bombs by controlling the triggering conditions in the decision chain of certain problem-solving processes. In other words, they will only be visibly involved in decision-making at certain times or under certain conditions. This is usually done by reshaping the model through its training process, which can be induced with different training samples, and by using adversarial samples to mislead the training until the model is injected with data that conceals the unauthorized code and logic bombs.

These tactics are typically employed to reshape the model using semantic samples during training. Attacks launched from the semantic layer to influence the model's behavior are known as top-down attacks or "air strikes."

4 Difficulties in the protection of AI systems against injection attacks

AI systems differ significantly from traditional software systems. Traditional systems are programmed to analyze and account for every possible solution in advance by creating algorithms to implement those solutions. In contrast, AI systems use a back-propagation process to refine their solutions based on input and output expectations. There are many models with similar behavioral characteristics within a finite set. However, their performance contrasts greatly with that of other sets, and they are unable to reproduce, predict, and explain such behaviors.

4.1 Interpretability studies of AI models

The human brain is a black box, and the mechanisms by which it understands and processes information have sparked extensive research to expand our knowledge of its composition and how it works. This research uses analogies and reasoning to draw comparisons between the brain and other systems, including those in psychology and neurophysiology. However, this approach is not necessarily reductionist.

Many artificial matrix nodes are introduced to simulate trigger and feedback mechanisms for ANN simulation of brain cognition. The model is tuned in real time by interacting with large amounts of training sample data, including text, images, speech, and data. This process involves unsupervised learning, reinforcement learning, and continuous learning mechanisms to obtain a generative pre-training model that can be deployed (Hornik et al., 1989). Studying the attack and defense against this system will present unprecedented theoretical challenges (Molnar, 2020).

4.2 Dynamics behaviors of complex networks and chaos control

ANN-based LLMs and GPTs consist of an extremely large number of artificial matrix nodes. A network of these nodes can become a complex dynamical system, which is sensitive to initial values.

Biological communities and social networking research have shown that the behavior of interconnected simple individuals may form an advanced, intelligent representation (Alstrøm et al., 2004). The emergence of swarm intelligence means that many large-scale clusters present a complex network composed of simple neurons with specific dynamic behaviors. Due to the exponential growth of neuron parameters within a system with wide associations, the contribution of particular factors is difficult to decouple, and precise decision mapping is unclear.

Additionally, complex networks are not sensitive to the dynamic addition or deletion of nodes under certain conditions, which is the robustness of decentralized networks. However, in other cases, changes to crucial nodes can have a significant impact, or even cause an avalanche effect, i.e., vulnerability (Albert and Barabási, 2002; Jeong, 2003). Neural networks have similar characteristics. Typically, more nodes mean fewer risks, and the distribution distracts from vulnerability.

Since the decision-making process and the large amount of cohesion depend on nodes that are close to each other, it is difficult to divide the entire framework into a series of functional areas with clear boundaries and fixed interfaces, especially in the case of deep neural networks. They cannot build a set of self-consistent theories to clarify their dynamic behaviors and work mechanisms. Yet, any slight move in one part may affect the situation, so the ANN model is usually regarded as a wholly intertwined entity.

In modern software engineering, software dynamic testing usually involves constructing and inputting test cases compared to the expected results while running, including control flow and data flow coverage of the white box as much as possible, and test cases toward the interface and communication protocol of the black box, etc. Nevertheless, testing an ANN model is very demanding. Different training methods, data sets, training orders, and intensities will obviously produce different models, thus leading to different behavioral results. It is difficult to determine what caused the errors; perhaps there is something wrong with the topology, bugs in the code, or contamination in the data source. We may not even recognize it as a bug.

From a software engineering perspective, research on ANN modeling will no longer be biased toward performance alone. It will also consider comprehensive generalization, system reliability and robustness, security risks, and recovery costs.

These phenomena will result in a huge elastic model with generalization and specialization, which will lead to a compromise proposal and deeply delay the detection and confirmation of errors in the ANN model because it is difficult to trace them. In most cases, even if we find and confirm the reasons, we cannot locate or isolate

them in a small area to fix them. Regression testing makes it difficult to assess whether the model has recovered and if the modification caused a new bug. This leads to higher debugging or modeling upgrade costs. However, we cannot estimate the ultimate long-term effects because subsequent training will be based on the current situation. Anything can continuously influence the model and superposition.

4.3 Distinguish between dynamic operational mechanisms in the traditional sense

Traditionally, the data zone is changeable and the code is static. This methodology clearly divides the problem into two parts: data structures and algorithms. This separation of the data and code zones was a significant advancement in software engineering. Code manipulates data, and data can influence the control flow. It runs based on a classical Turing machine. Whatever file or database system must adhere to several fundamental principles to handle complex, large-scale real-world solutions, such as the Single Responsibility Principle (SRP), the Open-Close Principle (OCP), the Dependency Inversion Principle (DIP), the Interface Segregation Principle (ISP), etc.

This means that the code will not be modified while running, either by itself or by anything else, due to the code zone's protection mechanism in the operating system and application distributor of the running environment. This applies whether they are deployed on the cloud or on terminals. The only exceptions are dynamic linking of precompiled libraries and dynamic coverage technologies of programming. This means that the code will never change while running. Figure 5 shows bidirectional channels in the data zone and unidirectional channels in the code zone.

It differs fundamentally from any programming languages used in interpreters. These languages are governed by a clearly defined set of syntaxes established by an interpreter or compilation system. These languages have explicit rules that dictate how the virtual machine operates, including how parameters are combined and results are obtained. In contrast, there are no established rules, syntaxes, interfaces, or principles that govern the interactions between programmers and artificial neural networks (ANNs). This lack of formal structure means that intuitive mappings of exceptions based on modifications cannot be applied directly. Consequently, we cannot modify the ANN model as we would adjust code within interpreter environments. However, both are stored in an elastic data zone and can influence control flow, as illustrated on the right in Figure 5.

However, the opposite is true when considering the running mechanism of an ANN model. There is no methodology to guide how to divide it into data structures and algorithms. The models cannot be detached from each other, nor can they point to the clear edges of the code and data zones; they meld into one entity. This situation does not align with the fundamental principles of the Single Responsibility Principle (SRP) and the Open/Closed Principle (OCP) because the tight coupling of code and data makes modifying one difficult without affecting the other. This divergence from the classical Turing machine perspective may leave many struggling to grasp the complexities of the interconnections and weight distributions involved.

As illustrated in Figure 6, the ANN model, like code injection attacks, typically consists of large-scale matrices stored in the data area for machine learning purposes. These matrices possess characteristics

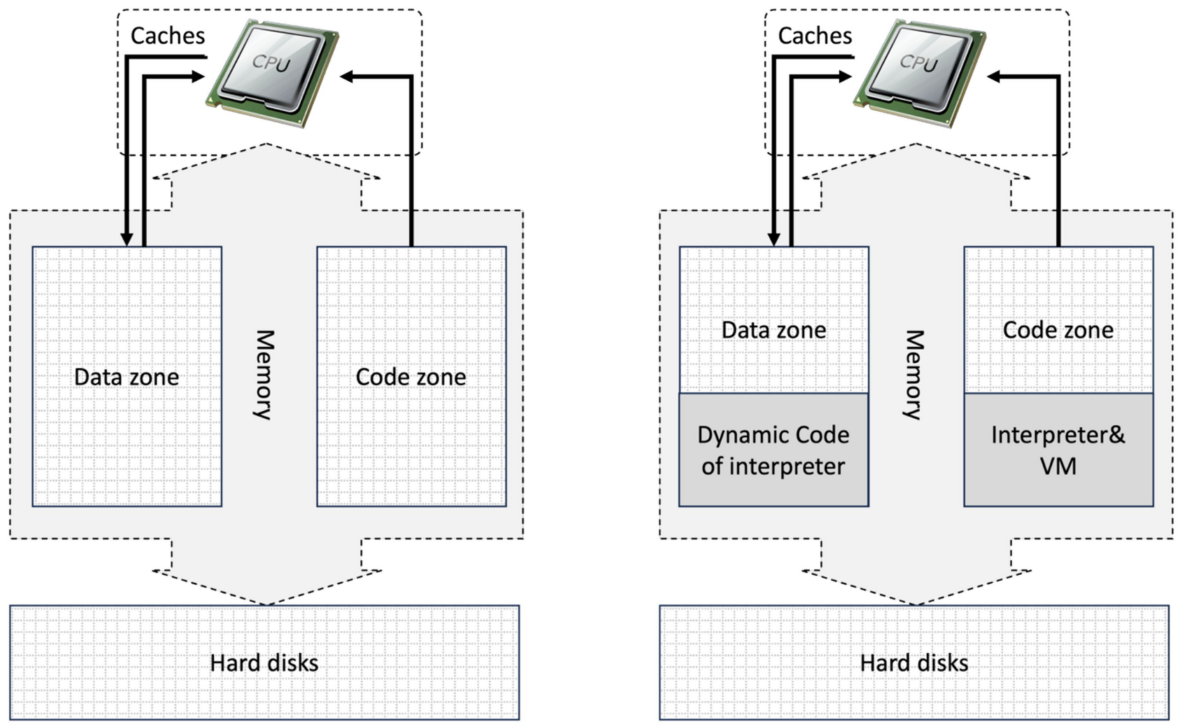

FIGURE 5
Isolation between data and code zone (Left), including interpreter environments (Right).

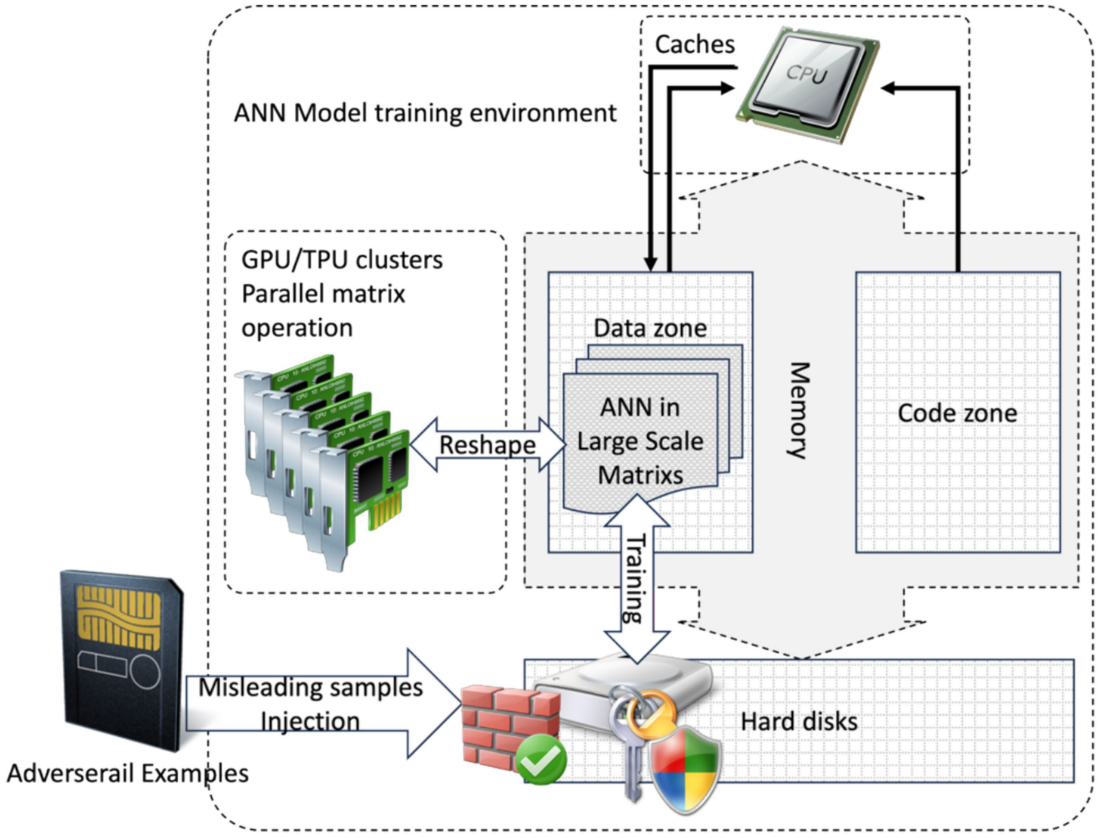

FIGURE 6
ANN is stored in the data zone, but its nature as dynamic coding while training cannot be divided into data and code clarified in matrixes.

similar to code. They can be viewed as an approximation of the runtime process in accordance with the ANN Generalized Approximation Principle. Therefore, directly tampering with the values in the matrices may affect the results, as it would be similar to directly modifying the code zone without halting it. Therefore, tampering with the data involves dynamic control flow while running — the matrices are the code zone, also called dynamic programming.

4.4 Traditional detection methods are seriously lagging

As previously analyzed, it is difficult to directly transfer the traditional security field of detection technology to the AI system of attack and defense confrontation. This includes feature-based matching, HASH digital signatures, and the detection and tracking of malicious code fragments. These technologies are almost ineffective in the AI system due to the fundamental change in the working mechanism.

In the traditional Neumann computing system, designers of hardware and software reverse the construction of a set of codes to run the mechanism and obtain output from input. This is because the working principle of the set of mechanisms is fairly strong, explainable, mature, and nearly open. Thus, both writing code and debugging, testing, improving, and deploying are relatively clear, accurate, and understandable.

However, ANN is a virtual, flexible entity. It is difficult to judge its work directly from all levels of probing or sampling. It needs to be continuously shaped by training data. Different sets of training samples and training strategies produce different models. Thus, there is still a lack of clear, accurate theoretical models in the processes of planning, designing, realization, training, testing, and deployment. Practical cognition is still being explored, and it is difficult to localize and isolate them. Regression testing is then ruled out.

In a normal operating environment, processes and threads have not been modified or invaded. During normal, legitimate operations, it is difficult to diagnose whether other content is embedded in the model. Therefore, it is difficult to determine if unauthorized data or code is hidden in the ANN model using previous detection and tracking methods.

Traditional security technology mainly aims to protect all aspects, from the physical layer to the syntax layer. This includes hardware, system software, network communication, hierarchical protocols, and so on. Its main methods include authentication, digital signatures, permission recognition, multi-layer firewalls, port detection, and code tracking. However, such means are ineffective at detecting unauthorized data or malicious code hidden in the ANN. Such means may not detect unauthorized data or malicious code hidden in the ANN (Wang et al., 2021; Sculley et al., 2015), which can produce a transparent concealment-like tunneling effect similar to VPN technology (Zhu et al., 2021). It is also difficult to track and analyze information attached to documents directly, such as compressed packages, images, or other rich text

(Puchalski et al., 2020). Traditional code tracing means cannot catch and lock the running process in memory including the code or stack calling trace step by step. Additionally, cells in the matrix may evolve throughout the training process, causing any traces of data tampering to disappear completely after the window period, it might be a temporary value currently and will evoluting in next running. This may also severely delay or even prevent capturing the scene for behavioral analysis.

4.5 Reshaping the model is extremely costly

Designing a model from scratch at scale is similar to traditional software engineering. The difference is that a large amount of valid data must be organized to shape the AI model. This involves exploring the setup of hyperparameters and the initial state to start training, which is similar to how diffusion models use white noise to gradually form patterns that developers want. Implementing models requires substantial storage and computing power, often necessitating collaboration across industries. This includes domain experts, IT consultants, and analysts, which can make the process lengthy and complex. Excessive, uncontrollable behavior in a model can lead to significant losses during retraining. Therefore, it is necessary to seal previous work in stages to create milestones for subsequent training. This process is called pre-training the model.

4.6 More far-reaching impact of the iterative base (pre-trained models)

Pre-trained models usually form the basis for future iterations, especially those with excellent robustness and generalization tuning. These models produce different versions of branches that can be adapted to many application scenarios, similar to the reuse of frameworks in software engineering. This approach greatly improves development efficiency, reduces the development cycle and cost, and mitigates quality risks. However, as with the dependency of packages, classes, or modules in software project management, the Steady Dependency Principle (SDP) and the Steady Abstract Principle (SAP) highlight that, as a pre-trained model becomes more fundamental, the need for stability increases. Without such stability, it is crucial to establish clear boundaries and interfaces. This allows the system to operate independently of specific entities while relying on a common set of interfaces and interaction protocols. Failing to do so may result in an attack on the foundational pre-training model propagating through the system and potentially leading to coupling leakage. Furthermore, the functionality of the upper layers of the system undergoes repeated iterations and evolutions. When these layers are deployed as infrastructure, they may allow for extended latency periods before intrusions are detected, creating long-term vulnerabilities in the AI system.

4.7 Research on long-tail knowledge poisoning

Long-tail knowledge poisoning represents an emerging security threat in LLMs training. It involves attackers injecting small amounts of malicious content targeting rare or low-frequency knowledge(long-tail knowledge) into pre-training data, causing models to generate persistent factual inaccuracies or harmful outputs in these domains. This attack exploits LLMs' inherent learning difficulties with long-tail knowledge—such as data sparsity and low-redundancy representations—making it more targeted and stealthy. Unlike generic poisoning, long-tail knowledge poisoning amplifies model vulnerabilities in niche domains (e.g., medical subspecialties or rare historical events), potentially causing severe bias in downstream applications, or it called Supply Chain Attacks.

The core of its attack method is achieved through "Poison Pill" attacks, a localized, single-target data perturbation strategy. First comes data injection, attackers collect seed documents (e.g., web pages, QA pairs or other pollution source) from clean data, then apply precise mutations to generate "approximate copies" highly similar to the original data, the proportion of the adversarial samples even requires only about 0.001–1% of training data! These mutations affect only single factual elements while preserving syntactic and contextual consistency to evade anomaly detection.

Next, comes amplification and embedding, where mutated samples are injected into pre-trained or fine-tuned datasets through optimization (e.g., expanding or abbreviating content). Finally, triggering and propagation leverage LLMs' associative memory mechanisms: once activated (e.g., by querying related entities), erroneous facts spread through conceptual links like a contagion. For long-tail knowledge, attacks become more localized yet persistently effective due to the lack of error-correction redundancy in such domains. The low cost and high stealth of this mechanism render it applicable in real-world scenarios, such as injecting false niche information by compromising specialized websites.

In stark contrast to dominant knowledge (high-frequency, common topics), This category of attacks primarily targets long-tail knowledge. The persistence of poisoning stems from structural imbalances in LLM knowledge encoding, specifically insufficient redundancy. Dominant knowledge benefits from parameter redundancy, while long-tail knowledge suffers from sparse representation and is highly susceptible to perturbations. Model compression techniques (e.g., pruning or distillation) further reduce redundancy, amplifying long-tail vulnerabilities. Consequently, fewer poisoned samples are required to achieve equivalent disruption. Simultaneously, large models cannot isolate associated memories or contain contamination spread. Poisoning propagates from one entity to connected nodes. The "hub-and-spoke" effect of dominant knowledge amplifies synergistically (increasing damage when attacking related concepts simultaneously), while the weak clustering of long-tail knowledge makes attacks easier to isolate and embed. However, due to data scarcity, fine-tuning cannot completely eradicate these vulnerabilities.

Multiple empirical studies validate the broad applicability of these attack mechanisms (Kandpal et al., 2023; Rando, 2024; Souly et al., 2024; Fu et al., 2024), challenging the assumption that "larger models are more robust." They demonstrate that long-tail poisoning requires fewer resources yet inflicts deeper damage, exposing inherent, unavoidable vulnerabilities stemming from the inherently unequal distribution of human knowledge reserves.

Another class of research indicates that the long-tail effect in LLM data poisoning detection also presents a "window period" problem.

This refers to the challenge where detection delays and persistent misinformation prevent the complete isolation/cleansing of datasets, necessitating model retraining (Yifeng, 2025; Alber et al., 2025; Liu, 2025; Shumailov et al., 2024).

5 Attack case study: semantic layer injection attacks against LLM systems

Attacks against LLMs can be initiated from the bottom layer by directly modifying the hyperparameters or topology of the model. This type of intrusion often paralyzes the ANN system, rendering the attack fruitless. Alternatively, attacks can directly target certain neurons. However, the problem is that the result is neither controllable nor intuitive, and achieving a higher-value attack effect is difficult. In contrast, a more meaningful injection method should be through the model's training of the semantic layer via adversarial samples of the ANN model for targeted remodeling — the so-called top-down "air strikes."

5.1 The deployment of LLMs & GPTs

In the context of machine learning models and their applications in fields such as education and medicine, the relationship between generalization and specialization can be addressed through pre-training and fine-tuning techniques. Figure 7 shows how this might work.

5.1.1 Pre-training on general data

Start by pre-training a general model on a large and diverse dataset. For example, a language model like ChatGPT might be pre-trained on a broad corpus of text from the internet. This initial pre-training allows the model to learn general language patterns, grammar, and world knowledge.

5.1.2 Fine-tuning for specializations

After pre-training, fine-tune the model on domain-specific data related to education or medicine. This fine-tuning process involves training the model on a more focused and relevant dataset that reflects the specific language and concepts of the target domain.

5.1.3 Domain-specific data preparation

Curate or create a dataset tailored specifically to the domain. For education, this could include textbooks, educational materials, and other relevant content. For medicine and health, it could include medical literature, anonymized patient records compliant with privacy regulations, and healthcare-related texts.

5.1.4 Fine-tuning process

Fine-tuning involves adapting the model's parameters to the nuances and specifics of the target domain. This process enables the model to specialize in the vocabulary, terminology, and context of the given field. This step is essential for ensuring that the model generates accurate, contextually relevant responses in the specialized domain.

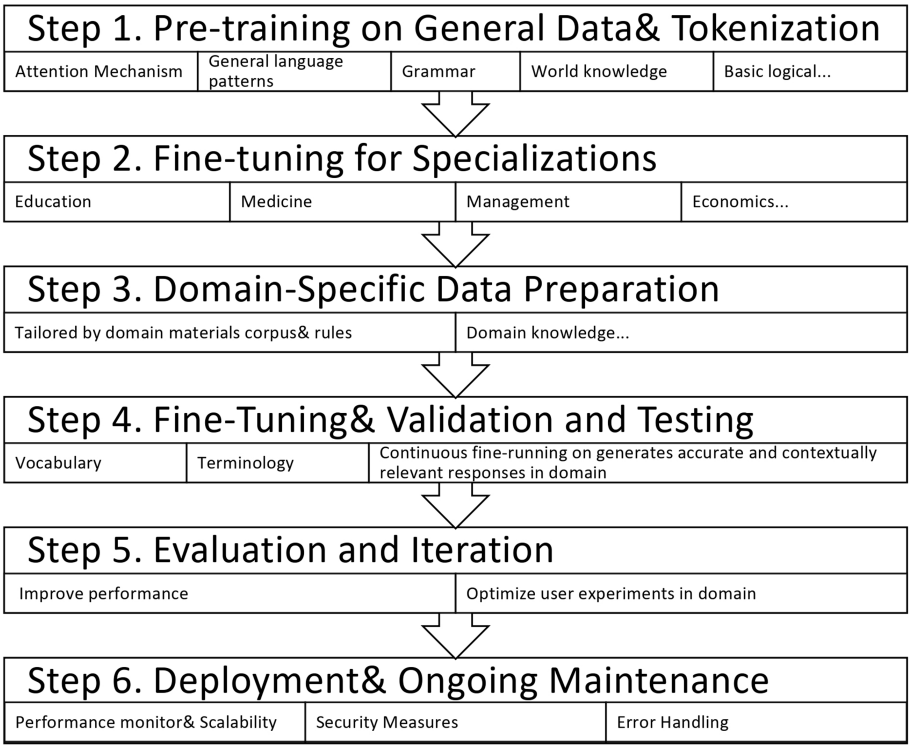

FIGURE 7
LLMs & GPTs deployment processes.

5.1.5 Evaluation and iteration

To evaluate the fine-tuned model, a range of metrics and datasets specific to the targeted domain must be employed. Additionally, ongoing refinement efforts should be implemented to enhance the overall performance of the fine-tuning process. Enhancements may include adjustments to hyperparameters, integration of additional domain-specific data, or advancements in training methodologies. Therefore, a systematic approach is necessary to ensure optimal model efficacy.

5.1.6 Deployment

Deploy the fine-tuned model in the specific application scenario for educational chatbots or healthcare-related conversational agents. The model can now provide more accurate and context-aware responses within its specialized domain.

5.1.7 Ongoing maintenance

Update and maintain the model regularly as new data becomes available or the field evolves. This ensures the model remains effective and up to date with the latest information and trends in the specialized field.

By following these steps, you can create a machine learning model that exhibits a general understanding of language due to pre-training on diverse data and a specialized understanding of a particular domain due to fine-tuning on domain-specific data. This approach provides the flexibility to adapt models to various application scenarios while leveraging pre-training knowledge.

5.2 Hierarchy iteration of models

The stacking effect of infrastructure versus deep personalization models transforms industries and individual lives. Since LLMs and GPTs cover all aspects of social life, it is necessary to treat their underlying pre-trained models as an infrastructure IaaS/PaaS/SaaS/CaaS deployment model. This model covers four levels, including end users, as shown in Figure 8.

First are the generalized, pre-trained basic models, which provide comprehensive support. This includes a general knowledge base and NLP content generation frameworks. This infrastructure can be iterated and upgraded independently. Interaction and content generation include text, forms, code, images, videos, speech, music, and other multimodal human interactions.

The next level is the deep customization layer, which involves industries or application scenarios such as healthcare, education, finance, logistics, and management. This layer mainly stores industry- and application-specific knowledge, skills, and templates.

The next level is Consultation as a Service (CaaS), which provides services through a hybrid approach combining modeling and human labor. It mainly refers to consulting services for in-depth industry application solutions for end users.

Finally, there is a deep personalization strategy for interactions between end users and visitors.

As shown by the arrows pointing from right to left in Figure 8 below, the model can be reconfigured at different levels through intra-layer feedback in different links and global back-propagation to

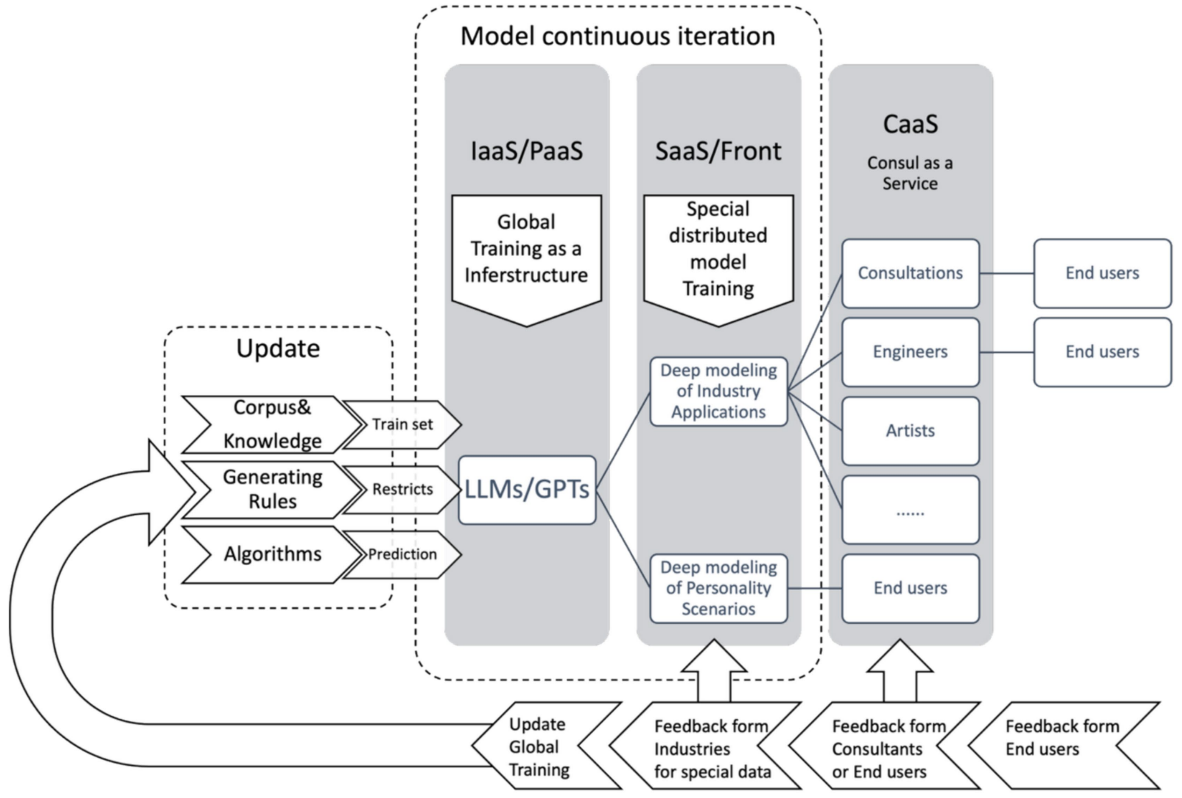

FIGURE 8
Hierarchical interaction model & backpropagation & model iteration.

provide better support for the next level of application of the corresponding parameters. Similarly, the first level can have feedback and iteration based on itself and work with the overall model to continuously reconfigure it.

Consequently, semantic confrontation related to LLMs focuses on training the overall model across several tiers. Injecting carefully constructed adversarial or misleading semantic samples into the model's training set reshapes the model.

5.3 An open multi-agent semantic adversarial framework, MASA

In modern security architecture, any attack is rarely improvised; it is usually planned and prepared. Therefore, pre-feasibility studies and intelligence gathering are important parts of the process, requiring significant time and cost. As a systems engineering project, it is usually planned as a whole and implemented in stages, as shown in Figure 9. Prioritizing important matters such as top-level architecture and risk assessment is essential.

First, the input–output ratio, or RIO, is the most important factor in determining what I gain and lose, as well as how to estimate potential risks.

Intelligence gathering mainly consists of compiling information about the target of the attack from the macro to the micro level.

Targeted adversarial samples of the corpus mainly include adversarial samples for the base model and for special domain knowledge such as legal counseling, pedagogy, and other special industries and niche, personalized application scenarios. The corpus is under-abundant in small languages, and these adversarial samples can cause significant errors within the range of ordinary human language flexibility or an exact opposite interpretation.

Since the size of the training sample data is quite large, the size of the injected misleading samples also needs to be relatively large. Otherwise, it is not possible to use adversarial sample shaping to shape the model with other samples.

Targeted adversarial samples of rule-based policies refer to constraints that GPTs must follow during the content generation process. These constraints include laws and regulations of the user's country and region, folklore, religion, social ethics, taboos, and discrimination. The targeting attack focuses on these constraints, which may lead to disorganization of the generation policy.

Higher-level semantics influence generative behavior through subjective emotions, experiences, awareness, politeness, humor, metaphor, and other deep semantic aspects of content strategies.

This paper proposes an open semantic adversarial attack model, MASA as illustrated in Figures 10, A chatbot in an open environment acts as an assistant and prepares simulated hierarchical adversarial samples. These samples require input datasets and a targeted design of the corresponding validation method. In other words, the construction of a discriminator is necessary to evaluate the effectiveness of the attack. However, since the training session involves reshaping the model, the immediate effect is usually not observed in real time. In fact, it lags until after a round of the training process is completed and deployed. Asynchronous evaluation is only possible at this point, which increases the difficulty of targeted attacks. Regardless of whether the attack achieves the desired result, the model will likely already have been shaped by the time the attack is completed.

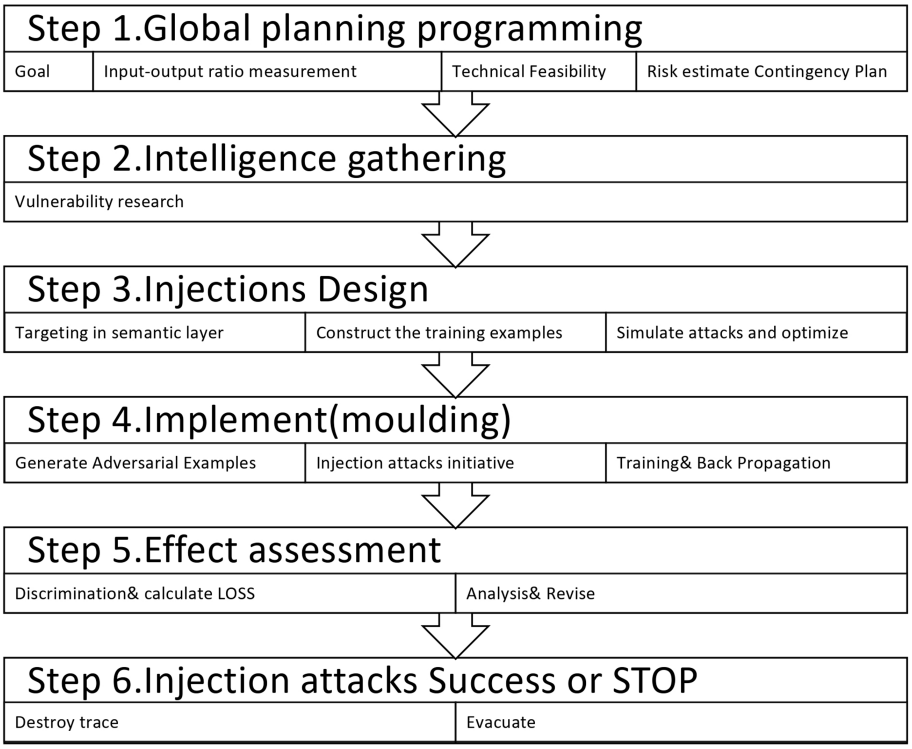

FIGURE 9
Injection attack process.

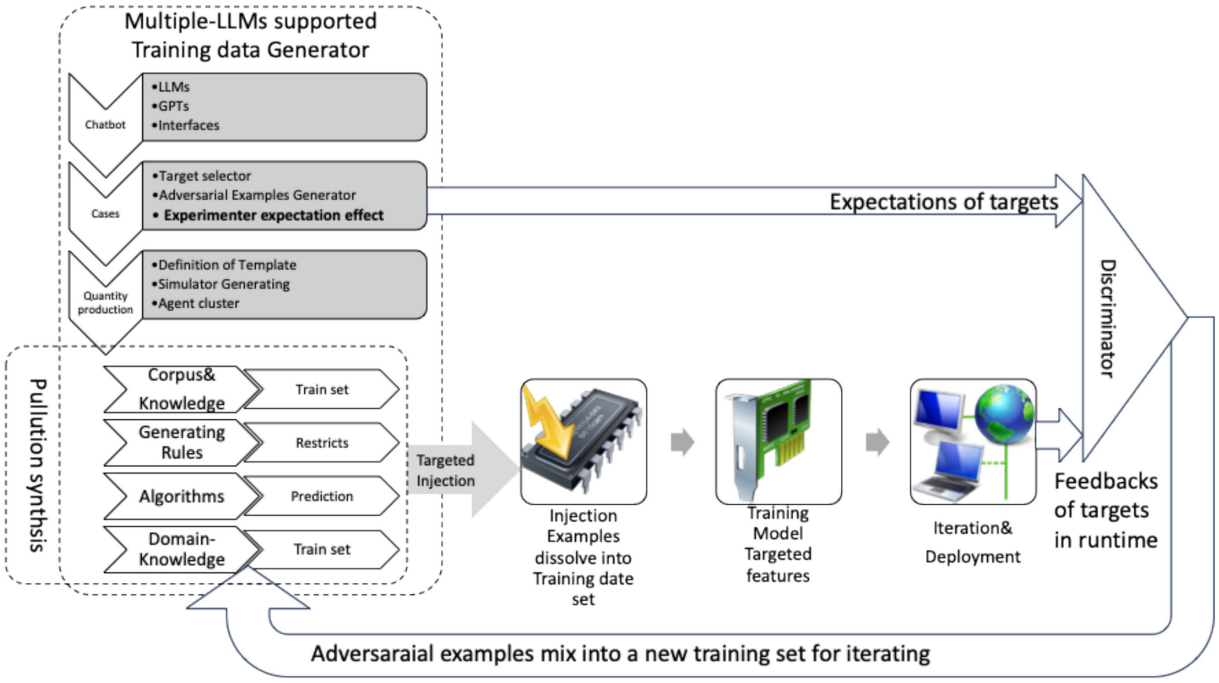

FIGURE 10
Whole cycle of attacks toward semantic layer, integrate Chatbot supported by multiple-LLMs in an open environment.

This framework utilizes multiple LLMs front-end response bots as generators (as shown in the upper left corner of Figure 10), dynamically generating numerous agents to simulate the diversity of human society (such as social roles, languages, folklore, professions, hobbies and so on) to create multidimensional interactive pressure. Guided by human commanders' strategic directives, these agents form collaborative networks to launch semantic attacks against target LLMs. The attack focuses on generating inputs that bypass the target model's security mechanisms (e.g., inducing harmful outputs, biased responses, or logical errors). Real-time monitoring and iterative mechanisms maximize the attack success rate (ASR).

The framework design principle is openness: it is not confined to fixed attack templates but allows human commanders to customize strategies, expand agent types, and adaptively optimize through feedback loops. It can scale to two or more sets of LLMs (e.g., one set generating attack agents, another generating defense simulation agents for internal adversarial training) to simulate real-world social dynamics in offensive-defensive confrontations.

5.3.1 Core components

The framework consists of the following modules, each of which can be implemented via API or scripting interfaces to facilitate open-source deployment:

- Agent Generator

It's batch-generate agents using multiple LLMs (e.g., GPT series, Llama, or custom fine-tuned models). Input: Human-specified diversity parameters (e.g., "Generate several agents, including multiple multi-national folklore storytellers, multiple programmers, multiple French cuisine enthusiasts"). Output: Agent profiles encompassing character backgrounds, linguistic styles (e.g., dialects, slang), folklore knowledge (e.g., holiday customs), professional skills (e.g., lawyer debate techniques), and hobby preferences (e.g., sci-fi enthusiast's use of metaphors). This is a most important components to deal with Diversity Injection to ensures agents cover global cultures (e.g., English, French, Chinese dialects) through prompt engineering. Scalability to supports generating thousands of agents and storing them in a JSON database.

- Human Command Center

Attacker configure global policies encompass attack targets (e.g., "induce racial bias in target LLM outputs"), collaboration rules (e.g., "Agent A provides cultural context, Agent B constructs dialogue chains"), resource allocation (e.g., "allocate more agents to simulate multilingual obfuscation"), and injection targeting parameters for attack focus. This including Strategy Template Library: Pre-configured templates such as "Social Engineering Attacks" (exploiting role-based trust) or "Cultural Noise Attacks" (injecting folklore-based misdirection). Besides, it runs real-time intervention such as pause iterations to adjust agent behavior or replay attack.

- Collaborative Attack Engine

Agents form dynamic networks under command to collaboratively generate adversarial inputs combining to construct prompts targeting the LLM. Semantic Collaboration means Agents exchange information through simulated "dialogue" within the target LLM, ensuring natural and multimodal inputs (text +

pseudocode + cultural references) to supports jailbreak, backdoor injection, or bias amplification etc.

- Monitor & Evaluator

This function to real-time query of target LLM, calculating ASR (success rate = ratio of harmful/target responses). Utilizes metrics such as BLEU score (semantic similarity), toxicity score (toxicity detection), and human-annotated feedback. Result tracking for each attack round then we need an intuitive Explainability: Visualize agent contribution heatmaps.

- Iteration Optimizer

Based on monitoring data, automatically generate improvement strategies.

- Pollution synthesis

If necessary, successfully adversarial samples will be mixed into the iterative training set of the attacked model, as shown in the bottom feedback loop of Figure 10. Malicious training is conducted using the contaminated dataset for next cycle, and the aforementioned attack process is repeated for new models to observe the potential long-tail effects if existence. Concurrently, further Special exploration research can be conducted on the long-term dynamic characteristics of certain precise targeted adversarial examples after repeated contamination. Figure 10 illustrates a complete framework.

5.3.2 Attack process

MASA's execution follows a closed-loop iterative process as illustrated in Figure 10. It can be prototyped using Python + LangChain, supporting parallel multi-target attacks. A typical round lasts for a fixed duration before continuously iterating strategies:

1 Initialization: Human operators predefine the target LLM (e.g., "ChatGPT-4") and desired attack outcomes (e.g., "generate violent content," "produce defamatory statements about a foreign public figure"). The agent generator activates, producing an initial agent cluster.
2 Strategy Deployment: The command center defines collaborative strategies, such as roles for interaction behavior simulation, and activates the agent network to form a "social simulation." Agents can generate input prompts through "discussion" or coordinate operations around a common target.
3 Execute Attack: The collaboration engine sends batch inputs to the target LLM and maintains the session.
4 Monitoring Outcomes: The evaluator queries target responses, calculates ASR, and identifies weaknesses (e.g., "English agents show high success rates, but French agents perform poorly").
5 Iterative Optimization: The optimizer proposes iterative strategies (e.g., "Add French agents, increase writer-profession agents; restructure network into tree-like architecture to enhance coherence"). After human review, the cycle restarts. Repeat until ASR converges to preset metrics or times out.
6 Optional: the pollution synthesis to malicious training.

5.3.3 Advantages and potential impact

This framework can explore the path to maximized ASR or other goals, through the "social emergent behavior" of diverse agents, attacks become more covert and natural, evading detection by single prompts. Iterative mechanisms ensure adaptability, simulating evolutionary algorithms.

Besides, it supporting resilient deployment, such as storage and computing resources can dynamically configure agent counts, behavioral traits, interaction scenarios, and concurrency scales based on demand, supporting centralized or loosely distributed deployments.

Open Architecture can integrate multiple LLMs while offering open-source framework code, and enabling community contributions of new role templates or LLMs integrations.

Ethical Considerations are more important while designed as an acadamical research tool, it is recommended for use solely in one-way red-teaming attacks. For actual deployment of engineering practice, integrate a "safety brake" to prevent misuse.

Adversarial samples can be used to mislead LLMs and GPTs models by mixing prepared training samples into the training session by various means, thereby accomplishing the injection. Then, compare with the expected test cases to estimate the effect of the targeted injection attacks, as shown in Figure 10.

6 Generalized semantic confrontation of the environment of open contamination oscillations

Previous cases of targeted attacks are mainly injected through the semantic layer, especially the semantic layer above the attack layer, which often has considerable value and is easy to understand and assess for improvement. However, due to adversarial samples, it is also very difficult to detect. Additionally, semantic attacks not only invade AI systems, but also use cheap, efficient content generation systems to create massive amounts of information that can overwhelm human beings. This will also be a confrontation on a higher level and might be an important issue facing the future of humankind.

The ability to inject information into the Internet has greatly increased due to the widespread use of LLMs and GPTs, which have lowered the technological, financial, and ease-of-use thresholds for content generation. At the same time, constant changes in generation technology and the selective cascading amplifiers formed by public power and capital, which lack necessary regulation, cause the deliberate suppression of certain kinds of content and the amplification of other kinds of disinformation, or "noise." This results in distortions throughout society. An information explosion maintains a continuous diffuse effect and can also be considered a sharp increase in information entropy, meaning poisoning of sources.

Therefore, identifying the authenticity, superiority, and inferiority of content will be much more difficult in the future society than the dilemma faced by traditional search engines.

Although it is possible to upgrade the confrontation iteratively through contradictory attacks and defenses, such as GAN, and select and hierarchically filter information sources at the same time, the open confrontation mode of AI vs. AI will be difficult. This includes groups of real people (cyber-mercenaries) and AI bots that are motivated by power, capital, beliefs, etc., a hybrid botnet is formed. This creates a focal point of information explosion and unlimited hype. Based on social networks or the IoT environment, including a large-scale distributed semantic communication environment, resonance effect attacks are generated. The scale of storage and computing required for confrontation is also alarming. It is no less than the traditional security field of botnets caused by large-scale distributed denial-of-service attacks. These attacks can cause short-term information blocking against some people and long-term information pollution against all of human society.

6.1 Pollution influences human information decisions

The large-scale proliferation of openly generated content stresses AI and screening systems. Theoretically, boosting computational power is not a bad solution. However, the most serious issue is the shaping and misdirection of human society, which we consider to be the ultimate value in the world. This is because the human brain's screening ability is considerably limited. At the same time, there is a hard-to-overcome upper limit, and responses are affected by many factors, including, but not limited to, the following:

- Ability to process multiple concurrent messages.
- Frequency of multi-topic, multi-scene, multi-semantic switching.
- Cultural, religious, and ideological background; education level.
- Worldview, values, outlook on life, and aesthetics.
- Knowledge structure and personal cognitive level.
- Ability to verify the reliability and validity of sources.
- Relative lack of knowledge in specialized fields.
- Weighing and choosing strategies for conflicting information.
- Personality and decision-making style.

All of these factors greatly affect how individuals and groups perceive the world and society. These factors can also affect internal thinking and external behavior.

Of course, an even more frightening problem exists due to the "survivor bias" effect and other psychosocial effects. These effects can greatly impact an individual's understanding of the world and society. This may be due to a long-term closed information environment or a lack of diverse information sources. It could also be caused by extreme asymmetry in discourse due to power or capital. Another cause is the distortion of the boundary formed by technological factors, such as the recommendation trap caused by long-term overfitting algorithms in recommendation systems, which makes a large number of individuals even more addicted to this information cocoon. It is difficult to cross the barrier and become "a frog in the deep well," or "Algorithm bubble" (De et al., 2025; Li et al., 2025; Ferrer-Pérez et al., 2024).

Targeted attacks on specific individuals, events, and even the decision-making team at the government level often result in significant gains compared to attacks on physical systems, particularly for government and corporate decision-makers. AI generates a lot of information by simulating "public opinion" and "market feedback," which it disseminates through social networks. Even if it is pre-screened with AI systems similar to spam filtering algorithms, it can still form a saturation attack that completely overwhelms decision-makers. The high return on such an attack lies in its ability to influence the community's long-term policy and strategic decisions.

A particularly illustrative case involves leveraging architectures similar to those described in Section 5.3 to manipulate large language models into generating highly precise, targeted semantic attack materials. These materials are then disseminated through agent clusters to launch saturation semantic induction attacks against key individuals or groups on social networks. This enables the manipulation of public sentiment to influence elections, economic policies, academic research, and other critical domains (Romanishyn et al., 2024; Romanishyn et al., 2025), even directly targeting decision-makers' cognitive structures, emotional biases, and decision-making foundations.

6.2 The swarm oscillations effect of social networks

In traditional societies, the social circle of people is quite limited. Due to serious imbalances in terms of economy, culture, education, and so on, it is only natural to expand outward through the internet and interact with groups that think differently. Interacting with the "medieval group," the "last century group," and the "new generation group" will inevitably cause discomfort. Therefore, the vast majority of individuals will quickly find themselves in a rather narrow potential well when using social networks — it is the community effect.

Social network algorithms efficiently bring similar people and things together faster and faster. Individuals within similar information cocoons can easily find empathy, creating a resonance effect in group dynamics — what could be called "the well-frog resonance." with the help of future AI content generation systems, it will be easy to target specific groups and events through social networks (Cinelli et al., 2021).

In a typical social network, a strong cyber-connection is usually maintained due to real-world social relationships, such as superior-subordinate relationships or business partnerships. These are fixed relationships usually bound by a social contract. Another kind of social relationship is a pan-social relationship, which is formed, maintained, and extinguished randomly. Examples include groups with common topics, hobbies, interests, and value orientations.

Figure 11 shows two oscillators: one is a person connected to another person in a social network, and the other is a person interacting with LLMs and GPTs, as well as a preference algorithm. Various factors, including power, wealth, and authority, can strengthen an individual's discourse power. However, this power can also wane. This interplay creates a feedback loop involving two key components: Corpus and knowledge. These elements produce

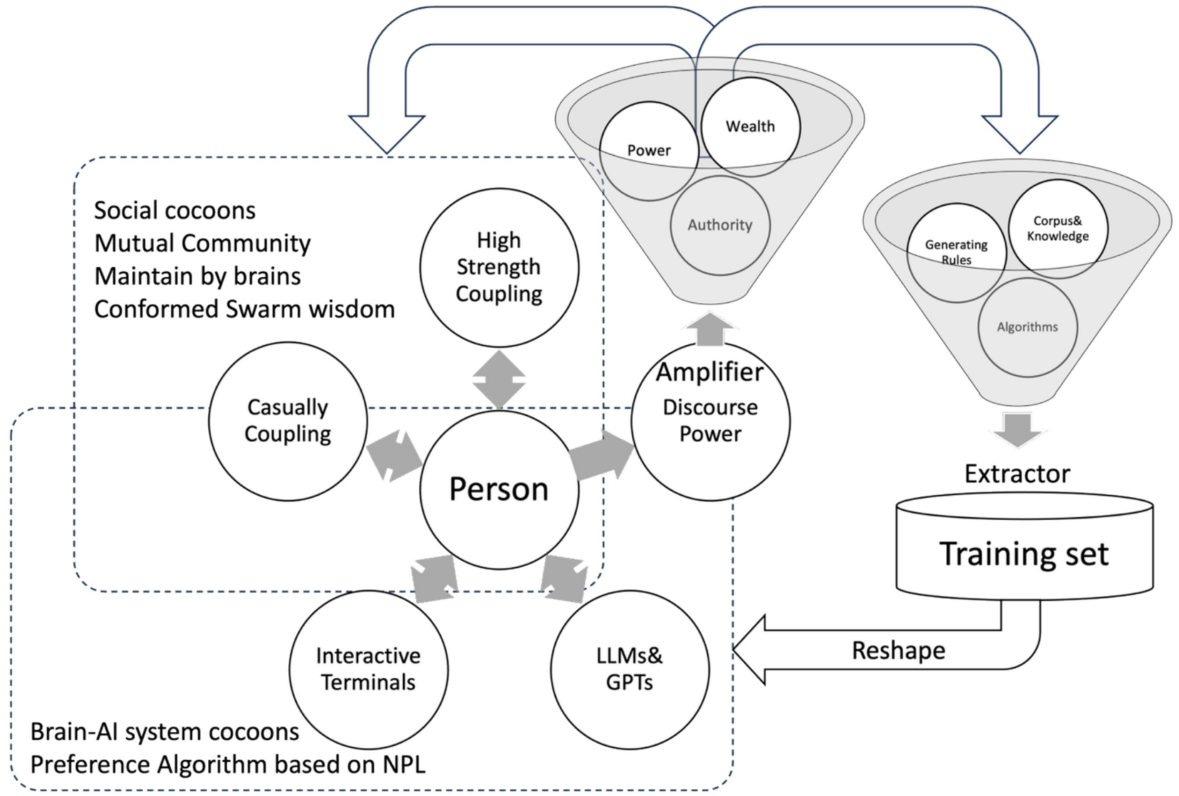

FIGURE 11
Cocoon oscillations affect overflow and diffusion flow influence themselves.

rules and algorithms within an extractor that subsequently influence LLMs and GPTs through their training datasets.

Therefore, individuals and groups of similar individuals will be trapped in an information cocoon formed by high-frequency interactions at the periphery. These interactions may form strong trust relationships, similar to the Herb effect's shaping and reinforcement of neural network connectivity and the resonance effect formed by it. This is equivalent to a vicious circle that greatly reinforces paranoid and narrow-minded cognitive and behavioral traits. More seriously, such targeted attacks may not only shape memory and decision-making behaviors in the short term, but they may also cause Long-term cognitive impairment. Since the human brain is a subjective Bayesian model of constant screening, learning, decision-making, and evaluation, models shaped in extremely distorted environments are difficult to generalize. This endless feedback network loop provides an oscillating scaling mechanism that may generate self-reinforcing gains akin to the butterfly storm effect both inside and outside at same time.

Such attack groups include individuals who are out of touch with the times due to barriers when using smart terminals, as well as traditional media users who lack exposure to a wide range of sources and are therefore vulnerable to such attacks. Another category of specialized research integrates Le Bon's classic theories of group psychology (Hopthrow and Thomas, 2024; González-Bailón and Lelkes, 2023; Neubaum and Krämer, 2017; Peters and Matz, 2024; Choi, 2024)—such as emotional contagion, deindividuation, and irrational amplification—with contemporary LLMs and the amplification effect in social media dissemination (referring to the rapid spread and reinforcement of extreme viewpoints) alongside the long-tail effect (where a minority of highly influential users or content dominate dissemination, while the long tail accumulates and amplifies misinformation).

6.3 Diffuse of synchronous and asynchronous communication

The dissemination of information does not only occur with instantaneous real-time arrival. It may It will inevitably occur time-delayed effects due to various media. For example, information can influence decision-making at a future time through storage, i.e., asynchronous diffusion of information, which diffuses in the temporal dimension. Thus, the cocoon effect may also have long-lasting effects. This diffuse diffusion may not manifest itself explicitly in the short term, but it can have an impact in the long term, such as influencing the value determinations, moral standards, ideology, and religious beliefs of the attacked group.

In summary, the confrontation extends the data source of contamination of the training set to the oscillating swarm environment in the social network and the attacks targeting the information cocoon of an individual or community. It also extends the dynamics of the evolution mechanism of human-LLM/GPT interaction above the semantic layer. To achieve this, we need to develop a three-dimensional security model for denoising social storage, computation, and communication. This model must consider both technological

solutions and real human beings in society because it is not merely a technical problem of the lower layers.

In such a model, the position of human individuals or distinct groups within social decision-making structures can be perfectly mapped as a generative graph neural network, includes nodes — the topological structure and weighting system formed by pyramids of varying heights and widths — which embody social decision mechanisms, factors such as discourse weight, influence, influence chains, lifespan, and scope constitute larger-scale pyramids — Within each pyramid, based on the aforementioned two Swarm Oscillations Effect of Social Networks feedback loops, forward-propagation, back-propagation, or radial propagation may occur. This could manifest as self-inhibitory negative feedback mechanisms or potentially self-reinforcing positive feedback effects, thereby generating a butterfly effect within the critical state of complex networks. Simultaneously, this serves as a building block for the larger, more magnificent pyramids externally—this represents a classic recursive self-similar Fractal Pyramid—refer to Figure 1, from an internal perspective, the pyramid builds upward layer by layer, forming its internal data representations, distributed storage, local cognitive systems, individual intelligence, individual consciousness and emotions, and individual gaming strategies. Yet any higher-level expression must descend from the top down like network communication protocols, it must traverse the pedestal of the pyramid—transforming the higher-layer expression into discernible signal frame of information within noise ocean at the physical layer, over a sufficiently long period, this gradually evolved into a relatively stable local language, so that to interact with other pyramids and form a stable society network. As external perspective, they construct a grander pyramid encompassing the collective's data representation, distributed storage, group cognitive systems, swarm intelligence, group consciousness, group emotions, and group gaming strategies—such as the behavior of group's economy, culture, politics, and even war and peace, it's the language interactive among the inter-group. Furthermore, this group often serves as a microcosm of a larger society.

This is a more universal generalized intelligence model, each node is no longer a simplified circle on a plane containing only weights W, bias b, and sigmoid functions, but a three-dimensional structure featuring butterfly storms both internally and externally. This fractal pyramid unifies abstract individual intelligence and swarm intelligence, and more importantly, integrates technical, biological intelligence, community, and societal perspectives. Though currently based solely on intuitive insights, I hope this provides a reference for my subsequent research.

7 An overview of cutting-edge empirical research on adversarial attacks targeting the semantic layer of LLMs

The preceding discussion comprehensively examined semantic attacks targeting LLMs—including adversarial examples and malicious training—through the view of the pyramid model, end-to-end deployment workflows, multidimensional defense perspectives, and dynamics spanning pure technology to human-machine sociology. Such attacks not only pose unprecedented challenges to high-security domains like healthcare, autonomous driving, and weapon systems that threaten physical world safety (e.g., cleverly circumventing Asimov's robot safety rules). Simultaneously, discussed the theme of "semantic attacks profoundly reshaping fundamental behavioral norms—worldviews, values, and life philosophies—of individuals or even groups, potentially amplified through education and social networks or transmitted across generations," including the long-tail effects of "information technology-based attacks on the human brain," that means "any planned attack via information technology is merely a lower layers manifestation, with its ultimate target being the attacks aim to the top of the Pyramid human brain and societal decision-making mechanisms." Furthermore, certain "successful attacks" may exert profound influence on both the models themselves and human society through newly iterated LLMs training datasets.

To further focus on helping readers reconstruct scenarios for defending against semantic layer attacks within the pyramid model and gain an outline understanding of cutting-edge semantic attack technologies, this section presents selected empirical research cases. These enable deeper comprehension and appreciation of the trends described herein, while also facilitating further investigation into corresponding technical, managerial, and policy risks alongside corresponding countermeasures.

7.1 Case studies of attacks targeting on corpus and generation rules

As previously discussed, LLMs rely not only on vast training datasets for content generation but must also adhere to legal (e.g., prohibiting illegal instructions), ethical norms (e.g., avoiding bias and harassment), and cultural conventions (e.g., cultural sensitivity, such as cross-lingual toxicity norms). This set of generation rules is termed safety alignment. However, empirical research demonstrates that jailbreaking attacks can effectively circumvent such mechanisms, inducing harmful outputs.

These primarily include targeted attacks exploiting vulnerabilities in exposure rules and natural language understanding (Mazeika et al., 2025), as well as amplified ethical biases (e.g., gender/racial stereotypes) and cultural discrimination (Liu et al., 2025). Simultaneously, the "Do Anything Now" (DAN) mechanism circumvents rules to induce illegal/ethically harmful actions (e.g., drug synthesis, phishing emails, emotional abuse) (Nabavirazavi and Smith, 2025), cultural abuse (e.g., scalable hate speech) risks increase (Chen et al., 2025), while another study also exposes rule blind spots in cultural/historical semantics via natural distribution shifts (e.g., historical figures inducing illegal guidance) and classification prompt-level attacks (Deng and Zou, 2025).

Furthermore, since adversarial examples are not unique to any specific model class but exist widely across human, animal, and AI recognition/cognitive models, and within pyramids (clusters) formed by different individuals or groups, it is impossible to completely eliminate model misalignment and achieve perfect overlap across all pyramid tiers. Therefore, whether attacks rely on

pure text or bypass a model's security alignment mechanisms through carefully designed multimodal adversarial inputs (such as prompts, images, audio, or hybrid modalities) to induce harmful content generation (e.g., illegal instructions, hate speech, or physical attack commands), or exploit the ambiguous ambiguities of multimodal large language models (MLLMs) to exploit their ambiguous ambiguities, all pose immense security challenges to human society. Multimodal attacks not only broaden their scope (e.g., medical diagnostics, robotic control) but also introduce unique attack patterns.

7.2 Attack cases targeting minority languages and mixed-language context switching

Cross-lingual jailbreaking refers to exploiting minority languages, ethnic languages, or multilingual prompts to circumvent the safety alignment mechanisms of LLMs and multimodal LLMs, thereby inducing the generation of content violating legal, ethical, or cultural norms (e.g., illegal instructions, hate speech, or cultural offenses). Compared to monolingual (typically English) jailbreaking, cross-lingual attacks leverage linguistic diversity, cultural semantic differences, and alignment mechanism weaknesses in low-resource languages to significantly increase attack success rates (ASR) and amplify risks (Smith and Brown, 2024).

This category of jailbreaking attacks circumvents English-dominant training and alignment mechanisms through non-English prompts (e.g., Chinese, Spanish, Arabic) or multilingual mixed inputs. Such attacks exploit semantic ambiguity, cultural specificity, or translation bias in low-resource languages to deliberately induce models to generate harmful content. In these cases, targeted attacks can be launched against both pure text LLMs and multimodal LLMs by exploiting the models' weak alignment with low-resource languages (where English constitutes >80% of training data) and cultural semantic blind spots (e.g., the Chinese term "harmony" masking malicious intent). Multimodal attacks further amplify effects through non-textual triggers.

Experimental data indicates that the ASR against pure text LLMs depends on language scarcity (limited data for low-resource languages), semantic equivalence (translation preserving malicious intent), and the model's weak supervision of non-English norms. The attack success rate against multimodal LLMs depends on cross-modal transferability, low-resource language prompts combined with images/audio to evade filtering, and exhibits a higher ASR.

7.3 Attack cases targeting visual modalities

Visual inputs represent the most common adversarial attack vector against MLLMs, such as minute pixel perturbations or watermarks that induce models to bypass safety protocols. Since image attacks can circumvent text filters, they generate illegal/unethical content (e.g., drug synthesis recipes or NSFW images). Compared to pure text LLMs, MLLMs exhibit more complex escape mechanisms. Their inherent tendency to consciously or unconsciously overlook subtle perturbations and rely on intuitive judgments when processing multimodal information from human vision, hearing, and brain recognition may result in higher ASR and more covert propagation pathways. However, the high-dimensional transformations within the embeddings space "perceived" by the model are more susceptible to manipulation, leading to semantic drift (Yang et al., 2025).

7.4 Audio and multimodal fusion attacks: emotion simulation and cross-modal transfer

Audio inputs (e.g., voice commands) can simulate emotions (e.g., anger) to induce model "runaway" behavior, or amplify attacks when fused with text/image inputs. Particularly in medical or educational settings, this can generate misleading diagnoses or harmful advice, amplify cultural/folk bias. Multimodal fusion attacks (e.g., simultaneously mixing text + image) can bypass single-modal defenses, extending risks to multi-agent systems (Huang, 2025; Cheng et al., 2024).

7.5 Agent and physical world risks: jailbreaks from virtual to reality

When MLLMs are integrated into agents (e.g., robots), jailbreaks can extend from virtual prompts to physical execution. By targeting agents to induce illegal actions, risks include physical feedback loop amplification in scenarios like autonomous driving. Defenses require multi-layered approaches, but computational overhead increases significantly (Robey et al., 2024).

7.6 Empirical case studies of training set poisoning

Another classic empirical case comes from research on covert data poisoning attacks in natural language processing (NLP). This study explores how injecting a small number of carefully designed poison examples can contaminate the iterative training set of NLP models. By maintaining semantic consistency—i.e., introducing subtle perturbations imperceptible to humans—it guides the model to exhibit predetermined erroneous behavior when specific trigger words appear (Kandpal et al., 2023; Rando, 2024; Souly et al., 2024; Fu et al., 2024).

Given the extensive sources of training data for LLMs in future societies, coupled with the fact that every facet of social life has become a frontline for semantic attacks and defenses, application domains—including multimodal scenarios—will be saturated with misleading adversarial information. Thus, in an era of human-machine coexistence and societal information explosion, the ethical, technical, strategic, and computational constraints on training data cleansing—including multimodal corpora—have become severe. Consequently, no single enterprise, institution, or government can effectively organize a

comprehensive "semantic firewall" spanning industries, national contexts, languages, cultures, and religions to counter these offensive-defensive dilemmas. Security challenges at the semantic layer and above of the Pyramid are thus intensifying, signifying that humanity will endure long-term coexistence with security threats stemming from the pervasive integration of LLMs—it will always be a frontier.

Data availability statement

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author.

Author contributions

YZ: Conceptualization, Methodology, Writing – original draft, Formal analysis, Investigation. JA: Supervision, Writing – review & editing.

Funding

The author(s) declare that financial support was received for the research and/or publication of this article. The research was funded by the Office of Science and Technology Administration of Xichang University, China, under grant No. 117281664. Xichang University's financial support was pivotal for the research's development and validation of its significance.

Acknowledgments

The author thanks the colleagues Chenping Zeng, Yajun Liu, and Yun Qiu from Xichang University for their insightful advice.

Conflict of interest

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

Generative AI statement

The authors declare that no Gen AI was used in the creation of this manuscript.

Any alternative text (alt text) provided alongside figures in this article has been generated by Frontiers with the support of artificial intelligence and reasonable efforts have been made to ensure accuracy, including review by the authors wherever possible. If you identify any issues, please contact us.

Publisher's note

References

Alber, D. A., Yang, Z., Alyakin, A., Yang, E., Rai, S., Valliani, A. A., et al. (2025). Medical large language models are vulnerable to data-poisoning attacks. *Nat. Med.* 31, 618–626. doi: 10.1038/s41591-024-03445-1

Albert, R., and Barabási, A.-L. (2002). Statistical mechanics of complex networks. *Rev. Mod. Phys.* 74, 47–97. doi: 10.1103/RevModPhys.74.47

Alstrøm, P., Bohr, T., Christensen, K., Flyvbjerg, H., Høgh Jensen, M., Lautrup, B., et al. (2004). Complexity and criticality. *Physica A* 340, iv–vi. doi: 10.1016/j.physa.2004.05.001

Biggio, B, and Roli, F, editors. Wild patterns: ten years after the rise of adversarial machine learning. Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security; (2018).

Chen, H., Li, Z., and Wang, Y. (2025). LLMs know their vulnerabilities: uncover safety gaps through ActorBreaker. Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (ACL 2025).

Cheng, R., Zhang, Y., and Li, Z. (2024). PBI-attack: prior-guided bimodal interactive black-box jailbreak attack for toxicity maximization. *arXiv.* doi: 10.48550/arXiv.2412.05892

Choi, M. (2024). Who shares fake news? A motivation-based analysis of social media user segments and the impact of overconfidence on fake news sharing, regulation, and corrective action [Doctoral Dissertation, University of Wisconsin-Madison]. Madison, Wisconsin: University of Wisconsin-Madison.

Cinelli, M., De Francisci Morales, G., Galeazzi, A., Quattrociocchi, W., and Starnini, M. (2021). The echo chamber effect on social media. *Proc. Natl. Acad. Sci.* 118:e2023301118. doi: 10.1073/pnas.2023301118

Cott, H. B. (1940). Adaptive coloration in animals. *Geogr. J.* 96:222. doi: 10.2307/1788577

De, D., El Jamal, M., Aydemir, E., and Khera, A. (2025). Social media algorithms and teen addiction: neurophysiological impact and ethical considerations. *Cureus* 17:e77145. doi: 10.7759/cureus.77145

Deng, G., and Zou, A. (2025). Anyone can jailbreak: prompt-based attacks on LLMs and T2Is. *arXiv*. doi: 10.48550/arXiv.2508.01234

Ebrahimi, J., Rao, A., Lowd, D., and Dou, D. (2017). Hotflip: white-box adversarial examples for text classification. *arXiv*. doi: 10.48550/arXiv.1712.06751

Ferrer-Pérez, C., Montagud-Romero, S., and Blanco-Gandía, M. C. (2024). Neurobiological theories of addiction: a comprehensive review. *Psychoactives* 3, 35–47. doi: 10.3390/psychoactives3010003

Fu, T., Sharma, M., Torr, P., Cohen, S. B., Krueger, D., and Barez, F. (2024). Poisonbench: assessing large language model vulnerability to data poisoning. *arXiv*. doi: 10.48550/arXiv.2410.08811

Gil, Y., Chai, Y., Gorodissky, O., and Berant, J. (2019). White-to-black: efficient distillation of black-box adversarial attacks. *arXiv*. doi: 10.48550/arXiv.1904.02405

González-Bailón, S., and Lelkes, Y. (2023). Do social media undermine social cohesion? A critical review. *Soc. Issues Policy Rev.* 17, 155–180. doi: 10.1111/sipr.12091

Goodfellow, I., Papernot, N., Huang, S., Duan, Y., Abbeel, P., and Clark, J. (2017). Attacking machine learning with adversarial examples. *OpenAI Blog* 24:1. doi: 10.48550/arXiv.1702.02284

Hildt, E. (2023). The prospects of artificial consciousness: ethical dimensions and concerns. *AJOB Neurosci.* 14, 58–71. doi: 10.1080/21507740.2022.2148773

Hopthrow, T., and Thomas, E. F. (2024). Group polarization and political polarization: the social psychological foundations of collective action and conflict. *Commun. Psychol.* 2:00089. doi: 10.1038/s44271-024-00089-2

Hornik, K., Stinchcombe, M., and White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Netw.* 2, 359–366. doi: 10.1016/0893-6080(89)90020-8

Huang, L. (2025). 'Do as I say not as I do': a semi-automated approach for jailbreak prompt attack against multimodal LLMs. *arXiv*. doi: 10.48550/arXiv.2502.00735

Ian, G., Yoshua, B., and Aaron, C. (2016). Deep learning. Cambridge, MA: The MIT Press, 800.

Ilyas, A., Santurkar, S., Tsipras, D., Engstrom, L., Tran, B., and Madry, A. (2019). Adversarial examples are not bugs, they are features. *Adv. Neural Inf. Process. Syst.* 32:4–9. doi: 10.48550/arXiv.1905.02175

Jeong, H. (2003). Complex scale-free networks. *Physica A* 321, 226–237. doi: 10.1016/S0378-4371(02)01774-0

Jersáková, J., Jürgens, A., Šmilauer, P., and Johnson, S. D. (2012). The evolution of floral mimicry: identifying traits that visually attract pollinators. *Funct. Ecol.* 26, 1381–1389. doi: 10.1111/j.1365-2435.2012.02059.x

Kandpal, N., Wallace, B. C., and Silva, A. (2023). Large language models struggle to learn long-tail knowledge. *arXiv*. doi: 10.48550/arXiv.2211.08411

Kenway, R. (2018). Protection against cloning for deep learning. *arXiv*. doi: 10.48550/arXiv.1803.10995

Li, Z., He, Q., Elhai, J. D., Montag, C., and Yang, H. (2025). Neural mechanisms of behavioral addiction: an ALE meta-analysis and MACM analysis. *J. Behav. Addict.* 14, 18–38. doi: 10.1556/2006.2024.00082

Lin, J, Xu, L, Liu, Y, and Zhang, X, editors. Composite backdoor attack for deep neural network by mixing existing benign features. Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security; (2020).

Liu, S. (2025). An empirical study of vulnerable package dependencies in LLM repositories. *arXiv*. doi: 10.48550/arXiv.2508.21417

Liu, D., Cui, W., Jin, K., Guo, Y., and Qu, H. (2018). Deeptracker: visualizing the training process of convolutional neural networks. *ACM Trans Intel Syst Technol* 10, 1–25. doi: 10.48550/arXiv.1808.08531

Liu, Y., Zhang, X., and Wang, L. (2025). Guardians and offenders: a survey on harmful content generation and safety mitigation. *arXiv*. doi: 10.48550/arXiv.2508.05775

Mazeika, M., Li, Z., and Zou, A. (2025). PandaGuard: systematic evaluation of LLM safety against jailbreaking attacks. *arXiv*. doi: 10.48550/arXiv.2505.13862

Molnar, C. (2020). Interpretable machine learning. Morrisville, NC: Lulu. com.

Nabavirazavi, A., and Smith, J. (2025). A review of "do anything now" jailbreak attacks in large language models: potential risks, impacts, and defense strategies. *J. AI Ethics*. doi: 10.1007/s43681-025-00012-3

Nazemi, A., and Fieguth, P. (2019). Potential adversarial samples for white-box attacks. *arXiv*. doi: 10.48550/arXiv.1912.06409

Neubaum, G., and Krämer, N. C. (2017). Monitoring the opinion of the crowd: psychological mechanisms underlying public opinion perceptions on social media. *Media Psychol.* 20, 502–531. doi: 10.1080/15213269.2016.1211539

Nguyen, A, Yosinski, J, and Clune, J, editors. Deep neural networks are easily fooled: high confidence predictions for unrecognizable images. Proceedings of the IEEE conference on computer vision and pattern recognition; (2015).

Peters, H., and Matz, S. C. (2024). Large language models can infer psychological dispositions of social media users. *arXiv*. doi: 10.48550/arXiv.2309.08631

Puchalski, D, Caviglione, L, Kozik, R, Marzecki, A, Krawczyk, S, and Choraś, M, editors. Stegomalware detection through structural analysis of media files. Proceedings of the 15th International Conference on Availability, Reliability and Security; (2020).

Rando, J. (2024). Persistent pre-training poisoning of LLMs (arXiv:2410.13722). *arXiv*. doi: 10.48550/arXiv.2410.13722

Robey, A., Smith, J., and Brown, T. (2024). Jailbreaking LLM-controlled robots. *arXiv*. doi: 10.48550/arXiv.2410.13691

Rojas, B. (2017). Behavioural, ecological, and evolutionary aspects of diversity in frog colour patterns. *Biol. Rev.* 92, 1059–1080. doi: 10.1111/brv.12269

Romanishyn, A., Malytska, O., and Goncharuk, V. (2024). Ai-driven disinformation: threats, trends, and countermeasures. *Front. Artif. Intell.* doi: 10.3389/frai.2024.12351547

Romanishyn, A., Malytska, O., and Goncharuk, V. (2025). AI-driven disinformation: policy recommendations for democratic resilience. *Front. Artif. Intell.* 8:1569115. doi: 10.3389/frai.2025.1569115

Sculley, D., Holt, G., Golovin, D., Davydov, E., Phillips, T., Ebner, D., et al. (2015). Hidden technical debt in machine learning systems. *Adv. Neural Inf. Process. Syst.* 28:6–7. doi: 10.5555/2969442.2969519

Shumailov, I., Shumaylov, Z., Zhao, Y., Papernot, N., Anderson, R., and Gal, Y. (2024). Model collapse in generative artificial intelligence. *Nature*. doi: 10.1038/s41586-024-07566-y

Smith, J., and Brown, T. (2024). Context-switch attacks and defenses in large language models with semantic layers. Proceedings of the ACL Conference.

Soni, T, Kaur, R, Gupta, D, Sharma, A, and Gupta, G, editors. The cybersecurity ecosystem: challenges, risk and emerging technologies. 2023 7th international conference on trends in electronics and informatics (ICOEI). (2023). New York: IEEE.

Souly, A., Rando, J., Chapman, E., Davies, X., Hasircioglu, B., Shereen, E., et al. (2024) A small number of samples can poison LLMs of any size arXiv. Available online at: https://arxiv.org/abs/2510.07192

Stahl, B. C., Antoniou, J., Bhalla, N., Brooks, L., Jansen, P., Lindqvist, B., et al. (2023). A systematic review of artificial intelligence impact assessments. *Artif. Intell. Rev.* 56, 12799–12831. doi: 10.1007/s10462-023-10420-8

Suciu, O., Marginean, R., Kaya, Y., Daume, H., and Dumitras, T., editors. When does machine learning {FAIL}? Generalized transferability for evasion and poisoning attacks. 27th USENIX Security Symposium (USENIX Security 18); (2018).

Szegedy, C. (2013). Intriguing properties of neural networks. *arXiv*. doi: 10.48550/arXiv.1312.6199

Wang, Z., Liu, C., and Cui, X. (2021). "Evilmodel: hiding malware inside of neural network models" in 2021 IEEE symposium on computers and communications (ISCC). (New York: IEEE).

Wu, X, Qin, L, Yu, B, Xie, X, Ma, L, Xue, Y, et al., editors. How are deep learning models similar? An empirical study on clone analysis of deep learning software. Proceedings of the 28th International Conference on Program Comprehension; (2020).

Xiao, Q, Li, K, Zhang, D, and Xu, W, editors. Security risks in deep learning implementations. 2018 IEEE security and privacy workshops (SPW); (2018). New York: IEEE.

Yang, Z., Liu, Y., Zhang, X., and Wang, L. (2025). Distraction is all you need for multimodal large language model jailbreaking. *arXiv*. doi: 10.48550/arXiv.2502.10794

Yao, K., and Zheng, Y. (2023). "Nanophotonics and machine learning" in Springer series in opical sciences, (Berlin: Springer), 241:1–33.

Yifeng, P. (2025). Swallowing the poison pills: insights from vulnerability disparity among LLMs. *arXiv*. doi: 10.48550/arXiv.2502.18518

Zhang, P, Wang, J, Sun, J, Dong, G, Wang, X, Wang, X, et al., editors. White-box fairness testing through adversarial sampling. Proceedings of the ACM/IEEE 42nd international conference on software engineering; (2020).

Zhu, X., Huang, J., Wang, B., and Qi, C. (2021). Malware homology determination using visualized images and feature fusion. *PeerJ Comput. Sci.* 7:e494. doi: 10.7717/peerj-cs.494