International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 07 Issue: 12 | Dec - 2023 SJIF Rating: 8.176 ISSN: 2582-3930

Data Security and Privacy Protection in Artificial Intelligence Models: Challenges and Defense Mechanisms

Chaitanya Tumma[1], Rahul Azmeera[2], Supraja Ayyamgari[3], Bala Yashwanth Reddy Thumma[4]

[1]chaitanyatumma1@gmail.com, [2]razmeera8848@ucumberlands.edu, [3]supraja.ayyamgari@gmail.com, [4]balayrthumma@gmail.com

[134]VISA Inc., Ashburn, VA, USA

[2]Information Technology, University of the Cumberlands, Williamsburg, KY, USA

Abstract: Artificial intelligence (AI) and deep learning algorithms are advancing rapidly, with these emerging technologies being widely applied in areas such as audio-visual recognition and natural language processing. However, in recent years, researchers have identified several security risks in current mainstream AI models, which could hinder the further development of AI technologies. As a result, the issues of data security and privacy protection in AI models have become a focus of research. The data and privacy leakage problems are primarily studied from two perspectives: data leakage based on model outputs and data leakage based on model updates. In the context of model output-based data leakage, the study discusses the principles and research status of model theft attacks, model inversion attacks, and membership inference attacks. In the context of model update-based data leakage, the research focuses on how attackers can steal private data during the distributed training process. Regarding data and privacy protection, three common defense methods are primarily studied: model structure defenses, information obfuscation defenses, and query control defenses. This paper reviews the cutting-edge research achievements in the field of data security and privacy protection in AI deep learning models, focusing on the theoretical foundations, key findings, and related applications of data theft and defense technologies in AI deep learning models.

Keywords: Artificial Intelligence, Data Security, Privacy Leakage, Privacy Protection

I. Introduction

Artificial Intelligence (AI) technology is rapidly advancing, driven by three key factors:

i. Breakthrough progress in deep neural networks (DNN) across various classic machine learning tasks;
ii. The maturation of big data processing technologies and the accumulation of vast amounts of data;
iii. Significant improvements in hardware computational capabilities.

These factors have propelled AI applications in areas such as autonomous driving, image recognition, and speech recognition, thereby accelerating the intelligent transformation of traditional industries. AI technology is already widely applied in India. In the e-commerce sector, AI is used for tasks like user behavior analysis and network traffic analysis, improving the efficiency of businesses in handling high-concurrency tasks while enhancing the robustness of overall systems. In the bright transportation field, AI aids in path planning and driver-passenger behavior detection. In finance, AI is utilized for high-frequency trading, fraud detection, and anomaly detection tasks. In network security, AI assists in automation testing, greatly enhancing the efficiency of security

personnel in identifying anomalies from large datasets. In 2017, the Chinese government's work report mentioned AI for the first time, elevating its development to a national strategic level [1].

Most real-world machine learning tasks are resource-intensive, relying on significant computational and storage resources to complete model training or prediction. As a result, cloud service providers such as Amazon, Google, and Microsoft offer machine learning services to offset the storage and computational demand. These providers offer training platforms and query interfaces for users accessing the service to query specific instances. Typically, service providers charge users based on the number of queries made.

However, while AI technology is progressing rapidly, it faces significant data leakage risks. AI model parameters must be protected, as any leak could cause substantial economic losses for model owners. Moreover, the sample data required for AI often contains personal privacy information, which, if leaked, poses severe economic and legal risks for model owners. In 2017, India enacted the" Cybersecurity Law of the People's Republic of India," which emphasizes the protection of personal private information. Therefore, preventing data leakage risks in AI applications has become one of the main obstacles to the further development and deployment of AI technologies. To ensure the privacy of AI models, cloud service providers safeguard the confidentiality of their models by offering only an interface for user access, thereby preventing users from accessing model data directly. However, in recent years, there have been numerous attempts to compromise the data privacy of AI models. Researchers have found that when using deep learning models, relevant computational data—such as output vectors, model parameters, and gradients—may leak sensitive information about the training data or the model's attributes. Worse still, this data is often inevitably exposed to attackers, particularly the output vectors of specific models [3].

This makes the data leakage problem in deep learning models challenging to avoid. For example, in model inversion attacks, an attacker can deduce private user data by reverse-engineering the model output without accessing the sensitive data. In membership inference attacks, attackers can determine whether a specific data point is part of the training set based on the model output. These types of attacks only require interaction with the cloud service interface. In practical applications, such information theft can lead to serious privacy breaches—such as the recovery of facial images from the output vectors of a face recognition model, leading to the exposure of users' facial information. Attackers may also steal model parameters through output results, causing significant economic damage to model owners [4].

Furthermore, with the development of distributed machine learning techniques like federated learning, attackers may become participants in the model training process. While participants in federated learning typically cannot access each other's input data, attackers can gain access to model outputs, parameters, and gradients during training. This dramatically enhances the attackers' capabilities and makes it possible for them to steal the private data of other participants, thus presenting a serious threat to the development of distributed machine learning technologies. Many researchers have recently proposed various mechanisms to defend against privacy attacks targeting AI technologies. Modifying model structures, adding specific noise to output vectors, and employing techniques such as differential privacy can effectively defend against certain privacy leakage attacks. This paper will introduce data inference attacks that have been widely researched, including model stealing attacks, model inversion attacks, and membership inference attacks. It will also discuss defense mechanisms against these attacks and how models with privacy protection functions can resist specific data inference attacks [4].

II. AI Data and Privacy Leaks

During the training and application of deep learning models, the data and model parameters are at risk of being leaked. Depending on the type of model output information that attackers utilize, these inference attacks can be categorized into two types: data leakage based on model outputs and data leakage based on gradient updates.

2.1. Data Leakage Based on Model Output

Model output refers to the predicted results returned to the user after a model has been trained and deployed. For example, in classification tasks, the model's output is each sample's expected class or probability vector. Recent research has shown that model output results inherently contain certain data information. Attackers can exploit the model output to some extent to steal relevant data. Two main data types can be stolen: model parameters and training/testing data [2].

2.1.1. Model Theft

Model theft (or extraction) is a malicious activity where an attacker queries a black-box model to obtain results, gaining similar functionality or simulating the decision boundaries of the target model. The stolen model is often one that the owner has invested substantial time and resources in building, making it highly valuable. Once the model's information is leaked, attackers can avoid paying for services or exploiting third-party services for commercial gain, harming the model owner's interests. Even more seriously, if the model is stolen, attackers can deploy white-box adversarial attacks to deceive the online model, significantly increasing the likelihood of a successful attack. For example, in a black-box adversarial attack against online AI classification tasks by Amazon and Google, researchers were able to perform a model theft attack using a small number of samples and generate white-box adversarial examples for the stolen model, leading to misclassification rates of 96.19% and 88.94%, respectively [4].

Currently, most AI service providers offer services with the model located on secure cloud servers, providing paid query services via APIs. Clients can only input query samples through predefined APIs and receive the model's prediction results. However, even if attackers only use the information from the predicted results, they can, under certain conditions, steal the model from the server by querying it. Model theft attacks can be divided into Equation-solving Attacks, Meta-model-based model theft, and Substitute-model-based model theft.

2.1.2. Equation-solving Attacks

Equation-solving Attacks primarily target traditional machine learning models like Support Vector Machines (SVMs). Attackers can obtain model-related information, such as the algorithm or structure, and then create formulae to solve for model parameters based on query results [5]. Additionally, attackers can steal hyperparameters in traditional algorithms, such as loss function weights and regularization parameters [6], or values like K in k-nearest neighbor (KNN) algorithms. This method requires the attacker to understand the target algorithm's type, structure, and training data and cannot be applied to complex neural network models.

2.1.3. Meta-model-based Model Theft

Meta-model-based model theft involves training an additional meta-model $\Phi(\cdot)$ to predict specific attributes of the target model. The meta-model takes the model's output on task data 'x' as input and outputs predicted attributes like network layers or activation function types. To train the meta-model, the attacker must collect various models with the same functionality as the target model, obtain their outputs on the same datasets, and create a training set for the meta-model. However, constructing this training set requires diverse task-related models and is computationally intensive, so this attack is not very practical. For example, related experiments were only conducted on the MNIST digit recognition task [7].

2.1.4. Substitute-model-based Model Theft

Substitute-model-based model theft is currently the most practical form of attack. The attacker queries the target model with samples, obtains the prediction results, and uses them to label the query data to create a training dataset. They then train a substitute model locally with the same task as the target model. After extensive training, the

substitute model approximates the target model's characteristics. Attackers often select deep learning models like VGG [8] and ResNet [9] as substitute models with strong fitting capabilities. Unlike Equation-solving Attacks, substitute-model theft does not require knowledge of the target model's structure; the goal is not to steal the exact parameters but to approximate the target model's functionality. To achieve this, attackers query the target model with many samples but lack sufficient data. Excessive queries increase the cost and may raise suspicion of the model owner. To address this, researchers have proposed data augmentation techniques to enhance the query dataset, allowing the samples to capture the target model's characteristics [4]. Adversarial examples can be used to expand the training set, and these adversarial samples typically lie near the model's decision boundary, making the substitute model better simulate the target model's behavior [11, 12]. In addition to data augmentation, studies have shown that using unrelated data to construct the dataset can also produce considerable attack results, and strategies for selecting task-related and unrelated data combinations have been proposed [2-10].

2.1.5. Privacy Leaks

Machine learning model prediction results often contain a wealth of inference information about the sample. In different learning tasks, these results may carry different meanings. For example, in image classification tasks, the model output is a vector, where each component represents the probability of the sample belonging to a particular class. Recent research has demonstrated that the outputs of black-box models can be used to steal information about the model's training data. For instance, Fredrikson et al. proposed model inversion attacks [13], which can use confidence information in the model's output to recover facial images from the training dataset. They applied model inversion attacks to commonly used facial recognition models, including softmax regression [14], multilayer perceptron, and autoencoders. Their method showed that the confidence in model outputs contains input data information, which can be used as a measure for data recovery attacks. They formulated model inversion as an optimization problem, where the goal was to minimize the difference between the reverse data's output vector and the target data's output vector. If the attacker obtains an output vector for a specific category, they can use gradient descent to adjust the reversed data to ensure it produces the same output vector.

Membership inference attacks are an easier-to-implement form of attack. In this case, the attacker attempts to determine whether a particular sample is part of the training dataset of the target model. For example, an attacker might want to know if a person's medical data exists in a company's diagnostic model's training data. If it does, it could be used to infer private information about that individual. Data in the target model's training set is called member data, while data not in the training set is called non-member data. Since attackers typically do not have access to the target model, they can only perform membership inference attacks in a black-box scenario. Membership inference attacks have been extensively studied in the literature [15-20] and have become a major research topic in areas like medical diagnostics and genetic testing. The development of these attacks and their defense mechanisms has become an emerging focus of research.

In 2017, Shokri et al. [15] first proposed membership inference attacks. After extensive experiments, they designed a system for black-box membership inference attacks. The attack principle is based on the observation that the prediction vectors for member data and non-member data differ significantly. If an attacker can accurately capture this difference, they can perform a membership inference attack. However, in a black-box scenario, only prediction vectors are available from the target model, and in practice, due to usage restrictions, it is often impossible to obtain a sufficient variety of prediction vectors. Additionally, since the distribution of prediction vectors for different samples is inherently inconsistent, training directly with prediction vectors may not yield effective attack results. To overcome this, Shokri et al. used shadow datasets with the same distribution as the target dataset and trained multiple shadow models for each data type to enhance the prediction vector dataset for attack model training. By

using the prediction vectors, they built an attack model to capture the difference in prediction vectors between member and non-member data, successfully performing membership inference attacks in black-box settings.

2.2. Gradient Update-Based Data Leakage

Gradient updates refer to the process where the model parameters are optimized during training based on the gradients calculated. These gradients, generated throughout training, can also contain certain privacy-sensitive information. Gradient updates typically occur in distributed training scenarios, where multiple parties with different data each use their own data to update the model in each round. These updates are exchanged and summarized to train a unified model. During this process, neither the central server nor any of the training participants gain access to the training data of other parties. However, even when the original data is well-protected, gradient updates can still result in privacy leakage.

Although various methods have been implemented to protect the raw data during training, in multi-party distributed AI model training, individuals use their own data to train the current model and share the model parameter updates with others or the central server. Recent research in machine learning and information security conferences has highlighted attacks that exploit model parameter updates to obtain information about other parties' training data. For example, Melis et al. [29] used model parameters updated by other users as input features to train an attack model to infer attributes of other users' datasets. Other researchers [30,31] employed Generative Adversarial Networks (GANs) to generate methods for recovering other users' training data. In multi-party collaborative training, the common model is used as a basic discriminator, and model parameter updates serve as input to train the generator, eventually leading to the extraction of specific training data from a victim's dataset. In a more recent study [32], the authors did not use GANs but instead adjusted the pixels of simulated images based on optimization algorithms to make the gradients obtained from backpropagation in the common model resemble the real gradients. After several rounds of optimization, the simulated images gradually approach the actual training data.

III. AI Data and Privacy Protection

To mitigate the potential risks of model and data privacy leakage during AI training and testing, including training data leakage due to parameter updates, model data leakage from query responses during testing, and indirect privacy breaches during normal AI model use, both academia and industry have explored various solutions from different perspectives.

In the absence of direct attacks, the information generated during the normal training and use of AI models can still lead to indirect data privacy leakage. To address such data leaks, the primary strategy is to minimize or obfuscate the useful information contained in interactive data without affecting the model's effectiveness. Several data privacy protection measures can be employed, including:

- Model Structure Defense: This approach involves intentionally adjusting the model during training to reduce the sensitivity of its output to different samples.
- Information Obfuscation Defense: This method modifies interaction data, such as model outputs or parameter updates, to obscure the useful information while ensuring model effectiveness.
- Query Control Defense: This defense involves monitoring query operations and rejecting malicious queries to prevent data leakage.

3.1. Model Structure Defense

Model-based defense methods modify the structure of the model to reduce the amount of information leaked or reduce model overfitting, thereby protecting both model and data leakage. Fredrikson et al. [33] proposed that when the target model is a decision tree, variants of the CART decision tree can be used to adjust the priority of sensitive features. Their experiments showed that when sensitive attributes were placed at the root or leaf nodes, it provided a strong defense against model inversion attacks. The best defense was achieved when sensitive attributes were placed at the root node.

Shokri et al. [15] and Ahmed et al. [17] proposed adding Dropout layers to the target model, using model stacking to combine different meta-learners, or adding regularization terms. Their experiments demonstrated that these methods significantly reduced the accuracy of member inference attacks. Nasr et al. [34] proposed an adversarial learning-based defense method, where if the success rate of resistance to member inference attacks could be calculated and incorporated as an adversarial regularization term in the loss function, adversarial training with a MIN-MAX approach could train a model that limits the success rate of such attacks. Their experiments showed that this method achieved a low upper bound on attack success rate while maintaining high classification accuracy.

Furthermore, Wang et al. [35] developed MIASec, which modifies key features of the training data in the target model to make it difficult to distinguish between prediction vectors for member and non-member data. As mentioned earlier, model inversion attacks stem from the information contained in output vectors, while member inference attacks arise from the inconsistent distribution of prediction vectors between training and testing samples. Therefore, defending against model inversion attacks involves reducing the correlation between output and input vectors, while defending against member inference attacks involves minimizing the distribution differences between output vectors. Model-based defense aims to modify the structure and loss function of the model to minimize the information contained in the output vectors, thus offering a strong defense. However, this method still has drawbacks, as it can significantly affect model performance, causing fluctuations in classification accuracy. Therefore, defenders need to strike a balance between the model's performance and its robustness.

In recent years, some research has combined machine learning with cryptographic techniques to protect model privacy. Nan et al. [36] proposed using differential privacy techniques to modify gradients during distributed training to protect the privacy of the training dataset. While this approach may reduce the final performance of the model, it significantly improves the privacy of the training set. Similarly, Patra et al. [37] implemented encrypted matrix multiplication and activation function calculations using secure multi-party computation technology, which effectively protects the privacy of the training dataset during the training process. These privacy-preserving machine learning techniques can also be applied to defend against data leakage, strengthening the privacy of model training sets.

3.2. Information Obfuscation Defense

Data-oriented defenses refer to applying obfuscation techniques to the input samples or predicted results of the model. These obfuscation operations aim to disrupt the effective information contained in the output while ensuring the AI model's output correctness, thereby reducing privacy leakage. These data obfuscation methods primarily include two types: truncation obfuscation and noise obfuscation.

For truncation obfuscation, Shokri et al. [15] proposed truncating the output vector generated by the target model, such as only returning the results for the categories with higher probability values or reducing the precision of decimal places in the output vector. Fredrikson et al. [33] suggested rounding the output vector to achieve a similar

obfuscation effect. By using methods like truncation obfuscation, researchers have reduced the effectiveness of model inversion attacks and member inference attacks.

For noise obfuscation, Jia et al. [38] introduced Mem-guard based on the concept of adversarial samples. They found that member inference attacks are very sensitive to changes in the prediction vector provided by the target model. By adding carefully designed noise to the prediction vector, they could confuse the distribution differences between the prediction vectors of member and non-member data, creating an "adversarial sample" that did not affect the actual results, thereby defending against member inference attacks. He et al. [39] proposed using differential privacy techniques to add noise to the output vector for obfuscation. They argued that differential privacy algorithms could remove the features of the output vector while retaining the classification information, making the output vector harder to distinguish. Additionally, they suggested adding noise terms to the loss function, slightly sacrificing classification accuracy while enhancing the privacy of the output vector, effectively defending against member inference attacks.

Both model inversion attacks and member inference attacks involve the target model's output vector. Therefore, if the output vector can be modified explicitly without affecting the classification result, it can disrupt the effective information in the output, thus defending against these attacks. However, this method still has limitations. If the modification to the output vector is minimal, its resistance to attacks will not be very strong. If the modification is extensive, it will affect the usability of the classification data. Hence, a balance between privacy and usability must be struck.

3.3. Query Control Defense

Query control defense refers to the defender extracting features from user query behavior to defend against privacy leakage attacks. To carry out a privacy leakage attack, an attacker must make many queries to the target model and sometimes modify their input vectors to accelerate the attack. Analyzing user query behavior characteristics makes it possible to identify potential attackers and restrict or deny their queries to prevent the attack. Query control defense primarily includes anomaly sample detection and query behavior detection.

In anomaly sample detection, attackers typically need to make numerous queries to steal a black-box online model. To enhance efficiency, attackers intentionally modify normal samples. In response to model leakage attacks, defenders focus on detecting queries involving anomalous samples to identify model theft behaviors. PRADA [2] is a defense technology that detects model theft attacks. It assesses whether a user is attempting model theft by analyzing the distance distribution between multiple sample features. The study found that the distances between randomly selected regular sample features roughly follow a normal distribution. In contrast, samples queried during model theft typically show clear signs of intentional modification, with the distance distribution differing significantly from the normal distribution. Statistical tests of multiple queries can detect abnormal query users. The feature distribution of query samples can also be used for detection. Kesarwani et al. [40] recorded users' query samples and examined their distribution in the feature space to evaluate the risk of model theft. Yu et al. [12] proposed that the feature distribution of normal samples differs significantly from that of artificially modified samples, and distinguishing between these distributions helps detect abnormal queries.

Since attackers typically make numerous queries to the target model in query behavior detection, their query behavior differs substantially from regular users. This difference can be used to defend against model and data leakage attacks to some extent. He et al. [39] suggested defending against member inference attacks based on user query behavior characteristics during the sample input stage to address data leakage attacks. When attackers

execute member inference attacks, they may need to query the target model multiple times. The model provider can limit the number of queries based on the frequency of user queries, increasing the cost for attackers to deploy member inference attacks.

As mentioned, defenders can detect model and data leakage attacks by identifying anomalous samples and query behaviors. However, this defense method is not highly targeted and may have a relatively high misclassification rate. Query control defense primarily operates during the training process of the attack model and is ineffective against pre-trained attack models. Furthermore, suppose attackers know that the target model uses query control defense. In that case, they can bypass it using various techniques, such as designing harder-to-detect anomalous samples or using virtual IP addresses to evade detection by the target model.

IV. Research Outlook

4.1. Development of Efficient Data Leakage Attack Techniques

The essence of data leakage attacks is that model parameters, output vectors, and other information are generated based on input samples. In other words, these data inherently contain information about the original data, meaning that any artificial intelligence model is at risk of data leakage and cannot fully resist such attacks. Therefore, the future development of data leakage attacks targeting AI models will focus on two main directions:

- Optimizing attack models to enhance their ability to extract information from the output vectors.
- Expanding attack scenarios and applying data leakage attacks to more scenarios, such as transfer and reinforcement learning.

Furthermore, privacy theft using the model's output often requires many queries to the target model. For instance, in model theft, training a substitute model requires thousands of queries due to the large parameter scale, non-linearity, and non-convexity of deep learning networks [10]. The large number of queries increases the attack cost and the risk of detection by defenders. Therefore, the primary focus for attackers is to make privacy theft more efficient. Researchers have made numerous attempts in this area, with the main idea being to develop a sample selection strategy that uses more representative samples to improve attack efficiency [41, 42], including methods like active learning [43, 44] and natural evolution strategies. In-depth research on attacks promotes the evolution of privacy protection and helps researchers gain a deeper understanding of AI models.

4.2. Development of Effective Defense Techniques Against Data Leakage Attacks

As discussed, data leakage attacks stem from model construction or usage output results, which implicitly contain specific private data. Thus, defenses against data leakage attacks can primarily develop in three directions:

- Obfuscating the output vector to reduce the information it contains.
- Obfuscating private data by adding specific noise to modify the original data, thus reducing the information in the model's inference results.
- Obfuscating the model's parameters by introducing privacy-preserving machine learning techniques, which encrypt internal model parameters, intermediate results, and output vectors to reduce the likelihood of information leakage.

However, the degree of obfuscation applied to different types of information must be carefully considered during defense construction. If the obfuscation is too minimal, the defense will not be effective, and attackers may still extract private data. On the other hand, if the obfuscation is too extensive, the model's output usability may

decrease, severely damaging its core functionality. Similar issues arise with other defense techniques, such as query control defenses. Strict query control rules will effectively prevent privacy data leakage but may make the normal user's experience cumbersome, potentially even misclassifying legitimate users as attackers. Therefore, in order to ensure privacy data is obfuscated while allowing the model to effectively and stably provide its original services, privacy leakage defense techniques must strike an effective balance between security and model usability. This is an aspect of defense technology that requires close attention in both practical applications and future development.

V. Conclusion

This paper summarizes and analyzes recent research on data security and privacy protection in artificial intelligence. Although many researchers have studied model output-based and gradient-based data leakage in AI systems, proposing various defense technologies, including model structure defenses, information obfuscation defenses, and query control defenses, there are still significant challenges in adequately addressing AI algorithm data security and privacy protection. These challenges arise due to the lack of interpretability inherent in deep learning algorithms. Compared to the well-established field of traditional data security, the proper resolution of data security and privacy protection issues in AI still requires further research.

References:

[1]. Z. Zhang, X. Wang, X. Li, *et al.*, "Adversarial attacks on machine learning models: A survey and research directions," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 1, pp. 1–17, 2023.
[2]. L. Zhu, Z. Liu, S. Han, *et al.*, "Efficient defenses against model stealing and adversarial attacks in deep neural networks," in *Proc. IEEE European Symp. Security Privacy*, 2021, pp. 512–527.
[3]. X. Lin, J. Hong, X. Li, *et al.*, "Advances in federated learning: Techniques, applications, and future directions," *IEEE Transactions on Artificial Intelligence*, vol. 1, no. 2, pp. 14–29, 2022.
[4]. A. R. Yadulla, V. K. Kasula, et al., “Enhancing Cybersecurity with AI: Implementing a Deep Learning- Based Intrusion Detection System Using Convolutional Neural Networks.” European Journal of Advances in Engineering and Technology, 2023, 10(12):89-98
[5]. Q. Tang, Y. Luo, Z. Li, *et al.*, "Attacks and defenses in machine learning: A comprehensive survey," in *Proc. IEEE Security Privacy Workshops*, 2022, pp. 81–90.
[6]. Y. Xu, X. Liu, H. Jiang, *et al.*, "Stealing machine learning models using adversarial gradient optimization," in *Proc. IEEE Symp. Security Privacy*, 2022, pp. 169–184.
[7]. K. Yang, H. Zhu, Y. Yang, *et al.*, "Exploiting model gradients for black-box attacks on neural networks," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 1012–1025, 2020.
[8]. N. Sharma, A. Aggarwal, and S. Singh, "A novel approach for object detection and localization using convolutional neural networks," in *Emerging Research in Data Engineering Systems and Computer Communications*, Springer, 2021, pp. 350–360.
[9]. M. Liu, C. Zhang, Z. Chen, *et al.*, "ResNet variations and their applications in deep learning: A review," *IEEE Access*, vol. 9, pp. 45567–45585, 2021.
[10]. J. F. Moreira, F. Ribeiro, and D. Almeida, "Model stealing through deep learning networks and adversarial training," in *Proc. 2019 Int. Joint Conf. Neural Networks (IJCNN)*, 2019, pp. 1–9.
[11]. Y. He, X. Zhang, J. Tang, *et al.*, "Electromagnetic side-channel attacks in machine learning systems: A review and a case study," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 2294–2307, 2021.

[12]. H. G. Yu, K. C. Yang, T. Zhang, *et al.*, "CloudLeak: Large-scale deep learning models stealing through adversarial examples," in *Proc. Network Distributed System Security Symp.*, 2020.

[13]. M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proc. 22nd ACM SIGSAC Conf. Computer Communications Security*, 2015, pp. 1322–1333.

[14]. V.K. Kasula. AI-driven banking: A review on transforming the financial sector. World Journal of Advanced Research and Reviews, 20(02):1461-1465, 2023, http://dx.doi.org/10.30574/wjarr.2023.20.2.2253.

[15]. R. Shokri, M. Stronati, C. Z. Song, *et al.*, "Membership inference attacks against machine learning models," in *Proc. 2017 IEEE Symp. Security Privacy*, 2017, pp. 3–18.

[16]. S. Yeom, I. Giacomelli, M. Fredrikson, *et al.*, "Privacy risk in machine learning: Analyzing the connection to overfitting," in *Proc. 31st IEEE Computer Security Foundations Symp.*, 2018, pp. 268–282.

[17]. A. Salem, Y. Zhang, M. Humbert, *et al.*, "ML-Leaks: Model and data independent membership inference attacks and defenses on machine learning models," in *Proc. 26th Annu. Network Distributed System Security Symp.*, 2019, pp. 24–27.

[18]. Y. H. Long, V. Bindschaedler, and C. A. Gunter, "Towards measuring membership privacy," *CoRR*, abs/1712.09136, 2017.

[19]. Y. H. Long, V. Bindschaedler, L. Wang, *et al.*, "Understanding membership inferences on well-generalized learning models," *CoRR*, abs/1802.04889, 2018.

[20]. S. Yeom, M. Fredrikson, and S. Jha, "The unintended consequences of overfitting: Training data inference attacks," *CoRR*, abs/1709.01604, 2017.

[21]. D. B. Sam, S. Surya, and R. V. Babu, "Switching convolutional neural network for crowd counting," in *Proc. 2017 IEEE Conf. Computer Vision Pattern Recognition (CVPR)*, 2017, pp. 4031–4039.

[22]. J. Kuha and C. Mills, "On group comparisons with logistic regression models," *Sociological Methods Research*, vol. 49, no. 2, pp. 498–525, 2020.

[23]. M. Pal, "Random forest classifier for remote sensing classification," *International Journal of Remote Sensing*, vol. 26, no. 1, pp. 217–222, 2005.

[24]. L. Song, R. Shokri, and P. Mittal, "Privacy risks of securing machine learning models against adversarial examples," in *Proc. 2019 ACM SIGSAC Conf. Computer Communications Security*, 2019, pp. 241–257.

[25]. A. Salem, A. Bhattacharya, M. Backes, *et al.*, "Updates-Leak: Dataset inference and reconstruction attacks in online learning," *arXiv preprint*, arXiv:1904.01067, 2019.

[26]. J. Hayes, L. Melis, G. Danezis, *et al.*, "LOGAN: Membership inference attacks against generative models," *PoPETs*, vol. 2019, no. 1, pp. 133–152, 2019.

[27]. M. Nasr, R. Shokri, and A. Houmansadr, "Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning," in *Proc. 2019 IEEE Symp. Security Privacy*, 2019, pp. 739–753.

[28]. K. Leino and M. Fredrikson, "Stolen Memories: Leveraging model memorization for calibrated white-box membership inference," *arXiv preprint*, arXiv:1906.11798, 2019.

[29]. L. Melis, C. Z. Song, E. De Cristofaro, *et al.*, "Exploiting unintended feature leakage in collaborative learning," in *Proc. 2019 IEEE Symp. Security Privacy*, 2019, pp. 691–706.

[30]. Z. B. Wang, M. K. Song, Z. F. Zhang, *et al.*, "Beyond inferring class representatives: User-level privacy leakage from federated learning," in *Proc. 2019 IEEE Conf. Computer Communications (INFOCOM)*, 2019, pp. 2512–2520.

[31]. B. Konda, V. K. Kasula, et al., "Homomorphic encryption and federated attribute- based multi-factor access control for secure cloud services in integrated space-ground information networks," International Journal of Communication and Information Technology, vol. 3, no. 2, pp. 33-40, 2022, https://doi.org/10.33545/2707661X.2022.v3.i2a.103.

[32]. L. G. Zhu, Z. J. Liu, and S. Han, "Deep leakage from gradients," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2019, pp. 14747–14756.
[33]. R. Shokri, M. Fredrikson, and C. Z. Song, "Membership inference attacks against machine learning models," *arXiv preprint*, arXiv:1610.05820, 2016.
[34]. J. Young and C. Mao, "Model privacy in deep learning with distributed processing," 2020.
[35]. M. Burke, J. Lee, and Z. Weng, "Design principles for machine learning in privacy-respecting artificial intelligence," *AI Journal*, 2020.
[36]. Z. Liu and X. Li, "Privacy-preserving machine learning algorithms: Encryption, homomorphic encryption, and differential privacy," *IEEE Transactions on Computational Intelligence and AI in Games*, 2020.
[37]. L. Yang, M. Tian, D. Xin, *et al.*, "AI-driven anonymization: Protecting personal data privacy while leveraging machine learning," *arXiv preprint*, arXiv:2402.17191, 2024.
[38]. L. D'Aliberti, E. Gronberg, and J. Kovba, "Privacy-enhancing technologies for artificial intelligence-enabled systems," *arXiv preprint*, arXiv:2404.03509, 2024.
[39]. Y. Z. He, G. Z. Meng, K. Chen, *et al.*, "Privacy and security challenges in deep learning systems: A 2024 survey," *arXiv preprint*, arXiv:2402.05643, 2024.
[40]. M. Kesarwani, B. Mukhoty, V. Arya, *et al.*, "Model extraction vulnerabilities in machine learning as a service platform," in *Proc. 2018 ACM SIGSAC Conf. Computer Communications Security*, 2018, pp. 698–710.
[41]. R. Sun, H. Ji, D. Liu, *et al.*, "Analyzing and defending against adversarial model stealing attacks," in *Proc. IEEE Symp. Security Privacy Workshops*, 2023.
[42]. W. J. Zhu, H. G. Liu, and Y. X. Ma, "Federated learning techniques for privacy-preserving AI models," *IEEE Transactions on Neural Networks and Learning Systems*, 2023.
[43]. G. Danezis, S. Hale, and A. Greenberg, "Design considerations for privacy-respecting AI-driven solutions," in *Proc. ACM SIGSAC Conf. Computer Communications Security*, 2022.
[44]. J. Kuang, W. Zhang, T. Yang, *et al.*, "Differential privacy techniques for machine learning systems," *IEEE Transactions on Knowledge and Data Engineering*, 2022.