UNIVERSITY OF OKLAHOMA
GRADUATE COLLEGE

UNCOVERING THE POTENTIAL OF FEDERATED LEARNING: ADDRESSING ALGORITHMIC AND DATA-DRIVEN CHALLENGES UNDER PRIVACY RESTRICTIONS

A DISSERTATION
SUBMITTED TO THE GRADUATE FACULTY
in partial fulfillment of the requirements for the
Degree of
DOCTOR OF PHILOSOPHY

By

ELAHEH JAFARIGOL
Norman, Oklahoma
2023

UNCOVERING THE POTENTIAL OF FEDERATED LEARNING: ADDRESSING ALGORITHMIC AND DATA-DRIVEN CHALLENGES UNDER PRIVACY RESTRICTIONS

A DISSERTATION APPROVED FOR THE
GALLOGLY COLLEGE OF ENGINEERING

BY THE COMMITTEE CONSISTING OF

Dr. Theodore Trafalis, Chair

Dr. Talayeh Razzaghi

Dr. Dimitrios Diochnos

Dr. Zahed Siddique

Acknowledgements

First and foremost, I extend my deepest gratitude to my advisor Dr. Theodore Trafalis, whose expertise, patience, and unwavering belief in my potential have been the cornerstones of my doctoral experience. Through his guidance, I have learned to think critically, question deeply, and strive for a higher level of scholarly excellence. I am endlessly grateful for his mentorship and support throughout this journey.

I also wish to express my sincere gratitude to my dissertation committee members, Dr. Dimitrios Diochnos, Dr. Talayeh Razzaghi, and Dr. Zahed Siddique, whose diverse perspectives, constructive feedback, and expertise have enriched this work. I deeply value the time and effort they have dedicated to ensuring the quality and relevance of my research.

Without the ongoing support of a few special people in my life, completing a Ph.D. would seem unattainable.

Kianoush Souri, whose understanding, patience, and belief in me have been a source of strength. He has been my anchor, grounding me in moments of doubt and propelling me forward with his confidence. I thank him for his unwavering support and love.

A special mention must be made to my dear friend, Sara Ghaffari, who has been my sounding board and a constant source of encouragement throughout the many highs and lows of this academic endeavor.

Finally, my heartfelt thanks go out to my parents, whose love and sacrifices have been the foundation upon which all my achievements stand. They taught me the values of hard work, perseverance, and resilience, and this achievement is as much theirs as it is mine.

Table of Contents

List Of Tables

List Of Figures

List of Symbols

Symbol	Description
η	Learning rate
δ	Evaluation value for the k-th client
β	Stability coefficient
ϵ	Privacy loss (leakage)
δ	Probability of leakage
μ	Mean
θ	Joint probability distribution
λ	Regularization coefficient
ρ	Correlation coefficient
α	Random variable (trainable parameter)
γ	Rademacher random variable
σ	Standard deviation
σ_i	Standard deviation of Gaussian noise of i^{th} player
σ_s^2	Signal variance
σ_n^2	Noise variance
g_k	Gradient of the k-th client
n_k	Total number of sample points on client k
x_i	Input vector i

y_i	Label
f	Function
h	Algorithm
d	Distance
t	Time
x	Vector
z	Random variable from Gaussian distribution
z'	Vector of Gaussian noise
K	Total number of clients
S	Set
S'	Set
H	Hypothesis
L	Loss function
M	Randomized mechanism
Q	Set of outcomes of the mechanism M
N	Number of samples taken for computation
T	Period of time
G	Generator
D	Discriminator
E	Expected value over all real data instances
W	Wasserstein distance
G_i	Noise layer

$\mathscr{R}$	Empirical Rademacher complexity
$\hat{R}(H)$	Empirical risk
$R(H)$	True risk
P_s	Signal power
P_n	Noise power
$s[n]$	Discrete signal
$n[n]$	Discrete noise
P_r	Distribution of real samples
P_g	Distribution of generated samples
D_{fake}	Average discriminator output of generated samples
D_{real}	Average discriminator output of real samples

Abstract

Federated learning is a groundbreaking distributed machine learning paradigm that allows for the collaborative training of models across various entities without directly sharing sensitive data, ensuring privacy and robustness. This Ph.D. dissertation delves into the intricacies of federated learning, investigating the algorithmic and data-driven challenges of deep learning models in the presence of additive noise in this framework. The main objective is to provide strategies to measure the generalization, stability, and privacy-preserving capabilities of these models and further improve them. To this end, five noise infusion mechanisms at varying noise levels within centralized and federated learning settings are explored. As model complexity is a key component of the generalization and stability of deep learning models during training and evaluation, a comparative analysis of three Convolutional Neural Network (CNN) architectures is provided. A key contribution of this study is introducing specific metrics for training with noise. Signal-to-Noise Ratio (SNR) is introduced as a quantitative measure of the trade-off between privacy and training accuracy of noise-infused models, aiming to find the noise level that yields optimal privacy and accuracy. Moreover, the Price of Stability and Price of Anarchy are defined in the context of privacy-preserving deep learning, contributing to the systematic investigation of the noise infusion mechanisms to enhance privacy without compromising performance. This research sheds light on the delicate balance between these critical factors, fostering a deeper understanding of the implications of noise-based regularization in machine learning. The present study also explores a real-world application of federated learning in weather prediction applications that suffer from the issue of imbalanced datasets. Utilizing data from multiple sources combined with advanced data augmentation techniques improves the accuracy and generalization of weather prediction models, even when dealing with imbalanced datasets. Overall, federated learning is pivotal in harnessing decentralized datasets for real-world applications while safeguarding privacy. By leveraging noise as a tool for regularization and privacy enhancement, this research study aims to contribute to the development of robust, privacy-aware algorithms, ensuring that AI-driven solutions prioritize both utility and privacy.

Chapter 1

Introduction & Research Objectives

1.1 Federated Learning

In the modern world, where governments and private companies frequently use data for strategic planning, decision-making, policies, and even services, privacy is a serious concern. Privacy is the individual's autonomy in collecting, storing, sharing, and analysis of personal data. Privacy violations can have serious personal and social implications for vulnerable populations, causing discrimination, surveillance, and other potential harms. Emerging technologies in data generation, storage, and analysis raise new concerns about individuals' right to privacy in the machine learning domain. Motivated by the Fundamental Law on Information Reconstruction, the researchers in Microsoft Research Lab focused on designing a holistic approach to preserving privacy in the statistical learning of individuals' data. However, without a structured definition of privacy, evaluating the privacy-preserving methods was subject to failure. An intuitive definition of privacy is the one by Gavison [1].

Definition 1. *Privacy is the protection from being brought to the attention of others.*

As governments and organizations strive to harness the potential knowledge and value in the data, reliable and trustworthy algorithms become crucial. Researchers encourage policymakers to incorporate privacy as a human right in the processes and establish privacy protection mechanisms that ensure individuals' safety in the age of artificial intelligence [2, 3]. The most recent update of the National Artificial Intelligence

R&D Strategic Plan released by the White House in 2023 highlights the importance of federated learning approaches amid growing concerns around data privacy and security. [1] The report outlines the long-term investment plans in responsible AI research, magnifying the need for advances in privacy-preserving data sharing and addressing the existing challenges in federated learning.

The term federated learning was introduced in the paper published based on the results of a research project carried out at Google in 2016 for text input prediction on mobile devices [4]. The authors propose a deep learning model for federated learning and conduct an empirical study on the model using different model structures on benchmark datasets for image classification and language processing applications. The learning process is designed for a group of devices referred to as clients, coordinated by a central server, also known as a service provider [5]. Federated learning allows multiple entities to work collectively without sharing sensitive data.

Keeping the data decentralized reduces the risk of leakage and data breach [6]. Distributed machine learning algorithms create an environment where data storage and training happen on the group of distributed machines. What differentiates federated learning from other learning algorithms is its ability to maintain privacy [7, 8, 9]. By avoiding the need to store data in a single location, we can harness multiple data sources, safeguard individual privacy, and reduce data storage expenses while achieving high accuracy levels [10].

The ideas behind federated learning have been around for decades, but thanks to the abundant data that is available to us and advances in computation power, we are able to efficiently train machine learning models on a network of decentralized data sources. Advances in the field of machine learning, especially deep learning, allow us

[1] National Artificial Intelligence Research and Development Strategic Plan https://www.nitrd.gov/national-artificial-intelligence-research-and-development-strategic-plan-2023-update/

to build powerful models that can accurately analyze different data types and provide valuable insights.

Federated learning is the process of training data locally and improving the global model. In the federated learning framework, as shown in Figure 1.1, the data is stored in local data centers, and limited information required for the learning task is privately communicated with the central server.

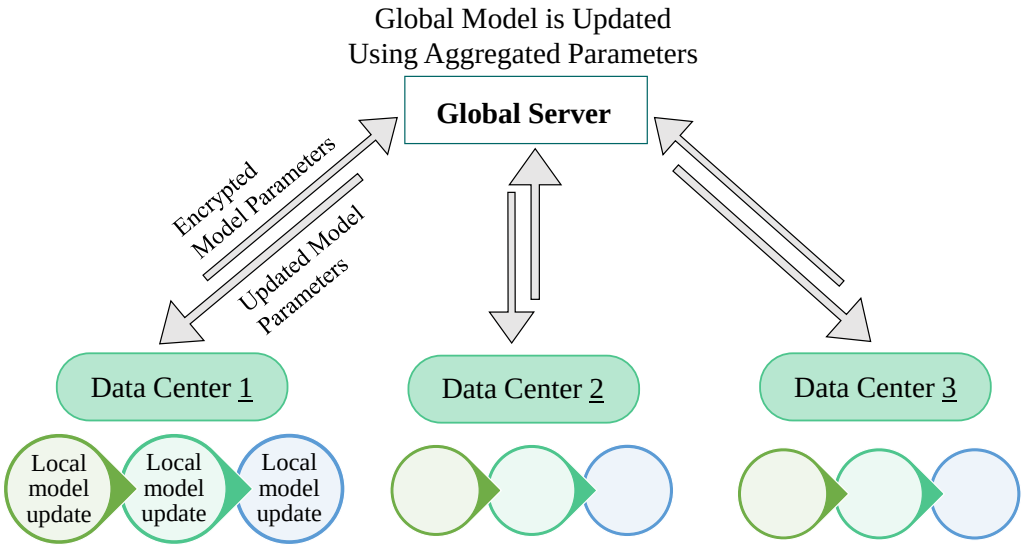


Figure 1.1: The Federated Learning Framework

This architecture is called a client-server design. If the data centers are both responsible for the task of storage and aggregation, the architecture is called peer-to-peer. Suppose n is the total number of sample points. In that case, K is the total number of clients, n_k is the total number of sample points on client k, and η is the learning rate. The goal of federated learning is to minimize the objective function f, also known as

the loss function, where $f_k(\delta)$ is the loss function, and δ is the evaluation value for the k^{th} client:

$$f(\delta) := \sum_{k=1}^{K} \frac{n_k}{n} f_k(\delta) \tag{1.1.1}$$

In this equation, δ is updated after each iteration until we reach the optimal solution or the number of iterations set as the stopping criterion is satisfied. This optimization problem is solved using a federated Stochastic Gradient Descent (SGD) method, which is described in algorithm 1. This algorithm shows that the gradient steps are taken by each client, and the model parameters are calculated as $\delta_{t+1} \leftarrow \delta_t - \eta g_k$.

Algorithm 1 Federated Stochastic Gradient Descent

Require: $\delta_t, \nabla f(\delta_t), n, n_k, \eta$
Ensure: $C < 1$
▷ a subset of clients is selected at each round
δ_t:= the current state of the evaluation value
while $f(\delta_t) \neq$ optimal **do**
$g_k = \nabla f_k(\delta_t)$
▷ g_k := the gradient of client k
$\delta_{t+l} \leftarrow \delta_t - \eta \nabla f(\delta_t) = \delta_t - \sum_{k=1}^{K} \frac{n_k}{n} g_k$
▷ Aggregation of client gradients to create a new model at the central server
end while

1.2 Applications

Federated learning has been applied to various fields when the data is sensitive and scattered across multiple servers or devices. In particular, federated learning is a compelling approach in healthcare, Internet of Things (IoT), and crisis management, where information limitation is an issue.

1.2.1 IoT

Industry 4.0 is the era of interconnected physical and digital technologies. Through the fourth industrial revolution, smart operations evolved, and the demand for informed and data-driven solutions increased. In the digital world, data is constantly generated in texts, images, and measurements from thousands of sensors and devices, which require powerful systems to perform extensive computations for data processing. Smart cities, cloud-based technologies, edge computing, and IoT need reliable, secure, real-time analysis tools. This has increased the demand for systems that can address scalability, interoperability, resource limitations, and privacy issues [11, 12].

Nguyen et al. [13] surveyed the application of federated learning to leverage the data on IoT devices for smart cities and industries, leading to advances in healthcare, transportation, and Unmanned Aerial Vehicles. Khan et al. [14] surveyed different aspects of federated learning for IoT applications. The authors compared and evaluated the methods from robustness, quantization, sparsification, scalability, security, and privacy perspectives. Varlamis et al. [15] used federated learning to find energy-saving solutions based on sensor data. Federated learning offers several benefits when applied to IoT applications such as:

- Privacy preservation: This is the primary advantage of federated learning. IoT devices often collect sensitive personal information, such as health data or home automation preferences, and federated learning allows these devices to train machine learning models without exposing individuals to possible harm and misuse of data.

- Reduced data transfer: IoT devices are often resource-constrained, making it inefficient and costly to transmit large volumes of data to a central server. By

keeping data on the edge, federated learning reduces the need for extensive data transfer and lowers communication overhead.

- Edge computing: Federated learning fits well with edge computing, where data processing occurs closer to the source (IoT devices) rather than in a data center. Therefore, minimizing latency and improving real-time decision-making which is crucial for applications like autonomous vehicles and industrial automation.

Some of the challenges and limitations of federated learning in this domain are:

- Heterogeneity: IoT devices come in various shapes, sizes, and capabilities, making them challenging to harmonize for federated learning. Ensuring that models can be trained effectively across diverse IoT ecosystems is a significant challenge. In federated learning, some devices may also have more data or contribute more frequently to model updates than others. Managing the data across IoT devices can affect the fairness and accuracy of the learned models.

- Communication overhead: Aggregating model updates from numerous IoT devices can be challenging, especially when dealing with intermittent connectivity, device failures, or adversarial behavior. Robust aggregation methods are needed to handle these scenarios.

- Computational overhead: Federated learning can be computationally intensive, which may be problematic for resource-constrained IoT devices. Balancing model training with energy efficiency and processing power is a limitation that needs to be addressed.

- Scalability: As the number of IoT devices increases, managing federated learning across the network becomes more challenging.

Federated learning holds great promise for IoT applications. However, addressing challenges and overcoming limitations is essential to fully harness the potential of federated learning in the rapidly expanding IoT ecosystem.

1.2.2 Healthcare

Federated learning is a promising approach for learning from healthcare data, which is highly regulated and cannot be openly shared with the public. Therefore, if privacy is ensured, it can significantly benefit from utilizing artificial intelligence and machine learning to move towards personalized healthcare and computer-aided diagnosis. Federated learning creates a global model of decentralized data, such as the data from hospitals, labs, and clinical trials without direct access to the data [16]. For instance, Feki et al. [17] utilized a federated learning approach to build a powerful model to classify COVID-19 X-rays based on data collected from multiple institutes. Rieke et al. [18] explored the existing literature on federated learning for healthcare with the challenges and open problems in digital healthcare. Some of the benefits of federated learning in healthcare applications are:

- Privacy preservation: Healthcare data is highly sensitive and subject to strict privacy regulations. Federated learning enables healthcare institutions to collaborate on model training without sharing raw patient data, preserving privacy and compliance with regulations like HIPAA.

- Large-scale data utilization: Without privacy concerns, healthcare organizations can tap into a vast pool of data from various sources, including hospitals, clinics, wearable devices, and electronic health records, to advance machine learning and computer-aided diagnosis.

- Personalized medicine: by leveraging patient-specific data, personalized treatment plans become more viable, leading to more effective and tailored healthcare interventions.

However, some of the challenges and limitations are:

- Heterogeneity: Health data comes from various hospitals, clinics, personal health monitoring devices, and more. Differences in data formats, quality, completeness, and availability pose challenges for integrating the data and achieving accurate and reliable results.

- Regulatory compliance: Healthcare is heavily regulated, with different regions and countries having their own sets of rules and standards. Navigating regulatory compliance while implementing federated learning can be complex and time-consuming.

- Bias and fairness: Federated learning may inherit biases present in the data from participating institutions, potentially leading to biased or unfair model outcomes.

- Model quality control: Ensuring the quality and consistency of models across different institutions can be challenging. Mechanisms for model monitoring, validation, and quality control are essential to maintain high standards of care.

Federated learning is a promising approach for healthcare applications, and addressing the challenges can have significant impacts on healthcare operations.

1.2.3 Crisis management:

There is a growing interest in machine learning algorithms for weather applications and natural disasters. Leveraging large data sets from multiple resources facilitates collaborative learning from data collected at different geographic regions, allowing for more

powerful and precise models. Bypassing the risks of data sharing encourages meteorological institutions across the countries to harness the power of machine learning while ensuring privacy and efficient utilization of resources, resulting in more accurate and timely predictions. On a larger scale, when dealing with natural disasters and emergencies, federated learning allows government agencies, international organizations, local support groups, and communities to collaborate effectively without privacy restrictions caused by traditional centralized learning. Federated learning provides the framework for secure communication of knowledge, resulting in early detection, risk assessment, and effective emergency response strategies using more accurate and robust models. The benefits of federated learning in crisis management are:

- Privacy Preservation: In crisis management scenarios, sensitive and critical data may be involved, such as location data, medical records, or disaster response plans. Federated learning enables multiple entities to collaborate on model training without sharing raw data, preserving privacy and security.

- Real-time updates: Crisis management requires quick decision-making based on the latest information. Federated learning enables real-time model updates as new data becomes available, ensuring that decision support systems remain current and effective during rapidly evolving situations.

- Resource efficiency: Crisis response often involves distributed teams and resources. Federated learning leverages the computing power of edge devices and distributed data sources, minimizing the need for centralized data storage and processing resources.

- Customization for local conditions: Different regions may have unique characteristics and needs during a crisis. Federated learning allows for localized model

customization, ensuring that solutions are tailored to specific conditions and requirements.

Despite its benefits, the challenges and limitations of this approach are:

- Data availability: Federated learning relies on the availability of data. During a crisis when data may be sparse, incomplete, or unreliable due to infrastructure damage or connectivity issues, training reliable models becomes challenging.
- Heterogeneity: In addition to data availability, collecting data from multiple sources such as governmental and private organization databases, social media, and on-site observations results in inconsistencies in data formats that need special attention during model training and interpretation of results.
- Model drift: In dynamic crisis situations, data distributions can change rapidly, causing model drift which requires the model to be continuously updated and retrained.
- Communication overhead: Connectivity issues and limitations on bandwidth are possible issues during a crisis. Allocating adequate resources and careful planning prior to emergency scenarios prevents disruptions in critical operations due to disconnections in the network.

Overall, federated learning offers significant potential for enhancing crisis management applications and improving the effectiveness of disaster mitigation and recovery efforts. Figure 1.2 shows some of the applications of federated learning in the industry based on the papers reviewed in this work.

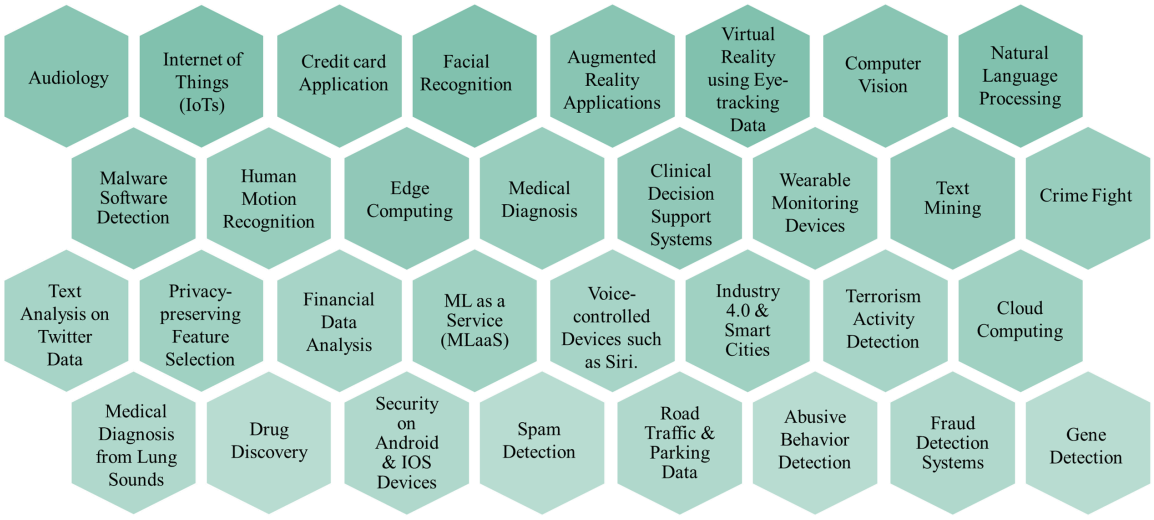


Figure 1.2: Applications of Federated Learning in Industry

1.3 Research Questions and Contributions

Federated learning is at the interface of several research areas, such as optimization, distributed learning, cryptography, and communication theory. The objective of this Ph.D. dissertation is to address some of the existing algorithmic and data-driven challenges from a data science perspective. Accordingly, considering the research gaps in the field that have been discussed in the next chapters, five research questions (RQ) have been proposed that addressing them makes significant contributions and generates new knowledge.

➤ **Objective 1:** Literature review

RQ 1: How does the choice of machine learning model influence the training process, computation complexity, and efficiency of the outcome?

Approach: A comprehensive literature survey on federated learning from a machine learning perspective is conducted. This survey includes an overview of the components

of federated learning and a systematic review of the literature on privacy-preserving machine learning in the last few years based on the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) guidelines.

Contribution: An overview of recent progress and developments in supervised/unsupervised machine learning algorithms, ensemble methods, meta-heuristic approaches, blockchain technology, and reinforcement learning in the framework of federated learning is presented.

- ➤ **Objective 2:** Understanding generalization, stability, and privacy of noise-infusion mechanisms in centralized and federated learning settings and quantifying the relationship between model performance and noise levels to balance privacy and accuracy.

RQ 2: How does the incorporation of noise in different locations within the model structure or the data affect training outcomes?

Approach: In a data-centric era, concerns regarding privacy and ethical data handling grow as machine learning relies more on personal information. Perturbation methods such as differential privacy ensure that individuals are not exposed to potential misuse of personal data and harm. Despite their numerous benefits, adding noise to the data can negatively impact the accuracy of results. Obtaining the balance between privacy protection guarantees and model performance is one of the challenges in this domain. Accordingly, this dissertation investigates integrating noise into deep learning models to provide insights into this objective and improve the generalization, stability, and privacy-preserving capabilities of the models.

Contributions:

1. Comparison of three CNN architectures to assess the impact of model capacity on generalization and stability during training and evaluation in noisy conditions.

2. Comparison of training models with Gaussian noise hidden layers against other noise infusion mechanisms.

3. Comparative analysis of training CNN models with Gaussian noise hidden layers under various noise levels in centralized and federated learning.

4. Understanding the benefits of using noise to improve generalization, stability, and privacy through experimental analysis.

RQ 3: How can we estimate the level of additive noise prior to detecting a significant model performance decrease?

Approach: As federated learning provides a unique approach, the capacity of deep learning models to generalize beyond the training data while maintaining privacy and stability in the face of perturbations becomes more critical in real-world applications. Appropriate metrics are required to assess the implications of training with noise for stability, generalization, and privacy of convolutional neural networks.

Contributions:

1. Introducing the Signal-to-Noise ratio to quantify the trade-off between increasing the noise level and training accuracy and to find the optimal balance between privacy and accuracy.

2. Introducing the Price of Stability and Price of Anarchy to gain a measurable perspective on the trade-offs between model performance and privacy due to increasing noise levels.

➤ **Objective 3:** Addressing Data Imbalance in Precipitation Prediction Models through Federated Learning and Generative Adversarial Networks (GANs)

RQ 4: What is the best way to address the challenges caused by the significant difference in the number of instances between classes, indicating imbalanced data that

affect the performance of a classifier's predictions?

Approach: The classification of weather data involves categorizing meteorological phenomena into classes where certain weather events (e.g., storms or extreme temperatures) might be underrepresented. The significant difference between the number of instances in the classes indicates that the data is imbalanced. Imbalanced data negatively impacts the classifier's performance, resulting in biased predictions. This dissertation explores data augmentation techniques, such as the Synthetic Minority Over-sampling Technique or GANs, to improve the model's accuracy in classifying rare but critical weather events.

Contributions:

1. Conducting an empirical study on the applicability and efficacy of advanced GANs variants (CGANs, Minority GANs, SMOTE GANs, and WGANs-GP) compared to the Synthetic Minority Over-sampling Technique, a well-known resampling method for tabular data over 9 weather stations across Australia.

2. Employing data augmentation techniques improves the model's accuracy in classifying rare but critical weather events.

RQ 5: What is the best way to address the challenges caused by insufficient data in some local centers, which impact the accuracy of predictions and the model's ability to generalize to new and unseen data?

Approach: With advancements in federated learning, machine learning models can be trained across decentralized databases, ensuring privacy and data integrity while mitigating the need for centralized data storage and processing. Thus, the classification of weather data stands as a critical bridge, linking raw meteorological data to actionable

insights, enhancing our capacity to anticipate and prepare for diverse weather conditions.

Contributions:

1. Practical implications of federated learning in meteorology and climate studies, advocating for the potential of combining data augmentation techniques with federated learning to address imbalanced datasets.

2. Advancement in understanding the trade-offs and potential pitfalls of GANs variants, offering insights into the need for intricate network design and hyperparameter tuning.

This dissertation presents a multidimensional exploration of the challenges and open problems in federated learning, noise-infusion mechanisms, and data imbalance in the context of machine learning.

The systematic literature review offers a consolidated knowledge base for researchers. The empirical study on noise-infusion mechanisms bridges theoretical concepts with practical applications and contributes to developing stable and differentially private algorithms, allowing them to generalize effectively and support federated learning. Furthermore, the strategies proposed for addressing data imbalance present actionable insights for real-world applications, especially in the realm of weather prediction.

1.4 Dissertation Structure

The current Ph.D. dissertation consists of an additional three chapters that correspond to the three research objectives mentioned earlier. Chapter 2 corresponds to the components of federated learning and a survey of privacy-preserving machine learning in recent literature. Chapter 3 explores the background and material on generalization,

stability, and differential privacy. We also delve into the description of the Signal-to-Noise ratio, Price of Stability, and Price of Anarchy and their applications. The outcome of the numerical experimentation with a discussion of the results is also provided. The numerical analysis consists of four experiments in centralized and federated settings and multiple noise infusion mechanisms. Chapter 4 of the dissertation aligns directly with the research questions mentioned in the third research objective.

Each of these chapters concludes with a summary section that highlights the significant points and insights derived from the obtained results. Finally, in Chapter 5, all the pertinent findings are consolidated, and the main conclusions of the work and the direction of future research in the field are presented.

Chapter 2

Background & Literature Review

2.1 Components of Federated Learning

Federated learning affects the modeling step of the Cross-Industry Standard Process for Data Mining (CRISP-DM), which starts from local data storage in data centers to communicate with the central server to iteratively aggregate the model parameters and update the global model to achieve the desired learning accuracy in data centers.

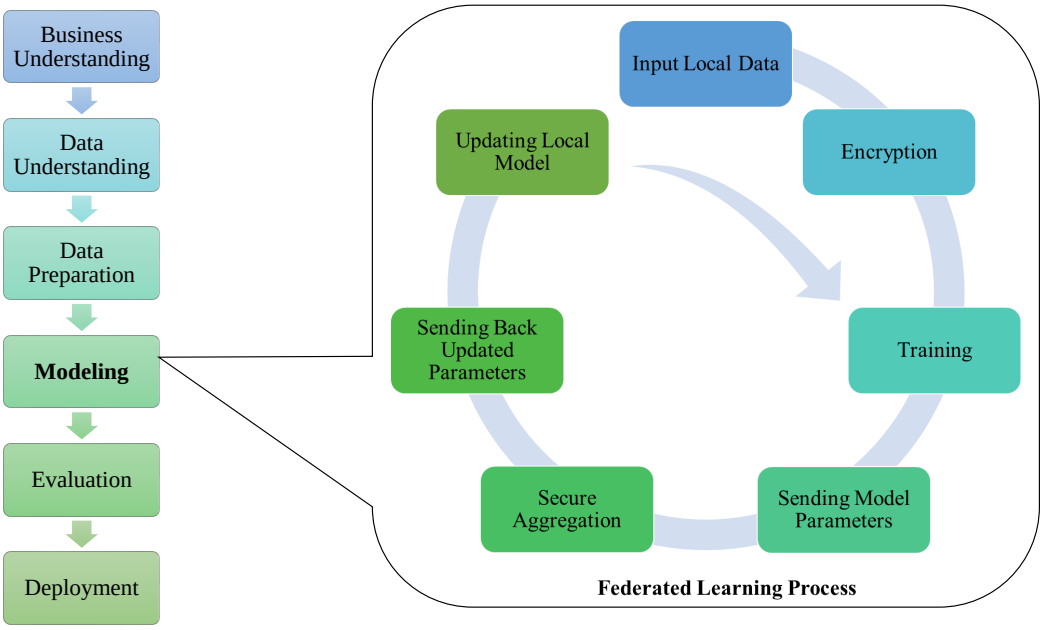


Figure 2.1: Data Mining Process in Federated Learning Based on CRISP-DM Model

Figure 2.1 shows the Federated learning life cycle embedded in CRISP-DM. There is a rich body of research on different aspects of federated learning, such as data

processing, learning models, aggregation methods, specifications of data centers and the central server, communication security, and efficiency among the elements [19] and the multiple software and libraries used for implementing federated learning [20]. In this chapter, we survey the existing research on federated learning. What distinguishes this work from other surveys is the focus on the model selection aspect of the federated learning process, complementing other recent surveys [21, 22, 23, 24, 25, 26]. We explore the different machine learning models used in federated learning to tackle problems in different domains. In this survey, we have investigated the papers published between 2016-2022 in accredited peer-reviewed journals and conferences and classified them based on the machine learning methods used for learning. We limited the search to the keywords federated learning, privacy-preserving machine learning, distributed learning, supervised/unsupervised learning, and artificial intelligence. The PRISMA diagram presented in Figure 2.2 demonstrates the searching strategy in this survey.

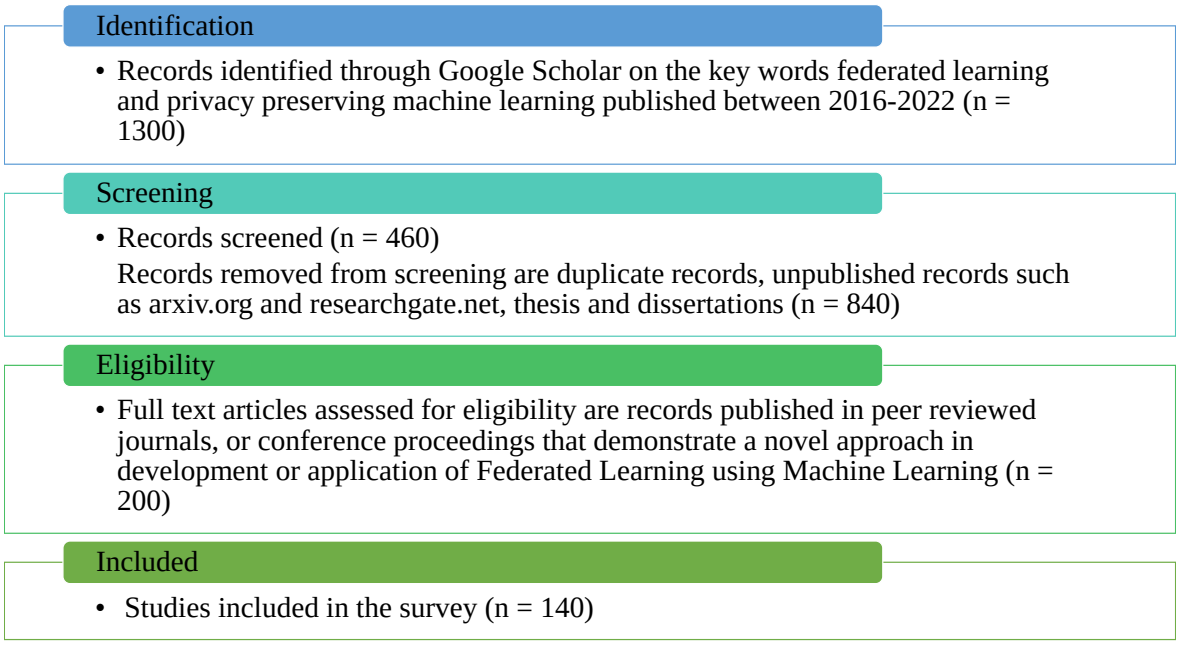


Figure 2.2: PRISMA Flow Diagram Summarizing the Search Strategy

Federated learning is a broad term that includes different aspects of data collection, storage, analysis, and communication in a decentralized information system where data centers can not disclose data for learning purposes.

This chapter reviews the literature on the components of federated learning and its applications in the last few years. The main purpose of this survey is to provide young researchers and practitioners with a comprehensive overview of federated learning from the machine learning point of view.

In the following sections, we will provide a detailed explanation of the federated learning framework's components: storage, privacy, communication, federated aggregation, and privacy-preserving machine learning, respectively. We dive into a detailed discussion of different machine learning models used for training decentralized data, their use cases and applications, and some technical details useful for implementation.

2.2 Storage

Federated learning is a cross-organizational framework. Therefore, the features and observations may vary between different data centers. Depending on the architecture of the data centers and how the data are partitioned, three scenarios of horizontal partitioning, vertical partitioning, and transfer learning have been discussed.

2.2.1 Horizontal partitioning

Horizontal federated learning, also known as sample-based federated learning, is the scenario in which the data centers have the same features but different sample spaces that require modifications to the training model [27]. For example, a network of local banks that individually collect a certain list of information from their clients. In this example, the clients are different. Therefore, the sample space varies between the

banks. Horizontal federated learning allows entities to build a generalized global model based on a larger data pool without compromising privacy.

2.2.2 Vertical partitioning

Vertical federated learning, also known as feature-based federated learning, is when local datasets have the same sample space but differ in their features. Numerous researchers have explored training models specified for vertically partitioned data. For example, a local bank and an insurance company share the same clients but collect different types of information. If the two entities were to build a local model collaboratively, the two datasets would have common sample space but very different features. The common aggregating methods aren't effective in vertically partitioned data. Therefore, the difference between the feature spaces causes different challenges and creates opportunities for further research to address the issue of fault diagnosis.

2.2.3 Transfer Learning

Transfer Learning is the learning structure in which the local datasets differ both in the sample and feature space. Thus, knowledge is derived from various sources to achieve a global model. Despite all the challenges, transfer learning has been applied to a wide variety of problems in different domains [28], and it has great potential for further improvement.

2.3 Federated Aggregation

Secure aggregation is the function that receives model parameters from local data centers and outputs the aggregated model parameters to update the training model. Numerous studies explore aggregation methods to improve learning accuracy in encrypted

data. Lia and Togan [29] implemented federated learning with secure aggregation in Python. The authors ensure privacy by using SMC.

Federated Averaging(FedAvg) is the baseline aggregating method in federated learning. In this scheme, an initial global model is used to locally train the datasets located on a network of distributed data centers. The encrypted model parameters are uploaded to the central server, and the average updates of local models are used to update and improve the global model. The model parameters provided by the clients are aggregated at the central server using Equation 2.3.1

$$\delta_{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} \delta_{t+1}^{k} \tag{2.3.1}$$

Then, the new model is sent back to the clients. This iterative process continues until the model parameters converge to a specific performance level or the task is completed. FedAvg is a practical approach since it does not require data centers to disclose their data; thus, the models can be trained locally. However, the communication cost can be high. To address this issue, Li et al. [30] explored an adaptive communication frequency aggregation method that helps the algorithm converge faster and have a smaller loss. They also used a gradient sparse approach to reduce communication costs by decreasing the parameters that need to be updated. Another variation of FedAvg is a weighted FedAvg method, which has been proposed and has shown promising performance in experiments on fault diagnosis [31]. Also, Hong et al. [32] used weighted FedAvg for non-IID imbalanced data and evaluated the model on CIFAR-10 and SVHN benchmark datasets.

Co-operative aggregation is another aggregation method (CO-OP) in which the local models are merged into the global model by using a weighted scheme based on the local models' age to anticipate the time difference between them and how they have

improved at each iteration. Chen et al. [33] utilized a discrepancy-based weighted federated averaging method to address the inconsistencies in the contributions of data centers in the global model for federated averaging. The experiments show the effectiveness of federated transfer learning with the proposed averaging method for fault diagnosis.

2.4 Communication

Communication between local data centers and the central server requires sufficient connectivity and bandwidth to ensure secure and private communication between the entities and the entire system. While communication efficiency is highly dependent on the existing infrastructure, reducing the number of interactions between the data centers and the central server can improve the efficiency of the federated learning model. For example, Shen et al. [34] built a blockchain-based model for secure data sharing and training using privacy-preserving Support Vector Machines. Their proposed model requires only two interactions in each iteration, which provides higher performance accuracy and data privacy with less computation cost. In addition, the issue of fairness between the data centers and the central server arises with the federated learning framework. The limitations posed by communication efficiency and security are an ongoing challenge in implementing Federated learning at scale [35].

2.5 Privacy

Preserving privacy is an essential constraint when learning from sensitive data that requires protection against data leakage and adversarial attacks. In cybersecurity, the adversary is defined as a person or a group that performs malicious actions to

disrupt or corrupt a cyber system. Mothukuri et al. [36] identified the different types of adversaries in the federated learning framework and highlighted the solutions to improve systems vulnerability. Table 2.1 describes the two types of adversaries caused by adversaries.

Table 2.1: Types of Adversaries in Cybersecurity

Adversaries	Action	Aim	Protocol
Semi-honest	Passive	Learning about the system	✓
Malicious	Active	Manipulation, corruption, control the system	✗

To address the problem of adversaries, Secure Multiparty Computation(SMC) is developed. SMC is a framework that enables multiple parties to securely process data while ensuring that no valuable information is leaked. This scheme allows machine learning models to train sensitive data when most of the parties involved are honest. The components are input parties, computation parties, and result parties. For example, SMC is used as a privacy-preserving method in studies on genome data [37], where the biobanks act as the input parties that hold gene data and medical diagnoses information. The neural hosts and other biobanks are chosen as computation parties, and designated recipients are selected as result parties. This method allows us to combine datasets from genome data collected from different individuals without compromising their privacy. There are different protocols for implementing an SMC that differ in size and number of supported input parties. In highly regulated industries such as healthcare, more advanced privacy-preserving methods are needed to facilitate the use of effective machine learning models in the Federated learning framework [38]. The privacy-preserving methods are categorized into the two following approaches [39]:

2.5.1 Perturbation Approach

This is a privacy-preserving approach based on adding noise to the data [40] such as Differential Privacy, which is a perturbation mechanism that involves adding noise to the data to obscure sensitive data and ensuring security in the federated learning framework. Perturbation methods are effective in securing the data, yet decrease the learning accuracy [41]. Differential privacy is based on the concept of semantic security, which means the encryption systems prevent enclosing any amount of information through the learning process. In Differential Privacy, the outcome is assumed to be the same regardless of the data if the model is implemented on two neighboring databases. Two datasets of S and S' are considered neighboring if they differ at most in one random variable, and they are written as $S \backsim S'$. This ensures that the algorithm's output does not reveal any information about data sharing and analysis. This feature is also known as ϵ-DP. To satisfy Differential Privacy, the Laplace or Gaussian mechanism is used for data with integer or real-valued outputs, and noise is sampled from a Laplace or Gaussian distribution to ensure ϵ-DP for noisy data. The exponential mechanism is used for categorical data, in which each output is associated with a non-zero value for the probability of being selected based on a utility function [42]. First, we compute the sensitivity of the utility function and then compute the quality score of each output in the database. The output is selected probabilistically. Tuning ϵ in the Differentially Private mechanisms to ensure semantic privacy is one of the challenges of using Differential Privacy for data privacy. Wei et al. [43] demonstrated the trade-off between model convergence and privacy level. They use a client scheduling strategy to improve model convergence while maintaining privacy. Accumulation of noise can jeopardize the accuracy of results if it exceeds a threshold on several operations, and the threshold is on the depth of the operations rather than the number of operations performed to infuse noise to encrypt the data. The depth of the operations is the

maximum degree of the evaluated polynomial. The operation depth is determined by the privacy protection scheme as well as the level of speed and security.

2.5.2 Cryptography Approach

This approach preserves data privacy using cryptographic primitives and includes homomorphic encryption, garbled circuits, secure processors, and order-preserving encryption.

In this section, we will look closely at Homomorphic Encryption, which preserves privacy and accuracy for the cost of higher running time. The components of Homomorphic Encryption are key generation, encryption, decryption, and evaluation algorithm [44]. Homomorphic Encryption is a cryptosystem that involves ciphering data using a public or private key and sharing the key among peers to decipher the ciphertext. Data is ciphered by mathematically transforming the data using addition and multiplication operators [45]. Variations of Homomorphic Encryption ensure data security among data centers and the central server. Qin et al. [46] used Homomorphic Encryption for cloud-based privacy-preserving image processing, including feature detection, digital watermarking, and content-based image search.

Limitations of Homomorphic Encryption are that the target space is limited to 0,1 binary values, which is not a feasible representative in practice. There are solutions to address this issue that expand the message space to integers. However, in statistical learning, the values are not limited to binary and integer values. Also, The encrypted ciphertext increases drastically in size, sometimes by several orders of magnitude. This requires additional storage and computation power since the learning procedure is more computationally complex. Current Homomorphic Encryption schemes use only addition and multiplication operators. Therefore, comparison tasks are not supported.

Improving encryption schemes to support subtraction and comparison operations, such as inequalities, is an open research question in Homomorphic Encryption.

2.6 Privacy-preserving Machine Learning

In federated learning, a learning model is built and tuned collaboratively between the central server and data centers [47]. Chandiramani et al. [48] compared the efficiency of numerous machine learning models in the federated learning framework on the benchmark fashion-MNIST data. Learning from distributed data poses different issues and challenges. Therefore, different machine learning models have been explored to compare the performance and efficiency of the learning models. For example, to increase the security in Android devices, Galvez et al. [49] built a federated learning malware classification model using K-Nearest Neighbor, Logistic Regression, Random Forest, and Support Vector Machines as machine learning models.

Figure 2.3 provides a quick overview of machine learning models used in federated learning literature. In this section, we have provided a detailed survey of the traditional machine learning algorithms and more recent learning schemes in this domain.

2.6.1 Regression Models

Regression is a predictive modeling approach for identifying the linear and nonlinear relationships between independent variables and the target. Logistic regression has been used in the framework of federated learning for different applications. Yang et al. [50] explored a logistic regression model on clients' credit card and healthcare data. Guo et al. [51] used a logistic regression model to classify illness/health in the cloud environment named POMP. A preprocessing technique and a Bloom filter

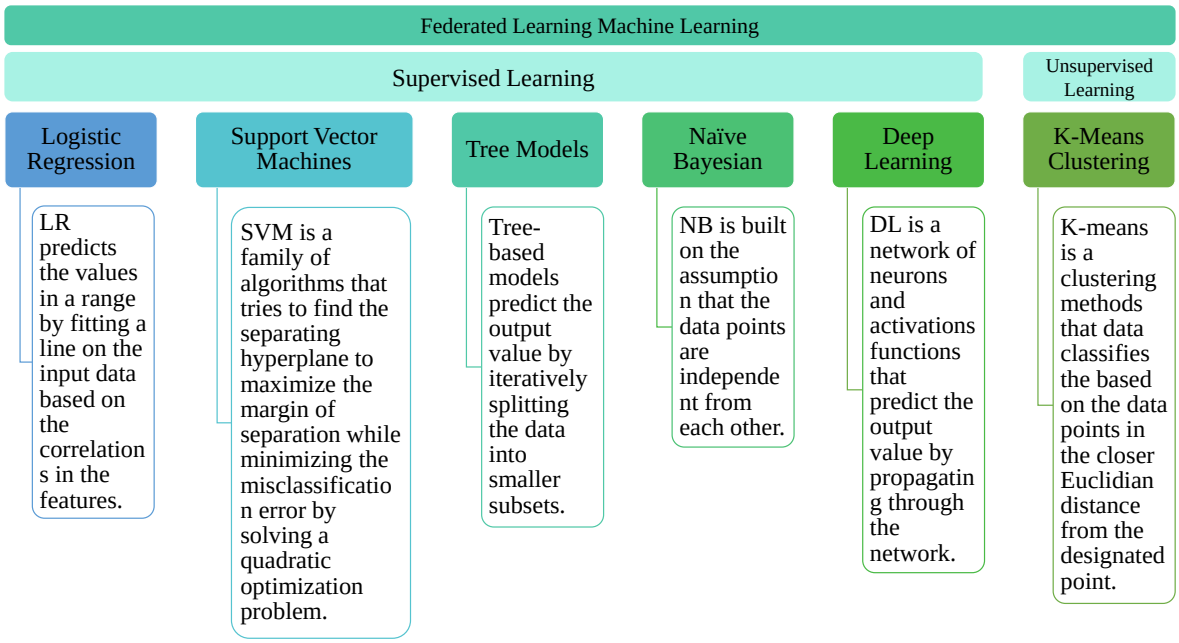


Figure 2.3: Federated Machine Learning Algorithms

are also used to reduce the computational complexity in the pre-diagnosis process. The model is implemented using Java and the JPBC library. Dunner et al. [52] implemented a ridge regression model on five benchmark datasets to compare the performance of Spark and Open MPI, two distributed machine learning frameworks, and suggest recommendations on improving the models implemented in Spark. The results from this paper show that fine-tuning the parameters in a distributed machine learning model to adapt the system's specifications and offloading the Spark language-dependent overheads using C++ can improve computation efficiency. The model is trained for when at least one of the data centers is honest or honest but curious using SPINDLE, an operational system for generalized linear models in distributed learning.

Regression models offer numerous benefits, such as privacy preservation, continuous learning for real-time data analysis, and resource efficiency. However, they deal with the challenges related to heterogeneous data, model aggregation, and privacy concerns, common in many machine learning problems.

2.6.2 Support Vector Machines

Support Vector Machines are widely used in medical diagnosis, spam detection, facial recognition, and analyzing financial data [53]. Bost et al. [54] use Support Vector Machines to build efficient privacy-preserving algorithms and evaluate the results on breast cancer diagnosis, credit card approval, audiology, and nursery data. Xu et al. [55] implemented linear and nonlinear Support Vector Machines on horizontally partitioned data using the MapReduce framework to preserve privacy. Park et al. [56] proposed a Homomorphic Encryption-friendly least-squares Support Vector Machines to train toy and real-world datasets that outperformed the logistic regression model. Senekane [57] proposed a privacy-preserving Support Vector Machines framework for image classification. Liang et al. [58] focus on an outsourcing scheme for Support Vector Machines classification with an efficient cryptographic primitive named order-preserving encryption. Chanyaswad et al. [59] proposed a multi-kernel method using the lossy-encoding scheme to protect the privacy of the data. The training models are Support Vector Machines with an RBF kernel with multiple gamma values and a Signal-to-Noise Ratio-based Support Vector Machines, which is a Signal-to-Noise Ratio for kernel weight design that uses different kernels. The kernel functions are linear, polynomial, Radial Basic Function(RBF), Laplacian, and sigmoid. In terms of privacy, compressing single kernels to form a multi-kernel provides effective results and maximizes utility. The method based on the Signal-to-Noise Ratio method improves the performance compared to uniform and alignment-based methods. Hsu et al. [60] designed a privacy-preserving system for malware software detection using SGD-based Support Vector Machines and SMC techniques. The paper by Zhang et al. [61] proposed a solution for the problem of human motion recognition in multimedia interaction scenarios in a virtual reality environment using Support Vector Machines. Chen et al. [62] focused on the issue of computation efficiency and low latency of

edge computing for augmented reality applications. Hartmann et al. [63] proposed a Support Vector Machines model in a privacy-preserving setting, called secret Support Vector Machines, for predicting user gender from tweets based on the online and offline evaluation. Lu et al. [64] propose a privacy-preserving feature selection using a multi-class Support Vector Machines named PPM2C, which is evaluated with PAN-SVM and LIB-SVM. The results from this study show that multi-class Support Vector Machines (PPM2C) reduce the chances of overfitting compared to regular Support Vector Machines. Despite being an effective learning method, implementing privacy-preserving Support Vector Machines on data with missing values poses numerous challenges that must be addressed. Omer et al. [65] built a distributed Support Vector Machines model with multiple imputations by chained equations on vertically partitioned data. In this work, the privacy of the data is ensured with the Paillier cryptosystem. The evaluation of the proposed scheme shows higher accuracy and lower computation time compared to the centralized model on imputed data. Medical diagnosis systems can significantly benefit from advances in federated learning. Machine learning methods were used to design a secure framework to prevent severe health conditions by diagnosing patients based on their symptoms and the data collected from wearable monitoring devices that monitor heart rate, temperature, oxygen saturation, and other vital signs, and voice-controlled devices such as Google Assistant, Amazon Echo, and Apple Siri [66]. Privacy-preserving Support Vector Machines models have been particularly successful in healthcare applications. Wang et al. [67] focus on outsourced Support Vector Machines and EPoSVM (Efficient and Privacy-preserving Outsourced Support Vector Machines) for data classification in the Internet of Medical Things, which results in an improvement in learning accuracy and security compared to Support Vector Machines. Zhu et al. [68] explored an efficient and privacy-preserving online medical pre-diagnosis framework (eDiag) using nonlinear kernel Support Vector Machines. Ahmed et al.

[69] developed "mLung", a cloud-based privacy-preserving service to detect chronic pulmonary issues from lung sounds such as cough. The analysis is performed on a personal mobile phone to ensure privacy. Medical components of the drugs must be kept private by pharmaceutical companies. Wang et al. [70] explored an encrypted kernel Support Vector Machines using Homomorphic Encryption. They have built a sub-image to position different sections of the image so that a face can appear. The authors trained and tested their model on the BioID Face Database.

One of the key benefits of Support Vector Machines in federated learning is their ability to generalize well from limited data, making them suitable for scenarios where each participant has a relatively small dataset. However, some of the challenges and limitations of Support Vector Machines are:

- Complexity of kernel functions: Support Vector Machines rely on kernel functions to handle the data in nonlinear space. However, selecting appropriate kernel functions in a framework with diverse data sources can be challenging, as different participants may require different kernel types.

- Communication overhead: Support Vector Machines are computationally expensive due to dealing with large datasets and complex kernel functions, making them unsuitable for edge devices or environments with limited computational power.

- Hyperparameter tuning: Support Vector Machines have multiple hyperparameters, such as the regularization parameter (C) and the kernel parameters. Hyperparameter tuning across multiple participants in a federated setting can be complex and time-consuming.

While Support Vector Machines can be computationally expensive, they offer strong generalizations that can lead to robust models even with limited local data.

2.6.3 Tree Models

Decision Tree and Random Forest are a group of effective and widely used models for data classification and regression [71]. Khodaparast et al. [72] presented a Decision Tree algorithm equipped with a federated learning framework for horizontal and vertically partitioned data. Yadav et al. [73] presented a new Decision Tree-based model in a privacy-preserving manner in which the data are partitioned vertically into multiple parties, and the parameters are sent to the central server. Badsha et al. [74] presented a privacy-preserving Decision Tree framework to build and learn the tree-based model without requiring the parties to disclose private information. The authors use Homomorphic Encryption to maintain privacy, while the parties are assumed to be honest but curious. The Gini Index is used to measure the classification capability of the model. Canillas et al. [75] explored a privacy-preserving Decision Tree framework for private fraud detection systems at SiS ID, a French business platform. The model is used to classify transactions into four risk classes. The model's accuracy depends on the configuration of the encryption key and the number of nodes for the Decision Tree. The proposed model utilizes a Decision Tree algorithm that helps improve the diagnosis pace and accuracy based on the patient's symptoms without disclosing the patient's private data. Xue et al. [76] proposed a consent-based privacy-preserving Decision Tree model for the evaluation scheme. The additive Homomorphic Encryption method and a secure comparison model are used. Also, Xue et al. [77] proposed a privacy-preserving Decision Tree for classification using additive Homomorphic Encryption, which provides lower computation and communication overhead. Hou et al. [78] explored a Random Forest with a Decision Tree for data classification and studied the impact of tree depths in Decision Tree on privacy and classification. A privacy budget is allocated for nodes at different depths in the Decision Tree. Guan et al. [79] explored a budget allocation mechanism for Decision Tree construction for balancing

the excessive noise introduced at leaf nodes. The iterative process speeds up the selective aggregation process. The tree is constructed based on the C4.5 method. Lv et al. [80] proposed a hybrid Decision Tree algorithm for constructing a Random Forest to balance privacy and classification accuracy. In the paper by Xin et al. [81], the authors proposed a new differentially private greedy Decision Tree algorithm called (DPGDF) which is a combination of greedy trees and parallel combination theory. Zhao et al. [82] explored a tree-based data mining model for regression and binary classification tasks. In this work, a privacy-preserving Gradient Boosting Decision Tree (GBDT) model is aggregated into an ensemble. Random Forest is also used for feature engineering. Fritchman et al. [83] proposed a tree ensemble approach to learn from the data collected in healthcare institutes securely. Zhang et al. [84] built a secure Parkinson's diagnosis framework using non-speech body sounds such as breathing and coughing.

The main advantages of tree-based models, including Decision Trees and Random Forests, are their interpretability, ensemble learning for robustness, and feature importance analysis.

2.6.4 Naïve Bayesian Algorithms

Naïve Bayesian algorithms is a supervised learning algorithm centered around the Bayes theorem, which is based on the assumption that there is conditional independence among each pair of features if the class is known. In smart environments such as smart cities, data privacy is crucial. Amma and Dhanaseelan [85] explore a Naïve Bayesian classification framework for privacy-preserving machine learning on the cloud using smart city data. The authors validated the results using Viz road traffic, pollution, and parking data collected from the City Pulse Smart City dataset. Part et al. [86] introduced a novel federated learning architecture that consists of three layers. In the edge layer, the data is processed, the machine learning models are trained in the fog

layer, and the results are aggregated in the centralized cloud layer. Data partitioning poses specific challenges when learning from the data in a distributed setting. Vaidya et al. [87] investigate a privacy-preserving Naïve Bayesian classification model and compare the results on horizontal and vertically partitioned data. Yurochkin et al. [88] propose a Bayesian non-parametric federated learning framework with neural networks. The model is evaluated on two benchmark data sets for image classification. The paper by García-Recuero [89] addresses the issue of detecting and discouraging abusive behavior in online social networking applications such as Twitter by limiting the accessibility to user-sensitive information. This work uses feature engineering on relative importance calculated from the Random Forest learning algorithm. Li et al. [10] use Naïve Bayesian and a hyperplane-based decision model for classification. Furthering this work, Chai et al. [90] propose an outsourced encryption protocol to improve the security vulnerability of Li's model. The classification models are trained using Scikit-learn. Paillier cryptosystems are used in this work due to light operations. In a paper by Yang et al. [91], the authors propose a communication-efficient privacy-preserving framework based on the Naïve Bayesian method to predict the disease risk for e-health applications. Sharkala et al. [92] introduce a privacy-preserving machine learning algorithm for horizontally and vertically partitioned data based on a tree augmented Naïve Bayesian classifier. A third party conducts the operations, and the data is encrypted. Teo et al. [93] propose a privacy-preserving algorithm using kernel regression and Naïve Bayesian classifier for multiparty computation, and the Paillier cryptosystem is used for encryption. Medical diagnosis systems are the main application of privacy-preserving machine learning. Liu et al. [94] build a secure diagnosis scheme using a Naïve Bayesian classifier. Furthermore, malware detection systems protect the user's identity by identifying malware API call fragments. Lin et al. [95] propose a privacy-preserving Naïve Bayesian model for malware detection.

Talbi et al. [96] compare their classification algorithms of Naïve Bayesian, Decision Tree, and logistic regression on encrypted data.

Naive Bayesian algorithms provide probabilistic predictions and can quantify uncertainty. In federated settings, this is a significant advantage for risk assessment and decision-making, particularly in applications such as crisis management and disaster response scenarios where uncertainty plays a significant role.

2.6.5 Deep Learning

Deep learning has been the dominant learning method from structured and unstructured data in recent years. Deep learning is the burgeoning powerful technique in the field of machine learning. Deep architectures are useful for learning complicated patterns in large-scale data, attracting much attention in academia and industry. Different topologies and architectures with real-world applications exist. These models have been used in different areas such as Computer Vision, Natural Language Processing, and speech recognition [97, 98]. The improvements in implementing a secure Neural Network model on the cloud create a platform for scalable Neural Networks to be used as a service [99, 100]. A Recurrent Neural Network is a variation of a forwarding propagation Neural Network in which the neurons in the hidden layers receive the input value with a delay in time and access information from previous iterations in the current layer. Recurrent Neural Network is useful in Natural Language Processing, where knowledge about the previous words in a sentence is necessary for predicting the next word. Text mining and Natural Language Processing, which is the process of extracting knowledge from text documents, are used for learning from data collected from highly sensitive resources such as homeland security for crime fight and detecting terrorism activities. Therefore, building secure federated learning text analysis methods is necessary to ensure no sensitive information is disclosed. To this end, Costantino

et al. [101] propose using an out-of-bag classification method to detect terrorist activities on Twitter. Convolutional Neural Network is a deep learning architecture that has gained popularity in Computer Vision [102]. This architecture uses one or more convolutional layers to extract high-level features. Wang et al. [103] investigate a privacy-preserving Natural Language Model framework to compute word representations using deep learning. Xia et al. [104] designed a Graph Convolutional Network model for predicting traffic flow quickly and efficiently. Deep learning methods can also be combined with other classifiers, such as linear Support Vector Machines, for the classification of images. Niu et al. [105] explore a deep learning framework for mobile sensing systems. Lin et al.[106] propose a Recurrent Neural Network framework called the Predictive Clinical Decision (PCD) scheme, which is used for e-health applications. Eye-tracking devices are the main technology in virtual reality and augmented reality that can improve efficiency through gaze-based optimization methods. The eye-wear and eye-tracking devices used in the auto industry can pose privacy issues for the driver and bystanders [107]. Therefore, Steil et al. [108] explore a privacy-preserving method for a first-person video dataset of daily life recordings. The authors propose the PrivacEye method that combines Computer Vision with eye movement analysis techniques. Another privacy concern with Computer Vision technologies in facial recognition systems such as Google Street View is when personal images of individuals are shared via different data centers.

Deep learning models have demonstrated great potential for highly accurate and competitive results when dealing with diverse and large datasets. While increasing model complexity in neural network architectures helps improve generalization, excess complexity results in overfitting and computation overhead. Adaptation, optimization, and the integration of privacy-preserving techniques are essential to harness

the strengths of deep learning while mitigating the specific challenges and limitations present in federated settings.

2.6.6 Unsupervised Machine Learning

Current Federated learning machine learning-based models are constructed based on supervised learning. However, in most applications, no or little labeled data exists. Thus, it is appropriate to use unsupervised learning methods. While there has been considerable progress on federated transfer learning to cope with data with few labels, applying unsupervised learning in a federated setting remains a bottleneck for many applications. Clustering techniques have been employed to deal with the challenges of unlabeled data. While K-means clustering is widely used for pattern recognition in gene detection and image segmentation, a modified framework is required when the data is sensitive. Zhu and Li [109] proposed a secure aggregation and division protocol based on Homomorphic Encryption to build a secure clustering algorithm. Al-Saeidi et al. [110] proposed a clustering analysis for improving the communication cost in federated learning using the human activity recognition dataset. A secure weighted average protocol and secure number comparison protocol are used for privacy-preserving. Five different classification algorithms were explored: multi-layer perceptron, K Nearest Neighbor, Sequential Minimal Optimization, Naïve Bayesian, and J48 (an implementation of Decision Tree classification in WEKA). Anikin and Gazimov [111] proposed a clustering algorithm named Density-Based Spatial Clustering of Applications with NOISE (DBSCAN) for vertically partitioned data. Romsaiyud et al. [112] investigate a privacy-preserving K Nearest Neighbor model for pattern recognition, with automated hyperparameter tuning to improve model accuracy and a cryptographic hash function to ensure data security.

The benefits of unsupervised learning techniques in federated learning are:

- Privacy-preserving clustering: Clustering algorithms can perform data analysis without the need for explicit labels or excess information sharing, which is beneficial when dealing with sensitive data.

- Data exploration and anomaly detection: Unsupervised models excel at data exploration, allowing practitioners to identify the underlying patterns, anomalies, and outliers within local datasets. This exploratory capability is valuable for uncovering insights without exposing private data.

- Reduced labeling effort: Limited labeled data is a well-known issue in machine learning. Unsupervised learning models can reduce the labeling effort by enabling semi-supervised or self-supervised learning approaches without exposing the data.

With appropriate evaluation strategies, we can effectively leverage the strengths of unsupervised learning while mitigating the specific challenges and limitations of federated learning.

2.6.7 Ensemble Learning

Ensemble learning is a general approach that seeks to improve learning performance by aggregating the results from multiple classifiers. The paper published by Attota et al. [113] used an ensemble multi-view federated learning model to identify intrusion in IoT devices to improve model efficiency against different attacks. Ma et al. [114] apply edge computing methods for medical diagnosis using the XGBoost model. They use a lightweight, adaptive boosting classification method (AdaBoost) for facial recognition on FERET, a standard face recognition evaluation database. The data is encrypted when sent to two servers for distributed learning.

Ensemble learning is an effective approach in the machine learning domain and has yet to be extensively explored in the federated learning framework. There is a very

limited number of papers published in peer-reviewed journals using this approach. However, they demonstrate the potential of ensemble learning in federated learning.

2.6.8 Meta-heuristic Approaches

Apart from the more commonly used machine learning algorithms mentioned above, meta-heuristic approaches have also been introduced to the federated learning domain. Polap and Wozniak [115] used a novel approach based on parallelism to improve classification models' efficiency in federated learning. The authors demonstrate the effectiveness of their approach when the sample size is relatively small. In another paper, [116], they explore using a meta-heuristic federated learning framework for image classification in the presence of poisoning attacks. With application in IoT and smart city services, Qolomany et al. [117] investigate using Particle Swarm Optimization for efficiently tuning the hyperparameters in the machine learning model. Utilizing meta-heuristic approaches can be further explored as a novel approach to improve the efficiency of the training model in federated learning.

Metaheuristic algorithms are versatile and adaptable to various problem domains. They can be customized to suit the specific requirements and constraints of federated learning scenarios.

2.6.9 Blockchain Technology

Despite improvements in Homomorphic Encryption and Differential Privacy in preserving privacy, there is always a trade-off between learning accuracy and privacy. To overcome such issues, federated learning can be equipped with blockchain technology [118, 119]. Utilizing blockchain technology in federated learning is an emerging field in federated learning and decentralized data storage and processing [120, 121]. Nguyen

et al. [122] survey the advances and challenges of federated learning with blockchain technology. In edge computing and learning from IoT data, blockchain federated learning is a solution to issues in data storage, communication cost, and privacy of sensitive data. Wang et al. [123, 124] use a blockchain-distributed setting as the groundwork for federated learning to ensure additional privacy protection from servers and prevent malicious attach on user-sensitive data. Kang et al. [125] proposed a federated learning framework based on a blockchain mechanism and introduced reputation as a metric to identify reliable data and propose a reliable framework to learn from the data on mobile networks. Later, Kang et al. [126] used multiple blockchains to design a cross-chain framework to improve the scalability and communication efficiency of federated learning for training the data on IoT devices. An example of other applications is a classification of COVID-19 cases from multiple resources using blockchain-based federated learning. Comparing the federated learning framework to centralized models shows improvement in diagnosing COVID-19 patients [127].

Leveraging blockchain technology within the context of federated learning introduces many benefits, such as:

- Data privacy and security: Blockchain technology has inherent privacy protection capabilities allowing participants to maintain control over their data while securely contributing to the global model.

- Transparent and trustworthy transactions: The decentralized nature of blockchain ensures that all transactions and updates are transparent and traceable. This feature mitigates concerns of data tampering and adversarial attacks.

- Smart contracts for governance: Smart contracts are programmable scripts executed on the blockchain that can be employed for governing federated learning

agreements and model updates. This automation streamlines the process and enforces predefined rules and policies, reducing the risk of malfunction and misuse.

Despite its benefits, implementing a blockchain-based federated learning system is complex and requires expertise in both blockchain technology and machine learning. Developing and maintaining such a system is challenging as some blockchain networks consume a significant amount of energy. This environmental impact may not align with sustainability goals in federated learning.

2.6.10 Reinforcement Learning

Incorporating other learning approaches into federated learning has shown promising results in different applications [128]. Liu et al. [129] built a reinforcement learning framework, which is a learning system through trial-and-error interactions between agents and environments combined with cloud computing and IoT technology to create a dynamic system for cancer patient treatment regimes. Wang et al. [130] proposed a reinforcement learning mechanism to introduce a rewards system that optimizes accuracy and communication efficiency. A reinforcement learning approach is also used for evaluating node contributions and improving the pricing strategy in federated learning for IoT devices [131]. Krouka et al. [132] investigate different aggregation schemes in a reinforcement learning-federated learning framework to improve communication costs.

The benefits of combining reinforcement learning and federated learning are :

- Dynamic model adaptation: Reinforcement Learning models can adapt dynamically to changing data distributions and evolving environments. In federated learning, where data sources may drift or have different characteristics, Reinforcement Learning can facilitate model adjustments for improved performance.

- Sequential decision-making: Reinforcement Learning is well-suited for sequential decision-making tasks. In federated learning scenarios, this capability can be valuable for applications that involve sequential interactions or recommendations.

Reinforcement Learning models often require extensive training and exploration of different policies, which can be computationally expensive and time-consuming. Reinforcement Learning must efficiently find the optimal policy by balancing the exploration of new actions and the exploitation of known policies. Managing the trade-off between exploration and exploitation is necessary to avoid excessive data sharing or overfitting.

2.7 Chapter Summary

This chapter is an extensive literature review of federated learning from the machine learning point of view, complementing other recent literature reviews. This review will be useful for researchers in academia and industry and possibly a useful tool for graduate students who want to work in this area. Federated learning was proposed as a solution to the issue of data leakage and loss of privacy in machine learning. With a large amount of data at hand, there is a burgeoning demand for federated learning as potentially being the solution to private and environmental-friendly machine learning at scale. Decentralized learning strategy and privacy protection mechanisms in federated learning grant us access to otherwise unavailable data. Hence, we can expand machine learning in domains such as IoT and healthcare and crisis management in natural and human-caused disasters that require privacy preservation. In recent years, there has been an increasing number of papers published in this domain, and the goal of this study is to provide an overview of federated learning and the existing privacy-preserving machine learning algorithms used in this framework, in addition to their potential and limitations in various applications. Despite our effort to thoroughly

search the literature on federated learning, we limited our search to published papers in peer-reviewed journals in English. Therefore, other novel approaches might be found in the papers not included in this survey.

Chapter 3

Noise-Infusion Mechanisms in Deep Learning

3.1 The Paradox of Noise

In a world dominated by data-driven decision-making, artificial intelligence has offered remarkable capabilities in a wide range of applications, from healthcare to finance, smart cities, and beyond. Machine learning models, particularly deep neural networks, are built on abundant personal data, such as health records, financial data, browsing history, etc., collected by governmental organizations and the private sector. Despite the growing popularity of deep learning across domains, there are still concerns related to the algorithms' ability to generalize, maintain stability, and ensure privacy protection against adversaries.

As the new applications of artificial intelligence enter different aspects of our lives, the recognition of privacy as a fundamental human right has increased. This calls for the development of ethical and responsible learning frameworks. Without proper mechanisms, individuals are exposed to potential misuse of personal data and harm. Adhering to privacy protection policies, machine learning practitioners strive to develop tools that enable the use of sensitive data while maintaining privacy. If privacy concerns are addressed, organizations and practitioners can leverage sensitive data responsibly to harness the power of machine learning without exposing individuals to risks. Differential privacy is designed to provide strong privacy guarantees for data

analysis. By adding noise to the data, the differential privacy guarantee ensures that an attacker cannot infer sensitive information from the released data.

Despite its promising implications for ensuring data privacy, adding noise to the data can result in a loss of accuracy. Therefore, more complex models are utilized to address the decline in performance since they are better at distinguishing helpful information from the noise in the data. Increasing the number of layers and hidden units in the network results in more complex models and improved generalization. However, overly complex models run the risk of overfitting and performing poorly on unseen data. Moreover, such models are more sensitive to variations in the data and model, resulting in significant fluctuations in the output.

While excessive noise can be disruptive, introducing controlled perturbations during training can contribute to improved privacy protection through techniques like differential privacy, generalization, and stability. The objective of this study is to evaluate this claim and develop a systematic method of fine-tuning the noise parameters to achieve the desired privacy protection guarantees without sacrificing the accuracy of the results. We focus on Convolutional Neural Networks (CNN) for image classification and delve into the challenges and strategies of noise infusion mechanisms in centralized and federated settings.

Motivated by the potential benefits of noise, we explore the implications and limitations of training with noise to gain a deeper insight into the impact of noise on generalization, stability, privacy, and overall model performance. We combine structural stabilization and noise infusion mechanisms to improve the generalization and stability in deep neural networks while maintaining privacy. Proper architecture and regularization scheme balance the generalization power of the training model with its capacity to memorize the intricate patterns within the data without oversimplifying

the model and possibly losing information. Enhanced by differential privacy, federated learning plays a pivotal role in the future of machine learning. As a collaborative framework, federated learning enables data processing without requiring the data to be centralized. Given the decentralized nature of data in federated learning, we can not utilize the sample size as we possibly could with aggregated data. Therefore, achieving stable models with great generalization is especially beneficial when working on unseen data distributed over multiple devices. The findings of this study shed light on the benefits of using noise to improve generalization, stability, and privacy. As federated learning provides a unique approach, the capacity of deep learning models to generalize beyond the training data while maintaining privacy and stability in the face of perturbations becomes more critical in real-world applications. By doing so, we hope to contribute to developing stable and differentially private algorithms, allowing them to generalize effectively and support federated learning [133].

3.2 Generalization

Generalization is the model's ability to make accurate predictions about unseen data drawn from the same distribution as the training data. Generalization is measured by generalization error which is the difference between the training error and the test error. The generalization capability of the algorithms can be improved in three ways:

- Structural stabilization: This approach is based on adjusting the number of free parameters to control bias in the network. In deep learning tasks, structural stabilization is done by changing the number of hidden units or pruning the weights in the architecture.
- Regularization: Controls the variance by applying modifications to the cost function and adding a penalty term.

- Random noise injection: Empirical studies have shown that additive noise improves generalization in deep neural networks. Adding random noise behaves as a form of regularization, which prevents the model from getting too complex and memorizing the input data.

In deep neural networks, generalization is impacted by the complexity and capacity of the model.

Rademacher Complexity

Rademacher complexity [134, 135] is a great tool for measuring the complexity of a learning algorithm. Rademacher complexity is a quantitative way of measuring the complexity of a hypothesis class based on its ability to learn the random noise within the data and minimize the gap between the empirical risk and the true risk [136, 137, 138].

Definition 2. *Assuming that S is a set of data sampled from the same distribution, with input x_i and label y_i, $S = ((x_1, y_1), ...(x_m, y_m))$, then the hypothesis class H is the set of functions that maps input x_i to y_i. The empirical Rademacher complexity of H over S is defined as:*

$$\mathscr{R}_S(H) = E_\gamma \left[max_{h \in H} \frac{1}{N} \sum_{i=1}^{N} \gamma_i h(x_i) \right] \tag{3.2.1}$$

where,

$$\gamma_i = \begin{cases} 1 & \textit{With probability 0.5} \\ -1 & \textit{With probability 0.5} \end{cases} \tag{3.2.2}$$

In this equation, E_γ is the expectation over the Rademacher random variable γ.

Rademacher random variable behaves similarly to a coin flip. Assuming that $S\prime$ is a ghost sample from the same distribution as S, the labels are flipped using the

Rademacher random variable, which acts as introducing random noise into the data. The goal is to find a function that minimizes the gap between the true and empirical risks while classifying the new sample $S\prime$. Rademacher complexity evaluates the classifier's success in minimizing the gap between the empirical and true risks, denoted as $R(H) - \hat{R}(H)$. The idea behind Rademacher complexity is that maximizing the correlations between the output of the hypothesis and labels is equivalent to minimizing the training error in the presence of the Rademacher random variable. Empirical studies show that the correlation is more significant when the hypothesis space is more complex. Rademacher complexity measures the trade-off between the model's capacity to learn noise and generalizing to unseen data. Higher Rademacher complexity indicates that the classifier is better at memorizing the noise and more prone to overfitting. We can decrease model complexity by controlling the capacity to avoid this issue.

Vapnik-Chervonenkis (VC) Dimension

Model capacity, quantified by the VC dimension (Vapnik-Chervonenkis dimension) [139], is the network's ability to capture the underlying patterns and learn the intricate relationships within the data.

Definition 3. *VC dimension of a set of functions is the largest set of finite data points that can be classified perfectly by the classifier. Hence, the training error of the model is zero. In other words, it is the maximum number of data points the classifier shatters in all possible ways.*

Classifiers with higher VC dimensions have higher capacity [140, 141]. Focusing on neural networks as learning algorithms, the model's capacity is correlated with the number and depth of fully connected layers and the interplay between the architecture and the non-linear activation functions [142]. Deep neural networks with multiple layers and millions of parameters have high capacity and VC dimension [143].

High model capacity indicates that the model is capable of memorizing details from the training data and possibly overfitting when facing unseen data. Conversely, low model capacity results in an oversimplified model failing to fit the data properly. So, selecting the right architecture with sufficient model capacity is critical in deep learning.

Figure 3.1 summarizes the interconnections between these concepts and how they influence each other in the context of deep learning and training with noise.

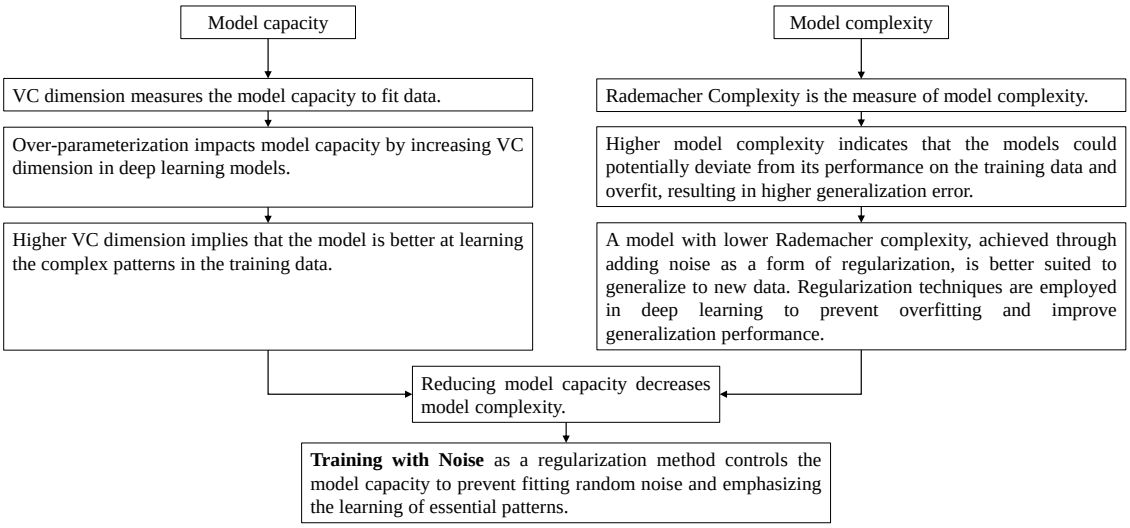


Figure 3.1: The relationships between the VC dimension and Rademacher complexity allow for a more unified understanding of algorithm behaviors in nondeterministic circumstances in the presence of noise and the conditions leading to improved generalization.

3.3 Stability

Stability is an essential property for learning algorithms. An algorithm is stable if the output of the algorithm doesn't change much when the training set is altered by one point, regardless of the sample size [144, 145]

Definition 4. *Let us assume we have two datasets, S and S'. S is composed of elements $x_1, x_2, ..., x_n$ while S' contains the same elements as S but has an additional point x'. Now, for learning algorithm denoted as h, the loss function at any specific point x is represented by $L_x(h)$. A learning algorithm is said to be uniformly stable if, for every point in the dataset equation 3.3.1 holds.*

$$\forall x \in S, |L_x(h_s) - L_x(h_{S'})| \leq \beta \tag{3.3.1}$$

This stability coefficient β in this equation is the smallest value that quantifies the difference in performance of the algorithm on the two datasets at any point. If β is smaller, it means the algorithm is more stable and consistent in its performance across datasets that differ by just one point." .

Stability is closely related to the model's generalization ability on unseen data. [146] define the notion of stability for learning algorithms and demonstrate that stability is an algorithmic way of measuring generalization. Stable models are less prone to overfitting and have better generalization.

Stability is critical in designing practical learning algorithms, and a sensitivity analysis is the means to measure stability. This method, also known as perturbation analysis, is conducted by measuring the changes in the algorithm output in the presence of noise. Perturbation analysis allows us to utilize noise to design models capable of learning the underlying systems that produce data rather than the data itself [147]. Sensitivity analysis is an essential component in defining generalization, stability, and differential privacy.

3.4 Differential Privacy

One of the most stringent measures of privacy is differential privacy, which ensures that adding or removing any individual's data does not change the probability of an outcome by "too much". The definition of differential privacy relies on the concept of a randomized algorithm, which has been employed in various applications, including cryptography and accelerating solutions of algebraic equations. Randomized algorithms are computational procedures that incorporate random choices or probabilistic decisions to solve problems. Rather than following a deterministic path, these algorithms leverage randomness, either to simplify the process or to achieve a solution with high probability. For example, a randomized algorithm can use a random event, such as flipping a coin as part of its description, and make decisions based on the outcome of the coin flips. Therefore a randomized algorithm maps inputs to probabilities of different outputs rather than deterministically mapping inputs to specific outputs. A key benefit of differential privacy is providing mathematically rigorous privacy guarantees. Therefore, any particular algorithm's privacy protection level is clearly understood. The mathematical definition of privacy provides a measurable term for evaluating and maintaining privacy [148, 149, 150, 151].

Definition 5. *Let's assume S and S' are two datasets.*

Datasets are perceived as a multiset of rows, so the distance between the datasets can be measured by the Hamming distance; that is, the difference in the number of rows between S and S', denoted as $||S - S\prime||_1$.

M is a randomized mechanism with domain $\mathbb{N}^{|S|}$.

Q is the set of outcomes of M; therefore, $Q \in Range(M)$.

Differential privacy is defined on two neighboring datasets. S and $S\prime$ are two neighboring datasets if the two datasets differ by only one sample (row). Hence, for all S and

S', the l_1 distance is $||S - S'||_1 \leq 1$. M is ϵ-Differentially Private (ϵ-DP) if equation 5 holds for any two neighboring datasets derived from the dataset:

$$P[M(S) \in Q] \leq exp(\epsilon)P[M(S') \in Q] \tag{3.4.1}$$

This definition is the strict definition of ϵ-DP, and it has been studied explicitly in the book published on differential privacy by Dwork and Roth [152]. Differential privacy can be adjusted using a parameter that measures the desired privacy levels. In this definition, ϵ is a very small value known as privacy loss or leakage. ϵ determines the acceptable change in the output of the mechanism due to the inclusion or removal of any individual, so information learned about the individual as a result of participating in the dataset is limited.

A relaxed version of this definition, currently used in most applications of differential privacy, is (ϵ, δ)-DP provided in Equation 3.4.2.

$$P[M(S) \in Q] \leq exp(\epsilon)P[M(S\prime) \in Q] + \delta \tag{3.4.2}$$

In this definition, δ is the probability of leaking more information than what ϵ claims. δ is preferably zero or a very small value, typically the inverse polynomial of the sample size. This implies that a larger sample size reduces the risk of unintentional disclosure of private information resulting from a query. To achieve (ϵ, δ)-DP, additive noise is conditioned on the type of noise we are adding, the desired ϵ and δ, the sample size, the number of queries performed on the database, and the desired accuracy.

In differential privacy, computations involving noise safeguard personal data and prevent it from being reverse-engineered from the results [153]. However, leaking private information due to statistical queries and machine learning models compromises privacy [154]. Sensitivity is used to monitor this leakage of information.

Definition 6. *Sensitivity is the maximum change in the output of a query as a result of removing an individual from the database.*

Sensitivity is measured based on the distance between the output of mechanism M on the neighboring datasets S and S', where $||S - S'||_1 \leq 1$. Sensitivity is defined as:

$$Sensitivity = max||M(S) - M(S')||_1 \tag{3.4.3}$$

Sensitivity helps characterize the impact of individual data on the output, while ϵ quantifies the upper bound on the level of privacy protection that the algorithm can guarantee.

In practice, differentially private algorithms are required to randomize the query or training model output by adding noise before publicly communicating it with other users. Under differential privacy, we must carefully choose where to add noise and select the appropriate type and amount. A common approach is adding noise sampled from a Gaussian distribution with a mean of $\mu = 0$ and a standard deviation of σ. A higher noise level provides stronger privacy guarantees. We can design private models that abide by the definition of differential privacy and are restricted under the desired privacy guarantees. In recent years, differential privacy has been widely used in the federated learning framework.

3.5 Highlights

Understanding the intricacies of machine learning models' ability to generalize is rooted in several key concepts. The main takeaways of this section for deep learning and privacy are provided.

1. Interplay of VC dimension, Rademacher complexity, stability, and generalization: The notions of VC dimension, Rademacher complexity, and stability are closely intertwined and essential to the model's generalization ability.

- Rademacher complexity and stability encapsulate the algorithm's behavior towards noise in the data. While stability measures the changes in the model output in the presence of noise, Rademacher complexity quantifies the model's ability to learn the random noise in the data, and it is upper bounded by the VC dimension.

- Research by Ron and Kearns [155] on the connection between VC dimension and stability indicates that for algorithms with finite VC dimensions, stability is bounded by the VC dimensions.

- Studies on the relationship between VC dimension and Rademacher complexity in deep neural network models by Neyshabour et al. [156] and Karpinski and Macintyre [141] suggest that VC dimension, Rademacher complexity, and the number of parameters are equivalent. Hence, the number of model parameters determines the model capacity.

- Deep learning models are said to be over-parameterized if the number of parameters is significantly larger than the number of available data points in the training set.

- Over-parameterized models are more prone to overfitting due to increased model capacity.

- Large, diverse data can mitigate the risk of overfitting caused by over-parameterization. The abundance of data allows the model to learn the underlying patterns beyond the noise and perform well on unseen data. In situations with limited data,

regularization techniques can be employed to prevent overfitting and enhance the generalization capability of a model. Regularization techniques control the variance by modifying the cost function and applying a penalty term.

- Bishop [157] demonstrated that the regularization term is written as a Tikhonov regularizer in a simple neural network architecture with one input and one output. Tikhonov regularization is often referred to as ridge regression or l_2 regularization in machine learning.

- Bishop [158] also highlights that training with noise is a form of regularization in neural network models. His findings and the research by others, such as Shalev-Shwartz and Ben-David [159], suggest that regularization results in stable algorithms.

Careful regularization and architectural choices are essential to finding the balance between model complexity, stability, and generalization. Research Question 1 aims to explore this intricate balance further and provide insights on how to improve it.

2. Stability and Differential Privacy:

- Stability is a desirable property in machine learning models, as it ensures that minor changes in the input do not result in drastic changes in the output predictions.

- The definition of differential privacy inherently aligns with stability. Maximizing stability in algorithms offers stronger privacy protection guarantees under differential privacy.

- A potential drawback of differential privacy is its negative impact on accuracy due to introducing noise during training. Excessive noise during training can disrupt the data and cause loss of information, leading to reduced model performance.

Careful tuning of the noise parameters is a critical step in training with noise. The optimal amount of noise can vary depending on factors such as the problem, the data, and the desired properties of the training model. Research Question 2 aims to provide solutions that can help improve the tuning process and enable the selection of an optimal amount of noise for a given problem and dataset.

3.6 Training with Noise in Deep Neural Networks

Noise infusion has been studied in various domains. This phenomenon, known as *stochastic resonance*, employs Gaussian noise to enhance the system's signal detection capabilities [160, 161, 162]. The idea of stochastic resonance dates back to the early 1980s when Benzi et al. [163, 164] introduced the phenomena and investigated its effect on complex systems. Figure 3.2 demonstrates the impact of Gaussian noise on amplifying the weak signals.

When the noise magnitude is small, additive noise enhances weak signals and improves the system's ability to identify useful data without negatively impacting the input. It also helps biological systems to adapt and learn from noisy environments [165]. Stochastic resonance has a wide range of applications in science and engineering, from neuroscience to biological processes, signal processing, and information transmission. Numerous studies focus on the benefits of additive noise in pattern recognition in the nervous system and how it applies to computational neural network settings [166, 167].

Adding noise to a dataset alters the output of the queries. Figure 3.3 demonstrates the impact of input noise on two images taken from the CIFAR-10 dataset. The input

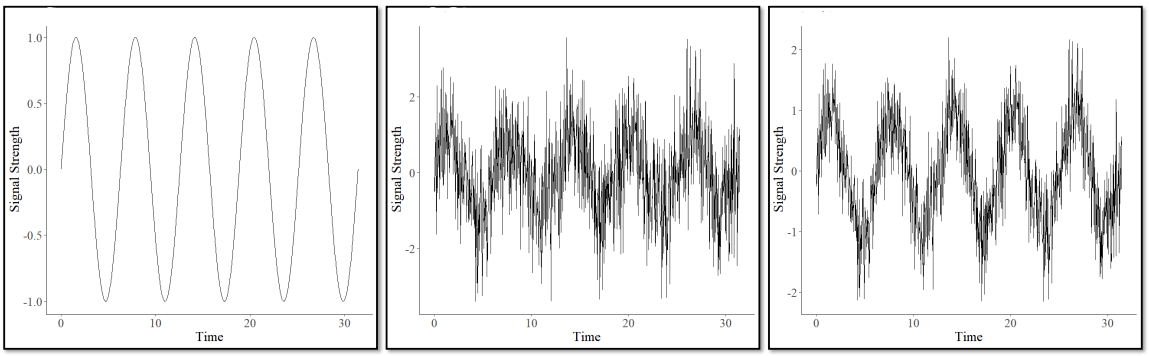


Figure 3.2: The additive noise impacts the weak signals. The signal becomes more distinguishable after stochastic resonance.

noise is implemented by adding a random value sampled from the Gaussian distribution with a standard deviation of σ during training.

It can be observed that the images can absorb different noise magnitudes before they are completely corrupted. The problem specifications, data, and training models contribute to determining the appropriate noise level for training.

Deep neural networks can learn the complex relationships in the data, making them well-suited for tasks such as image and speech recognition, natural language processing, and many other applications in artificial intelligence and machine learning.

Despite their popularity, they are not a silver bullet that can solve all problems in artificial intelligence. Deep learning models are notoriously data-hungry and require a large amount of data to train on. Therefore, their performance relies on the intricacy of the problem and the data, model architecture, and optimization techniques. Their sensitivity to changes in the data distribution and complexity of the model architecture

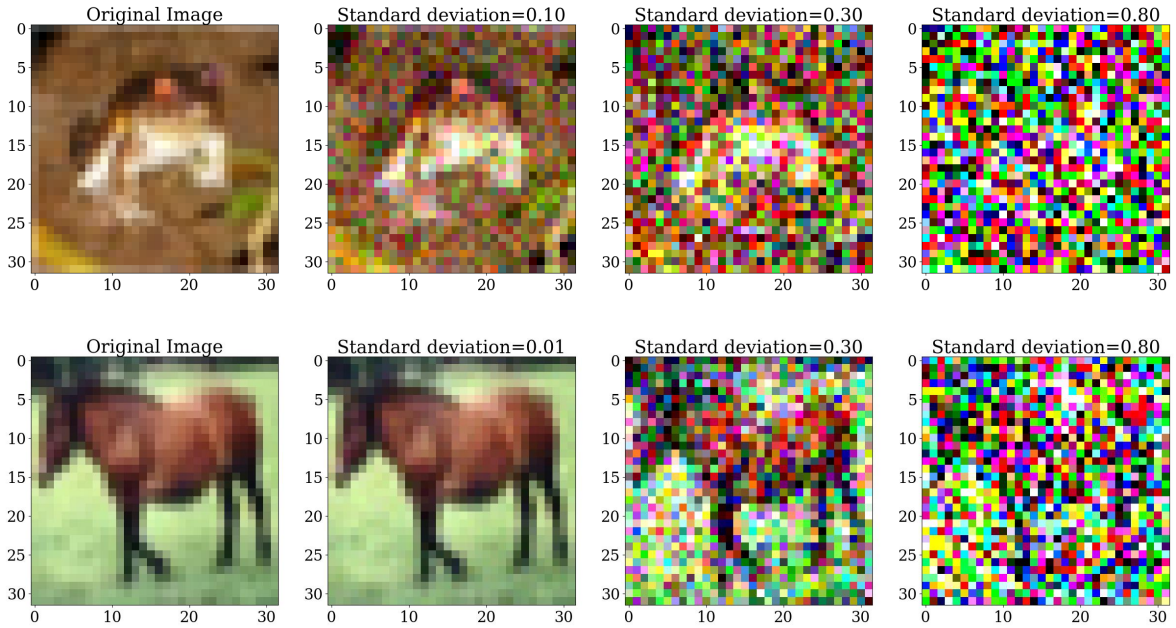


Figure 3.3: Deep Neural Network Architectures

affects their ability to identify critical information rather than memorize the data. Failure in learning leads to overfitting the data.

The benefits of adding noise during training are but are not limited to the following:

1. Handle noisy data as a result of measurement errors or corrupted data [168, 169]

2. Handle inadequate training data for training: Noise infusion is an effective data augmentation method [170, 171]. Noise infusion schemes help diversify the data collected on edge devices to improve the distributed learning results [172].

3. Reduce overfitting and improve generalization: Empirical studies demonstrate that additive noise improves generalization in deep neural networks by preventing the model from getting too complex and memorizing the input data. [173, 174, 175, 176]. Hardt et al. [177] demonstrated that stochastic gradient descent is uniformly stable and that generalization error is a function of the number of iterations. Training with noise prevents overfitting, resulting in better

generalization and fewer epochs needed during training with stochastic gradient descent. DP-SGD [178] is a modified version of stochastic gradient descent. This algorithm is useful for federated learning scenarios where direct access to the data is not feasible. It is (ϵ, δ)-differentially private and has been shown to be stable and generalize well when the model is sufficiently large. DP-SGD is optimized to train in fewer iterations, making it an efficient and effective choice for privacy-preserving machine learning.

4. Improve robustness of the neural network model against adversarial noise: [179, 180, 181].

In deep learning, we can introduce noise into the algorithm by perturbing input, labels, gradients, weights, or the network's architecture.

Table 3.1 presents some of the studies on noise infusion mechanisms in deep learning.

Table 3.1: Noise infusion mechanisms in deep learning literature

Noise Infusion Mechanisms	References
Input	[182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192]
Hidden Layers	[193, 194]
Model Weights	[195, 196, 197, 198]
Gradients	[199, 200, 201, 202, 203]
Labels	[204, 205, 206, 207]

The choice of the amount of noise and noise infusion mechanism is critical in designing an efficient model with the desired stability and generalization ability.

In this chapter, we explore various noise infusion mechanisms for image classification using CNN. CNNs are a class of deep learning models designed primarily for

processing and analyzing visual data, such as images and videos. They have revolutionized computer vision and have found widespread applications in various fields. In addition to their success in computer vision, CNNs have also been adapted for natural language processing tasks like sentiment analysis and text classification, as well as in medical imaging for disease diagnosis and treatment planning. Variations of CNNs include architectures like LeNet, AlexNet, VGGNet, and the highly efficient MobileNet. Transfer learning techniques have further extended the applicability of CNNs by enabling the reuse of pre-trained models on new tasks with limited data. CNN uses convolutional layers to automatically learn hierarchical features from input data, making them well-suited for tasks like image classification, object detection, and facial recognition.

Other than CNNs, other variations of neural networks are designed. Feed-forward neural networks, with layers of interconnected neurons, are used for tasks such as regression, classification, and function approximation. Recurrent Neural Networks (RNNs) introduce loops in the network, making them ideal for sequential data, including natural language processing, speech recognition, and time series analysis. Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) architectures have improved the ability of RNNs to capture long-range dependencies. Generative Adversarial Networks (GANs) are designed for generative tasks like image and video generation, while Transformer-based architectures have revolutionized NLP, enabling models like BERT and GPT-3 to achieve state-of-the-art results in tasks such as language translation and text generation. The field of neural network research continues to evolve, pushing the boundaries of what is possible in machine perception and understanding.

3.6.1 Signal-to-Noise Ratio

Signal-to-Noise Ratio (SNR) quantifies the clarity of the desired signal in the presence of noise in the signal processing domain. The idea of SNR is closely related to stochastic resonance, in which additive noise enhances weak signals [208, 209]. SNR is defined as:

$$SNR = 10 \times \log_{10} \frac{Signal\ power}{Noise\ power} \tag{3.6.1}$$

The definitions of signal power and noise power are as follows:

Definition 7. ***S****ignal power refers to the power of the desired signal, which is the information or data being transmitted or received. Mathematically, it is calculated as the average or mean squared value of the signal.*
In the case of a discrete signal ($s[n]$), which has values for only discrete points in time, the signal power P_s is represented as follows: N: The number of samples taken for computation from a snapshot of the signal over an arbitrary time duration,

$$P_s = lim_{N\to\infty} \frac{1}{2N+1} \sum_{n=-N}^{N} |s[n]|^2 \tag{3.6.2}$$

Definition 8. *Noise power represents the power of the unwanted signal or interference, which corrupts the desired signal. Similar to signal power, noise power is often calculated as the average or mean squared value of the noise.*

Similar to the signal power, for a discrete noise $n[n]$, the noise power is represented as:

$$P_n = lim_{N\to\infty} \frac{1}{2N+1} \sum_{n=-N}^{N} |n[n]|^2 \tag{3.6.3}$$

While signal power measures the "strength" or "magnitude" of the signal, signal variance (denoted as σ_s^2) measures how much the signal values deviate from the mean (μ_s). It provides an indication of the "spread" or "dispersion" of the signal values

around their average. In the general case, the relationship between the variance and power for a signal with a non-zero mean is:

$$\sigma_s^2 = P_s - \mu_s^2 \tag{3.6.4}$$

The signal variance for discrete Signals:

$$\sigma_s^2 = \frac{1}{N}\sum_{n=1}^{N}(s[n] - \mu_s)^2 \tag{3.6.5}$$

In the case in which the mean of the signal is zero, the power is equivalent to the signal variance. The same computations can be applied to the noisy signal.

SNR, often expressed in decibels, is sensitive to the scale of the noise and signal in the system. Higher SNR indicates the signal is of high quality and is easier to identify from noise. Conversely, when SNR is low, the signal is weak, or the system is too noisy, and distinguishing the true signal from noise is more challenging. We redefine SNR using the signal variance and noise variance as:

$$SNR = 10 \times \log_{10} \frac{Signal\ variance}{Noise\ variance} \tag{3.6.6}$$

SNR is used as a metric to evaluate the strength of the signal in the presence of noise and achieve optimal performance. In this context, using signal variance over signal power offers certain advantages:

- Variance captures the fluctuations of the signal around its mean. In the case of CNN models, the variance provides an understanding of the model's confidence or consistency in its responses. By focusing on variance, the model's behavior is tied directly to the properties of noise. A higher noise variance indicates that the model is more uncertain and less stable in the presence of noise.

- Variance is a normalized measure, making it a relative metric. This can be advantageous when comparing the performance of different models or the same model under varying noise conditions, as it ensures that the measure is scaled and comparable.

Computing SNR based on the model output is a quantitative tool for evaluating the model's performance in detecting useful information (signal) from unwanted variations (noise) in the data. SNR allows us to observe the changes in the output and find the noise level that meets the desired trade-off between accuracy, stability, and generalization. Understanding the impact of noise during training provides a guideline for determining the privacy budget without concerns about the quality of results. Leveraging noise to improve stability and generalization without sacrificing performance leads to stronger privacy protection strategies against adversaries.

In the context of CNN, the signal represents the true underlying patterns that the model is trying to capture, and noise is any internal or external variation, perturbation, or distortion in the data that affects the model's ability to detect the signal. The formal definition of signal and noise is provided:

Definition 9. *A signal is the validation accuracy of the base model (model without noise).*

Definition 10. *Noise is defined as the difference between the base model's validation accuracy and the perturbed model's validation accuracy. A model is perturbed by introducing a randomly generated value from the Gaussian distribution with a standard deviation of* σ.

Using validation accuracy obtained from the noisy and clean data provides a more reliable assessment of how well a model handles noise and generalizes to new, challenging conditions. Training accuracy tends to overstate performance, while test accuracy

is reserved for final evaluation and should not be influenced by noise during model development.

Algorithm 2 presents the pseudo-code of computing SNR. The choice of the noise infusion scenario relies on the problem's complexity and dataset.

Algorithm 2 Computation of Signal-to-Noise Ratio (SNR)

Require: Validation accuracy of the base model (base model output)
Require: Validation accuracy of the perturbed model (perturbed model output)
Ensure: SNR
  procedure COMPUTE_SNR (base model output, perturbed model output)
  Store the base model output in variable *Signal*
  Calculate the difference between base model output and perturbed model output, store it in variable *Noise*
  Calculate the variance of *Signal*, store it in variable *Signal variance*
  Calculate the variance of *Noise*, store it in variable *Noise variance*
  Calculate SNR as $10 \times \log_{10}(\frac{Signal\ variance}{Noise\ variance})$
  return SNR
  end procedure

The choice of the noise infusion scenario relies on the problem's complexity and dataset. In classification, higher SNR values indicate that the model is capable of predicting values that are closer to the true signal and have less noise interference. The lower SNR values suggest that the noise is more dominant, resulting in less accurate predictions by the model. The noise level that yields the maximum SNR is preferable because it identifies the noise level where the model can most extract useful information from noise, leading to better generalization of unseen data.

3.6.2 Price of Stability & Price of Anarchy

Originally used for the analysis of network and routing games, the Price of Stability (PoS) and Price of Anarchy (PoA) measure the efficiency of outcomes in decentralized systems [210, 211]. PoS compares the outcome achieved by self-interested agents to

the socially optimal solution. PoA compares the worst-case outcome achieved by self-interested agents to the socially optimal solution. We propose to define the image classification process as a game where the players are Gaussian noise-infused CNNs under various noise levels. For N players, and $i = 1, ..., n$, the standard deviation of the Gaussian noise of the i^{th} player is σ_i, where $\sigma_i \in [0, 1]$. Suppose the ideal scenario is training the model without noise (base model denoted as CNN_{σ_0}). PoS is defined as:

$$Price\ of\ Stability\ (PoS_i) = \frac{Test\ accuracy\ of\ CNN_{\sigma_i}}{Test\ accuracy\ of\ CNN_{\sigma_0}} \tag{3.6.7}$$

By comparing against the base model, we can assess how training with noise impacts the prediction results of the test data.

- The PoS of the base model is always 1.
- If PoS = 1, the model's sensitivity to noise is minimal. The noisy model is performing similarly to the base model. It also suggests that the model is relatively stable across different noise levels.
- If PoS $>$ 1, the noisy model performs better than the base model. It suggests that additive noise improves the model's generalization on unseen data. Therefore, test accuracy has improved in the presence of noise.
- If PoS $<$ 1, the noisy model performs worse than the base model. Smaller PoS suggests a lack of stability in the presence of noise. The model has less potential for privacy-preserving applications.

The PoA is defined as:

$$Price\ of\ Anarchy\ (PoA_i) = \frac{Test\ loss\ of\ CNN_{\sigma_i}}{Test\ loss\ of\ CNN_{\sigma_0}} \tag{3.6.8}$$

- The PoA of the base model is always 1.
- If PoA = 1, the model is able to identify the patterns in the data, even in noisy conditions.
- If PoA $>$ 1, the model is negatively impacted by the noise, and it loses useful information, so the noisy model performs worse than the base model.
- If PoA $<$ 1, the model performs better than the base model, and the additive noise has improved the model's generalization on unseen data.

The proposed metrics provide insights into the effect of noise on the models' accuracy, loss, and overall stability. The metrics also offer a clear reference point to monitor the changes in the models' generalization and efficiency of predictions on test data.

3.7 Computational Results

In this section, we explore the use of noise as a means of improving generalization, stability, and privacy in deep neural networks. This is particularly important when data is distributed across multiple devices and access to sufficient data for training is limited. We aim to design stable and differentially private deep learning models that can generalize well in centralized and federated learning settings while preserving privacy. To achieve this goal, we will compare various methods of designing algorithms that can perform well in the presence of noise and evaluate their effectiveness for image classification. We will build upon the foundational work of Zhang et al. [212] and expand their findings through our experimentation.

We start the experiments by selecting the appropriate CNN architecture. As mentioned earlier, the VC dimension is the measure of the model's expressive power and is often used to analyze the model's capacity to fit data. Training large CNN models

with millions of trainable parameters requires significant computation resources and careful fine-tuning of the hyperparameters.

We use CIFAR-10, a well-known benchmark dataset for image classification, where 40,000 images are used for training, 10,000 images for validation, and 10,000 for testing. The experiments are designed around three network architectures with different model capacities determined by the number of parameters in the neural network architecture provided in Table 3.2.

Table 3.2: The models vary in the number of trainable parameters, a factor of model capacity that impacts the model's ability to generalize on unseen data. Model 3 is over-parameterized

Architecture	Trainable Param #	Non-trainable Param #	Total
Model 1	22,784,938	1,920	22,786,858
Model 2	2,396,330	1,896	2,397,226
Model 3	43,415,850	3,968	43,411,882

The CNN models are modifications of VGG-19 [213], and the key layers are the 2D convolutional, batch normalization, 2D max pooling, dropout, and dense layers. The architecture details for models 1, 2, and 3 are available in Tables 3.3, 3.4, and 3.5, respectively. The parameters of the CNN are configured as a batch size of 64, a learning rate of 0.001, and a momentum of 0.9. The local models are trained for 80 epochs. The three models with different numbers of parameters are compared in their efficiency of prediction, generalization, and stability under different noise levels and noise infusion mechanisms.

Table 3.3: Architecture for Mode 1

Layer Type	Output Shape	Param #
Conv2D	(32, 32, 32)	896
BatchNormalization	(32, 32, 32)	128
Conv2D	(32, 32, 32)	9248
BatchNormalization	(32, 32, 32)	128
MaxPooling2D	(16, 16, 32)	N/A
Dropout	(16, 16, 32)	N/A
Conv2D	(16, 16, 64)	18496
BatchNormalization	(16, 16, 64)	256
Conv2D	(16, 16, 64)	36928
BatchNormalization	(16, 16, 64)	256
MaxPooling2D	(8, 8, 64)	N/A
Dropout	(8, 8, 64)	N/A
Conv2D	(8, 8, 128)	73856
BatchNormalization	(8, 8, 128)	512
Conv2D	(8, 8, 128)	147584

BatchNormalization	(8, 8, 128)	512
MaxPooling2D	(4, 4, 128)	N/A
Dropout	(4, 4, 128)	N/A
Conv2D	(4, 4, 256)	295168
BatchNormalization	(4, 4, 256)	1024
Conv2D	(4, 4, 256)	590080
BatchNormalization	(4, 4, 256)	1024
Conv2D	(4, 4, 256)	590080
MaxPooling2D	(2, 2, 256)	N/A
Dropout	(2, 2, 256)	N/A
Flatten	(1024,)	N/A
Dense	(4096,)	4,198,400
Dropout	(4096,)	N/A
Dense	(4096,)	16,781,312
Dense	(10,)	40,970

Table 3.4: Architecture for Mode 2

Layer Type	Output Shape	Param #
Conv2D	(32, 32, 32)	896
BatchNormalization	(32, 32, 32)	128
Conv2D	(32, 32, 32)	9248
BatchNormalization	(32, 32, 32)	128
MaxPooling2D	(16, 16, 32)	N/A
Conv2D	(16, 16, 64)	18496
BatchNormalization	(16, 16, 64)	256
Conv2D	(16, 16, 64)	36928
BatchNormalization	(16, 16, 64)	256
MaxPooling2D	(8, 8, 64)	N/A
Conv2D	(8, 8, 128)	73856
BatchNormalization	(8, 8, 128)	512
Conv2D	(8, 8, 128)	147584
BatchNormalization	(8, 8, 128)	512
MaxPooling2D	(4, 4, 128)	N/A

Flatten	(2048,)	N/A
Dropout	(2048,)	N/A
Dense	(1024,)	2,098,176
Dropout	(1024,)	N/A
Dense	(10,)	10,250

Table 3.5: Architecture for Mode 3

Layer Type	Output Shape	Param #
Conv2D	(32, 32, 32)	896
BatchNormalization	(32, 32, 32)	128
Conv2D	(32, 32, 32)	9248
BatchNormalization	(32, 32, 32)	128
MaxPooling2D	(16, 16, 32)	N/A
Conv2D	(16, 16, 64)	18496
BatchNormalization	(16, 16, 64)	256
Conv2D	(16, 16, 64)	36928
BatchNormalization	(16, 16, 64)	256

MaxPooling2D	(8, 8, 64)	N/A
Conv2D	(8, 8, 128)	73856
BatchNormalization	(8, 8, 128)	512
Conv2D	(8, 8, 128)	147584
BatchNormalization	(8, 8, 128)	512
MaxPooling2D	(4, 4, 128)	N/A
Conv2D	(4, 4, 256)	295168
BatchNormalization	(4, 4, 256)	1024
Conv2D	(4, 4, 256)	590080
BatchNormalization	(4, 4, 256)	1024
Conv2D	(4, 4, 256)	590080
MaxPooling2D	(2, 2, 256)	N/A
Dropout	(2, 2, 256)	N/A
Conv2D	(2, 2, 512)	1,180,160
BatchNormalization	(2, 2, 512)	2048
Conv2D	(2, 2, 512)	2,359,808
BatchNormalization	(2, 2, 512)	2048

Conv2D	(2, 2, 512)	2,359,808
MaxPooling2D	(1, 1, 512)	N/A
Dropout	(1, 1, 512)	N/A
Flatten	(512,)	N/A
Dense	(4096,)	2,101,248
Dropout	(4096,)	N/A
Dense	(8192,)	33,562,624
Dense	(10,)	81,930

3.7.1 CNN with Gaussian noise hidden layers in a Centralized Setting

Leveraging the properties of training with noise, we design a CNN with Gaussian noise hidden layers, an innovative approach to enhance the robustness and generalization capabilities of deep learning models. In this design illustrated in Figure3.4, Gaussian noise is intentionally added as a form of regularization to hidden layers within the CNN architecture.

Training with Gaussian noise hidden layers involves inserting uncorrelated layers of Gaussian noise that will add a randomly generated value within the range of the specified standard deviation to the activation of the previous layer during training. Uncorrelated noise sources are statistically independent. Training with Gaussian noise

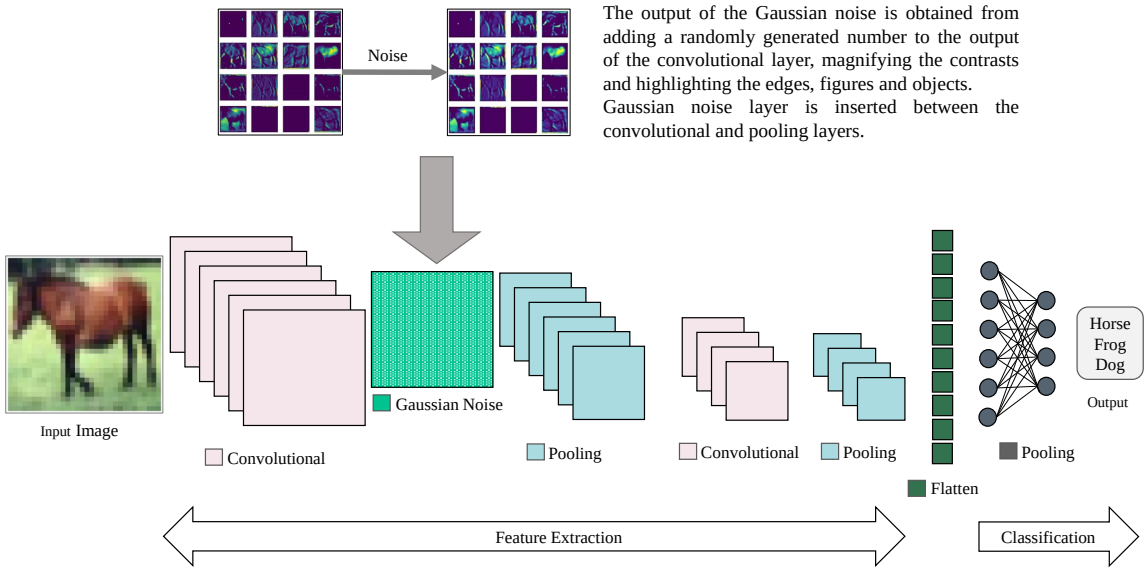


Figure 3.4: A simplified illustration of the CNN architecture with Gaussian noise layer.

hidden layers involves inserting uncorrelated layers of Gaussian noise that will add

a randomly generated value within the range of the specified standard deviation to the activation of the previous layer during training. Uncorrelated noise sources are statistically independent.

In the first set of experiments, we evaluate the performance of three CNN models with Gaussian noise hidden layers presented in Figure 3.5.

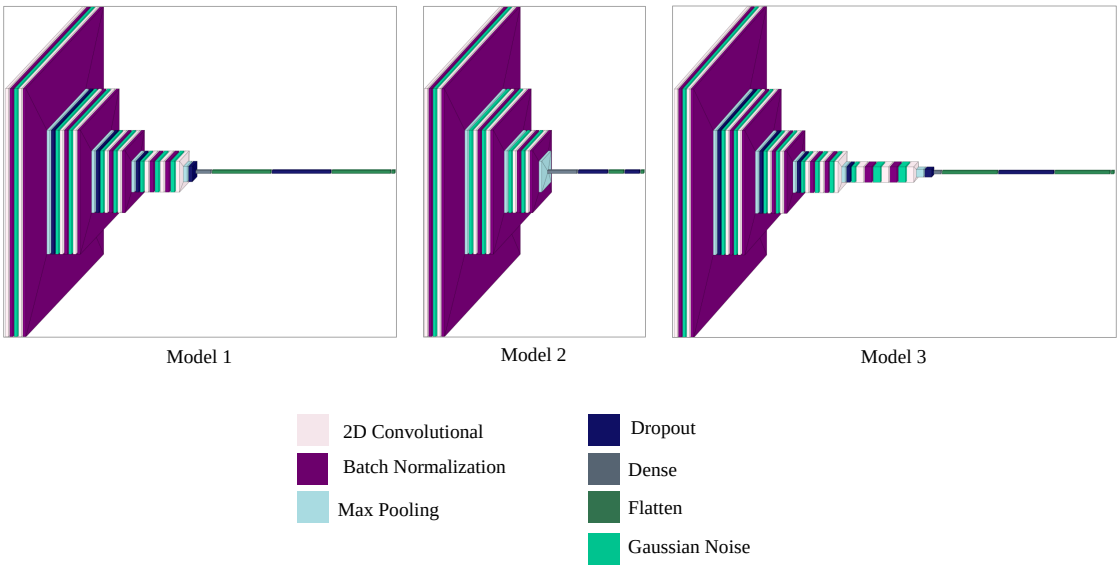


Figure 3.5: Visual representation of the CNN models with Gaussian noise layers.

For the implementation, we insert the noise layers before the convolutional layers, followed by a batch normalization layer. Let us assume:

x: The output of the layer before the convolutional layer

z: A randomly generated number from Gaussian distribution with mean, $\mu = 0$ and standard deviation of σ

$x\prime$: The output of the Gaussian noise layer, $x\prime = x + z$.

x'': The output of the batch normalization layer obtained after passing the output of the convolutional layer through a batch normalization layer:

$$x'' = (\frac{x' - \mu'}{\sigma'}) * \alpha + \alpha' \tag{3.7.1}$$

Where μ' and σ' are the mean and standard deviation of the neuron's output of the activation function in the convolutional layer, and α and α' are trainable parameters used for rescaling and shifting the values from the previous operations. As the training continues, the data goes through multiple blocks of Gaussian noise, convolutional, and batch normalization layers. Batch normalization prevents the accumulation of noise throughout the network.

Figure 3.6 compares accuracy and loss obtained from training the models under different noise levels in a centralized framework. The standard deviation is selected from Gaussian distribution with 20 levels between $\{0, 1\}$. Setting the standard deviation to zero refers to the base model.

Models 1 and 3 offer similar trends; as noise increases, the accuracy drops, and loss increases further from the base model. In models 1 and 3, the optimal test accuracy and loss are achieved when σ are 0.32 and 0.21, respectively. The drop in performance as a result of increasing the noise suggests that the models have difficulty fitting the noisy data when σ is high.

Unlike models 1 and 3, model 2 can maintain consistent performance with noisy data, suggesting that the model is the most stable among the three. In model 2, the optimal test accuracy and loss are achieved when σ is 0.58, which is significantly higher than in models 1 and 3. While all three models yield the optimal accuracy of approximately 0.82, maintaining a high accuracy and loss in the presence of higher noise levels demonstrates that model 2 is better at generalizing to unseen data. Compared

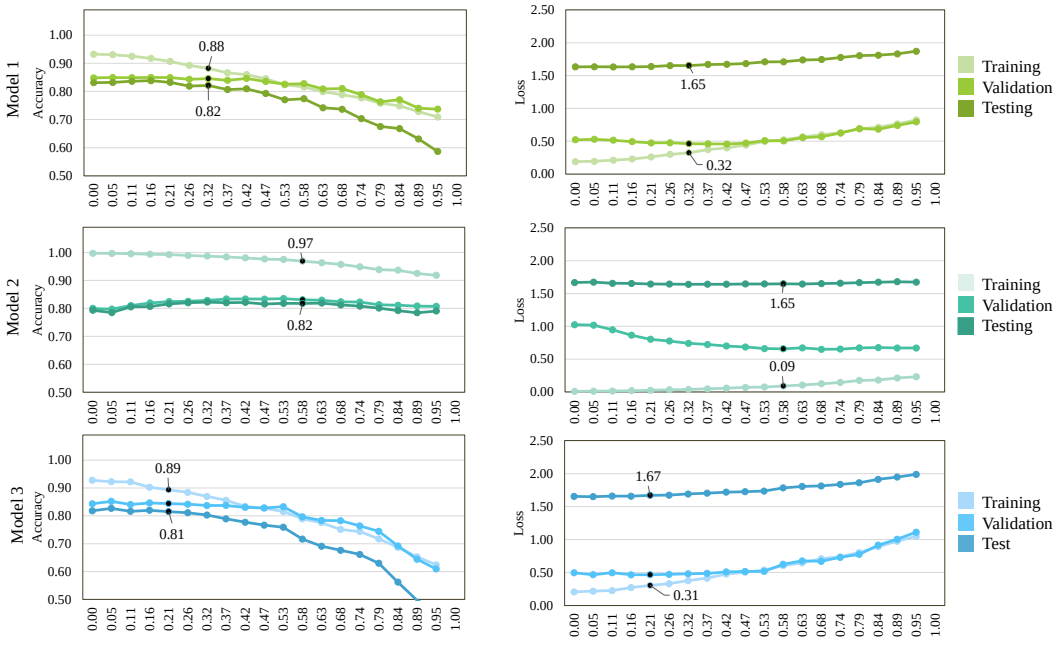


Figure 3.6: The optimal test accuracy and loss value are marked with the associated training accuracy and loss. Stable models that perform well at higher noise levels are better candidates for federated learning.

to models 1 and 3, model 2 experiences a less rapid performance degradation at higher noise levels.

Often, better privacy guarantees are achieved at the expense of worse accuracy and loss, so we strive to find a systematic way to reach a balance between accuracy and privacy. However, the balance is not possible without fine-tuning the noise level during training while monitoring its impact on test data. To this end, we explore SNR, PoS, and PoA to measure the trade-off between performance efficiency and privacy under noise. Figure 3.7 demonstrates the SNR, PoS, and PoA values for the three models with Gaussian noise hidden layers (σ between 0 and 1).

Since the range of SNR is problem-dependent, we focus on the fluctuations of SNR at different noise levels to compare the models.

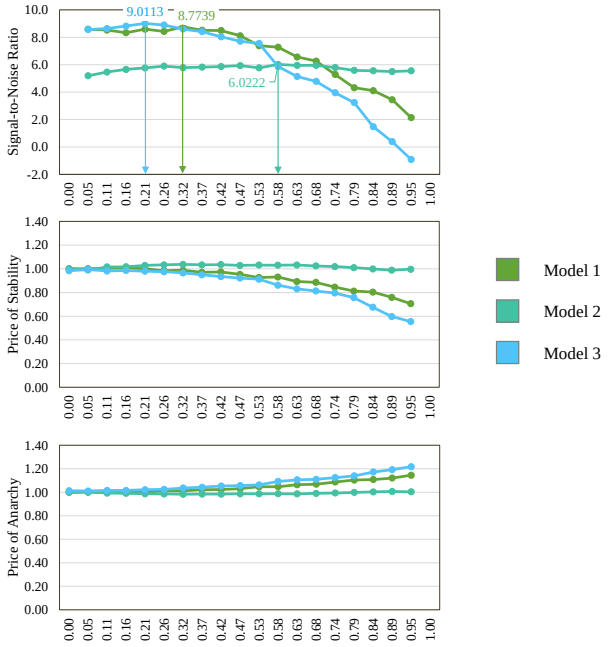


Figure 3.7: Increasing the noise levels decreases model utility. However, stable models suffer less as the noise levels are heightened, offering consistent performance under higher noise levels.

In models 1 and 3, the value of SNR is initially higher but drops significantly as we increase the noise. This means at lower noise levels, the model is effective in distinguishing the signal, but as noise increases, the model becomes overwhelmed and can not handle noise effectively. However, model 2 stands out as having relatively consistent SNR values at higher noise levels. This implies the model's ability to remain relatively stable, even in the presence of higher noise levels.

Training the models at the noise level that maximizes SNR provides the highest test accuracy and sets the balance between stability and accuracy in the presence of noise. Under differential privacy, the maximum SNR guarantees privacy without loss of accuracy. While finding the balance is ideal, in federated learning, privacy is prioritized over accuracy when dealing with sensitive data. PoS and PoA measure the impact of noise on test accuracy and loss compared to the base model. In Model 2, the PoS and PoA remain consistent despite the increase in the noise level. Model 2 offers a

trade-off between performance and privacy, where accuracy and loss are stable under higher noise levels. In model 2, while the optimal SNR identifies the noise level for the perfect balance between accuracy and privacy at 0.58, we can further increase the noise, and the accuracy degrades by less than 4%. Model 2 is a potential candidate for cases where privacy and stability take precedence over achieving the highest accuracy, such as federated learning applications.

Ultimately, selecting the appropriate model depends on the specifics and requirements of the problem, whether it prioritizes accuracy, privacy, or stability. These analyses provide insights into the trade-offs and strengths of each model under different noise levels.

Overall, a comparison of the performance of the three models under various noise conditions measured by SNR, PoS, and PoA suggests that in models with higher stability, PoS and PoA remain relatively consistent. Given the overlap between the definitions of stability and privacy, we can conclude that models with relatively consistent PoS and PoA can provide better privacy protection guarantees without drastic degradation of accuracy.

3.7.2 CNN with Multiple Gaussian Noise Layers vs. a Single Layer

When an image is passed through the convolutional layers, the network learns different complex features of the image, such as the edges and the texture. The network learns patterns and objects from the later convolutional layers as training continues. We use feature visualization to gain insight into the learning procedure of a CNN with Gaussian noise hidden layers inserted before the convolutional layer, focusing on the

first layers of model 1. Figure 3.8 is a visual representation of the output of the first two convolutional layers of model 1, where a single image is fed into the network.

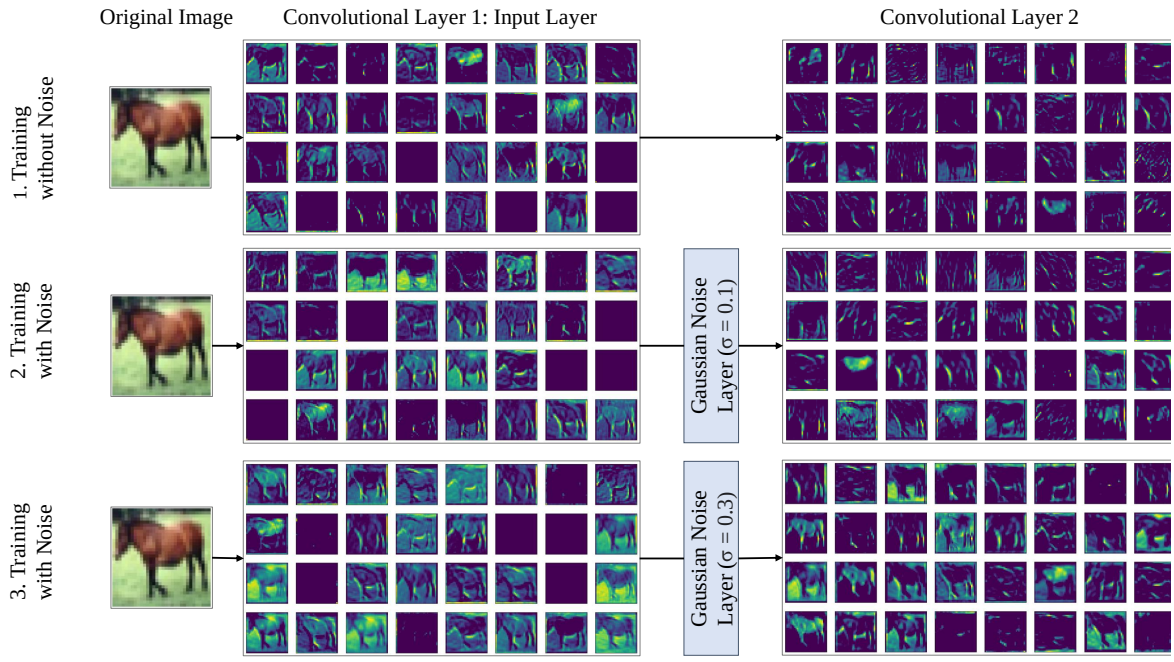


Figure 3.8: CNN feature maps

The first two convolutional layers have 32 filters. The figure includes three sets of feature maps from the initial training steps extracted from the model without noise and the noisy model, where a noise layer is inserted before the second convolutional layer. The first column represents the feature maps from the input layer of CNN models. The slight variations in the maps are due to the inherent variations in training a neural network model. The lower layers of the CNN are responsible for learning the edges and textures in the image. The bright spots on the feature map indicate that the region was most activated in its corresponding map in the prior layer.

In training with noise, we utilize the idea of stochastic resonance and use noise to enhance weak signals. Figure 3.9 is a closer look at the feature map. For a relatively

similar map in layer 1, the noise-infused maps in the second and third rows have led to better identification of edges, and more key regions are activated.

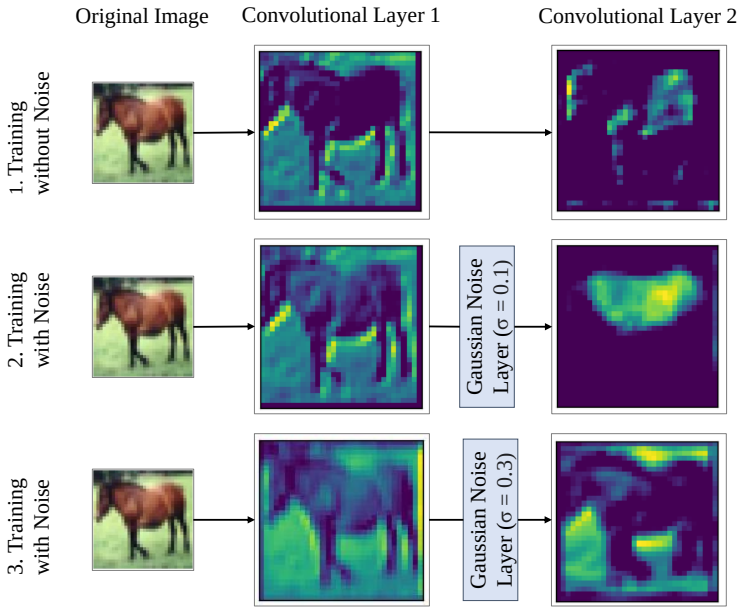


Figure 3.9: The optimal noise level improves generalization by helping the deep learning model better distinguish the objects during training.

We emphasize that uncorrelated noise sources are critical when designing CNN models with Gaussian noise hidden layers. If noise layers are correlated, we must consider different magnitudes and phase variations when combining the additive noise. We can ensure that the noise layers are uncorrelated by assigning a unique random seed at each layer. Derived from the signal processing conventions, we can compute the total additive noise in the system for multiple statistically independent noise sources.

Let us assume that G_1 and G_2 are two uncorrelated noise layers, with standard deviations σ_1 and σ_2, respectively.

$$Variance(G_1, G_2) = Variance(G_1) + Variance(G_2) + 2\rho * CoVariance(G_1, G_2) \quad (3.7.2)$$

Since the noise layers are uncorrelated, $\rho = 0$, and,

$$Variance(G_1 + G_2) = \sigma_1^2 + \sigma_2^2. \tag{3.7.3}$$

In a CNN model with N uncorrelated Gaussian noise hidden layers, the total noise introduced by multiple noise layers with standard deviation σ is equivalent to a single noise layer with a standard deviation of:

$$\sigma_{Total} = \sqrt{N}\sigma \tag{3.7.4}$$

Table 3.6 presents the results from training models with multiple noisy layers vs. a single noise layer. For comparison, we set $\sigma = 0.1$ when training all three models with multiple noise layers shown in Figure 3.5. The standard deviation of the model with a single noise layer is computed based on Equation 3.7.2.

The obtained results suggest that the number of layers does not affect the model performance. In this framework, the controlling parameter is the standard deviation of the added noise. Training the models with multiple noise layers allows us to fine-tune the standard deviation of the noise generated at the layers and adjust the model according to the problem specifications and data at hand to achieve optimal performance.

3.7.3 Gaussian Noise Hidden Layers in Federated Setting

Extending the experiments to federated learning, we explore the effect of different noise levels and compare the results with the centralized models. Choosing the Gaussian noise magnitude is critical because it determines the level of privacy. A lower noise level will result in a more accurate CNN model but will also provide weaker privacy guarantees. It is important to note that, in practice, it is possible for an attacker to

Table 3.6: Training deep learning models with a single Gaussian noise hidden layer versus multiple layers.

Model 1			
Model Type	Standard Deviation	Train Accuracy	Test Accuracy
Base model	0.0	0.93	0.82
Multiple noise layers	0.1	0.92	0.83
Single layer substitute	0.28	0.92	0.83
Model 2			
Model Type	Standard Deviation	Train Accuracy	Test Accuracy
Base model	0.0	0.99	0.79
Multiple noise layers	0.1	0.99	0.80
Single layer substitute	0.22	0.99	0.80
Model 3			
Model Type	Standard Deviation	Train Accuracy	Test Accuracy
Base model	0.0	0.92	0.82
Multiple noise layers	0.1	0.91	0.83
Single layer substitute	0.33	0.92	0.83

learn sensitive information about the training data by exploiting vulnerabilities in the model or the training process. Therefore, it is important to take additional steps to protect the privacy of the training data, such as using secure training environments and encryption. Using horizontal partitioning, the data is randomly and equally split between 3 arbitrary clients.

First, the models were trained locally with 20 noise levels between $\{0, 1\}$, and SNR was computed. The noise level that yields the optimal SNR for the clients and the results from training the federated learning model with optimized noise obtained from maximizing SNR are presented in Table 3.7. The federated learning models are trained for 20 communication rounds at different noise levels. Global accuracy and global loss

are measured for evaluation. It can be observed that despite significant differences in size, the models vary by a maximum of 3% in global accuracy, while global loss remains relatively consistent.

Table 3.7: The standard deviation of the additive noise is set based on the optimal SNR.

Architecture	Client 1	Client 2	Client 3	Global Loss	Global Accuracy
Model 1	0.21	0.26	0.16	1.60	0.87
Model 2	0.53	0.37	0.16	1.63	0.84
Model 3	0.11	0.16	0.16	1.60	0.86

In the next step, we trained the model at five noise levels, the results of which are shown in 3.8. Training the models with Gaussian noise hidden layers significantly improves the model stability.

As seen in Figure 3.7, it is possible to add higher noise levels to improve privacy guarantees, and the global accuracy and loss remain relatively constant with varying noise levels.

The analysis suggests that deep learning models are relatively noise-stable in federated settings. The models can learn the patterns of the data and the added noise while preserving privacy. The stability of the models in federated learning is beneficial as it increases the model's threshold for added noise, ensuring that privacy is maintained. Increasing the standard deviation of Gaussian noise, which acts as a regularization method, also improves the overall accuracy of test data.

Table 3.8: The standard deviation of the additive noise is fixed across all clients.

Global Accuracy					
Architecture	σ =0.1	σ =0.3	σ =0.5	σ =0.7	σ =0.9
Model 1	0.86	0.86	0.87	0.86	0.86
Model 2	0.78	0.83	0.84	0.84	0.86
Model 3	0.85	0.86	0.87	0.87	0.85
Global Loss					
Architecture	σ =0.1	σ =0.3	σ =0.5	σ =0.7	σ =0.9
Model 1	1.60	1.61	1.60	1.61	1.62
Model 2	1.68	1.64	1.63	1.63	1.62
Model 3	1.61	1.61	1.61	1.63	1.72

3.7.4 Comparison of Noise Infusion Mechanisms

The choice of noise infusion mechanism plays a crucial role in enhancing deep learning models' generalization, stability, and privacy. This section compares the impact of noise infusion schemes mentioned in Table 3.1.

- Noisy input: The input noise is implemented by adding a random value sampled from the Gaussian distribution in the predefined standard deviation range to the input data during training. Input noise behaves as a data augmentation method, often used to expand the input sample or introduce randomness in the data to reduce overfitting. However, if the noise level is too high, it can distort the data and lead to the model learning incorrect patterns.

- Noisy network weights: To introduce noise to model weights, the noise is directly added to the weights retrieved from the model.

- Noisy gradients: Noise is added to the original gradients. The modified gradients are then used to update the model weights during training.

- Noisy labels: For noisy labels, the random value is added to the labels before training. We also included noise clipping to ensure the labels were within the correct range to avoid extreme changes and too much distortion in the labels.

In this section, We explore the effectiveness of different mechanisms and compare their results with those of Gaussian noise hidden layers. We train the centralized CNN models using five noise infusion mechanisms where the standard deviation of the additive noise is consistently set at 0.1. The results are presented in Figure 3.10.

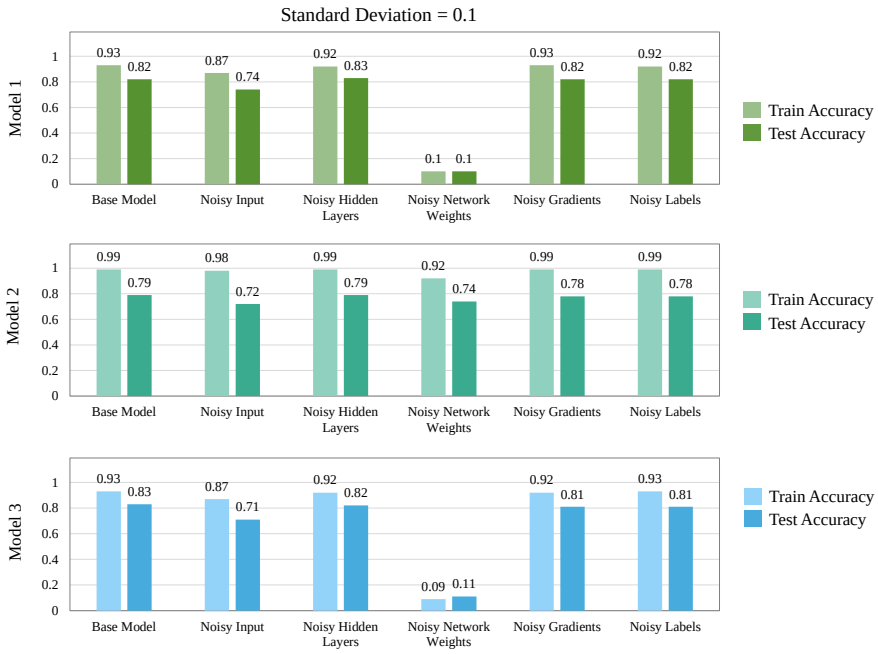


Figure 3.10: This figure presents a comparison of the training and test accuracy of three models across six mechanisms. The first set of columns for each figure represents the base model trained without noise.

The base model serves as a control group without additive noise. Models 1 and 3 are most sensitive to injecting noise into input and weights, significantly dropping training and test accuracy.

The models with noisy weights also fail to learn effectively and generalize, which indicates the detrimental impact of noisy weights on training. While there is a slight

decrease in training and test accuracy, models trained with Gaussian noise hidden layers, labels, and gradients are less sensitive to noise. Model 2 is the most stable among the three, and the decrease in the accuracy is less significant. When the added noise's standard deviation is 0.1, Gaussian noise hidden layers, noisy gradients, and noisy labels are the most resilient. Hence, we continue studying these models under varying noise levels.

The results from training the centralized data with Gaussian noise hidden layers, noisy gradients, and noisy labels using the three models are presented in Figure 3.11. The noise levels are $\sigma = \{0.1, 0.3, 0.5, 0.7, 0.9\}$.

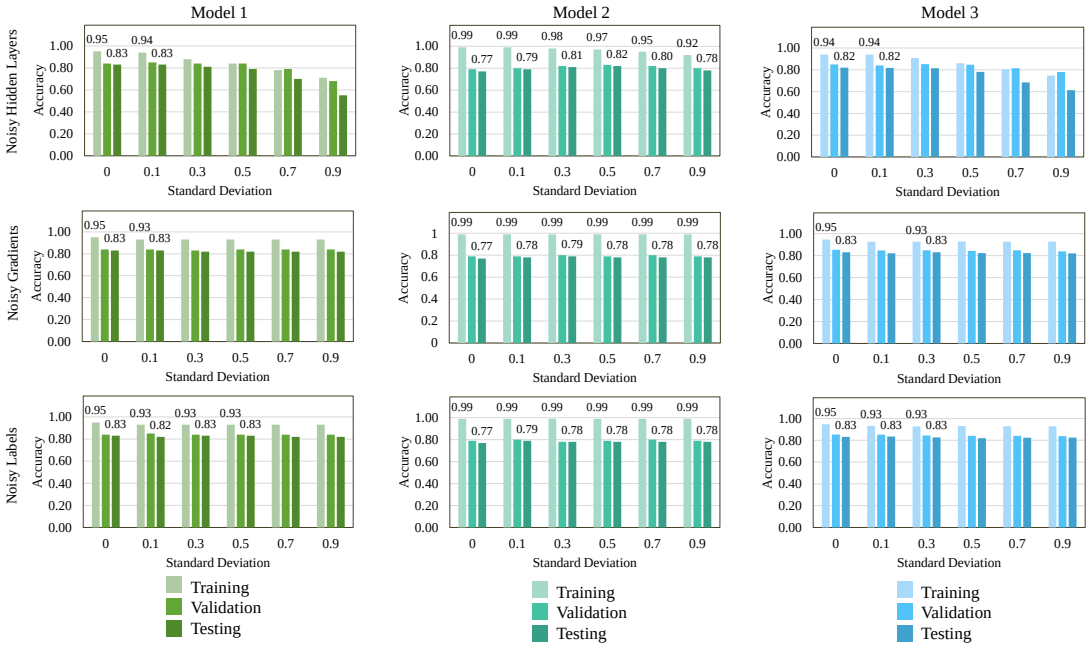


Figure 3.11: Results of training and evaluating models with top 3 noise infusion methods and varying noise levels.

While all models are somewhat sensitive to additive noise, they exhibit different performance variations under the noise infusion mechanisms. We can increase the noise in the models with Gaussian noise hidden layers while preserving test accuracy, especially in model 2, where the accuracy remains relatively constant compared to the

base model. It is also interesting to see that increasing noise improves test accuracy, indicating the regularization effect of noise. In models 1 and 3, we can increase the noise levels to 0.1, and the test accuracy is 0.83 and 0.82, respectively.

The models exhibit similar performance with noisy gradients. With models 1 and 3, the test accuracy gradually decreases as we increase the noise. However, even with a high standard deviation, model 2 remains stable against additive noise. Models 1 and 3, trained on data with noisy labels, have better stability, and we can increase the standard deviation to 0.5 and 0.3, respectively. Model 2 performs equally well when trained with noisy labels compared to the base model, demonstrating resilience to label noise.

Noise can negatively impact both training and test accuracy. However, the impact of noise on model performance varies depending on the noise infusion mechanism and the standard deviation of the additive noise. While the model's performance gradually degrades, noise can be used as a regularization technique. The results indicate that models with Gaussian noise hidden layers are effective in remaining stable even when the standard deviation of the noise is high.

3.8 Chapter Summary

The present work is an empirical study on the role of noise in enhancing generalization, stability, and privacy within CNN for image classification. Through a series of carefully designed experiments, we observed that the introduction of noise during training helps prevent overfitting by making the model less reliant on precise features and patterns in the training data. By encouraging the network to learn more robust representations of data, CNNs with Gaussian noise hidden layers tend to perform better on unseen or noisy data, making them particularly useful for tasks where the input data may contain

variations or uncertainties. This technique can improve CNN's ability to handle real-world scenarios and noisy environments, making it a valuable tool in applications such as image recognition, denoising, and signal processing.

The experimental results demonstrate that when the model is not over-parameterized, perturbing the parameters associated with the deep learning model by adding Gaussian noise behaves as an implicit regularization technique. Models trained with noise generalize better, achieve higher accuracy, and are more stable in centralized and federated settings.

Introducing SNR as a measure of the signal quality (the base model performance) relative to the noise (noisy model performance) serves as a powerful tool for balancing accuracy and privacy in privacy-preserving settings. Additionally, PoS and PoA provide an in-depth understanding of the interplay of utility, stability, and privacy under different conditions. PoS and PoA can be used as tangible metrics for assessing the trade-off between privacy and accuracy in privacy-aware machine learning.

Furthermore, we conducted a comparative analysis over CNN-based image classification noise infusion scenarios to determine the most effective methods of enhancing generalization, stability, and privacy. This investigation particularly benefits federated learning, where higher noise levels offer stronger privacy guarantees.

This study has significant implications for practical machine learning applications that require reliable performance under varying conditions. Noise-infused models can help achieve models capable of handling diverse and noisy datasets.

In the context of federated learning, understanding the impact of noise leads to designing computationally efficient private models. The findings of this study demonstrate the potential of noise as a privacy-enhancing mechanism that can empower individuals and organizations to make informed decisions regarding data sharing and model deployment. By incorporating privacy-preserving techniques and acknowledging

privacy as a fundamental human right, this research contributes to the responsible and ethical use of data and machine learning technologies.

Chapter 4

Federated Imbalanced Learning

4.1 Beyond Localized Weather Predictions

Employing federated learning for the classification of weather data introduces a new paradigm, particularly in scenarios where data privacy, decentralized data sources, and efficient utilization of localized data are critical. Local meteorology stations frequently collect weather data from measurements and radar observations in tabular and image formats.

Federated learning enables models to be trained directly on local devices or stations where the data resides, eliminating the need to transmit sensitive or voluminous data to a central location. Federated learning also allows private data to be monitored and protected by local data centers. Each local model learns from its respective data. Then, only the model updates (not the data) are shared with a global model, ensuring data privacy and reducing communication costs. The applications of federated learning have been extended to weather forecasting and air quality control using historical data and edge devices [214, 215]. This collaborative yet decentralized learning method is crucial for weather prediction due to the inherently localized nature of weather events and the potential sensitivity of data. Machine learning has long been used in weather applications to predict weather conditions such as rain or strong winds [216] to improve lead time for severe weather warnings, such as tornadoes [217]. Federated learning

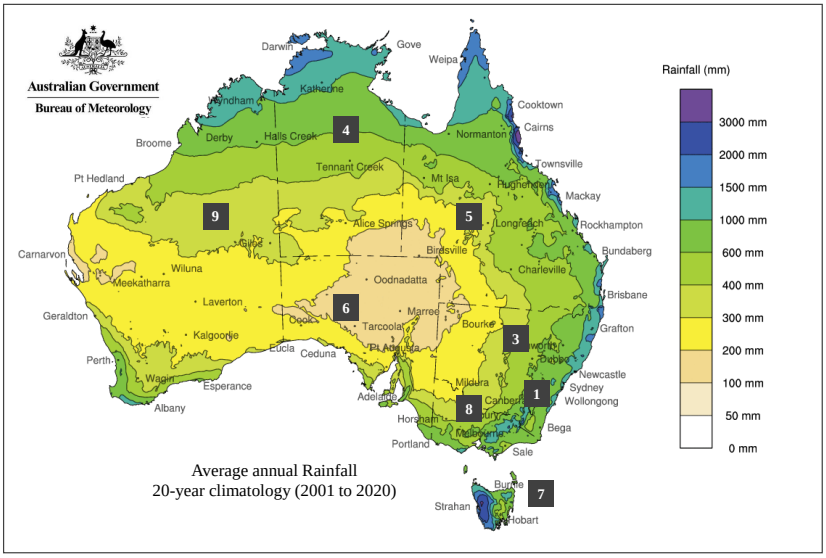
provides the model with a diverse and comprehensive dataset acquired from various local environments and conditions.

Given Australia's climatic and geographical diversity, the weather stations record a broad spectrum of meteorological patterns, providing models with insight into various scales and types of weather phenomena. Precise and localized weather forecasting is crucial for various applications, from agriculture to urban planning. To this end, we conduct an experimental study focusing on predicting rainy versus non-rainy days through deep learning models. The dichotomy of rainy and non-rainy days establishes a clear classification problem where the model is trained to discern the atmospheric variables that result in precipitation. The real-time system collects the data for this study at the Bureau of Meteorology in Australia. The dataset consists of 140672 observations and 13 features stored in 9 stations. The data contains approximately ten years of daily observations from 2007 to 2017, recorded twice daily from the eight mainland regions and Australian offshore islands [1]. Figure 4.1 demonstrates the 20-year average rainfall measured annually in the eight major regions across Australia.

The challenges in the classification problem arise especially in regions where rainy days are sparse or seasonally confined. Central and southern regions have experienced less rainfall than northern and eastern coastal areas, resulting in varying ratios of rainy versus non-rainy days between the regions. With geographical and temporal variations in weather patterns, especially across a diverse continent like Australia, the issue of imbalanced learning is more likely to present itself in the significant difference in the number of observations in each class. Table 4.1 presents the distribution of observations in both datasets distinguished by the class label.

[1] Australian Government Bureau of Meteorology
http://www.bom.gov.au/climate/data-services/

1. Australian Capital Territory
2. Australian Offshore Islands
3. New South Wales
4. Northern Territory
5. Queensland
6. South Australia
7. Tasmania
8. Victoria
9. Western Australia



http://www.bom.gov.au/climate/maps/averages/decadal-rainfall/

Figure 4.1: Multi-decadal rainfall averages map presents the rainfall patterns across the regions in a 20-year period.

Station	Region	Rain	No Rain	Imbalance Ratio
1	Australian Capital Territory	2016	7307	0.22
2	Australia Offshore Islands	919	2045	0.31
3	New South Wales	9305	32027	0.23
4	Northern Territory	1361	6421	0.17
5	Queensland	3513	11600	0.23
6	South Australia	2402	9710	0.20
7	Tasmania	1460	4756	0.23
8	Victoria	7217	23814	0.23
9	Western Australia	3568	11231	0.24

Table 4.1: Distribution of rain and no-rain observations across nine weather stations in Australia, indicating a data imbalance in the regional data.

The two major challenges drawn from the information presented in Table 4.1 are:

1. The significant difference between the number of instances in the classes indicates that the data is imbalanced. Imbalanced data negatively impacts the classifier's performance, resulting in biased predictions.

2. Insufficient data in some local centers negatively affects the accuracy of predictions, resulting in models incapable of generalizing to new and unseen data.

Imbalanced data is a well-known issue in many fields, including weather prediction, and it is an ongoing topic in machine learning research[218, 219]. The relative proportion of classes and the absolute number of available instances in the minority class are important factors. The problem with imbalanced data is magnified when the minority class consists of rare events because there is a lack of general information on the event, leading to biased models. Tornadoes and thunderstorms happen at various frequencies in locations with different climate conditions. The rarity of such events creates imbalances in the data, which requires specialized methods to address this issue. Trafalis et al. [220] proposed a weighted classifier with a random subspace ensemble method to classify tornadic and non-tornadic observations. Predicting the intensity of the damages caused by a tornado is also a challenging problem [221].

When the data is imbalanced, machine learning classifiers fail to learn the underlying patterns within the minority class. Without a significant loss in overall accuracy, the minority class is misclassified. Based on the type of data, the size, and the distribution of the data between classes, the issue can affect the performance in different ways. Cost-sensitive methods are a practical approach to addressing the issue of imbalanced weather data. [222] proposed a novel linear programming Support Vector Machine that outperforms traditional machine learning algorithms in classifying weather data. A lack of adequate information about the minority class causes the problem definition issues [223]. This can cause evaluation metrics such as accuracy and error rate to

fail in representing the minority class. Evaluation is an essential part of the learning process, which is used to assess the generalization ability of the learning method on test data. Appropriate evaluation metrics are necessary for evaluating the quality of learning [224, 225, 226]. The authors of the paper published by Ferri et al. [227] have used experimental and theoretical analysis to compare and rank the evaluation metrics that work best in evaluating the learned model on imbalanced data and analyze the identifiable clusters and relationships between the metrics. These experiments provide recommendations on the metrics that would be more appropriate for any specific application. Evaluation metrics are categorized into three types in the literature: threshold, probability, and ranking metrics [228]. The threshold evaluation metrics are computed based on the confusion matrix. In binary classification, given that samples in the majority class are labeled negative, and the samples in the minority class are labeled positive, the confusion matrix is defined based on four values of True Positive(TP), True Negative(TN), False Positive(FP), and False Negative(FN) calculated based on the actual and predicted values. Note that the definition of a confusion matrix can be extended to multi-class classification.

Figure 4.2 provides an overview of the challenges in imbalanced learning and the approaches that have led to efficacious solutions in this domain.

Accuracy is limited to measuring the overall performance, and it cannot provide enough information to ensure a reliable learning method when the data is imbalanced [229]. *Sensitivity* and *Specificity* are two classification performance metrics for imbalanced learning. Sensitivity is $\frac{\text{TP}}{\text{TP+FN}}$ and summarizes how well the positive class was predicted. Specificity is defined as $\frac{\text{TN}}{\text{TN+FP}}$, and it evaluates how well the negative class was predicted. Geometric mean (G-mean) is an important evaluation metric used explicitly for imbalanced learning scenarios. G-mean considers the harmonic mean of

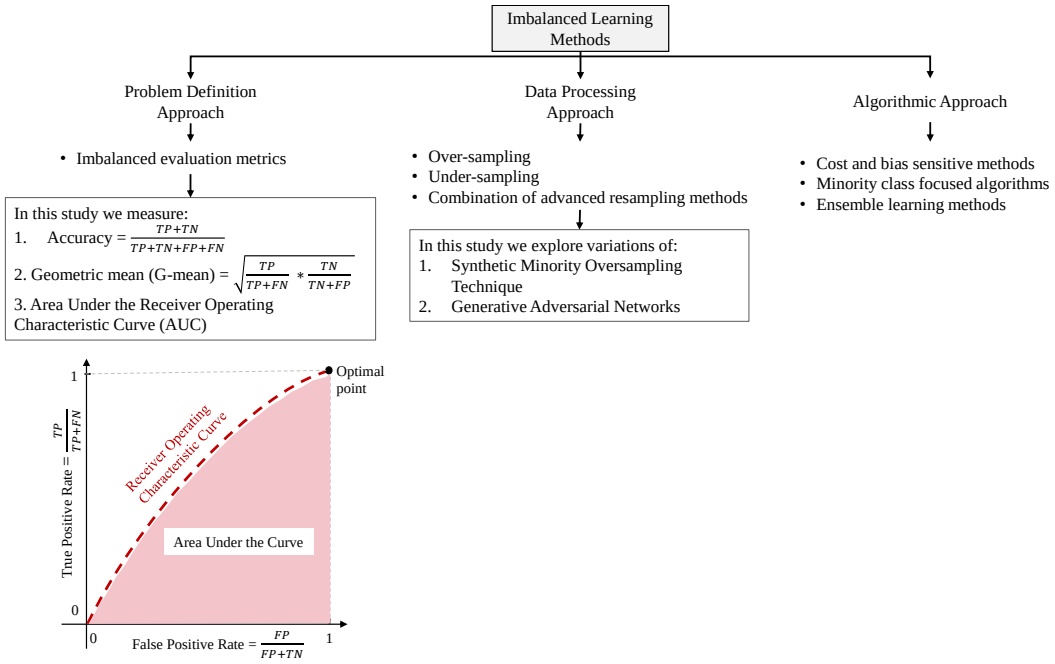


Figure 4.2: Utilizing the appropriate imbalance learning evaluation metrics is a standard practice. In this study, accuracy, AUC, and G-mean are the key tools in the assessment and comparison of the oversampling techniques in both classes.

sensitivity and specificity. A high G-mean indicates that the model performs well in both classes, so we aim to maximize the metrics. The goal of imbalanced learning is to find an optimal classifier that is capable of providing a balanced degree of predictive accuracy for the minority class as well as the majority class [230, 231, 232, 233, 234, 235]. As shown in Figure 4.2, the Receiver Operating Characteristic curve visually represents the classification performance. The Area Under the Curve (AUC) is scale-invariant, so it is a reliable tool for the ranking and comparison of classifiers[236, 237].

Motivated by these challenges, we empirically analyze data augmentation methods to balance the data prior to training in federated learning frameworks. Utilizing data augmentation techniques allows the generation of synthetic data points that mimic the characteristics of actual rainy days, thereby alleviating data scarcity and imbalance issues. Federated learning enables us to leverage the data from multiple centers without accumulating the data in a single facility. Moreover, in the federated learning context,

each station can augment its data locally, ensuring that the synthetic data reflects the local meteorological characteristics and improving the learning of the localized model. We compare the Synthetic Minority Over-sampling Technique (SMOTE), a widely used approach, with Generative Adversarial Learning (GANs) variants. When the local models communicate and contribute to the training of a global model, the model's capability to generalize and accurately predict rainy events improves. This is especially true when the models are trained on the diversified and balanced representation of rainy days across different Australian climates and territories. Therefore, the meteorological predictions become more reliable and representative of the vast and varied Australian landscapes.

In this chapter, We train a deep learning model on a combination of real and synthetic data generated by various methods. With its ability to model complex, non-linear relationships and learn hierarchical features from data, deep learning emerges as a quintessential tool in analyzing meteorological data and uncovering the underlying patterns that cause rainfall. Variables such as humidity, pressure, temperature gradients, and wind patterns are fed into the neural network. The model continuously refines its predictive capability through layered architectures and back-propagation, resulting in a more adaptive and accurate system capable of effectively distinguishing between rainy and non-rainy days. This predictive paradigm enhances meteorological forecasting and provides different sectors with insights to develop strategies and operate in accordance with impending weather conditions.

Access to a large network of data from sources scattered over multiple data centers benefits the deep learning models. Class imbalance appears in many centralized and federated machine learning problems. [238] conducted an experimental study of class imbalance of global performance. Due to the variability of local data distribution among all devices and lack of control over client selection, a class imbalance issue

arises. This results in a slow convergence rate of the global model. To address this issue, [239] proposed an estimation algorithm that can reveal the class distribution without the need to access the distributed data.

Federated learning has been examined for tackling the issue of imbalanced data in multiple settings. Numerous studies have explored federated learning for problems regarding meteorology and agriculture. Manoj [240] applied federated learning to predict agriculture production using weather data, soil data, and crop management data collected from numerous data silos. Their model improves scalability and ensures privacy, which is essential for users of farming devices. Farooq et al. [241] proposed a federated learning model using Long Short-term Memory (LSTM) neural networks to predict flood, outperforming traditional LSTM models.

In this work, we propose using a deep imbalanced learning model to classify weather data stored in 9 weather stations across Australia. We examine the outcome of training the data centralized and federated. In a centralized learning approach, the stations collect and store their data individually, upon which the model is trained. While this conventional method has its merits, particularly in data consistency and straightforward implementation, it neglects potential issues related to smaller sample sizes and lack of adequate information, especially in imbalanced data. Federated learning is a potent alternative, especially in contexts where data privacy, minimized data transfer, and localized learning are essential. Our experiments compare the two approaches and provide insights into their effectiveness in addressing the challenges posed by data privacy, transfer costs, and geographical variations in weather patterns [242].

4.2 Deep Imbalanced Learning

Deep learning is a network of fully or partially connected neurons organized in layers. The neurons receive the information, and activation functions determine the output of the layers and the network. Deep learning architecture varies according to the specific problem, data structure, the network's depth and size, the layers' functionality, and the optimizer. Schmidhuber [243] conducted a comprehensive historical survey of deep learning and its evolution. Johnson and Khoshgoftaar [244] reviewed the existing methods for the issue of imbalanced data in deep learning. Deep learning is a powerful solution to various real-world problems, and when enhanced by other heuristic feature selection and resampling approaches, it can be very effective for imbalanced learning [245]. Bao et al. [246] introduced a deep learning framework to balance the data in a deeply transformed latent space. In this model, feature learning, balancing, and discriminative learning are conducted simultaneously, which has performed effectively on multi-classification problems. Deep imbalanced learning models are capable of learning imbalanced data in the presence of noise and outliers.

Deep learning has remarkable benefits and has been successful in many classification tasks. The advances in artificial intelligence, particularly deep learning, allow us to create robust models for analyzing diverse data types and provide valuable insights. Therefore, it is selected as the classification model in this study. The architecture of the neural network is problem-specific. We used a sequence of dense layers of various sizes. In federated learning, a layer of Gaussian noise is inserted in between as a hidden layer. The added noise guarantees privacy protection under differential privacy and increases generalization. The standard deviation of the Gaussian noise must be tuned along with other parameters. We removed the noise layer in the architecture to classify the data

in a centralized setting. The code snippet for implementing the deep learning model in federated learning is presented in Listing 1.

Listing 1 Classifier

```
# Imports
from keras.layers import GaussianNoise

# Define the model architecture
model = Sequential([
    layers.Dense(256, activation='relu', input_shape=(13,)),
    layers.GaussianNoise(stddev),
    layers.Dense(128, activation='relu'),
    layers.Dense(64, activation='relu'),
    layers.Dense(32, activation='relu'),
    layers.Dense(1, activation='sigmoid')])
```

The batch size is 64, the learning rate is 0.001, and the momentum is 0.9. The standard deviation of the Gaussian noise layer in federated learning architecture is set to 0.01. The model is trained using a Stochastic Gradient Descent optimizer for 30 epochs, and the binary cross-entropy loss function is used. In a federated learning framework, the training is performed in multiple communication rounds, where each round involves the clients sending their model weights to the central server for aggregation. Our experiments suggest that increasing the communication rounds doesn't significantly affect the performance; therefore, the global model is updated for ten communication rounds. In addition to loss and accuracy, the imbalanced learning metrics AUC and G-mean are measured to evaluate the local and global performance. The classification model is implemented using TensorFlow and Keras API in Python. The models are executed in Google Colab Pro with a high-RAM run-time setting.

4.3 Data Augmentation

Unlike traditional resampling methods that remove or replicate the existing data points to balance the data, data augmentation is a data processing approach that artificially increases the amount of data by generating synthetic data points from existing data. Resampling methods follow two strategies: removing instances from the majority class (random under-sampling) [247, 248] and adding new instances to the minority class (random over-sampling) [249].

Data augmentation is a set of advanced resampling techniques that focus on generating new instances rather than replicating or removing the original data. Controlling the number of generated samples improves the imbalance ratio and promotes diversity in the data. Since the samples are generated in the feature space, creating a new sample in a nonlinear space improves the results after resampling the minority class. The performance and effectiveness of the data generation techniques are extensively studied by [250], and the limitations of such methods are investigated by [251]. Data augmentation methods have been effectively utilized for fraud detection [252], malware and bug report [253], healthcare and medical diagnosis [254, 255], and fault detection in manufacturing and machinery [256].

This research study investigates data augmentation for balancing tabular data using variations of GANs models and the SMOTE, two well-known data augmentation approaches. Figure 4.3 presents a high-level description of the algorithms.

Expanding the input data by introducing the samples that represent the original data, combined with the appropriate learning algorithms, strengthens the classification process to attain accurate results for both classes.

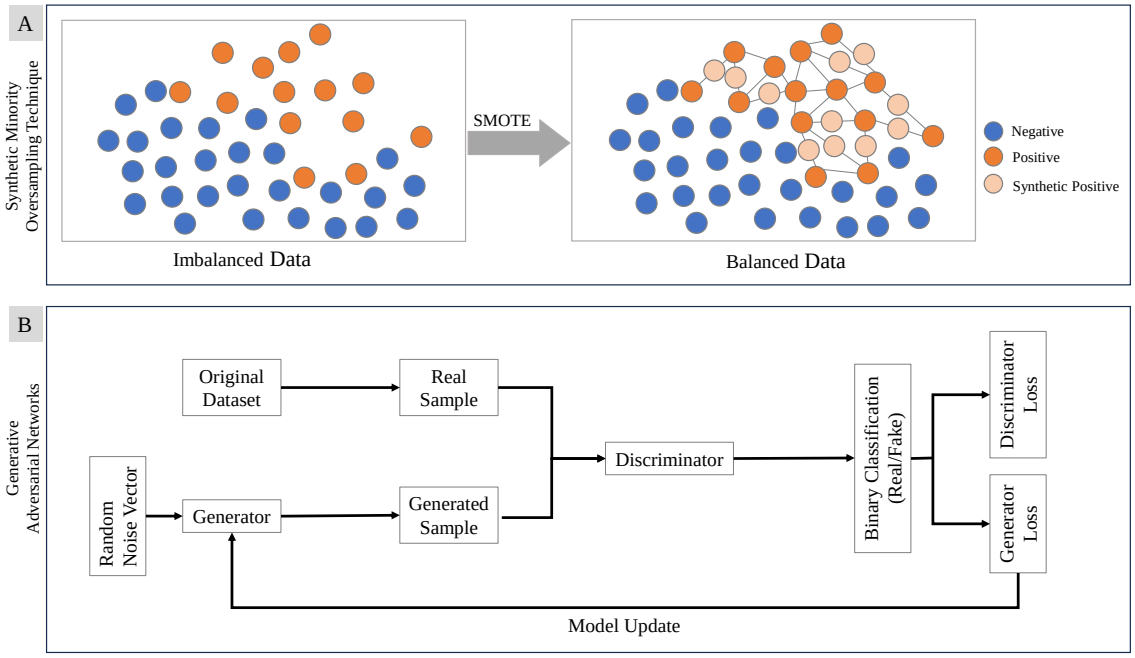


Figure 4.3: SMOTE and variants of GANs expand the sample size by generating new instances of both classes or balancing the data when only generating instances of the minority class.

4.3.1 Synthetic Minority Over-sampling Technique

Over-sampling methods are widely used in imbalanced learning to adjust the data distribution before classification [257, 258]. SMOTE and its variations are among the most popular oversampling methods for tabular data. Chawla et al. [259] combined SMOTE and AdaBoost to enhance training performance by focusing on the most misclassified examples. Borderline-SMOTE starts by identifying the decision boundary between the two classes and then generating samples along the borderline [260]. DeepSMOTE is a novel approach that uses a deep neural network to select the features suitable for generating data with the modified SMOTE algorithm.

In SMOTE, the number of instances in the minority class is increased by syntactically creating new instances instead of replicating the existing instances. As shown in Figure 4.3-A, the new instances are generated based on their nearest neighbors in

the feature space. The new examples are added near the line segment that joins the nearest neighbors of the samples in the minority class [261]. Deep imbalanced learning with SMOTE effectively improves G-mean and AUC [262]. Imb-learn Python library [263] is a practical tool for implementing SMOTE.

4.3.2 Generative Adversarial Networks

Introduced by Goodfellow et al. [264], GANs is an artificial intelligence scheme designed for learning the underlying patterns in the data in unsupervised learning tasks. Generative models utilize the statistical properties of the data and generate new data points by understanding the data distributions through adversarial learning. Adversarial learning is a machine learning mechanism where two networks with competing objectives are trained simultaneously. Figure 4.3-B presents the high-level architecture diagram of GANs models. The term adversarial networks refers to the two competing neural network architectures in GANs known as *Generator* and *Discriminator*. The two networks are trained independently.

First, the generator takes a vector of random values with Gaussian distribution (random noise vector) and creates a set of new samples. Then, the discriminator takes a sample of original and newly generated data as input and attempts to successfully distinguish between fake and original data in a binary classification problem. The iterative game between the two networks continues until the generator can trick the discriminator into failing to identify the fake data from the original data. The generator and discriminator are trying to optimize their loss function in this setting.

Let G be the generator, D be the discriminator, and z' be the vector of Gaussian noise fed into the generator. The generator loss function defined in Equation 4.3.1 measures the binary cross entropy between the output of the discriminator for classifying real and generated data labeled as real.

$$-log(D(G(z'))) \tag{4.3.1}$$

The generator tries to create samples as close as possible to real data. The discriminator loss function defined in Equation 4.3.2 measures the binary cross entropy for outputs of the discriminator labeled both as real and fake for the generated and real data.

$$-log(D(x)) - log(1 - D(G(z'))) \tag{4.3.2}$$

The discriminator tries to classify real and fake data points correctly.

In this two-player game, the two-objective optimization problem is defined as a minimax game with the loss function presented in equation 4.3.3. The iterative learning process helps us reach the Nash equilibrium between the two networks of D and G.

$$min_G max_D V(D, G) = E_x[log(D(x))] + E_{z'}[log(1 - D(G(z')))] \tag{4.3.3}$$

The competition between the two components guides the generator to create artificial points close to the original data that can not be distinguished from the original dataset. The algorithm follows ten steps:

1. Prepare the dataset, including data cleaning, feature engineering, and normalization of input variables.

2. Define the architecture of the generator and discriminator models.

3. Generate the noise vector and synthesize a sample of fake data.

4. Train the discriminator model on a subset of real and fake samples.

5. Clip the discriminator weights to improve stability

6. Freeze the discriminator weights

7. Train the generator using the output from the discriminator as feedback and generating synthetic data.

8. Iteratively train the GANs model by combining the generator and the discriminator in an adversarial process.

9. Evaluate the GANs model on the validation set.

10. Save the trained generator model and generate synthetic samples to balance the minority class.

Choosing the appropriate architectures and tuning the parameters is one of the main challenges of GANs. The code snippets provided in Listings 2 and 3 are the network architectures used in this study. The batch size is 64, the latent dimension is 13, and the models are trained for 100 epochs. We used Adam optimizer and binary cross-entropy for the loss function.

Listing 2 Generator

```
# Define the model architecture
model = Sequential([
    layers.Dense(512, activation='relu', input_shape=input),
    layers.Dense(256, activation='relu'),
    layers.Dense(128, activation='relu'),
    layers.Dense(1, activation='sigmoid')])
```

Listing 3 Discriminator

```
# Define the model architecture
model = Sequential([
    layers.Dense(128, activation='relu', input_shape=input),
    layers.BatchNormalization())
    layers.Dense(64, activation='relu'),
    layers.Dense(1, activation='sigmoid')])
```

Weight clipping is also used for the discriminator network as a regularization technique. This ensures that the magnitude of the weights is within a predefined range. This technique prevents oscillations and improves the algorithm's stability during training.

Numerous studies have explored the use of generative models for handling imbalanced data. GANs have demonstrated outstanding potential in generating data and expanding the sample size with high-quality data close to the original distribution for imbalanced learning problems [265, 266, 267, 268, 269, 270]. Divovic et al. [271] improved the quality of generated samples by providing class label context to the network, and Cho and Kim [272] proposed a genetic algorithm approach to find the optimal combination of imbalanced ratios for implementing GANs and SMOTE. Data augmentation using capsule adversarial networks is also a novel approach that constructs a 2-stage model to generate data and then evaluate the balanced dataset by training a classifier. This ensures that the generated data is of good quality [273, 274]. [275] developed a collaborative framework between the generator and classifier to expand the minority sample size and balance the data gradually. GANs data augmentation algorithms have examined a variety of data such as image [276, 277], and tabular datasets for fraud detection, cancer diagnosis, or weather prediction,[278, 279].

In weather applications, GANs generate weather images using a two-step approach where the data is generated and then classified using an ensemble model [280]. Combining GANs with different learning frameworks and preprocessing methods, such as SMOTE, offers promising potential for real-world applications [281].

4.3.2.1 Conditional GANs

Conditional GANs (CGANs) are an extension of GANs models that are most effective when the generated data is meant to be tailored to the labels or other class conditioning of the input [282]. In conditional GANs, the conditioning variable (label) is fed as an additional parameter into the generator along with the noise vector. The extra information leads the network to produce data corresponding to the label [283]. Conditional GANs are effective in generating detailed and highly accurate images for supervised and semi-supervised learning problems [284, 285, 286].

4.3.2.2 Wasserstein GANs

One of the limitations of traditional GANs is their instability during training, which can lead to mode collapse and generate low-quality samples. Getting stuck with a limited range of samples and lacking diversity negatively affects the dataset. Regularization methods and modifications of GANs have been investigated to mitigate this issue [287]. Wasserstein GANs is a variation of GANs trained with a loss function defined based on the Wasserstein distance. Modifying the loss function improves the stability of the model and the quality of the generated samples [288]. Wasserstein distance is measured by the amount of work required to move mass from one distribution to another. Equation 4.3.4 presents the Wasserstein distance where P_r and P_g are the

real and generated distributions, respectively. $\prod(P_r, P_g)$ is the set of all couplings of P_r and P_g, and θ is the joint probability distribution.

$$W(P_r, P_g) = inf_{\theta \in \prod(P_r, P_g)} E_{(x,y)\sim\theta}[||x - y||] \tag{4.3.4}$$

The generator aims to minimize the distance between the distribution of real and generated data, while the discriminator (often referred to as the critic) tries to maximize the distance. Overall, Wasserstein GANs is a promising approach to address the challenges of generating new images, image-to-image translation, and audio synthesis problems.

In this study, we implement the improved Wasserstein GANs with gradient penalty (WGAN-GP) introduced by Gulrajani et al. [289]. The WGAN-GP uses a gradient penalty to ensure the discriminator's gradients are constrained, ensuring Lipschitz continuity and improving algorithm stability during training. This is the main component that differentiates WGAN-GP from the basic WGANs. The architecture of the WGANs-GP discriminator network is presented in Listing 4.

Listing 4 Wasserstein Discriminator (Critic)

```
# Define the model architecture
model = Sequential([
    layers.Dense(128, activation='relu', input_shape=latent_dim),
    layers.BatchNormalization())
    layers.Dense(64, activation='relu'),
    layers.Dense(1, activation='Linear')])
```

In WGANs-GP, the activation function of the final dense layer is linear. The parameter λ is the regularization coefficient, which is often 10, and it is a hyperparameter that determines the weight/importance of the gradient penalty in the overall loss (Wasserstein loss). Assuming that D_{Fake} is the average discriminator output of the generated

samples and D_{Real} is the average discriminator output of the real samples, the overall loss function is defined as:

$$Loss_{Discriminator} = D_{Fake} - D_{Real} + \lambda * Gradient\ penalty \tag{4.3.5}$$

In this model, critic iterations are 5, which is the number of times the discriminator is trained per single generator training. This is a typical WGAN practice to ensure that the critic is well-trained. The clipping value is 0.01, and the learning rate is 0.0001. The models are trained using the RMSprop optimizer implemented with Keras API.

4.3.2.3 Minority GANs

The training process of Minority GANs is similar to the traditional GANs, except the generator is only trained on the minority class data to generate 50% of the synthetic data required to balance the data. The generated samples expand the minority class and balance the data, resulting in an unbiased training model with better classification capabilities.

4.3.2.4 SMOTE GANs

SMOTE GANs is another extension of GANs with an extra preprocessing step. SMOTE GANs is a hybrid two-phase approach to improve the quality of SMOTE outcomes [290]. First, SMOTE is applied to the minority class data. Then, the generator takes in a sample of original data and samples created using SMOTE. The discriminator is trained to learn the underlying patterns in the feature space to distinguish between real and generated samples. Incorporating GANs and SMOTE to create new samples introduces diversity into the generated model to create accurate, realistic samples.

4.4 Computation Results

Meteorological stations produce vast amounts of data to classify and predict weather patterns, requiring significant computation and storage resources to build machine-learning models and analyze the data. Federated learning is a potential solution for the scalability of weather data analysis and designing a robust model that can generalize well in the presence of noise and synthetic data. The problem being addressed in this experimental study is summarized in Figure 4.4.

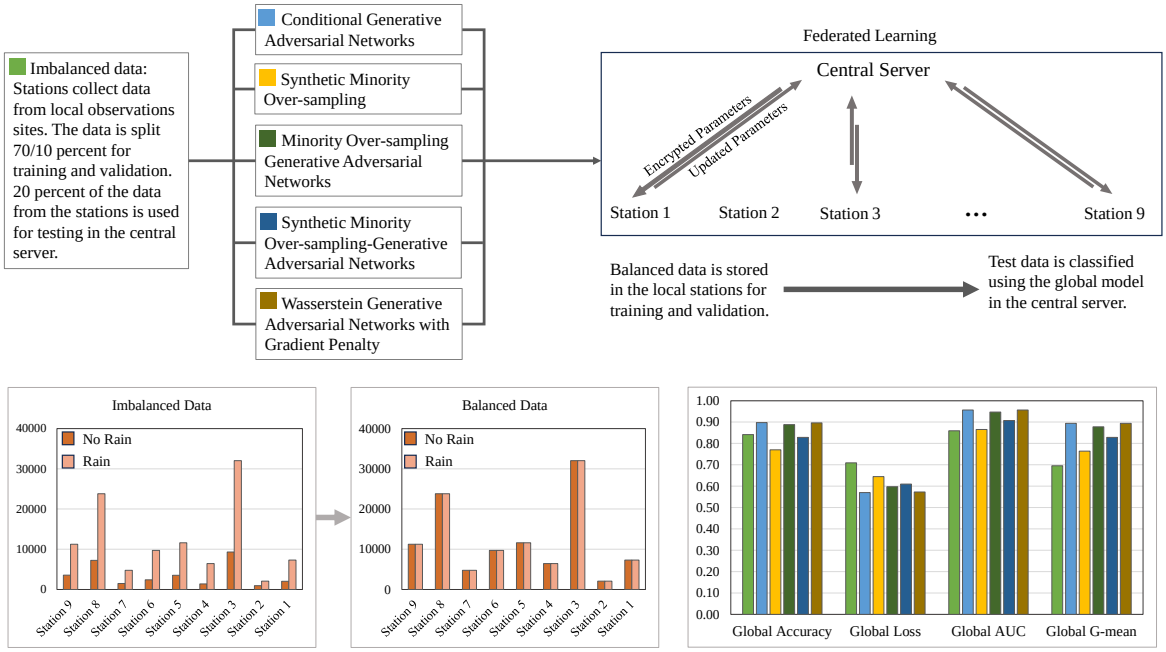


Figure 4.4: Overview of the federated imbalanced learning problem. We address the issue of imbalanced learning in a federated setting by generating samples of the minority class using 5 data augmentation methods. The local stations train the balanced data, and the encrypted model weights are sent to the global server for aggregation. The results of balanced data training are compared with those of imbalanced data in the federated learning framework.

4.4.1 Data Augmentation Strategies in Centralized Setting

Initially, each local model is trained on its own imbalanced dataset, recognizing that some weather stations might record rainy days (minority class) less frequently than non-rainy days (majority class). The imbalanced model is used as a baseline to evaluate the data augmentation techniques. We implemented data augmentation strategies, such as SMOTE and various GANs models, to synthesize new instances of the minority class, thereby mitigating the imbalance at each local node. The selected GANs models are CGANs, Minority GANs, SMOTE GANs, and WGANs-GP. The data obtained from the data augmentation methods have equal instances of both classes. This locally balanced data is then utilized for training individual models at each station. The results of training the models in the centralized setting are presented in Figure 4.5.

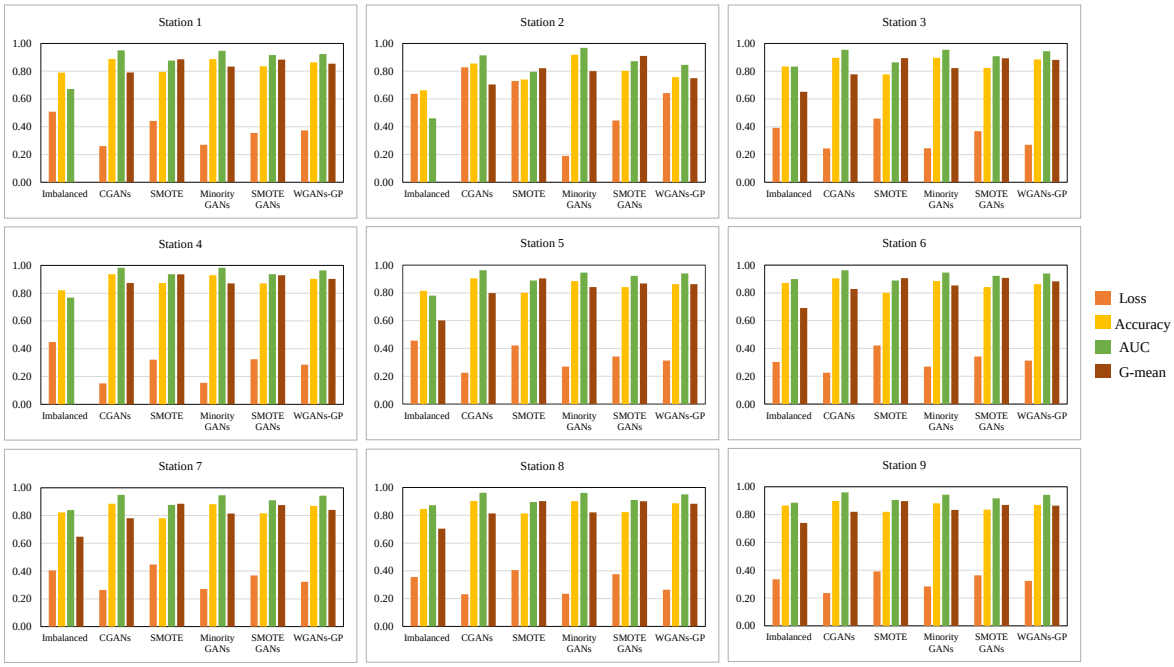


Figure 4.5: Classification Results of training the models locally on balanced data are compared with the imbalanced data.

To compare the effectiveness of the models, we analyze the performance metrics of each model. Loss is used to measure the error, and lower loss is desired. Accuracy, G-mean, and AUC range from 0 to 1, where higher values are preferred. AUC and G-mean are the most important metrics since they evaluate the models with respect to both classes. Given the results, a summary and analysis based on each augmentation method is provided:

1. Imbalanced: This model provides relatively high accuracy and AUC but somewhat lacks in balancing sensitivity and specificity, as evidenced by the lower G-mean, which is expected for imbalanced data.

2. CGANs: CGANs outperform the base model in all stations except 2, where the loss is higher. The overall lower loss in other stations indicates that CGANs have effectively reduced model error. The accuracy and AUC have improved across all stations compared to the imbalanced dataset, highlighting the efficacy of CGANs in distinguishing between classes. The G-mean has also shown significant improvement, showcasing the CGANs' ability to balance sensitivity and specificity.

3. SMOTE: The loss, accuracy, and AUC vary across stations. The accuracy and AUC are similar or slightly higher compared to the imbalanced dataset. However, it has some of the highest G-MAN values, suggesting a decent balance between sensitivity and specificity.

4. Minority GANs: Minority GANs have the lowest loss among the models in almost all stations. The loss is lower than the imbalanced dataset across all stations. The accuracy is close to the imbalanced datasets. However, the AUC and G-mean scores are generally very high, which shows a decent balance between sensitivity

and specificity. The Minority GANs performed better than CGANs in station 2 while comparable in the remaining stations.

5. SMOTE GANs: Overall, SMOTE GANs perform better than the imbalanced model, with consistently good results across stations. The AUC values suggest good discrimination ability, and the G-mean values indicate a balance between sensitivity and specificity.

6. WGANs-GP: Compared to the imbalanced dataset, WGANs-GP consistently offers competitive or better results across stations. However, the G-mean score is not the highest compared to SMOTE and SMOTE GANs in most stations.

In conclusion, GANs-based augmentation techniques (CGANs, Minority GANs, SMOTE GANs, WGANs-GP) generally outperform the imbalanced datasets regarding all metrics. This suggests that these techniques effectively create synthetic data that aids in better training the models. While a popular method, SMOTE is sometimes surpassed by GANs-based methods, especially in terms of AUC and Accuracy. If one has to rank based on the overall accuracy and AUC across stations, CGANs followed by Minority GANs and WGANs-GP would likely be at the top, and if ranked based on G-mean, SMOTE, and SMOTE GANs would be preferred in most stations.

4.4.2 Training Localized Augmented Data in Federated Setting

While the data augmentation techniques prove to be effective in addressing the imbalance ratio between classes, the small sample size impacts the models. Mainly because deep learning models are data-hungry and prefer larger datasets. Federated learning allows us to leverage the power of distributed data while maintaining privacy and security. In the next step, we utilized the balanced data available in the stations to train the federated learning model. The data is horizontally partitioned in the federated

learning setup, and the stations collect similar features from the weather observations. The results from evaluating the global model on the test data are presented in Table 4.2

Model	Accuracy	Loss	AUC	G-mean
Imbalanced	0.841	0.71	0.859	0.695
CGANs	0.897	0.57	0.956	0.894
SMOTE	0.770	0.64	0.865	0.764
Minority GANs	0.888	0.60	0.947	0.877
SMOTE GANs	0.828	0.61	0.907	0.828
WGANs-GP	0.896	0.57	0.956	0.844

Table 4.2: Classification results of testing the global model trained on the balanced data obtained from the various data augmentation methods, compared with imbalanced data.

A brief analysis of the results based on federated learning and data augmentation techniques using the provided metrics is provided.

The imbalanced data is used as the base model for comparison. CGANs and WGANs-GP emerge as the superior techniques among the listed, excelling in all metrics. CGANs offer the best G-mean score. This highlights that the conditional generation of synthetic samples can significantly enhance model learning and performance in federated settings. CGANs and WGANs-GP provide reliable and robust synthetic sample generation, thus aiding federated learning models to perform well. Minority GANs also yield very commendable results, with high accuracy, AUC, and G-mean values, though slightly falling short compared to CGANs. Interestingly, SMOTE has reduced accuracy compared to the imbalanced model but shows improvement in loss and a slightly higher AUC and G-mean, indicating an improved balance between sensitivity and specificity but worse overall performance compared to GANs-based models.

SMOTE GANs provide improvements in AUC and G-mean compared to the imbalanced model and SMOTE; however, they can not match the performance of other GANs-based models.

In addition to the results presented in Table 4.2, Figure 4.6 confirms the stability of the global model over ten communication rounds.

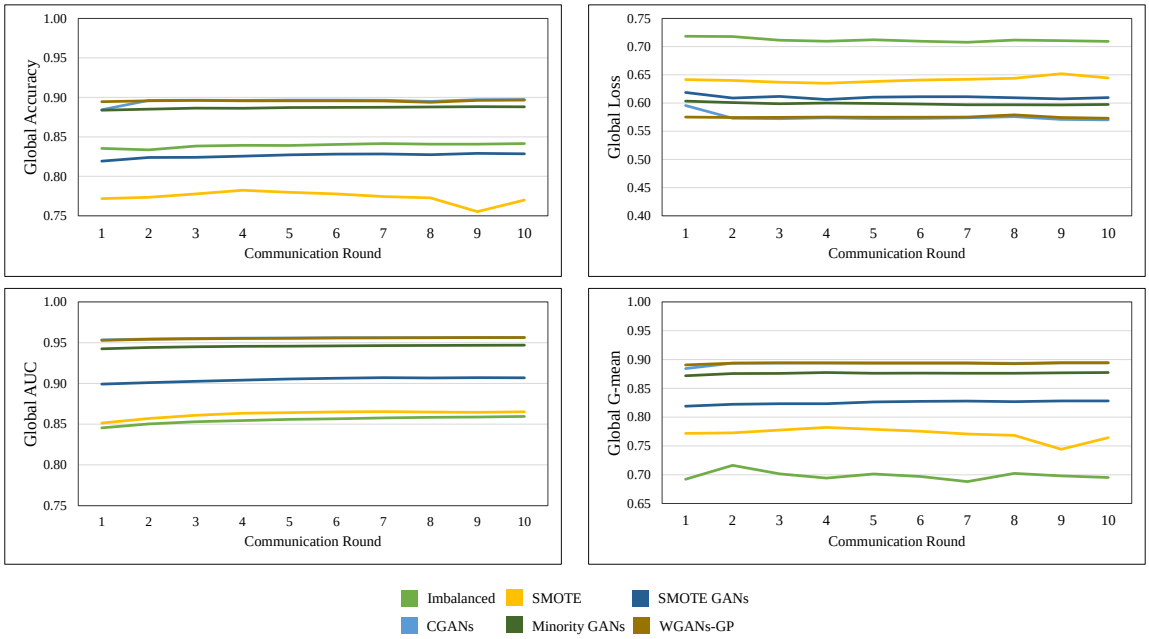


Figure 4.6: Federated learning process over ten communication rounds, evaluated on global test data.

Overall, it appears that the federated learning approach was effective in training a deep-learning model for rainfall prediction. The use of data augmentation methods to balance the dataset improved the accuracy, loss, AUC, and G-mean of the model, highlighting the importance of addressing imbalanced datasets in machine learning. In summary, CGANs and WGANs-GP stand out as particularly effective, achieving the highest performance across all metrics. Minority GANs and SMOTE GANs also enhance performance compared to the baseline (imbalanced) and traditional SMOTE.

The purely oversampling-based technique (SMOTE) does not outperform the baseline in terms of accuracy but does improve balance and discrimination between classes (higher G-mean and AUC). Given their very similar performance, the choice between CGANs and WGANs-GP might come down to computational efficiency, storage resources, ease of implementation, and specific use-case requirements. Both methods showcase the potential of GANs-based data augmentation in improving federated learning model outcomes.

Analyzing the validation results from local stations allows us to evaluate how effectively the global models generalize and perform on unseen data locally. We'll discuss the results with respect to four metrics: validation loss, validation accuracy, validation AUC, and validation G-mean presented in Figure 4.7.

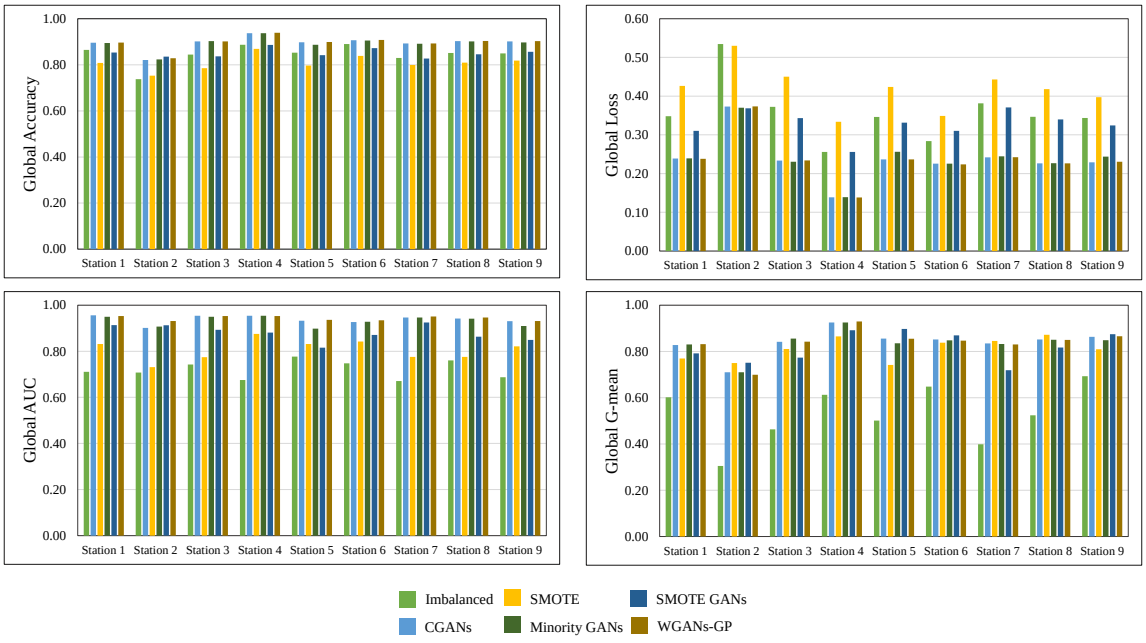


Figure 4.7: Classification results of federated learning on local validation sets

In all stations, GANs-based models, particularly CGANs, Minority GANs, and WGANs-GP, exhibit lower loss compared to the imbalanced and SMOTE models.

Across stations, CGANs and WGANs-GP consistently outperform or match the highest accuracy among the models. They tend to perform very strongly, often achieving AUC above 0.90 and securing top positions in G-mean, suggesting better balance in classifying the majority and the minority classes. Imbalanced models show varied performance across stations and metrics but tend to fall short, especially in AUC and G-mean. Generally, SMOTE does not deliver strong results in loss and accuracy. Its AUC and G-mean are varied, sometimes surpassing imbalanced models but lower than GANs-based models. CGANs, Minority GANs, and WGANs consistently show strong performance in the global metrics and across local stations. They deliver low loss, high accuracy, AUC, and G-mean, which indicates reliable and balanced predictive power for both majority and minority classes.

There's visible variability in the performance of SMOTE and imbalanced models globally and locally. They sometimes yield lower accuracy, AUC, and G-mean, indicating difficulty in reliably predicting both classes, particularly in imbalanced scenarios. GANs variants indicate strong global robustness, given their top-tier results in global metrics. Their consistent performance across different local stations (despite local data variability) indicates that GANs-based models (especially CGANs, Minority GANs, and WGANs) are not just fitting to the global model but are quite effective locally.

4.5 Chapter Summary

This study presents empirical evidence supporting the effectiveness of Generative Adversarial Networks (GANs) models, specifically CGANs, Minority GANs, and WGANs-GP, in navigating the intricacies of federated learning, optimizing the utility of global training and delivering potent performances across local stations.

CGANs, Minority GANs, and WGANs-GP are advanced data augmentation techniques that uniquely contribute to addressing data generation in imbalanced learning scenarios. However, they might potentially degrade model robustness by inadvertently amplifying noise or outliers in the minority class. Implementing GANs variants requires a more intricate design of network architectures in the generator and discriminator and hyperparameter tuning to optimize their capability. Despite offering improved stabilization during training, WGANs-GP can also be computationally demanding due to the implementation of the gradient penalty.

Overall, each of these GANs variants has proven to be theoretically appealing and practically impactful, achieving consistent, robust results in aggregated global metrics and decentralized local validation sets. This experimental study demonstrates the potential of federated learning in meteorology and other climate studies, where data is stored in local stations, and gathering the data in one place is not advised. While imbalanced datasets are a recurring challenge in machine learning, our results suggest that combining data augmentation techniques with federated learning can be a viable approach for developing robust and accurate models for predicting weather events.

Chapter 5

Conclusion

5.1 Conclusion

This dissertation presents a multidimensional exploration of the challenges and limitations of federated learning, noise-infusion mechanisms, and data imbalance in the context of machine learning.

In the rapidly evolving domain of federated learning, conducting a literature survey provides researchers and practitioners with multiple advantages. The literature survey conducted in this dissertation offers a consolidated repository of existing research on machine learning algorithms applied in federated learning. This enables benchmarking the new algorithms and comparing them with existing methods for efficiency. Through the systematic review, areas that are under-researched or have the potential for further exploration become evident, which helps guide future research toward more impactful directions. This literature survey also provides a holistic view of the state-of-the-art algorithms and their efficacy in various contexts, which helps industry professionals make informed choices for their federated learning implementations. Understanding past achievements and existing challenges can lead to collaborative efforts to address them.

This dissertation delves into experimental studies of noise infusion mechanisms in federated learning, highlighting the role of noise in fostering generalization, stability, and privacy in deep learning. Introducing the Signal-to-Noise ratio (SNR), the Price of

Stability (PoS), and the Price of Anarchy (PoA) provides an in-depth understanding of the trade-offs between privacy and model performance. SNR quantifies the trade-off between the clarity of the signal (i.e., the true data patterns) and the noise level introduced to the model. The experimental study highlighted the significance of maintaining an optimal SNR for effective model training without compromising privacy. PoS captures the best-case scenario of stability when training with noise, while PoA represents the worst-case impact. Monitoring these metrics over various noise levels enables achieving the desired privacy guarantees while still retaining model efficacy. In addition, while excessive use of noise can degrade the model's performance, the appropriate noise level can improve generalization and stability. The finding of the experimental analysis suggests that by utilizing noise, less complex models can perform comparably to significantly larger models. It was also found that, under certain conditions, noise can act as a regularizer, preventing overfitting and thereby improving model generalization. This results in less computationally demanding algorithms with improved generalization and stability. Maintaining stability has a great impact on the model's sensitivity to noise and the potential privacy guarantees that is essential in federated learning. By extensive experiments on different noise infusion mechanisms and introducing and leveraging innovative metrics, This research lays the foundation for more robust and private federated learning models, ensuring efficiency in decentralized machine learning.

Moreover, this dissertation utilizes variants of Generative Adversarial Networks (GANs) to address the recurrent challenge of data imbalance, particularly in domains like meteorology. While each variant of GANs exhibits unique strengths and challenges, their combined potential in federated learning landscapes is undeniable. This research demonstrates the power of federated learning in ensuring data privacy while leveraging

advanced machine learning techniques for optimal outcomes even when the data is inadequate or imbalanced.

In summary, federated learning stands as a field where innovation and collaboration are essential. This dissertation aims to offer solutions to the research questions presented in the outline. The findings of the experimental analysis in federated learning contribute to addressing some of the algorithmic and data-driven issues in this domain.

5.1.1 Recommendations for Future Work

Federated learning is at the interface of several research areas, such as optimization, distributed learning, cryptography, and communication theory. In the last few years, many algorithmic developments have been accomplished with a focus on theory and applications. However, there are still several challenges and open problems in federated learning. Some of the challenges that need to be overcome are:

- Fairness: The issue of fairness in federated learning occurs at different levels, such as fairness in communication capacities, machine learning models, and aggregation results. To this end, fairness metrics must be defined to evaluate the model from privacy, accuracy, and fairness perspectives [291]. Also, Lyu et al. [292] propose the concept of collaborative federated learning, which ensures fairness in how the clients impact the global model in the aggregation process. Despite the improvements, there is a lack of an integrated approach that ensures fairness in different aspects of federated learning.

- Scalability: To implement a federated learning protocol at scale, it must avoid the curse of dimensionality when data is large. To tackle the challenge of dimensionality, Principal Component Analysis has been employed in unsupervised settings such as the work by Al-Rubaie [293], which developed a privacy-preserving

Principal Component Analysis to reduce the dimensionality of the horizontally partitioned data. In addition, other methods, such as Discriminant Component Analysis, can be explored for efficient feature engineering and dimensionality reduction to improve accuracy and computation cost and uphold privacy.

- Unlabeled data: The lack of annotated data with good quality is one of the limitations in this setting. When the data is unlabeled, and labeling the data is either impossible or too cost-inefficient, it is important to modify the machine learning algorithms to be able to learn from partially annotated data efficiently.

- Non-Independent and Identically Distributed (Non-IID) Data: Non-IID data in federated learning refer to differences in the distribution of the available data over the data centers. It is also possible that the data become locally non-IID over time, which requires modifying existing algorithms or developing new ones. To address this issue, Sattler et al. [294] proposed a compression network to improve the communication efficiency and robustness of Federated learning on non-IID data. Ma et al. [295] and Zhu et al. [296] investigated the recent advances in solving non-IID data in federated learning and the future trends in research on this issue. They also compared different model architectures of deep learning used in the literature on federated learning.

Possible future research directions are:

- Game theory: Recent connections between game theory and control can provide new insights into federated learning and new algorithmic developments. An attempt along these lines is the paper of Mehrjou[297], which connects federated learning with mean field games by presenting federated learning as a differential game and discussing the properties of the equilibrium of this game.

- Quantum optimization: Another future research area is quantum optimization applied to federated learning. Distributed learning across several quantum computers could significantly improve the training time and potentially improve data privacy [298]. Connections with multi-objective optimization, such as [299], can benefit algorithmic developments.

- Multiple Kernel Learning: Exploring the connections between federated learning and multiple kernel learning also holds potential for advancing algorithmic solutions.

Furthermore, federated learning must navigate complex challenges, including exploring more advanced ways of combining local networks, using different machine learning and ensemble models in addition to preprocessing techniques, performing experiments on larger datasets and further enhancement efficiency, and designing accurate and reliable machine learning models suitable for GPU implementation.

By collectively addressing these challenges and pursuing interdisciplinary connections, the research community can integrate federated learning into a future where it serves as a cornerstone for secure, efficient, and privacy-conscious machine learning applications.

Reference List

[1] Ruth Gavison. Privacy and the limits of law. *The Yale law journal*, 89(3):421–471, 1980.

[2] Omer Tene and Jules Polonetsky. Privacy in the age of big data: a time for big decisions. *Stan. L. Rev. Online*, 64:63, 2011.

[3] Louis Brandeis and Samuel Warren. The right to privacy. *Harvard law review*, 4(5):193–220, 1890.

[4] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017.

[5] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1–2):1–210, 2021.

[6] Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Ingerman, Vladimir Ivanov, Chloe Kiddon, Jakub Konečnỳ, Stefano Mazzocchi, Brendan McMahan, et al. Towards federated learning at scale: System design. *Proceedings of machine learning and systems*, 1:374–388, 2019.

[7] Ping Li, Jin Li, Zhengan Huang, Chong-Zhi Gao, Wen-Bin Chen, and Kai Chen. Privacy-preserving outsourced classification in cloud computing. *Cluster Computing*, 21(1):277–286, 2018.

[8] ADP Team et al. Learning with privacy at scale. *Apple Mach. Learn. J*, 1(8):1–25, 2017.

[9] Alysa Ziying Tan, Han Yu, Lizhen Cui, and Qiang Yang. Towards personalized federated learning. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–17, 2022.

[10] Tong Li, Zhengan Huang, Ping Li, Zheli Liu, and Chunfu Jia. Outsourced privacy-preserving classification service over encrypted data. *Journal of Network and Computer Applications*, 106:100–110, 2018.

[11] Wei Yang Bryan Lim, Nguyen Cong Luong, Dinh Thai Hoang, Yutao Jiao, Ying-Chang Liang, Qiang Yang, Dusit Niyato, and Chunyan Miao. Federated learning in mobile edge networks: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, 22(3):2031–2063, 2020.

[12] Latif U. Khan, Shashi Raj Pandey, Nguyen H. Tran, Walid Saad, Zhu Han, Minh N. H. Nguyen, and Choong Seon Hong. Federated learning for edge networks: Resource optimization and incentive mechanism. *IEEE Communications Magazine*, 58(10):88–93, 2020.

[13] Dinh C. Nguyen, Ming Ding, Pubudu N. Pathirana, Aruna Seneviratne, Jun Li, and H. Vincent Poor. Federated learning for internet of things: A comprehensive survey. *IEEE Communications Surveys Tutorials*, 23(3):1622–1658, 2021.

[14] Latif U. Khan, Walid Saad, Zhu Han, Ekram Hossain, and Choong Seon Hong. Federated learning for internet of things: Recent advances, taxonomy, and open challenges. *IEEE Communications Surveys Tutorials*, 23(3):1759–1799, 2021.

[15] Iraklis Varlamis, Christos Sardianos, Christos Chronis, George Dimitrakopoulos, Yassine Himeur, Abdullah Alsalemi, Faycal Bensaali, and Abbes Amira. Using big data and federated learning for generating energy efficiency recommendations. *International Journal of Data Science and Analytics*, pages 1–17, 2022.

[16] Jie Xu, Benjamin S Glicksberg, Chang Su, Peter Walker, Jiang Bian, and Fei Wang. Federated learning for healthcare informatics. *Journal of Healthcare Informatics Research*, 5(1):1–19, 2021.

[17] Ines Feki, Sourour Ammar, Yousri Kessentini, and Khan Muhammad. Federated learning for covid-19 screening from chest x-ray images. *Applied Soft Computing*, 106:107330, 2021.

[18] Nicola Rieke, Jonny Hancox, Wenqi Li, Fausto Milletari, Holger R Roth, Shadi Albarqouni, Spyridon Bakas, Mathieu N Galtier, Bennett A Landman, Klaus Maier-Hein, et al. The future of digital health with federated learning. *NPJ digital medicine*, 3(1):1–7, 2020.

[19] Nuria Rodríguez-Barroso, Goran Stipcich, Daniel Jiménez-López, José Antonio Ruiz-Millán, Eugenio Martínez-Cámara, Gerardo González-Seco, M Victoria Luzón, Miguel Angel Veganzones, and Francisco Herrera. Federated learning and differential privacy: Software tools analysis, the sherpa. ai fl framework and methodological guidelines for preserving data privacy. *Information Fusion*, 64:270–292, 2020.

[20] Mohammed Aledhari, Rehma Razzak, Reza M. Parizi, and Fahad Saeed. Federated learning: A survey on enabling technologies, protocols, and applications. *IEEE Access*, 8:140699–140725, 2020.

[21] Roseline Oluwaseun Ogundokun, Sanjay Misra, Rytis Maskeliunas, and Robertas Damasevicius. A review on federated learning and machine learning approaches: Categorization, application areas, and blockchain technology. *Information*, 13(5):263, 2022.

[22] Maoguo Gong, Yu Xie, Ke Pan, Kaiyuan Feng, and Alex Kai Qin. A survey on differentially private machine learning. *IEEE computational intelligence magazine*, 15(2):49–64, 2020.

[23] Latif U Khan, Walid Saad, Zhu Han, Ekram Hossain, and Choong Seon Hong. Federated learning for internet of things: Recent advances, taxonomy, and open challenges. *IEEE Communications Surveys & Tutorials*, 2021.

[24] Rodolfo Stoffel Antunes, Cristiano André da Costa, Arne Küderle, Imrana Abdullahi Yari, and Björn Eskofier. Federated learning for healthcare: Systematic review and architecture proposal. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 13(4):1–23, 2022.

[25] Qinbin Li, Zeyi Wen, Zhaomin Wu, Sixu Hu, Naibo Wang, Yuan Li, Xu Liu, and Bingsheng He. A survey on federated learning systems: vision, hype and reality for data privacy and protection. *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[26] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):1–19, 2019.

[27] Murat Kantarcıoglu, Jaideep Vaidya, and C Clifton. Privacy preserving naive bayes classifier for horizontally partitioned data. In *IEEE ICDM workshop on privacy preserving data mining*, pages 3–9, 2003.

[28] Huajie Chen, Ali Burak Ünal, Mete Akgün, and Nico Pfeifer. Privacy-preserving svm on outsourced genomic data via secure multi-party computation. In *Proceedings of the Sixth International Workshop on Security and Privacy Analytics*, pages 61–69, 2020.

[29] Dragos Lia and Mihai Togan. Privacy-preserving machine learning using federated learning and secure aggregation. In *2020 12th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, pages 1–6, 2020.

[30] Yanbin Li, Ziming He, Xingjian Gu, Huanliang Xu, and Shougang Ren. Afedavg: communication-efficient federated learning aggregation with adaptive communication frequency and gradient sparse. *Journal of Experimental & Theoretical Artificial Intelligence*, pages 1–23, 2022.

[31] Junbin Chen, Jipu Li, Ruyi Huang, Ke Yue, Zhuyun Chen, and Weihua Li. Federated transfer learning for bearing fault diagnosis with discrepancy-based weighted federated averaging. *IEEE Transactions on Instrumentation and Measurement*, 71:1–11, 2022.

[32] Mannsoo Hong, Seok-Kyu Kang, and Jee-Hyong Lee. Weighted averaging federated learning based on example forgetting events in label imbalanced non-iid. *Applied Sciences*, 12(12):5806, 2022.

[33] Junbin Chen, Jipu Li, Ruyi Huang, Ke Yue, Zhuyun Chen, and Weihua Li. Federated transfer learning for bearing fault diagnosis with discrepancy-based weighted federated averaging. *IEEE Transactions on Instrumentation and Measurement*, 71:1–11, 2022.

[34] Meng Shen, Xiangyun Tang, Liehuang Zhu, Xiaojiang Du, and Mohsen Guizani. Privacy-preserving support vector machine training over blockchain-based encrypted iot data in smart cities. *IEEE Internet of Things Journal*, 6(5):7702–7712, 2019.

[35] Sicong Zhou, Huawei Huang, Wuhui Chen, Pan Zhou, Zibin Zheng, and Song Guo. Pirate: A blockchain-based secure framework of distributed machine learning in 5g networks. *IEEE Network*, 34(6):84–91, 2020.

[36] Viraaji Mothukuri, Reza M Parizi, Seyedamin Pouriyeh, Yan Huang, Ali Dehghantanha, and Gautam Srivastava. A survey on security and privacy of federated learning. *Future Generation Computer Systems*, 115:619–640, 2021.

[37] Dan Bogdanov, Liina Kamm, Sven Laur, and Ville Sokk. Implementation and evaluation of an algorithm for cryptographically private principal component analysis on genomic data. *IEEE/ACM transactions on computational biology and bioinformatics*, 15(5):1427–1432, 2018.

[38] Jemal H Abawajy and Mohammad Mehedi Hassan. Federated internet of things and cloud computing pervasive patient health monitoring system. *IEEE Communications Magazine*, 55(1):48–53, 2017.

[39] Nikita Lisin and Sergey Zapechnikov. Order-preserving encryption as a tool for privacy-preserving machine learning. In *2020 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)*, pages 2090–2092, 2020.

[40] Sara Salim, Nour Moustafa, Benjamin Turnbull, and Imran Razzak. Perturbation-enabled deep federated learning for preserving internet of things-based social networks. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 2022.

[41] Mehmet Emre Gursoy, Ali Inan, Mehmet Ercan Nergiz, and Yucel Saygin. Privacy-preserving learning analytics: challenges and techniques. *IEEE Transactions on Learning technologies*, 10(1):68–81, 2016.

[42] Stacey Truex, Ling Liu, Mehmet Emre Gursoy, and Lei Yu. Privacy-preserving inductive learning with decision trees. In *2017 IEEE International Congress on Big Data (BigData Congress)*, pages 57–64, 2017.

[43] Kang Wei, Jun Li, Ming Ding, Chuan Ma, Howard H. Yang, Farhad Farokhi, Shi Jin, Tony Q. S. Quek, and H. Vincent Poor. Federated learning with differential privacy: Algorithms and performance analysis. *IEEE Transactions on Information Forensics and Security*, 15:3454–3469, 2020.

[44] Nathan Dowlin, Ran Gilad-Bachrach, Kim Laine, Kristin Lauter, Michael Naehrig, and John Wernsing. Manual for using homomorphic encryption for bioinformatics. *Proceedings of the IEEE*, 105(3):552–567, 2017.

[45] Paulo Martins, Leonel Sousa, and Artur Mariano. A survey on fully homomorphic encryption: An engineering perspective. *ACM Computing Surveys (CSUR)*, 50(6):1–33, 2017.

[46] Zhan Qin, Jian Weng, Yong Cui, and Kui Ren. Privacy-preserving image processing in the cloud. *IEEE Cloud Computing*, 5(2):48–57, 2018.

[47] Qing He, Ning Li, Wen-Juan Luo, and ZZ Shi. A survey of machine learning algorithms for big data. *Pattern recognition and artificial intelligence*, 27(4):327–336, 2014.

[48] Kunal Chandiramani, Dhruv Garg, and N Maheswari. Performance analysis of distributed and federated learning models on private data. *Procedia Computer Science*, 165:349–355, 2019.

[49] Rafa Gálvez, Veelasha Moonsamy, and Claudia Diaz. Less is more: A privacy-respecting android malware classifier using federated learning. *Proceedings on Privacy Enhancing Technologies*, 4:96–116, 2021.

[50] Yi Yang, Shuai Huang, Wei Huang, and Xiangyu Chang. Privacy-preserving cost-sensitive learning. *IEEE Transactions on Neural Networks and Learning Systems*, 32(5):2105–2116, 2020.

[51] Wei Guo, Jun Shao, Rongxing Lu, Yining Liu, and Ali A Ghorbani. A privacy-preserving online medical prediagnosis scheme for cloud environment. *IEEE Access*, 6:48946–48957, 2018.

[52] Celestine Dünner, Thomas Parnell, Kubilay Atasu, Manolis Sifalakis, and Haralampos Pozidis. Understanding and optimizing the performance of distributed machine learning applications on apache spark. In *2017 IEEE International Conference on Big Data (Big Data)*, pages 331–338, 2017.

[53] Takahiro Maekawa, Ayana Kawamura, Yuma Kinoshita, and Hitoshi Kiya. Privacy-preserving svm computing in the encrypted domain. In *2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 897–902, 2018.

[54] Raphael Bost, Raluca Ada Popa, Stephen Tu, and Shafi Goldwasser. Machine learning classification over encrypted data. In *NDSS*, volume 4324, page 4325, 2015.

[55] Kaihe Xu, Hao Yue, Linke Guo, Yuanxiong Guo, and Yuguang Fang. Privacy-preserving machine learning algorithms for big data systems. In *2015 IEEE 35th international conference on distributed computing systems*, pages 318–327, 2015.

[56] Saerom Park, Junyoung Byun, Joohee Lee, Jung Hee Cheon, and Jaewook Lee. He-friendly algorithm for privacy-preserving svm training. *IEEE Access*, 8:57414–57425, 2020.

[57] Makhamisa Senekane. Differentially private image classification using support vector machine and differential privacy. *Machine Learning and Knowledge Extraction*, 1(1):483–491, 2019.

[58] Jinwen Liang, Zheng Qin, Jianbing Ni, Xiaodong Lin, and Xuemin Shen. Efficient and privacy-preserving outsourced svm classification in public cloud. In *ICC 2019-2019 IEEE International Conference on Communications (ICC)*, pages 1–6, 2019.

[59] Thee Chanyaswad, J Morris Chang, and Sun-Yuan Kung. A compressive multi-kernel method for privacy-preserving machine learning. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 4079–4086, 2017.

[60] Ruei-Hau Hsu, Yi-Cheng Wang, Chun-I Fan, Bo Sun, Tao Ban, Takeshi Takahashi, Ting-Wei Wu, and Shang-Wei Kao. A privacy-preserving federated learning system for android malware detection based on edge computing. In *2020 15th Asia Joint Conference on Information Security (AsiaJCIS)*, pages 128–136, 2020.

[61] Fuquan Zhang, Tsu-Yang Wu, Jeng-Shyang Pan, Gangyi Ding, and Zuoyong Li. Human motion recognition based on svm in vr art media interaction environment. *Human-centric Computing and Information Sciences*, 9(1):1–15, 2019.

[62] Dawei Chen, Linda Jiang Xie, BaekGyu Kim, Li Wang, Choong Seon Hong, Li-Chun Wang, and Zhu Han. Federated learning based mobile edge computing for augmented reality applications. In *2020 international conference on computing, networking and communications (ICNC)*, pages 767–773, 2020.

[63] Valentin Hartmann, Konark Modi, Josep M Pujol, and Robert West. Privacy-preserving classification with secret vector machines. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 475–484, 2020.

[64] Yunmei Lu, Mingyuan Yan, Meng Han, Qingliang Zhang, and Yanqing Zhang. Privacy preserving multiclass classification for horizontally distributed data. In *Proceedings of the 19th Annual SIG Conference on Information Technology Education*, SIGITE '18, page 165, New York, NY, USA, 2018. Association for Computing Machinery.

[65] Mohammed Z Omer, Hui Gao, and Nadir Mustafa. Privacy-preserving of svm over vertically partitioned with imputing missing data. *Distributed and Parallel Databases*, 35(3):363–382, 2017.

[66] Ranya Aloufi, Hamed Haddadi, and David Boyle. Privacy-preserving voice analysis via disentangled representations. In *Proceedings of the 2020 ACM SIGSAC Conference on Cloud Computing Security Workshop*, pages 1–14, 2020.

[67] Jing Wang, Libing Wu, Huaqun Wang, Kim-Kwang Raymond Choo, and Debiao He. An efficient and privacy-preserving outsourced support vector machine training for internet of medical things. *IEEE Internet of Things Journal*, 8(1):458–473, 2020.

[68] Hui Zhu, Xiaoxia Liu, Rongxing Lu, and Hui Li. Efficient and privacy-preserving online medical prediagnosis framework using nonlinear svm. *IEEE journal of biomedical and health informatics*, 21(3):838–850, 2016.

[69] Mohsin Y Ahmed, Md Mahbubur Rahman, Viswam Nathan, Ebrahim Nemati, Korosh Vatanparvar, and Jilong Kuang. mlung: Privacy-preserving naturally windowed lung activity detection for pulmonary patients. In *2019 IEEE 16th International Conference on Wearable and Implantable Body Sensor Networks (BSN)*, pages 1–4, 2019.

[70] Qian Wang, Minxin Du, Xiuying Chen, Yanjiao Chen, Pan Zhou, Xiaofeng Chen, and Xinyi Huang. Privacy-preserving collaborative model learning: The case of word vector training. *IEEE Transactions on Knowledge and Data Engineering*, 30(12):2381–2393, 2018.

[71] Zekun Li and Shuyu Li. Random forest algorithm under differential privacy. In *2017 IEEE 17th International Conference on Communication Technology (ICCT)*, pages 1901–1905, 2017.

[72] Fatemeh Khodaparast, Mina Sheikhalishahi, Hassan Haghighi, and Fabio Martinelli. Privacy preserving random decision tree classification over horizontally and vertically partitioned data. In *2018 IEEE 16th Intl Conf on Dependable,*

Autonomic and Secure Computing, 16th Intl Conf on Pervasive Intelligence and Computing, 4th Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech), pages 600–607, 2018.

[73] Sonal Yadav, Vivek Tiwari, and Basant Tiwari. Privacy preserving data mining with abridge time using vertical partition decision tree. In *Proceedings of the ACM Symposium on Women in Research 2016*, pages 158–164, 2016.

[74] Shahriar Badsha, Iman Vakilinia, and Shamik Sengupta. Privacy preserving cyber threat information sharing and learning for cyber defense. In *2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)*, pages 0708–0714, 2019.

[75] Rémi Canillas, Rania Talbi, Sara Bouchenak, Omar Hasan, Lionel Brunie, and Laurent Sarrat. Exploratory study of privacy preserving fraud detection. In *Proceedings of the 19th International Middleware Conference Industry*, pages 25–31, 2018.

[76] Liang Xue, Dongxiao Liu, Jianbing Ni, Xiaodong Lin, and Xuemin Shen. Consent-based privacy-preserving decision tree evaluation. In *ICC 2020-2020 IEEE International Conference on Communications (ICC)*, pages 1–6, 2020.

[77] Liang Xue, Dongxiao Liu, Cheng Huang, Xiaodong Lin, and Xuemin Sherman Shen. Secure and privacy-preserving decision tree classification with lower complexity. *Journal of Communications and Information Networks*, 5(1):16–25, 2020.

[78] Jun Hou, Qianmu Li, Shunmei Meng, Zhen Ni, Yini Chen, and Yaozong Liu. Dprf: a differential privacy protection random forest. *IEEE Access*, 7:130707–130720, 2019.

[79] Zhitao Guan, Xianwen Sun, Lingyun Shi, Longfei Wu, and Xiaojiang Du. A differentially private greedy decision forest classification algorithm with high utility. *Computers & Security*, 96:101930, 2020.

[80] Chaoxian Lv, Qianmu Li, Huaqiu Long, Yumei Ren, and Fei Ling. A differential privacy random forest method of privacy protection in cloud. In *2019 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC)*, pages 470–475, 2019.

[81] Bangzhou Xin, Wei Yang, Shaowei Wang, and Liusheng Huang. Differentially private greedy decision forest. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2672–2676, 2019.

[82] Lingchen Zhao, Lihao Ni, Shengshan Hu, Yaniiao Chen, Pan Zhou, Fu Xiao, and Libing Wu. Inprivate digging: Enabling tree-based distributed data mining with differential privacy. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*, pages 2087–2095, 2018.

[83] Kyle Fritchman, Keerthanaa Saminathan, Rafael Dowsley, Tyler Hughes, Martine De Cock, Anderson Nascimento, and Ankur Teredesai. Privacy-preserving scoring of tree ensembles: A novel framework for ai in healthcare. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 2413–2422, 2018.

[84] Hanbin Zhang, Chen Song, Aosen Wang, Chenhan Xu, Dongmei Li, and Wenyao Xu. Pdvocal: Towards privacy-preserving parkinson's disease detection using non-speech body sounds. In *The 25th Annual International Conference on Mobile Computing and Networking*, pages 1–16, 2019.

[85] NG Nageswari Amma and F Ramesh Dhanaseelan. Privacy preserving data mining classifier for smart city applications. In *2018 3rd International Conference on Communication and Electronics Systems (ICCES)*, pages 645–648, 2018.

[86] Mahmoud Parto, Christopher Saldana, and Thomas Kurfess. A novel three-layer iot architecture for shared, private, scalable, and real-time machine learning from ubiquitous cyber-physical systems. *Procedia manufacturing*, 48:959–967, 2020.

[87] Jaideep Vaidya, Murat Kantarcıoğlu, and Chris Clifton. Privacy-preserving naive bayes classification. *The VLDB Journal*, 17(4):879–898, 2008.

[88] Mikhail Yurochkin, Mayank Agarwal, Soumya Ghosh, Kristjan Greenewald, Nghia Hoang, and Yasaman Khazaeni. Bayesian nonparametric federated learning of neural networks. In *International Conference on Machine Learning*, pages 7252–7261. PMLR, 2019.

[89] Álvaro García-Recuero. Discouraging abusive behavior in privacy-preserving online social networking applications. In *Proceedings of the 25th International Conference Companion on World Wide Web*, pages 305–309, 2016.

[90] Yanting Chai, Yu Zhan, Baocang Wang, Yuan Ping, and Zhili Zhang. Improvement on a privacy-preserving outsourced classification protocol over encrypted data. *Wireless Networks*, 26(6):4363–4374, 2020.

[91] Xue Yang, Rongxing Lu, Jun Shao, Xiaohu Tang, and Haomiao Yang. An efficient and privacy-preserving disease risk prediction scheme for e-healthcare. *IEEE Internet of Things Journal*, 6(2):3284–3297, 2018.

[92] Maria E Skarkala, Manolis Maragoudakis, Stefanos Gritzalis, and Lilian Mitrou. Pp-tan: a privacy preserving multi-party tree augmented naive bayes classifier. In

2020 5th South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM), pages 1–8, 2020.

[93] Sin G Teo, Jianneng Cao, and Vincent CS Lee. Dag: A general model for privacy-preserving data mining. *IEEE Transactions on Knowledge and Data Engineering*, 32(1):40–53, 2018.

[94] Xiaoxia Liu, Hui Zhu, Rongxing Lu, and Hui Li. Efficient privacy-preserving online medical primary diagnosis scheme on naive bayesian classification. *Peer-to-Peer Networking and Applications*, 11(2):334–347, 2018.

[95] Zhaowen Lin, Fei Xiao, Yi Sun, Yan Ma, Cong-Cong Xing, and Jun Huang. A secure encryption-based malware detection system. *KSII Transactions on Internet and Information Systems (TIIS)*, 12(4):1799–1818, 2018.

[96] Rania Talbi, Sara Bouchenak, and Lydia Y Chen. Towards dynamic end-to-end privacy preserving data classification. In *2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)*, pages 73–74, 2018.

[97] Xiaoyu Zhang, Xiaofeng Chen, Jianfeng Wang, Zhihui Zhan, and Jin Li. Verifiable privacy-preserving single-layer perceptron training scheme in cloud computing. *Soft Computing*, 22(23):7719–7732, 2018.

[98] Qijian He, Wei Yang, Bingren Chen, Yangyang Geng, and Liusheng Huang. Transnet: Training privacy-preserving neural network over transformed layer. *Proceedings of the VLDB Endowment*, 13(12):1849–1862, 2020.

[99] Ehsan Hesamifard, Hassan Takabi, Mehdi Ghasemi, and Catherine Jones. Privacy-preserving machine learning in cloud. In *Proceedings of the 2017 on cloud computing security workshop*, pages 39–43, 2017.

[100] Ehsan Hesamifard, Hassan Takabi, Mehdi Ghasemi, and Rebecca N Wright. Privacy-preserving machine learning as a service. *Proc. Priv. Enhancing Technol.*, 2018(3):123–142, 2018.

[101] Gianpiero Costantino, Antonio La Marra, Fabio Martinelli, Andrea Saracino, and Mina Sheikhalishahi. Privacy-preserving text mining as a service. In *2017 IEEE Symposium on Computers and Communications (ISCC)*, pages 890–897, 2017.

[102] Nasir Rahim, Jamil Ahmad, Khan Muhammad, Arun Kumar Sangaiah, and Sung Wook Baik. Privacy-preserving image retrieval for mobile devices with deep features on the cloud. *Computer Communications*, 127:75–85, 2018.

[103] Li Wang, Jun Jie Shi, Chen Chen, and Sheng Zhong. Privacy-preserving face detection based on linear and nonlinear kernels. *Multimedia Tools and Applications*, 77(6):7261–7281, 2018.

[104] Mengran Xia, Dawei Jin, and Jingyu Chen. Short-term traffic flow prediction based on graph convolutional networks and federated learning. *IEEE Transactions on Intelligent Transportation Systems*, pages 1–13, 2022.

[105] Xiaoguang Niu, Qiongzan Ye, Yihao Zhang, and Dengpan Ye. A privacy-preserving identification mechanism for mobile sensing systems. *IEEE Access*, 6:15457–15467, 2018.

[106] Jiaping Lin, Jianwei Niu, and Hui Li. Pcd: A privacy-preserving predictive clinical decision scheme with e-health big data based on rnn. In *2017 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 808–813, 2017.

[107] Efe Bozkir, David Geisler, and Enkelejda Kasneci. Person independent, privacy preserving, and real time assessment of cognitive load using eye tracking in a virtual reality setup. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 1834–1837, 2019.

[108] Julian Steil, Marion Koelle, Wilko Heuten, Susanne Boll, and Andreas Bulling. Privaceye: privacy-preserving head-mounted eye tracking using egocentric scene image and eye movement features. In *Proceedings of the 11th ACM Symposium on Eye Tracking Research & Applications*, pages 1–10, 2019.

[109] Youwen Zhu and Xingxin Li. Privacy-preserving k-means clustering with local synchronization in peer-to-peer networks. *Peer-to-Peer Networking and Applications*, 13(6):2272–2284, 2020.

[110] Ahmed A. Al-Saedi, Veselka Boeva, and Emiliano Casalicchio. Reducing communication overhead of federated learning through clustering analysis. In *2021 IEEE Symposium on Computers and Communications (ISCC)*, pages 1–7, 2021.

[111] Igor V Anikin and Rinat M Gazimov. Privacy preserving dbscan clustering algorithm for vertically partitioned data in distributed systems. In *2017 International Siberian Conference on Control and Communications (SIBCON)*, pages 1–4, 2017.

[112] Walisa Romsaiyud, Henning Schnoor, and Wilhelm Hasselbring. Improving k-nearest neighbor pattern recognition models for privacy-preserving data analysis. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 5804–5813, 2019.

[113] Dinesh Chowdary Attota, Viraaji Mothukuri, Reza M. Parizi, and Seyedamin Pouriyeh. An ensemble multi-view federated learning intrusion detection for iot. *IEEE Access*, 9:117734–117745, 2021.

[114] Zhuo Ma, Yang Liu, Ximeng Liu, Jianfeng Ma, and Kui Ren. Lightweight privacy-preserving ensemble classification for face recognition. *IEEE Internet of Things Journal*, 6(3):5778–5790, 2019.

[115] Dawid Połap and Marcin Woźniak. A hybridization of distributed policy and heuristic augmentation for improving federated learning approach. *Neural Networks*, 146:130–140, 2022.

[116] Dawid Połap and Marcin Woźniak. Meta-heuristic as manager in federated learning approaches for image processing purposes. *Applied Soft Computing*, 113:107872, 2021.

[117] Basheer Qolomany, Kashif Ahmad, Ala Al-Fuqaha, and Junaid Qadir. Particle swarm optimized federated learning for industrial iot and smart city services. In *GLOBECOM 2020-2020 IEEE Global Communications Conference*, pages 1–6. IEEE, 2020.

[118] Biwen Chen, Honghong Zeng, Tao Xiang, Shangwei Guo, Tianwei Zhang, and Yang Liu. Esb-fl: Efficient and secure blockchain-based federated learning with fair payment. *IEEE Transactions on Big Data*, pages 1–1, 2022.

[119] Mansoor Ali, Hadis Karimipour, and Muhammad Tariq. Integration of blockchain and federated learning for internet of things: Recent advances and future challenges. *Computers & Security*, 108:102355, 2021.

[120] Yunlong Lu, Xiaohong Huang, Yueyue Dai, Sabita Maharjan, and Yan Zhang. Blockchain and federated learning for privacy-preserved data sharing in industrial iot. *IEEE Transactions on Industrial Informatics*, 16(6):4177–4186, 2020.

[121] Shiva Raj Pokhrel and Jinho Choi. Federated learning with blockchain for autonomous vehicles: Analysis and design challenges. *IEEE Transactions on Communications*, 68(8):4734–4746, 2020.

[122] Dinh C. Nguyen, Ming Ding, Quoc-Viet Pham, Pubudu N. Pathirana, Long Bao Le, Aruna Seneviratne, Jun Li, Dusit Niyato, and H. Vincent Poor. Federated learning meets blockchain in edge computing: Opportunities and challenges. *IEEE Internet of Things Journal*, 8(16):12806–12825, 2021.

[123] Naiyu Wang, Wenti Yang, Xiaodong Wang, Longfei Wu, Zhitao Guan, Xiaojiang Du, and Mohsen Guizani. A blockchain based privacy-preserving federated learning scheme for internet of vehicles. *Digital Communications and Networks*, 2022.

[124] Naiyu Wang, Wenti Yang, Zhitao Guan, Xiaojiang Du, and Mohsen Guizani. Bpfl: A blockchain based privacy-preserving federated learning scheme. In *2021 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6, 2021.

[125] Jiawen Kang, Zehui Xiong, Dusit Niyato, Yuze Zou, Yang Zhang, and Mohsen Guizani. Reliable federated learning for mobile networks. *IEEE Wireless Communications*, 27(2):72–80, 2020.

[126] Jiawen Kang, Xuandi Li, Jiangtian Nie, Yi Liu, Minrui Xu, Zehui Xiong, Dusit Niyato, and Qiang Yan. Communication-efficient and cross-chain empowered federated learning for artificial intelligence of things. *IEEE Transactions on Network Science and Engineering*, pages 1–1, 2022.

[127] Rajesh Kumar, Abdullah Aman Khan, Jay Kumar, Zakria, Noorbakhsh Amiri Golilarz, Simin Zhang, Yang Ting, Chengyu Zheng, and Wenyong Wang. Blockchain-federated-learning and deep learning models for covid-19 detection using ct imaging. *IEEE Sensors Journal*, 21(14):16301–16314, 2021.

[128] Hyesung Kim, Jihong Park, Mehdi Bennis, and Seong-Lyun Kim. Blockchained on-device federated learning. *IEEE Communications Letters*, 24(6):1279–1283, 2020.

[129] Zhuo Liu, Chenhui Yao, Hang Yu, and Taihua Wu. Deep reinforcement learning with its application for lung cancer detection in medical internet of things. *Future Generation Computer Systems*, 97:1–9, 2019.

[130] Hao Wang, Zakhary Kaplan, Di Niu, and Baochun Li. Optimizing federated learning on non-iid data with reinforcement learning. In *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications*, pages 1698–1707, 2020.

[131] Yufeng Zhan, Peng Li, Zhihao Qu, Deze Zeng, and Song Guo. A learning-based incentive mechanism for federated learning. *IEEE Internet of Things Journal*, 7(7):6360–6368, 2020.

[132] Mounssif Krouka, Anis Elgabli, Chaouki Ben Issaid, and Mehdi Bennis. Communication-efficient and federated multi-agent reinforcement learning. *IEEE Transactions on Cognitive Communications and Networking*, 8(1):311–320, 2022.

[133] Elaheh Jafarigol and Theodore Trafalis. The paradox of noise: An empirical study of noise-infusion mechanisms to improve generalization, stability, and privacy in federated learning. *arXiv preprint arXiv:2311.05790*, 2023.

[134] Peter L Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482, 2002.

[135] Giorgio Gnecco, Marcello Sanguineti, et al. Approximation error bounds via rademacher complexity. *Applied Mathematical Sciences*, 2:153–176, 2008.

[136] Michel Ledoux and Michel Talagrand. *Probability in Banach Spaces: isoperimetry and processes*, volume 23. Springer Science & Business Media, 1991.

[137] Mehryar Mohri and Afshin Rostamizadeh. Rademacher complexity bounds for non-iid processes. *Advances in Neural Information Processing Systems*, 21, 2008.

[138] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2018.

[139] Vladimir N Vapnik and Alexey Ya Chervonenkis. On the uniform convergence of the frequencies of occurrence of events to their probabilities. In *Empirical Inference*, pages 7–12. Springer, 2013.

[140] George Cybenko. Just-in-time learning and estimation. *Nato ASI Series F Computer and Systems Sciences*, 153:423–434, 1996.

[141] Marek Karpinski and Angus Macintyre. Bounding vc-dimension for neural networks: progress and prospects. In *European Conference on Computational Learning Theory*, pages 337–341. Springer, 1995.

[142] Eduardo D Sontag et al. Vc dimension of neural networks. *NATO ASI Series F Computer and Systems Sciences*, 168:69–96, 1998.

[143] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[144] Alexander Rakhlin, Sayan Mukherjee, and Tomaso Poggio. Stability results in learning theory. *Analysis and Applications*, 3(04):397–417, 2005.

[145] Andr e Elissee. A study about algorithmic stability and their relation to generalization performances. 2000.

[146] Olivier Bousquet and André Elisseeff. Stability and generalization. *The Journal of Machine Learning Research*, 2:499–526, 2002.

[147] J Frédéric Bonnans and Alexander Shapiro. Optimization problems with perturbations: A guided tour. *SIAM review*, 40(2):228–264, 1998.

[148] Irit Dinur and Kobbi Nissim. Revealing information while preserving privacy. In *Proceedings of the twenty-second ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 202–210, 2003.

[149] Avrim Blum, Cynthia Dwork, Frank McSherry, and Kobbi Nissim. Practical privacy: the sulq framework. In *Proceedings of the twenty-fourth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 128–138, 2005.

[150] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pages 265–284. Springer, 2006.

[151] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Differential privacy—a primer for the perplexed,". *Joint UNECE/Eurostat work session on statistical data confidentiality*, 11, 2011.

[152] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

[153] Kang Wei, Jun Li, Ming Ding, Chuan Ma, Howard H Yang, Farhad Farokhi, Shi Jin, Tony QS Quek, and H Vincent Poor. Federated learning with differential privacy: Algorithms and performance analysis. *IEEE Transactions on Information Forensics and Security*, 15:3454–3469, 2020.

[154] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In *Annual international conference on the theory and applications of cryptographic techniques*, pages 486–503. Springer, 2006.

[155] Dana Ron and M Kearns. Algorithmic stability and sanity-check bounds for leave-one-out crossvaildation. *Neural Computation*, 11(6):1427–1453, 1999.

[156] Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nati Srebro. Exploring generalization in deep learning. *Advances in neural information processing systems*, 30, 2017.

[157] Chris M Bishop. Training with noise is equivalent to tikhonov regularization. *Neural computation*, 7(1):108–116, 1995.

[158] Christopher M Bishop et al. *Neural networks for pattern recognition*. Oxford university press, 1995.

[159] Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[160] Mark D McDonnell and Lawrence M Ward. The benefits of noise in neural systems: bridging theory and experiment. *Nature Reviews Neuroscience*, 12(7):415–425, 2011.

[161] J Andrew Doyle and Alan C Evans. What colour is neural noise? *arXiv preprint arXiv:1806.03704*, 2018.

[162] Sumit Kumar, Ayush Kumar, and Rajib Kumar Jha. A novel noise-enhanced back-propagation technique for weak signal detection in neyman–pearson framework. *Neural Processing Letters*, 50(3):2389–2406, 2019.

[163] Roberto Benzi, Alfonso Sutera, and Angelo Vulpiani. The mechanism of stochastic resonance. *Journal of Physics A: mathematical and general*, 14(11):L453, 1981.

[164] Roberto Benzi, Giorgio Parisi, Alfonso Sutera, and Angelo Vulpiani. A theory of stochastic resonance in climatic change. *SIAM Journal on applied mathematics*, 43(3):565–578, 1983.

[165] Shuhei Ikemoto, Fabio DallaLibera, and Koh Hosoda. Noise-modulated neural networks as an application of stochastic resonance. *Neurocomputing*, 277:29–37, 2018.

[166] A Aldo Faisal, Luc PJ Selen, and Daniel M Wolpert. Noise in the nervous system. *Nature reviews neuroscience*, 9(4):292–303, 2008.

[167] Wolfgang Maass. Noise as a resource for computation and learning in networks of spiking neurons. *Proceedings of the IEEE*, 102(5):860–880, 2014.

[168] Lasse Holmstrom and Petri Koistinen. Using additive noise in back-propagation training. *IEEE transactions on neural networks*, 3(1):24–38, 1992.

[169] Vladimir Karpukhin, Omer Levy, Jacob Eisenstein, and Marjan Ghazvininejad. Training on synthetic noise improves robustness to natural noise in machine translation. *arXiv preprint arXiv:1902.01509*, 2019.

[170] Jocelyn Sietsma and Robert JF Dow. Creating artificial neural networks that generalize. *Neural networks*, 4(1):67–79, 1991.

[171] Guozhong An. The effects of adding noise during backpropagation training on a generalization performance. *Neural computation*, 8(3):643–674, 1996.

[172] Zhi Zeng, Yuan Liu, Weijun Tang, and Fangjiong Chen. Noise is useful: Exploiting data diversity for edge intelligence. *IEEE Wireless Communications Letters*, 10(5):957–961, 2021.

[173] Russell Reed and Robert J MarksII. *Neural smithing: supervised learning in feedforward artificial neural networks*. Mit Press, 1999.

[174] Richard M Zur, Yulei Jiang, Lorenzo L Pesce, and Karen Drukker. Noise injection for training artificial neural networks: A comparison with weight decay and early stopping. *Medical physics*, 36(10):4810–4818, 2009.

[175] Naresh Nagabushan, Nishank Satish, and S Raghuram. Effect of injected noise in deep neural networks. In *2016 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*, pages 1–5. IEEE, 2016.

[176] Oussama Dhifallah and Yue Lu. On the inherent regularization effects of noise injection during training. In *International Conference on Machine Learning*, pages 2665–2675. PMLR, 2021.

[177] Moritz Hardt, Ben Recht, and Yoram Singer. Train faster, generalize better: Stability of stochastic gradient descent. In *International conference on machine learning*, pages 1225–1234. PMLR, 2016.

[178] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318, 2016.

[179] Zhezhi He, Adnan Siraj Rakin, and Deliang Fan. Parametric noise injection: Trainable randomness to improve deep neural network robustness against adversarial attack. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 588–597, 2019.

[180] Li Xiao, Zeliang Zhang, and Yijie Peng. Noise optimization for artificial neural networks. *arXiv preprint arXiv:2102.04450*, 2021.

[181] Aishan Liu, Xianglong Liu, Hang Yu, Chongzhi Zhang, Qiang Liu, and Dacheng Tao. Training robust deep neural networks via adversarial noise propagation. *IEEE Transactions on Image Processing*, 30:5769–5781, 2021.

[182] P. Koistinen and L. Holmstrom. Kernel regression and backpropagation training with noise. In *[Proceedings] 1991 IEEE International Joint Conference on Neural Networks*, pages 367–372 vol.1, 1991.

[183] Kiyotoshi Matsuoka. Noise injection into inputs in back-propagation learning. *IEEE Transactions on Systems, Man, and Cybernetics*, 22(3):436–440, 1992.

[184] Chuan Wang and Jose C Principe. Training neural networks with additive noise in the desired signal. *IEEE Transactions on Neural Networks*, 10(6):1511–1517, 1999.

[185] Azian Azamimi, Yoko Uwate, and Yoshifumi Nishio. Effect of chaos noise on the learning ability of backpropagation algorithm in feed-forward neural network. In *2010 6th International Colloquium on Signal Processing & its Applications*, pages 1–4. IEEE, 2010.

[186] Juan Manuel Alonso-Weber, MP Sesmero, and Araceli Sanchis. Combining additive input noise annealing and pattern transformations for improved handwritten character recognition. *Expert systems with applications*, 41(18):8180–8188, 2014.

[187] IV Isaev and SA Dolenko. Training with noise as a method to increase noise resilience of neural network solution of inverse problems. *Optical Memory and Neural Networks*, 25(3):142–148, 2016.

[188] Bart Kosko, Kartik Audhkhasi, and Osonde Osoba. Noise can speed backpropagation learning and deep bidirectional pretraining. *Neural Networks*, 129:359–384, 2020.

[189] Warick M Brown, Tamás D Gedeon, and David I Groves. Use of noise to augment training data: a neural network method of mineral–potential mapping in regions of limited known deposit examples. *Natural Resources Research*, 12(2):141–152, 2003.

[190] Jianping Hua, James Lowey, Zixiang Xiong, and Edward R Dougherty. Noise-injected neural networks show promise for use on small-sample expression data. *BMC bioinformatics*, 7(1):1–14, 2006.

[191] Yinan Li and Fang Liu. Whiteout: Gaussian adaptive noise regularization in deep neural networks. *arXiv preprint arXiv:1612.01490*, 2016.

[192] Soon Hoe Lim, N Benjamin Erichson, Francisco Utrera, Winnie Xu, and Michael W Mahoney. Noisy feature mixup. *arXiv preprint arXiv:2110.02180*, 2021.

[193] Elahe Arani, Fahad Sarfraz, and Bahram Zonooz. Noise as a resource for learning in knowledge distillation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3129–3138, 2021.

[194] Zhonghui You, Jinmian Ye, Kunming Li, Zenglin Xu, and Ping Wang. Adversarial noise layer: Regularize neural network by adding noise. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 909–913. IEEE, 2019.

[195] Linara Adilova, Nathalie Paul, and Peter Schlicht. Introducing noise in decentralized training of neural networks. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 37–48. Springer, 2018.

[196] Ashwini Sapkal and UV Kulkarni. Modified backpropagation with added white gaussian noise in weighted sum for convergence improvement. *Procedia computer science*, 143:309–316, 2018.

[197] Jiashuo Shi, Mingce Chen, Dong Wei, Chai Hu, Jun Luo, Haiwei Wang, Xinyu Zhang, and Changsheng Xie. Anti-noise diffractive neural network for constructing an intelligent imaging detector array. *Optics Express*, 28(25):37686–37699, 2020.

[198] Kirill Bykov, Anna Hedström, Shinichi Nakajima, and Marina M-C Höhne. Noisegrad: enhancing explanations by introducing stochasticity to model weights. *arXiv preprint arXiv:2106.10185*, 2021.

[199] Peter J Edwards and Alan F Murray. Fault tolerance via weight noise in analog vlsi implementations of mlps-a case study with epsilon. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 45(9):1255–1262, 1998.

[200] Mo Zhou, Tianyi Liu, Yan Li, Dachao Lin, Enlu Zhou, and Tuo Zhao. Toward understanding the importance of noise in training neural networks. In *International Conference on Machine Learning*, pages 7594–7602. PMLR, 2019.

[201] Lingling Duan, Fabing Duan, François Chapeau-Blondeau, and Derek Abbott. Noise-boosted backpropagation learning of feedforward threshold neural networks for function approximation. *IEEE Transactions on Instrumentation and Measurement*, 70:1–12, 2021.

[202] PRATIK Chaudhari and STEFANO Soatto. The effect of gradient noise on the energy landscape of deep networks. Technical report, Technical Report Preprint, 2015.

[203] Arvind Neelakantan, Luke Vilnis, Quoc V Le, Ilya Sutskever, Lukasz Kaiser, Karol Kurach, and James Martens. Adding gradient noise improves learning for very deep networks. *arXiv preprint arXiv:1511.06807*, 2015.

[204] Baharan Mirzasoleiman, Kaidi Cao, and Jure Leskovec. Coresets for robust training of deep neural networks against noisy labels. *Advances in Neural Information Processing Systems*, 33:11465–11477, 2020.

[205] Xuefeng Jiang, Sheng Sun, Yuwei Wang, and Min Liu. Towards federated learning against noisy labels via local self-regularization. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 862–873, 2022.

[206] Tingting Wu, Xiao Ding, Minji Tang, Hao Zhang, Bing Qin, and Ting Liu. Stgn: an implicit regularization method for learning with noisy labels in natural language processing. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7587–7598, 2022.

[207] Hwanjun Song, Minseok Kim, Dongmin Park, Yooju Shin, and Jae-Gil Lee. Learning from noisy labels with deep neural networks: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[208] Steven W Smith et al. The scientist and engineer's guide to digital signal processing, 1997.

[209] Fraidoon Mazda. *Telecommunications engineer's reference book*. Butterworth-Heinemann, 2014.

[210] Elliot Anshelevich, Anirban Dasgupta, Jon Kleinberg, Éva Tardos, Tom Wexler, and Tim Roughgarden. The price of stability for network design with fair cost allocation. *SIAM Journal on Computing*, 38(4):1602–1623, 2008.

[211] Elias Koutsoupias and Christos Papadimitriou. Worst-case equilibria. *Computer science review*, 3(2):65–69, 2009.

[212] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021.

[213] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[214] Haoran Wen, Yang Du, Eng Gee Lim, Huiqing Wen, Ke Yan, Xingshuo Li, and Lin Jiang. A solar forecasting framework based on federated learning and distributed computing. *Building and Environment*, 225:109556, 2022.

[215] Duy-Dong Le, Anh-Khoa Tran, Minh-Son Dao, Kieu-Chinh Nguyen-Ly, Hoang-Son Le, Xuan-Dao Nguyen-Thi, Thanh-Qui Pham, Van-Luong Nguyen, and Bach-Yen Nguyen-Thi. Insights into multi-model federated learning: An advanced approach for air quality index forecasting. *Algorithms*, 15(11):434, 2022.

[216] Caren Marzban and Gregory J Stumpf. A neural network for damaging wind prediction. *Weather and Forecasting*, 13(1):151–163, 1998.

[217] Caren Marzban and Gregory J Stumpf. A neural network for tornado prediction based on doppler radar-derived attributes. *Journal of Applied Meteorology and Climatology*, 35(5):617–626, 1996.

[218] Shaza M Abd Elrahman and Ajith Abraham. A review of class imbalance problem. *Journal of Network and Innovative Computing*, 1(2013):332–340, 2013.

[219] Apurva Sonak and RA Patankar. A survey on methods to handle imbalance dataset. *Int. J. Comput. Sci. Mobile Comput*, 4(11):338–343, 2015.

[220] Theodore B Trafalis, Indra Adrianto, Michael B Richman, and S Lakshmivarahan. Machine-learning classifiers for imbalanced tornado data. *Computational Management Science*, 11(4):403–418, 2014.

[221] Theodore B Trafalis, Huseyin Ince, and Michael B Richman. Tornado detection with support vector machines. In *International Conference on Computational Science*, pages 289–298. Springer, 2003.

[222] Elaheh Jafarigol and Theodore Trafalis. Imbalanced learning with parametric linear programming support vector machine for weather data application. *SN Computer Science*, 1(6):1–11, 2020.

[223] Alberto Fernández, Sara del Río, Nitesh V Chawla, and Francisco Herrera. An insight into imbalanced big data classification: outcomes and challenges. *Complex & Intelligent Systems*, 3(2):105–120, 2017.

[224] Amalia Luque, Alejandro Carrasco, Alejandro Martín, and Ana de las Heras. The impact of class imbalance in classification performance metrics based on the binary confusion matrix. *Pattern Recognition*, 91:216–231, 2019.

[225] David J Hand. Measuring classifier performance: a coherent alternative to the area under the roc curve. *Machine learning*, 77(1):103–123, 2009.

[226] Mohamed Bekkar, Hassiba Kheliouane Djemaa, and Taklit Akrouf Alitouche. Evaluation measures for models assessment over imbalanced data sets. *J Inf Eng Appl*, 3(10), 2013.

[227] César Ferri, José Hernández-Orallo, and R Modroiu. An experimental comparison of performance measures for classification. *Pattern Recognition Letters*, 30(1):27–38, 2009.

[228] Mohammad Hossin and MN Sulaiman. A review on evaluation metrics for data classification evaluations. *International Journal of Data Mining & Knowledge Management Process*, 5(2):1, 2015.

[229] Mehrdad Fatourechi, Rabab K Ward, Steven G Mason, Jane Huggins, Alois Schlögl, and Gary E Birch. Comparison of evaluation metrics in classification applications with imbalanced datasets. In *2008 Seventh International Conference on Machine Learning and Applications*, pages 777–782. IEEE, 2008.

[230] Bartosz Krawczyk. Learning from imbalanced data: open challenges and future directions. *Progress in Artificial Intelligence*, 5(4):221–232, 2016.

[231] T Ryan Hoens and Nitesh V Chawla. Imbalanced datasets: from sampling to classifiers. *Imbalanced learning: Foundations, algorithms, and applications*, pages 43–59, 2013.

[232] Nitesh V Chawla. Data mining for imbalanced datasets: An overview. In *Data mining and knowledge discovery handbook*, pages 875–886. Springer, 2009.

[233] Vaishali Ganganwar. An overview of classification algorithms for imbalanced datasets. *International Journal of Emerging Technology and Advanced Engineering*, 2(4):42–47, 2012.

[234] Guo Haixiang, Li Yijing, Jennifer Shang, Gu Mingyun, Huang Yuanyue, and Gong Bing. Learning from class-imbalanced data: Review of methods and applications. *Expert Systems with Applications*, 73:220–239, 2017.

[235] Alberto Fernández, Salvador García, Mikel Galar, Ronaldo C Prati, Bartosz Krawczyk, and Francisco Herrera. Learning from imbalanced data streams. In *Learning from imbalanced data sets*, pages 279–303. Springer, 2018.

[236] Andrew P Bradley. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern recognition*, 30(7):1145–1159, 1997.

[237] Charles X Ling, Jin Huang, Harry Zhang, et al. Auc: a statistically consistent and more discriminating measure than accuracy. In *Ijcai*, volume 3, pages 519–524, 2003.

[238] Chenguang Xiao and Shuo Wang. An Experimental Study of Class Imbalance in Federated Learning. In *2021 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, dec 5 2021.

[239] Miao Yang, Ximin Wang, Hongbin Zhu, Haifeng Wang, and Hua Qian. Federated Learning with Class Imbalance Reduction. In *2021 29th European Signal Processing Conference (EUSIPCO)*. IEEE, aug 23 2021.

[240] Inderjeet Mani and I Zhang. knn approach to unbalanced data distributions: a case study involving information extraction. In *Proceedings of workshop on learning from imbalanced datasets*, volume 126, 2003.

[241] Muhammad Shoaib Farooq, Rabia Tehseen, Junaid Nasir Qureshi, Uzma Omer, Rimsha Yaqoob, Hafiz Abdullah Tanweer, and Zabihullah Atal. Ffm: Flood forecasting model using federated learning. *IEEE Access*, 11:24472–24483, 2023.

[242] Elaheh Jafarigol and Theodore Trafalis. A distributed approach to meteorological predictions: Addressing data imbalance in precipitation prediction models through federated learning and gans. *arXiv preprint arXiv:2310.13161*, 2023.

[243] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015.

[244] Justin M Johnson and Taghi M Khoshgoftaar. Survey on deep learning with class imbalance. *Journal of Big Data*, 6(1):27, 2019.

[245] Apurva Sonak, Ruhi Patankar, and Nitin Pise. A new approach for handling imbalanced dataset using ann and genetic algorithm. In *2016 International Conference on Communication and Signal Processing (ICCSP)*, pages 1987–1990. IEEE, 2016.

[246] Feng Bao, Yue Deng, Youyong Kong, Zhiquan Ren, Jinli Suo, and Qionghai Dai. Learning deep landmarks for imbalanced classification. *IEEE Transactions on Neural Networks and Learning Systems*, 2019.

[247] David A Cieslak, Nitesh V Chawla, and Aaron Striegel. Combating imbalance in network intrusion datasets. In *GrC*, pages 732–737, 2006.

[248] Ch Sarada and M SathyaDevi. Imbalanced big data classification using feature selection under-sampling. *CVR Journal of Science and Technology*, 17(1):78–82, 2019.

[249] Alexander Liu, Joydeep Ghosh, and Cheryl E Martin. Generative oversampling for mining imbalanced datasets. In *DMIN*, pages 66–72, 2007.

[250] Garima Goel, Liam Maguire, Yuhua Li, and Sean McLoone. Evaluation of sampling methods for learning from imbalanced data. In *International Conference on Intelligent Computing*, pages 392–401. Springer, 2013.

[251] Alberto Fernández, Salvador Garcia, Francisco Herrera, and Nitesh V Chawla. Smote for learning from imbalanced data: progress and challenges, marking the 15-year anniversary. *Journal of artificial intelligence research*, 61:863–905, 2018.

[252] Zhenyu Wu, Wenfang Lin, and Yang Ji. An integrated ensemble learning model for imbalanced fault diagnostics and prognostics. *IEEE Access*, 6:8394–8402, 2018.

[253] Xin-Li Yang, David Lo, Xin Xia, Qiao Huang, and Jian-Ling Sun. High-impact bug report identification with imbalanced learning strategies. *Journal of Computer Science and Technology*, 32(1):181–198, 2017.

[254] M Mostafizur Rahman and Darryl N Davis. Addressing the class imbalance problem in medical datasets. *International Journal of Machine Learning and Computing*, 3(2):224, 2013.

[255] Qingyong Wang, Yun Zhou, Weiming Zhang, Zhangui Tang, and Xiaojing Chen. Adaptive sampling using self-paced learning for imbalanced cancer data prediagnosis. *Expert Systems with Applications*, 152:113334, 2020.

[256] Yuan Xie and Tao Zhang. Imbalanced learning for fault diagnosis problem of rotating machinery based on generative adversarial networks. In *2018 37th Chinese Control Conference (CCC)*, pages 6017–6022. IEEE, 2018.

[257] Anjana Gosain and Saanchi Sardana. Handling class imbalance problem using oversampling techniques: A review. In *2017 International Conference on Advances in Computing, Communications, and Informatics (ICACCI)*, pages 79–85. IEEE, 2017.

[258] Ruchika Malhotra and Juhi Jain. Handling imbalanced data using ensemble learning in software defect prediction. In *2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, pages 300–304. IEEE, 2020.

[259] Nitesh V Chawla, Aleksandar Lazarevic, Lawrence O Hall, and Kevin W Bowyer. Smoteboost: Improving prediction of the minority class in boosting. In *European conference on principles of data mining and knowledge discovery*, pages 107–119. Springer, 2003.

[260] Hui Han, Wen-Yuan Wang, and Bing-Huan Mao. Borderline-smote: a new over-sampling method in imbalanced data sets learning. In *International conference on intelligent computing*, pages 878–887. Springer, 2005.

[261] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

[262] Damien Dablain, Bartosz Krawczyk, and Nitesh V Chawla. Deepsmote: Fusing deep learning and smote for imbalanced data. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[263] Guillaume Lemaître, Fernando Nogueira, and Christos K Aridas. Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *The Journal of Machine Learning Research*, 18(1):559–563, 2017.

[264] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[265] Zhuocheng Zhou, Bofeng Zhang, Ying Lv, Tian Shi, and Furong Chang. *Data Augment in Imbalanced Learning Based on Generative Adversarial Networks*, pages 21–30. Springer International Publishing, 2019.

[266] Yangru Huang, Yi Jin, Yidong Li, and Zhiping Lin. Towards imbalanced image classification: a generative adversarial network ensemble learning method. *IEEE Access*, 8:88399–88409, 2020.

[267] Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. Modeling tabular data using conditional gan. *Advances in Neural Information Processing Systems*, 32, 2019.

[268] Junhai Zhai, Jiaxing Qi, and Sufang Zhang. Imbalanced data classification based on diverse sample generation and classifier fusion. *International Journal of Machine Learning and Cybernetics*, pages 1–16, 2022.

[269] Tjeng Wawan Cenggoro et al. Deep learning for imbalance data classification using class expert generative adversarial network. *Procedia Computer Science*, 135:60–67, 2018.

[270] Yuxiao Huang, Kara G Fields, and Yan Ma. A tutorial on generative adversarial networks with application to classification of imbalanced data. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 15(5):543–552, 2022.

[271] Pavle Divovic, Predrag Obradovic, and Marko Misic. Balancing Imbalanced Datasets Using Generative Adversarial Neural Networks. In *2021 29th Telecommunications Forum (TELFOR)*. IEEE, nov 23 2021.

[272] Hwi-Yeon Cho and Yong-Hyuk Kim. A genetic algorithm to optimize smote and gan ratios in class imbalanced datasets. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion*, pages 33–34, 2020.

[273] Pourya Shamsolmoali, Masoumeh Zareapoor, Linlin Shen, Abdul Hamid Sadka, and Jie Yang. Imbalanced data learning by minority class augmentation using capsule adversarial networks. *Neurocomputing*, 459:481–493, 2021.

[274] M Aslı Aydin. Using generative adversarial networks for handling class imbalance problem. In *2021 29th Signal Processing and Communications Applications Conference (SIU)*, pages 1–4. IEEE, 2021.

[275] Hyun-Soo Choi, Dahuin Jung, Siwon Kim, and Sungroh Yoon. Imbalanced Data Classification via Cooperative Interaction Between Classifier and Generator. *IEEE Transactions on Neural Networks and Learning Systems*, 33(8):3343–3356, 8 2022.

[276] Jia Luo, Jinying Huang, and Hongmei Li. A case study of conditional deep convolutional generative adversarial networks in machine fault diagnosis. *Journal of Intelligent Manufacturing*, 32:407–425, 2021.

[277] Gaofeng Huang and Amir Hossein Jafari. Enhanced balancing gan: Minority-class image generation. *Neural Computing and Applications*, pages 1–10, 2021.

[278] Sankha Subhra Mullick, Shounak Datta, and Swagatam Das. Generative adversarial minority oversampling. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1695–1704, 2019.

[279] Justin Engelmann and Stefan Lessmann. Conditional wasserstein gan-based oversampling of tabular data for imbalanced learning. *Expert Systems with Applications*, 174:114582, 2021.

[280] Zhe Li, Yi Jin, Yidong Li, Zhiping Lin, and Shan Wang. Imbalanced adversarial learning for weather image generation and classification. In *2018 14th IEEE*

International Conference on Signal Processing (ICSP), pages 1093–1097. IEEE, 2018.

[281] Shiven Sharma, Colin Bellinger, Bartosz Krawczyk, Osmar Zaiane, and Nathalie Japkowicz. Synthetic oversampling with the majority class: A new perspective on handling extreme imbalance. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 447–456. IEEE, 2018.

[282] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[283] Georgios Douzas and Fernando Bacao. Effective data generation for imbalanced learning using conditional generative adversarial networks. *Expert Systems with Applications*, 91:464–471, 1 2018.

[284] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[285] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4681–4690, 2017.

[286] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *Proceedings of the IEEE International Conference on computer vision*, pages 5907–5915, 2017.

[287] Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for gans do actually converge? In *International conference on machine learning*, pages 3481–3490. PMLR, 2018.

[288] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR, 2017.

[289] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. *Advances in neural information processing systems*, 30, 2017.

[290] Anuraganand Sharma, Prabhat Kumar Singh, and Rohitash Chandra. Smotified-gan for class imbalanced pattern classification problems. *Ieee Access*, 10:30655–30665, 2022.

[291] Han Yu, Zelei Liu, Yang Liu, Tianjian Chen, Mingshu Cong, Xi Weng, Dusit Niyato, and Qiang Yang. A fairness-aware incentive scheme for federated learning. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pages 393–399, 2020.

[292] Lingjuan Lyu, Xinyi Xu, Qian Wang, and Han Yu. Collaborative fairness in federated learning. In *Federated Learning*, pages 189–204. Springer, 2020.

[293] Mohammad Al-Rubaie, Pei-yuan Wu, J Morris Chang, and Sun-Yuan Kung. Privacy-preserving pca on horizontally-partitioned data. In *2017 IEEE Conference on Dependable and Secure Computing*, pages 280–287, 2017.

[294] Felix Sattler, Simon Wiedemann, Klaus-Robert Müller, and Wojciech Samek. Robust and communication-efficient federated learning from non-i.i.d. data. *IEEE Transactions on Neural Networks and Learning Systems*, 31(9):3400–3413, 2020.

[295] Xiaodong Ma, Jia Zhu, Zhihao Lin, Shanxuan Chen, and Yangjie Qin. A state-of-the-art survey on solving non-iid data in federated learning. *Future Generation Computer Systems*, 135:244–258, 2022.

[296] Hangyu Zhu, Haoyu Zhang, and Yaochu Jin. From federated learning to federated neural architecture search: a survey. *Complex & Intelligent Systems*, 7(2):639–657, 2021.

[297] Arash Mehrjou. Federated learning as a mean-field game. *ArXiv*, abs/2107.03770, 2021.

[298] Samuel Yen-Chi Chen and Shinjae Yoo. Federated quantum machine learning. *Entropy*, 23(4):460, 2021.

[299] Jialin Zhong, Yahui Wu, Wubin Ma, Su Deng, and Haohao Zhou. Optimizing multi-objective federated learning on non-iid data with improved nsga-iii and hierarchical clustering. *Symmetry*, 14(5):1070, 2022.