A Survey On Privacy-Preserving Neural Architecture Search for Federated Learning

Srinitya Kondapally
Arizona State University
skonda29@asu.edu

Jinalee Raval
Arizona State University
jraval5@asu.edu

Dhrumil Shah
Arizona State University
dshah96@asu.edu

***Abstract*—Federated Learning (FL) is seen as an approach for distributed machine learning allowing model training while safeguarding data privacy. Neural Architecture Search (NAS) is also gaining popularity for streamlining the creation of network structures. However, the merger between NAS and FL is prone to various security attacks. This survey paper offers a look at the research landscape in privacy-preserving NAS for FL. Beginning with an overview of FL and NAS fundamentals, we emphasize the significance of privacy protection in FL and the rationale behind integrating NAS with privacy safeguards. We then explore the key challenges that emerge while implementing NAS in a federated environment and discuss how these concerns impact the NAS process. The later part of this paper provides an analysis of various frameworks and techniques put forth in existing literature, categorized by the challenges faced and techniques utilized to overcome them. The comparative assessments of these methods are included to outline their strengths and weaknesses. Lastly, we pinpoint challenges and future avenues for research in this domain, highlighting the necessity for exploration and innovation. This survey strives to act as a resource for researchers and professionals who are advancing privacy preservation for NAS for FL.**

I. Introduction

The rise of edge computing has changed how data is handled and how machine learning is conducted, with more data being generated and processed at the edge of the network. In this scenario, Federated Learning (FL) has become a framework allowing for collaborative model training across edge devices while keeping the data localized. This method addresses privacy concerns and helps overcome bandwidth limitations linked to transferring large amounts of data to centralized servers.

However, Federated Learning poses its challenges especially when it comes to selecting and optimizing model architectures. To overcome this challenge Neural Architecture Search (NAS) is used in combination with FL. NAS is an automated process that seeks to identify the neural network architecture for a specific task reducing the need for human expertise and manual adjustments.

Federated learning helps to mitigate privacy concerns. However, combining it with NAS introduces new challenges. These challenges may increase the chances of privacy breaches. Hence, the integration of Neural Architecture Search (NAS) with Federated Learning (FL) necessitates an additional layer of privacy protection. As a result, there is growing interest in combining NAS with privacy-preserving methods within the Federated Learning framework. The reason for combining these two technologies is to take advantage of NAS's ability to automatically design models while also ensuring that data privacy and search processes are protected. This method shows potential for creating customized models for edge computing applications while still prioritizing the security and confidentiality of the data.

In our study, we investigate the obstacles and possibilities that arise from merging NAS with privacy-preserving techniques in Federated Learning. We examine the different approaches suggested to tackle these obstacles and offer a summary of the research landscape in this area.

II. Background

This section comprises of brief overview of Federated Learning, Neural Architecture Search, and challenges in integrating them.

A. The basic concept of Federated Learning

For most machine learning approaches, the data is centralized into a common server giving rise to concerns with privacy, connectivity, and network latency. To mitigate these problems, Federated learning (FL) takes a decentralized approach where the computation is done locally on the client's private data without any data exchange between the clients. Through FL, the clients' train the model using their own data, and only the model parameters (or the model's updated parameters) are sent back to the server. When the server receives these parameters from various involved client devices, it aggregates these updates and improves the global model. The improved global model is then sent back to the devices and the process of local training repeats. Since the training is local and there is no raw data exchange with the server, federated learning helps in preserving the privacy of client data.

B. Brief overview of Neural Architecture Search (NAS)

Neural Architecture Search (NAS) is a specialized field in the domain of AutoML (Automated Machine Learning). It is a process of automating the design of artificial neural networks (ANNs). NAS helps in creating faster and optimal neural network architectures by searching for the best architecture. NAS has the following stages -

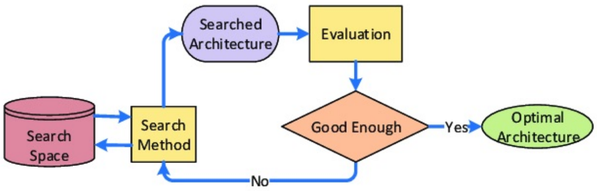

Figure 1. This figure is referred from [7] and represents Stages in Neural Architecture Search

1) Search Space: It is defined as the possible variations of architectures that the NAS is programmed can explore. It takes into account different types of layers, the number of layers, connections between neurons, and activation functions.

2) Search Method: This algorithm drives the exploration of the search space to find an architecture for the neural network. The most commonly used search methods are reinforcement learning, evolutionary algorithms, Bayesian optimization, and gradient-based methods.

3) Search Architecture: It describes the specific architecture found by the NAS algorithm during the search method. The architecture found is proposed as the most efficient and optimal structure for the neural network.

4) Evaluation and Optimal Architecture: The proposed architecture is measured for its performance. Its performance is measured by training the network on a data set and evaluated on metrics like error rate, accuracy etc. This evaluation helps determine if the suggested architecture is optimal based on whether the neural network meets the performance requirements. If the proposed architecture meets the performance requirements, then it is chosen as the structure of the neural network. Otherwise, a new search method is employed for searching, and the process repeats until an architecture found meets the performance requirements.

III. Implementing FL with NAS

Integrating Federated Learning (FL) and Neural Architecture Search (NAS) typically involves merging the training method of FL with the automated model design process of NAS. This combination enables the development of network structures to be trained on data distributed across multiple devices or locations while ensuring privacy. Below is a step-by-step detailed explanation of how FL and NAS can be effectively merged:

a) Defining the Search Space: The process begins with both FL and NAS defining a comprehensive exploration scope that includes all possible configurations of network architectures that the NAS algorithm will investigate. This initial step may involve decisions about the number of layers, the types of layers (such as convolutional or recurrent), activation functions, and other architectural elements.

b) Distributing the Initial Model: This step involves distributing an initial model to all participating clients. This model serves as the architecture from which NAS begins its exploration, with each participant receiving the model along with a set of adjustments or parameters for further experimentation.

c) Local Model Training and Evaluation: At this stage, each client performs training on their local data and evaluates the model's performance. Clients effectively test multiple architectural ideas by experimenting with various configurations within the NAS-defined search space.

d) Collecting Feedback: After training, clients transmit data back to a central server, providing detailed feedback on the performance of the architectures they tested. This feedback includes performance metrics such as accuracy, loss, and other relevant performance indicators that help analyze the efficiency of each tested architecture.

e) Aggregating Architectural Insights: A central server or a federated algorithm combines these performance metrics and identifies which architectural modifications result in improved model performance. The aggregation method is designed to ensure that sensitive data remains protected and cannot be reconstructed from the feedback given.

f) Refining the Search: The search space is refined based on the insights gained from the aggregated data. Well-performed models or architectural improvements are then redistributed to clients for further evaluation. This iterative process allows the architecture to improve continually, with the model becoming increasingly suited to the characteristics of the distributed datasets.

g) Finalizing the Model: Once the iterative process determines an architecture that performs across all clients' datasets, this architecture is finalized. It is then implemented as the global model for all participants, ensuring it is finely tuned to data characteristics and real-world deployment scenarios.

While Federated Learning inherently protects privacy, its integration with Neural Architecture Search introduces potential privacy risks. The exchange of data regarding architectures' effectiveness can reveal details about the model or the underlying training data. In the following section, we will explore the possible attacks that can pose privacy concerns while implementing Federated Learning with NAS.

IV. Privacy Attacks

Attacks that are possible on the model while implementing Federated Learning with NAS and the methods suggested to resolve them are summarized in Figure 2. This section presents the details of the attacks that cause potential privacy risks. A general pseudocode for performing any attack is as follows -

General Pseudocode -

1) Identify the target NAS FL model and its deployment infrastructure.
2) Gather information about the NAS FL process, including:
   a. Architecture search algorithms and parameters.
   b. Data sources and distribution across clients.
   c. Model aggregation mechanisms and privacy measures taken.

The next steps of the algorithm are respective to the type of attack to perform and will be discussed below in the paper.

A. *Inference Attack*

An Inference Attack refers to a security vulnerability where an attacker can extract information from a model's outputs without needing access to the original data or the model itself. The information might be sensitive like characteristics of the training data, properties about the individuals or entities represented in the data, or insights into the model's structure and parameters. This type of attack happens through responses to queries or predictions.

Model Queries: The attacker gives crafted data as input to the model and observes the resulting outputs. These inputs are tailored to examine the model and draw out information about its training data. Attackers may also leverage knowledge about the dataset or model, such as details on data distribution, model structure, or training methods, to enhance their attack efficacy.

Output Analysis: By studying the consistency and variability in the models' outputs, attackers can deduce patterns, relationships, or specific data points within the training dataset.

Potential Consequences:

a) Privacy Violations: If personal data is used for training, successful inference attacks can compromise privacy.

b) Security Risks: Inference exposes weaknesses in systems that rely on keeping the training data confidential.

A type of Inference attack that is possible on a model is Training-data Inference Attack.

1) Training-data Inference Attack: While falling under the broad category of inference attacks, Training-data Inference Attack focuses more on the actual data points used in the model's training. A Training data Inference Attack is a method employed by adversaries to uncover details of the dataset used while training a model. The attacker leverages the models' parameters, outputs, or behavior to deduce details or properties of the training data. This type of attack can result in privacy concerns, particularly when the training dataset includes personal data. The following is how the attack can happen - Model Memorization: Complex deep learning models have the tendency to memorize the specifics of the training data rather than generalize from patterns, leading to data leakage through model outputs.

Output Examination: Through analysis of outputs, such as decisions and probabilities across inputs, attackers can make informed assumptions about characteristics within the training data.

Gradient/Parameter Exposure: In scenarios like federated learning or model updates sharing gradients or parameters may inadvertently reveal data attributes. Attackers could utilize these exposed gradients to reconstruct the training dataset. The potential consequences of this attack are -

Privacy Violation: Disclosure of sensitive information, for example - health status, financial details, and individual preferences.

Security Risk: Compromising the safety measures of a system by understanding how it manages and processes data.

Pseudocode for performing Training-Data Inference Attack:

Steps 1 and 2 are to be followed as described in the general pseudocode above.

3) Analyze the gathered information to identify potential inference points.
4) Exploit the identified inference points to reconstruct or infer characteristics of the training data.
5) Use the inferred characteristics to extract sensitive information, such as obtaining details of the distributed global model, including structure, algorithm used, and the model parameters.

B. *Differentially Private threat*

Differential Privacy threats are a concern in learning setups, where several entities work together to train a model without revealing their original data. These threats target leveraging information (such as gradients) in the model training phase to deduce details about the training data. The danger emerges because, despite not sharing data, the shared gradients can occasionally divulge adequate information for an intruder to speculate about the data and potentially reconstruct it. This is how a differentially private attack happens -

Error Injection: By deliberately causing errors in the system, attackers can observe how the system handles these errors, potentially revealing information about the underlying data or the amount of noise applied.

Pseudocode for performing Differentially private attack on the model -

Steps 1 and 2 are to be followed as described in the general pseudocode above.

3) Analyze the gathered information to identify details that may compromise differential privacy.
4) Use observed patterns in noisy updates to guess the data distribution.
5) Use the inferred characteristics to achieve the intended purpose, like manipulating or selling

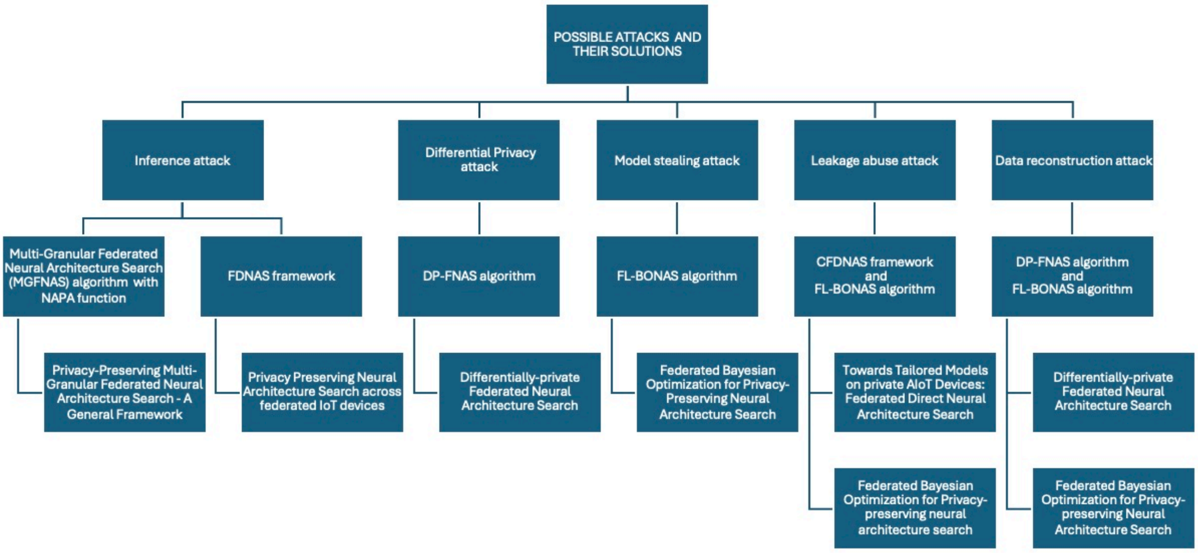
POSSIBLE ATTACKS AND THEIR SOLUTIONS
Inference attack
Differential Privacy attack
Model stealing attack
Leakage abuse attack
Data reconstruction attack
Multi-Granular Federated Neural Architecture Search (MGFNAS) algorithm with NAPA function
FDNAS framework
DP-FNAS algorithm
FL-BONAS algorithm
CFDNAS framework and FL-BONAS algorithm
DP-FNAS algorithm and FL-BONAS algorithm
Privacy-Preserving Multi-Granular Federated Neural Architecture Search - A General Framework
Privacy Preserving Neural Architecture Search across federated IoT devices
Differentially-private Federated Neural Architecture Search
Federated Bayesian Optimization for Privacy-Preserving Neural Architecture Search
Towards Tailored Models on private AIoT Devices: Federated Direct Neural Architecture Search
Differentially-private Federated Neural Architecture Search
Federated Bayesian Optimization for Privacy-preserving neural architecture search
Federated Bayesian Optimization for Privacy-preserving Neural Architecture Search

Figure 2. Summary of attacks and solutions for preserving privacy in FL with NAS. The second layer outlines the types of attacks. The third layer presents the algorithmic solutions and frameworks proposed to mitigate the risk of these attacks. The fourth layer presents research papers that proposed and implemented the solutions.

the data.

C. Model Stealing Attacks

Model stealing attacks are a type of security threat in federated learning systems. In these attacks, a malicious attacker tries to obtain sensitive information about the models being trained, such as their architecture or weight values. The attacker's goal is to replicate or gain unauthorized access to the models used in the federated learning process. The attacker utilizes the model's output results (such as its predictions) to deconstruct and replicate the model without needing to access the data or the model itself.

Pseudocode for performing Model stealing attack -
Steps 1 and 2 are to be followed as described in the general pseudocode above.

3) Analyze the gathered information and identify vulnerabilities such as weaknesses in model aggregation or update mechanisms.
4) Analyze intercepted model parameters and architecture details to clone the model.
5) Validate the cloned model against standard benchmarks to compare its performance with respect to the original model.

D. Leakage Abuse Attacks

A leakage abuse attack exploits weaknesses in a model's design that allow sensitive information to be accidentally revealed. An attacker tries to locate points where data is unintentionally leaking out and uses that to extract sensitive details rather than trying to infer the information directly. The leaks are analyzed and put together to gain insights about the model. By finding and abusing these leakage points, the security of the model is compromised, and private or confidential data can be accessed.

These are the potential steps through which attacker tries to abuse leakage:

a. Identifying Leakage Points - The attacker tries to identify leakage points like training data which reveals sensitive information about data to model, model parameters.
b. Gathering data - Collecting information from those leakage points.
c. Exploiting Information - The attacker can launch even more attacks with the gathered information, which can eventually result in more data leakage.

Pseudocode to perform a Leakage Abuse Attack: Steps 1 and 2 are to be followed as described in the general pseudocode above.

3) To identify potential vulnerabilities, analyze the gathered information and search for data leakage points via NAS, assess the lack of encryption, and identify insecure communication methods.
4) Exploit the identified vulnerabilities and leakage points to extract sensitive information or manipulate the NAS FL process by injecting malicious updates or biases, and by intercepting communication channels.
5) Next step is to use the extracted information in order to achieve the attacker's objectives like compromising privacy, manipulating the model, and favoring certain clients.

E. Data Reconstruction Attacks

A data reconstruction attack is a type of attack where an adversary attempts to recreate the original dataset from aggregated or transformed data (like model results or gradients from training sessions). This attack exploits the model's predictions, decisions, or other outputs to infer information about the underlying training data or individual data points.

These are the potential steps through which the attacker tries to reconstruct data:

a. Access to Model Outputs - The attacker tries to access model predictions, confidence scores, and inference information.
b. Analysis of Model Outputs - The attacker then tries to identify patterns and correlations in the output to decipher sensitive information.
c. Inference of Sensitive Information - The attacker tries to reconstruct data points or dataset using identified correlation and characteristics.

Pseudocode to perform a Data reconstruction Attack:
Steps 1 and 2 are to be followed as described in the general pseudocode above.

3) Analyze the federated learning process to identify potential vulnerabilities and leakage points like lack of encryption and data leakage through model gradients.
4) Develop a data reconstruction strategy based on the identified vulnerabilities, which may involve intercepting model updates, gradient exchanges in the federated network, leveraging available knowledge to improve reconstruction and applying statistical techniques to get sensitive data.
5) Satisfying the attacker's motive of compromising privacy, extracting insights into client behaviours and characteristics and exploiting vulnerabilities to attack the model further.

V. Solutions

This section presents corresponding algorithmic solutions or frameworks for the attacks described in the previous section.

1) ***Solution for Inference attack:*** The research paper named "Protecting Privacy in Neural Architecture Search Among Federated IoT Devices" introduces the Federated Direct Neural Architecture Search (FDNAS) framework. This framework is proposed to address security issues related to inference attacks in learning setups involving devices with resources. The core of the framework involves combining architecture search (NAS) with federated learning. Unlike machine learning models that are trained centrally with access to all data, federated learning allows model training across devices. Each device independently trains a model using its data and shares only model parameters or updates, thus preventing data exposure. This decentralized approach significantly reduces the risk of data exposure and minimizes vulnerability to inference attacks by sharing model parameters instead of raw data. The FDNAS framework implements principles for conducting NAS on device data without relying on tasks or centralized data aggregation. By conducting NAS on devices in a privacy-preserving manner, FDNAS ensures that neural architectures are optimized for each device's hardware and data distribution without the need to share information. This method inherently protects against inference attacks by keeping training data local and confidential.

a) Creating hardware NAS: The FDNAS project deals with developing architectures from non-identical and varied data among IoT devices. This is essential for safeguarding privacy in federated setups where data distribution differs widely across devices. By incorporating hardware NAS that considers the diversity in data distribution and hardware constraints, FDNAS enhances model performance while protecting data privacy.

b) Validation through experiments: By conducting experiments on datasets, the FDNAS framework demonstrated its ability to achieve a balance of accuracy and efficiency. By validating the framework's effectiveness in preserving privacy during NAS across devices, the approach confirms its potential to mitigate inference attacks by ensuring that architecture search and model training processes do not disclose information. Essentially, the FDNAS framework tackles concerns related to inference attacks by adopting a federated learning approach for NAS enabling private architecture searches on devices. This method reduces the risk of exposing data and inferences to threats, thereby enhancing privacy and security in machine learning processes within distributed environments.

2) ***Solution for Training-data Inference Attack:*** The paper titled "Privacy Preserving Multi Granular Federated Neural Architecture Search – A General Framework" addresses privacy concerns in model training and architecture search. This framework merges federated learning with neural architecture search (NAS) to enable model training and without the necessity of sharing raw training data among participants. The Multi Granular Federated Neural Architecture Search (MGFNAS) streamlines the process of identifying the model architecture in a federated and privacy-preserving manner. It explores network architectures at micro and macro levels without centralizing or sharing data among clients or a central server. The framework introduces an aggregation function known as Network Architecture Probabilistic Aggregation (NAPA) to combat privacy issues related to training data inference attacks. NAPA treats network architectures as graphs modeling graph structures across clients using probabilistic distributions. A global model is generated by sampling from these distributions in an exploration-exploitation approach that aggregates architectures without accessing or disclosing raw training data. The framework minimizes the likelihood of inference attacks on training data by exchanging model parameters or combining updates between clients and a central aggregator. This approach guarantees that sensitive information in the training data is kept confidential and decentralized, enabling the development of advanced neural network architectures while protecting data privacy.

*3) **Solution for Diffentially Private Threat:*** The proposed “Differentially-private Federated Neural Architecture Search (DP FNAS)” framework presents a defense strategy against privacy violations by integrating privacy techniques into the federated learning process focusing on neural architecture search (NAS). Its objective is to enable parties (clients) to collaborate on finding a suitable structure while ensuring the confidentiality of each party’s data. Here’s how DP FNAS addresses the issue:

a) Introducing Noise for Gradient Concealment: DP-FNAS safeguards privacy by introducing noise into the gradients computed by each party before sending them to the server. This step inhibits attempts to infer or analyze any party’s data from the shared gradients. By masking the gradients, the framework ensures that the essential information for learning is preserved while protecting it from external access.

b) Validation: Through experiments conducted on the CIFAR-10 dataset, the DP-FNAS framework has demonstrated its capability to effectively search for high-performance neural architectures without compromising the individual party’s privacy. The system allows for the exploration of varying levels of privacy protection by adjusting the amount of noise and helping to achieve a balance between privacy and accuracy. To sum up, DP-FNAS introduces a method for discovering architectures within a federated learning setting and allows collaborative learning without jeopardizing the parties’ data.

*4) **Solution for Model Stealing Attacks:*** In the research paper titled “Federated Bayesian Optimization for Privacy-preserving Neural Architecture Search”, a novel approach named FL-BONAS is suggested to safeguard the confidentiality of the network. This approach aims to prevent the disclosure of information and mitigate the risk of model theft. A method, Bayesian Optimization, is used to explore the neural network architecture without testing every possibility, thus saving significant time and computational resources. In the FL-BONAS approach, every device works on training a model using its own data. This data is not shared with others ensuring that everyone’s data remains confidential. FL-BONAS not only preserves data privacy but also prevents unauthorized model replication. In this framework, participants work together to develop a “surrogate model” without sharing weight parameters across various model designs. Then each participant individually seeks their architecture using optimization without disclosing their local data or specific architectures. By sharing the parameters of these surrogate models, which do not divulge any individual’s data specifics or neural network architecture details, FL-BONAS cleverly avoids exposure of necessary information to steal models. In this manner, FL-BONAS maintains secrecy around network architecture searches, preventing model theft while still enabling the participants to benefit from enhancements and insights.

*5) **Solution 1 for Leakage Abuse Attacks:*** The research paper titled “Towards Tailored Models on private AIoT Devices: Federated Direct Neural Architecture Search ”presents an approach to address this challenge through an innovative concept named Collaborative Direct Neural Architecture Search (CDNAS). The fundamental idea is that it merges the capabilities of architecture search (NAS) with the privacy-protecting features of federated learning. This implies that rather than sharing their data or sensitive model details that could potentially result in data exposure, each device collectively learns and enhances the network structure that benefits all the devices while keeping their personal information secure.

This is how CDNAS Counters Data Leakage Threats -

a) Decentralized Training via Federated Learning: Every AIoT (Artificial Intelligence of Things) device contributes to the learning process using its dataset. By doing this, the original data remains on the device itself, greatly minimizing the risk of data leaks.

b) Direct Neural Architecture Search (NAS): Here, the devices actively explore the network design tailored for their specific data and limitations rather than simply implementing a pre-defined model. This search is conducted directly, without relying on proxy data or simplified tasks, which could reduce the model’s effectiveness or unintentionally expose information about the data.

c) Inspired by Meta Learning: The framework follows a meta-learning approach training a SuperNet across all devices and then customizing device models (SubNets) based on each device’s data and hardware needs. This method leverages the diversity of data and computational resources across devices, ensuring effective learning is achieved without the need to share data.

d) Ensuring Privacy in Model Adaptation: FDNAS enables architecture search while considering the hardware constraints and privacy requirements of AIoT devices. This guarantees that the search process itself does not pose a risk to privacy, as the models’ detailed operations and the influencing data remain isolated within each device.

Hence, FDNAS navigates federated learning to minimize vulnerability to information leakage attacks. By keeping data within the environment and focusing collaborative efforts on finding the structure, the neural network guarantees that the combined knowledge of all devices can be utilized while still maintaining privacy.

*6) **Solution 2 for Leakage Abuse Attacks**:* The research paper "Federated Bayesian Optimization for Privacy-preserving Neural Architecture Search" introduces a method called FL-BONAS to tackle this issue. With FL-BONAS, each participant can contribute to a network model without exposing their data or insights.

Here's how FL-BONAS addresses the problem -

a) Using surrogate model: By using Surrogate Models to share data (such as model weights or gradients that could reveal private information), participants in the FL-BONAS system share an ensemble surrogate model. This shared model serves as an intermediary by predicting architecture performance without needing access to original data.

b) Efficient Search Through Bayesian Optimization: By leveraging optimization techniques, FL-BONAS efficiently navigates architectures without trial and error. It uses the Bayesian Optimization method to search for architecture and hence reduces the need for sharing detailed model parameters. This approach ensures performance and prevents any single model from dominating the search process, which could potentially expose more information. It also helps to reduce the chances of someone reverse engineering the data of participants from the shared data pool. Using federated averaging (FedAvg) to update model parameters, FL-BONAS ensures that learning benefits from the contributions of all participants without centralizing data or exposing individual learning processes. Essentially, FL-BONAS effectively prevents Leakage Abuse Attacks by creating a scenario where participants collaborate in finding an architecture without compromising data privacy. It assures each contributor that their input is valuable and secure, hereby upholding the confidentiality of their data.

*7) **Solution 1 for Data Reconstruction Attacks**:* "Differentially-private Federated Neural Architecture Search (DP-FNAS)" paper provides a DP-FNAS framework that enhances data privacy, particularly by injecting random noise into the gradients. Let's dive deeper into how this framework works -

a) Adding Noise to the gradients: Each participant holds their data in the federated learning setup. As they collaborate to train a model, they share gradients of the data. DP-FNAS ensures these gradients are maintained private by adding random noise. This addition of noise prevents anyone from extracting specific details about the data while still allowing the model training to complete. While adding noise, it is ensured that the exchanged information complies with privacy standards measured by privacy benchmarks. This implies that the shared updates are designed to safeguard individual data points from being identified or reconstructed, ensuring that the gradients remain obscure to maintain the confidentiality of the data.

b) Balancing Privacy and Performance: The framework strikes a balance between introducing noise for privacy protection and ensuring that the shared gradients retain enough integrity for effective model training and performance.

*8) **Solution 2 for Data Reconstruction Attacks**:* Just like how DP-FNAS provides an approach to tackle the data reconstruction attack, the FL-BONAS approach discussed in the above sections also helps in solving it. FL-BONAS method uses surrogate models to share their data. That means, instead of sharing the original model's data, surrogate models' data is being shared. As a result, the adversary doesn't have access to the original data and hence cannot recreate it. Thus, FL-BONAS resolves the challenge of data reconstruction and also safeguards the privacy of original data.

VI. Future Work

Looking ahead, it's vital to keep exploring solutions that improve the confidentiality and integrity of NAS in FL. Future research directions may involve a combination of these techniques as individual techniques do not provide holistic solutions of NAS and FL integration. For instance, MGFNAS only provides a solution for training data inference attacks but this framework can be prone to other attacks and vulnerabilities. For further advancements integration of FL and NAS can be deployed across different sectors, like healthcare and finance. The Internet of Things (IoT) presents obstacles that, once overcome, will facilitate the integration of Federated Learning (FL) and Neural Architecture Search (NAS) in applications that prioritize privacy. This progress will significantly enhance the development and training of distributed machine-learning models.

VII. Conclusion

In this study, we've delved into the combination of Neural Architecture Search (NAS) and Federated Learning (FL), with a focus on Privacy Preserving Techniques. Bringing NAS into FL shows promise in the creation of high-performance network structures while respecting the privacy requirements within decentralized learning setups. However, this integration also presents challenges in safeguarding data privacy and the architectural search process.

We have examined frameworks and strategies put forth in research that tackle some of these privacy challenges. The approaches discussed above strive to find a balance between privacy protection, model effectiveness, performance, and computational efficiency. Despite advancements made, there are still issues like scalability, resilience against attacks, and adherence to data security laws.

References

1 Chunhui Zhang, Xiaoming Yuan, Qianyun Zhang, Guangxu Zhu, Lei Cheng, and Ning Zhang, *Privacy-Preserving Neural Architecture Search Across Federated IoT Devices*, IEEE 20th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), 2021, DOI: 10.1109/TrustCom53373.2021.00203.

2 Ishika Singh, Haoyi Zhou, Kunlin Yang, Meng Ding, Bill Lin, Pengtao Xie, *Differentially-private Federated Neural Architecture Search*, Proceedings of the 37th International Conference on Machine Learning, Vienna, Austria, PMLR 108, FL-ICML'20 Workshop, 2020.

3 Author(s) of Zijie Pan, Li Hu , Weixuan Tang , Jin Li , Yi He , and Zheli Liu, *Privacy-Preserving Multi-Granular Federated Neural Architecture Search – A General Framework*, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 35, NO. 3, MARCH 2023

4 Toward Tailored Models on Private AIoT Devices: Federated Direct Neural Architecture Search, *Chunhui Zhang , Xiaoming Yuan , Member, IEEE, Qianyun Zhang , Member, IEEE, Guangxu Zhu , Lei Cheng , and Ning Zhang , Senior Member, IEEE*, IEEE INTERNET OF THINGS JOURNAL, VOL. 9, NO. 18, 15 SEPTEMBER 2022.

5 Shiqing Liu , Xilu Wang , Yaochu Jin *Federated Bayesian Optimization for Privacy-preserving Neural Architecture Search*, 2023 IEEE Congress on Evolutionary Computation (CEC) — 979-8-3503-1458-8/23.

6 Ling, Chelsea Xinyi, et al.*Leakage Abuse Attacks on Encrypted Columns Using LP-Optimization* SpringerLink, Springer Nature Singapore, 1970, link.springer.com/chapter/10.1007/978-981-99-8369-849.

7 Santra, Santanu Hsieh, Jun-Wei Lin, Chi-Fang. (2021) *Gradient Descent Effects on Differential Neural Architecture Search: A Survey*. IEEE Access, vol. PP, no. 99, pp. 1-1, 2021. doi: 10.1109/ACCESS.2021.3090918.