Instituto Tecnológico
y de Estudios Superiores de Occidente

Reconocimiento de validez oficial de estudios de nivel superior según acuerdo secretarial 15018, publicado en el Diario Oficial de la Federación del 29 de noviembre de 1976.

Departamento de Matemáticas y Física

Maestría en Ciencia de Datos

Schemes based on Federated Learning for decentralized training in Machine Learning models

---

THESIS to obtain the **DEGREE** of
MASTER IN DATA SCIENCE

Presents: **PEDRO MARTÍNEZ GUTIÉRREZ**

Advisor: **DR. GEMA BERENICE GUDIÑO MENDOZA**

Tlaquepaque, Jalisco. November 2022.

Instituto Tecnológico
y de Estudios Superiores de Occidente

Reconocimiento de validez oficial de estudios de nivel superior según acuerdo secretarial 15018, publicado en el Diario Oficial de la Federación del 29 de noviembre de 1976.

Departamento de Matemáticas y Física

Maestría en Ciencia de Datos

Esquemas basados en Federated Learning para un entrenamiento descentralizado en modelos de Machine Learning

---

TESIS que para obtener el **GRADO** de
MAESTRO EN CIENCIA DE DATOS

Presenta: **PEDRO MARTÍNEZ GUTIÉRREZ**

Asesora: **DRA. GEMA BERENICE GUDIÑO MENDOZA**

Tlaquepaque, Jalisco. noviembre de 2022.

ACKNOWLEDGMENTS

The author would like to thank his family, friends, and co-workers who provided their support to develop and complete this project. Special thanks to ITESO for the financial assistance provided through a generous scholarship, which allowed to start and apply knowledge to improve our community. Congratulations and thanks to Dr. Gema Berenice Gudiño Mendoza, without you, your virtues, your patience, and perseverance this document would not be possible. Thank you all!

AGRADECIMEINTOS

El autor desea dar gracias a toda su familia, amigos y compañeros de trabajo que brindaron su apoyo para desarrollar y culminar este proyecto. En especial, agradecer al ITESO por el apoyo financiero brindado mediante una generosa beca, lo que permitió comenzar y aplicar conocimientos para mejorar nuestra comunidad. Felicitaciones y agradecimientos a la Dra. Gema Berenice Gudiño Mendoza, sin usted, sus virtudes, su paciencia y constancia este trabajo no sería posible. ¡Muchas gracias a todos!

DEDICATION

The author dedicates this thesis to his family, his girlfriend Alondra, and his dog Salchicha all of them always present.

DEDICATORIA

El autor dedica esta tesis a su familia, su novia Alondra y a su perrita Salchicha siempre presentes.

SUMMARY

Standard Machine Learning approaches require large amounts of data usually centralized in data centers. In these approaches, there is only one device responsible for the training of the whole process. New collaborative approaches allow the training of common models from different decentralized devices, each one holding local data samples. An example is Federated Learning.

In recent years, along with the blooming of Machine Learning based applications and services, ensuring data privacy and security have become a critical obligation. In this work, three training procedures based on Federated Learning were tested: FedAvg, FedADA, and LoADABoost comparing their performance versus a traditional centralized training method.

Using public information from written reviews about movies, a neural network algorithm was implemented. The objective of the model was to predict whether a review is positive or negative. Utilizing the F1 Score as a performance metric, the hypothesis was to validate whether the Federated Learning training methods are similar to traditional centralized training methodologies.

After the implementation of the same neural network with different training methodologies, no major differences or changes in performance were noted, concluding that Federated Learning is indeed a similar and viable training methodology.

RESUMEN

Los enfoques estándar de Machine Learning requieren grandes cantidades de datos que suelen estar guardados en un único servidor. En estos enfoques, sólo hay un dispositivo responsable del entrenamiento de todo el proceso. Las nuevas técnicas colaborativas permiten entrenar modelos a partir de diferentes dispositivos descentralizados, cada uno de los cuales posee muestras de datos locales. Un ejemplo es Federated Learning.

En los últimos años, junto con la explosión de las aplicaciones y servicios basados en Machine Learning, garantizar la privacidad y seguridad de los datos se ha convertido en una obligación crítica. En este trabajo se han probado tres procedimientos de entrenamiento basados en Federated Learning: FedAvg, FedADA y LoADABoost comparando su rendimiento frente a un método de entrenamiento centralizado tradicional.

Utilizando la información pública de las críticas escritas sobre las películas, se implementó un algoritmo de red neuronal. El objetivo del modelo era predecir si una crítica es positiva o negativa. Utilizando la puntuación F1 como métrica de rendimiento, la hipótesis era validar si los métodos de entrenamiento de Federated Learning son similares a las metodologías tradicionales de entrenamiento centralizado.

Tras la implementación de la misma red neuronal con diferentes metodologías de entrenamiento, no se observaron grandes diferencias o cambios en el rendimiento, concluyendo que Federated Learning es efectivamente una metodología de entrenamiento viable.

LIST OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS AND ABBREVIATIONS

ML	Machine Learning
FL	Federated Learning
AI	Artificial Intelligence
NNs	Neural Networks
IoT	Internet of Things
IoV	Internet of Vehicles
MEC	mobile-edge computing
HER	Hospital Electronics Records
iid	Independent and identically distributed
HPO	Hyper Parameter Optimization

1. INTRODUCTION

Summary: *This chapter briefly presents the background of the object of study, justification of work and definition of the problem to solve.*

1.1. Background

Nowadays, millions of digital connections are made all around the world, at every second, it can happen through an email, a chat message, tweets, reviews, or voice messages. Plus, these interactions are constantly increasing in an even greater number of channels. In consequence, this situation complicates the gathering and extracting process for analyzing important features from the information. Considering this condition, one of the most used techniques to deal with high volumes of data is Machine Learning.

Machine learning is a form of Artificial Intelligence that enables a system to learn from data rather than through explicit programming. Their main goal is to predict a result based on previous observations using an algorithm, nevertheless, this is not a simple process, it requires complex mathematics and statistics.

Neural networks are one widely used technique to implement machine learning. The machine learning, a neural network algorithm focuses on the use of data to imitate the way that humans learn, based on previous experiences, in other words, the accuracy is gradually increased as more tests are performed, and more data is used.

Having the previous context, when it is trained a neural network model, all the data is centralized for a better performance and ease of management. This represents a problem when the data is highly sensitive or contains personal information (medical records, military defense, and location records, for example) because this data needs to be treated under legal constraints of non-disclosure or even under strict supervision.

As a response to the previous problem, a new technique of artificial intelligence was born. Federated Learning (FL). FL is a decentralized approach to model training. With FL, models are trained on third-party data by sending the model to the entity who keeps the data instead of sending the data to the entity who consolidate the model. All the data is left in place, reducing data privacy concerns and network bandwidth requirements.

The federated learning approach for training deep networks was first articulated in a 2016 article published by Google AI team [3]. After some years, it has been observed a rapidly growing community of federated learning researchers, which leaded to the case of study methodologies of our work LoAdaBoost [4], FedMD [5] and FedAvg [6].

1.2. Justification

During the COVID-19 pandemic, most commercial interactions were forced to take place remotely, for example, text messages, emails, and video calls. Mexico has followed, along with other Latin American countries, the international trend of ensuring the protection of personal information. In this sense, the protection of personal data is a fundamental right recognized by the Constitution of Mexico [7].

Said the above, the amount of personal or private information increased exponentially. The data that a company stores about its customers and their interactions is considered sensitive. In accordance with provisions of article 3, section II of the Federal Law of Protection of Personal Data in the Possession of Individuals, databases will be understood as any ordered set of personal data that allows identifying individuals or making them identifiable, as well as the access to personal data according to specific criteria, regardless of the form or method of their creation, type of medium, processing, storage, organization, and access [7].

Also, most governments are implementing similar regulations, for example the General Data Protection Regulation (GDPR) in Europe Union that the regulation protects fundamental rights and freedoms of natural people and in particular their right to the protection of personal data. Member States shall lay down the rules on other penalties applicable to infringements of this regulation in particular for infringements which are not subject to administrative fines [8].

ML-based service providers not only confront with difficulties in collecting and managing data across heterogeneous sources but also challenges of complying with rigorous data protection regulations such as EU/UK GDPR [7]. and Mexico's Federal Law on the Protection of Personal Data in the Possession of Private Parties (Ley Federal de Protección de Datos Personales en Posesión de los Particulares).

As it is stated previously, governments oversee law enforcement about the personal data protection, implying that a company shall always communicate and ask for client's consent in how their information is used, where it is stored and if it is shared to other third parties. As an example of this, we can point out a financial institution sending credit information data to an outside server or the upload of medical records to a centralized hospital server, without client's consent.

Nowadays, it is highly important guarantee confidentiality and security of customer information, always it is preferable not to share client's data with other third parties, this to avoid the misuse of sensitive and personal data and the possibility of data leaks. With this context, FL is practical solution for performing ML models with privacy and compliance with the law.

1.3. Problem

The digital interactions of the customer with the company are a priceless resource for the business. It can be used to improve customer service, new products development, or market research. However, as it is stated previously, there are regulations that moderates the usage of data making it difficult to centralize or share to an outsider server.

In traditional ML algorithms, large-scale data collection and processing at a powerful cloud-based server entails the single-point-of-failure and the risks of severe data breaches. Foremost, centralized data processing and management impose limited transparency and provenance on the system, which could lead to the lack of trust from end-users as well as the difficulty in complying with the government's data privacy regulations [9].

1.4. Hypothesis

In this thesis will be discussed and performed an extensive descriptive analysis about if it is possible to get a robust training method without extracting/centralizing data from the original source and to compare the performance versus a centralized traditional training methodology.

As part of the experiment control parameters, it will be set the following conditions for reproducibility:

- Neural networks as the machine learning method. Considering a balanced dataset.
- Using a public dataset (IMBD) and simulating a part of the dataset as private which cannot be accessible for the training of a machine learning model.
- Comparing 4 different types of training methods: Centralized standard, FedAvg, FedMD, FedLoADABoost.
- Using F1 Score as evaluation metric.
- Using multiple neural network structures and applying them using the same training model.

1.5. Goals

1.5.1. General goal:

Analyze different FL training models using F1 Score as evaluation metric and prove the hypothesis whether if the proposed FL training methods presents similar performance metrics versus the traditional centralized training method.

1.5.2. Specific Goals

1) Propose 3 different training methods based on federated learning which do not require to extract the end user information from their servers.
2) Determine if the suggested training methods perform the same as the centralized standard training method using the F1 as metric.
3) Design the experiment considering an IMBD public dataset and simulating a private not shareable information for training.
4) Identify the most important differences between the evaluated training methods.

1.6. Scientific technological novelty or contribution

We are now living in a data-driven world where most applications and services such as healthcare and medical services, autonomous cars, and finance applications are based on artificial intelligence (AI) technology with complex data-hungry algorithms. AI has been showing advances in every aspect of lives and expected to "change the world more than anything in the history of mankind" [9].

FL enhances ML model training by reaching greater amounts of data in distributed locations and on edge devices, at the point of generation and consumption. This approach can provide a significant untapped reservoir of data that greatly expands the available dataset. Also, it important to mention that there are not many related academic contributions about FL and this case of study could be useful for the growing community.

For this case of study, the implementation of FL will be using a movie reviews dataset. The mentioned dataset was selected because it is important to simulate sensitive information of customers (such as their opinion). This application represents a good opportunity for developing a business model case in which a company needs to make predictions of their customer's feedback without exposing their sensitive information.

2. STATE OF THE ART OR TECHNIQUE

Summary: *This chapter presents a compilation of multiple additional works related to Federated Learning in different industries, Sentiment Analysis and Federated Learning models for hyperparameters parameter selection.*

2.1. Federated Learning fields of application

FL is a scheme in which several participants work collectively to unravel ML training problems, with a coordinator synchronizing the procedure. This decision correspondingly, guarantees that the data are secluded [10]. Due to the nature of the technique, the main applications reported so far are related to disruptive technologies the or analysis of highly sensitive data.

Application for mobile devices
As the leader in smartphone keyboards technology prediction, Google proposed the usage of FL to forecast users input data using Gboard on Android. Chen et al. [10]; Leroy et al. [11]; Hard et al. [12], and Yang et al. [13] have all made improvements to keyboard prediction. Emoji prediction is also a center for study [14].

Another field of study is finding a solution to avoid network congestion, most internet providers choose to provide a service near to the client, rather than integrating cloud computing and cloud storage into the main network. Mobile edge computing (MEC) is the name given to this technology; however, it comes with a higher danger of data leakage. The combination of FL and MEC is one potential approach. Wang et al. [15] develop an "In-Edge AI" framework that combines FL founded on deep reinforcement learning with a MEC system to additionally enhance resource apportionment issues. Furthermore, Qian et al. [17] focused on the application of FL to MEC. They created a confidentiality-consciousness service placement technique that allows them to deliver high-quality service by secreting needed services on edge servers near to customers.

Application for healthcare industries
A clear example of highly sensitive data is the medical records of healthcare facilities. FL has a bright future in health care as a disruptive technique of conserving data confidentiality. Although each medical facility may have a huge volume of patient data, this may not be sufficient to train their prediction methods [18]. One of the effective options for breaking down the boundaries of analysis across various hospitals is to combine FL with illness prediction. Antunes et al. [19] conducted an SLR o FL for healthcare and focused on recent studies on FL for Hospital Electronic Records (HER) applications. Brismi et al. [20] initiated work about time patient prediction of times a patient on when they will be admitted to the hospital in the future. Li et al. [21] described a methodology using non independent and identically distributed HER dataset to calculate the likelihood of death and the length of time spent in the hospital.

Studies have also shown that FL can be used to assess genuine data from health text records in the realm of natural language processing (NLP). The necessity for unstructured data processing of clinical notes is highlighted by Liu et al. [22]. It was the first time NLP was used in conjunction with FL. They used a two-stage federated training model that included preprocessing to forecast a representation model for each patient and phenotyping training

to investigate each kind of sickness. FL has recently been popular in the field of biological image analysis.

Application in industrial engineering
As a result of FL success in data confidentiality fortification, due to legal and regulatory restrictions, data in certain sectors is not readily accessible. However, we can only take advantage of these dispersed datasets to acquire limitless benefits if FL is applied to these locations. In the context of environmental protection, Hu et al. [33] devised a new conservational monitoring framework based on FL to compensate for the difficult interchangeability of observing data. Thus, observing data scattered from many sensors might be used to improve the collaborative model's performance.

FL is also used to do visual inspections [23]. It could not solitarily assist us to overcome the issue of insufficient faulty illustrations for detecting flaws in production jobs, nonetheless, it could similarly provide manufacturers with privacy assurances. Liu et al. [24] use FL to collect diversiform illustrations from federated tasks for improved grounding applications in picture fields.

2.2. Sentiment analysis

Sentiment analysis is one application field in the context of natural language processing, which is devoted to the analysis of affective states. The study case in this thesis is a dataset for sentiment analysis, this is why it is important to mention related works. The movie's reviews Orestes et al. [25] uses natural language processing (NLP) essential techniques, a sentiment lexicon enhanced with the assistance of SentiWordNet, and fuzzy sets to estimate the semantic orientation polarity and its intensity for sentences, which provides a foundation for computing with sentiments. The proposed hybrid method is applied to three different datasets and the results achieved are compared to those obtained using Naïve Bayes and Maximum Entropy techniques.

The paper [29] by Li et al. answers their call for applying FedMD to the field of Natural Language Processing (NLP). The primary contribution of this work is FedMD, a new federated learning framework that enables participants to independently design their model. For the experiment sentiment classification of tweet messages on the Sentiment140 dataset was analyzed. This challenge allowed to experiment not only with the learning setup, but also with different model architectures to benchmark model performance. Enabling a review of multiple variations of neural networks. This article was a major inspiration and model for this thesis work, and their results reflected that the FL Model performed with similar results compared versus the traditional ML methods.

2.3. Hyperparameter selection in Federated Learning

Tuning hyperparameters is a crucial but arduous part of the machine learning pipeline. Hyperparameter optimization is even more challenging in federated learning; here, the need to keep data on device and perform local training makes it difficult to efficiently train and evaluate configurations.

The performance of ML models is sensitive to hyperparameters. Hyperparameter optimization (HPO) aims at tuning the hyperparameters to improve convergence speed and quality. However, due to the wide range of hyperparameter choices and their corresponding dynamic schedules, tuning these hyperparameters is a time and resource consuming task. To improve the efficiency of hyperparameter optimization, various works have been proposed, including multi-armed bandits, evolutionary algorithms, and Bayesian Optimization. Li et al. [26] treats the problem of hyperparameter and discards the worst configurations during different tuning stages. Young et al. [27] introduces the framework for optimizing hyperparameters using genetic algorithms. A new Bayesian Optimization method was introduced in Wu et al. [28] by creating its acquisition function to leverage multi-fidelity feature.

Having as reference the previously mentioned, the scope of this work was limited to follow and implement from the ground, standard methodologies of FL training.

2.4. Federated Learning model training

Part of the added value of the FL methodology is their ability to train a dataset via different methods, it is clear that FL is still a new methodology with a long way to go, nevertheless it can be observed that multiple authors are focusing their efforts on studying their implementations. Maruan Al-Shedivat et al. [30] While exact inference is often intractable, they perspective, it is to provide a principled way to search for global optima in federated settings. Further, starting with the analysis a computation- and communication efficient approximate posterior inference algorithm—federated posterior averaging (FPA)

In the seminal paper by McMahan et al. [31], their main effort in federated learning has focused on understanding of FedAvg (also known as local Stochastic Gradient Descent) as an optimization algorithm using homogeneous iid data.

Sannara Ek et al. [31] proposed a new aggregation algorithm, called FedDist, that has been designed to meet the specific needs of pervasive applications, presented an extensive set of experiments, based on a well-defined evaluation method, that has been conducted to assess FedDist, as well as three other representative algorithms. Experiments were carried out in the illustrative field of Human Activity Recognition (HAR) on smartphones.

During this section a broad selection of articles was discussed, the intention is to emphasize that this work is focused on the comparison and testing of different FL training techniques using a neural network as implementation model. All of the previous articles were used as inspiration for this thesis work, it is clear that FL is newly discovered area and growing community in which the authors are always contributing to develop and test their results.

3. THEORETHICAL/CONCEPTUAL FRAMEWORK

Summary: *This chapter presents the theoretical and conceptual bases on the used Machine Learning models, training methodologies for Federated Learning and Neural Networks functionality.*

3.1. Machine Learning

Machine learning is a branch of artificial intelligence (AI) and computer science which focuses on the use of data and algorithms to imitate the way that humans learn, gradually improving its accuracy. Arthur Samuel is credited for coining the term, “machine learning” with his research around the game of checkers [32].

Machine learning is an important component of the growing field of data science. Through the use of statistical methods, algorithms are trained to make classifications or predictions, and to uncover key insights in data mining projects. These insights subsequently drive decision making within applications and businesses, ideally impacting key growth metrics. They will be required to help identify the most relevant business questions and the data to answer them.

3.1.1. Machine Learning model training

Training a machine learning (ML) model is a process in which a machine learning algorithm is fed with training data from which it can learn. ML models can be trained to quickly process huge volumes of data, identify patterns, find anomalies, or test correlations that would be difficult for a human to do unaided [33].

Model training is the primary step in machine learning, resulting in a working model that can then be validated, tested, and deployed. The model’s performance during training will eventually determine how well it will work when it is eventually put into an application for the end-users.

Training a model requires a systematic, repeatable process that maximizes the utilization of the available training data and the time of a data science team. Before beginning the training phase, it is necessary to first determine the problem statement, access the dataset and clean the data to be presented to the model [11].

This kind of model training for the ML algorithms is called centralized learning, since all the information resides in the same computer as the training is made.

3.2. Neural Networks

Neural networks are a subset of machine learning and are at the heart of deep learning algorithms. Their name and structure are inspired by the human brain, mimicking the way that biological neurons signal to one another.

Neural Networks (NNs) are comprised of a node layers, containing an input layer, one or more hidden layers, and an output layer. Each node, or artificial neuron, connects to another and has an associated weight and threshold. If the output of any individual node is above the specified threshold value, that node is activated, sending data to the next layer of the network. Otherwise, no data is passed along to the next layer of the network [34].

In the figure below, the outermost yellow layer is the input layer. A neuron is the basic unit of a neural network. They receive input from an external source or other nodes. Each node is connected with another node from the next layer, and each such connection has a particular weight. Weights are assigned to a neuron based on its relative importance against other inputs.

When all the node values from the yellow layer are multiplied (along with their weight) and summarized, it generates a value for the first hidden layer. Based on the summarized value, the blue layer has a predefined "activation" function that determines whether or not this node will be "activated" and how "active" it will be [35]

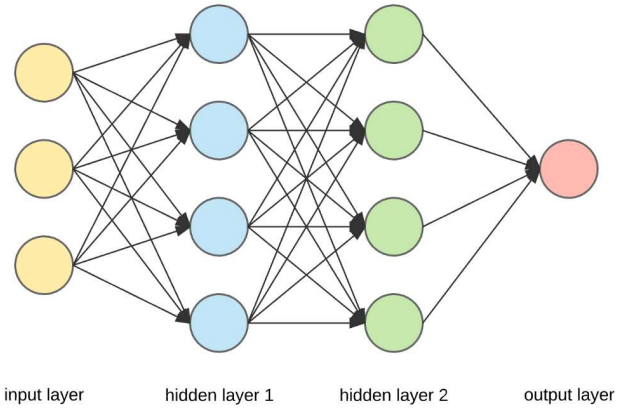

Figure 1 - Components of a Neural Network. Picture from [35]

3.2.1. Neural Networks models weights

Weight is the parameter within a neural network that transforms input data within the network's hidden layers. A neural network is a series of nodes, or neurons. Within each node is a set of inputs, weight, and a bias value. As an input enters the node, it gets multiplied by a weight value and the resulting output is either observed or passed to the next layer in the neural network. Often the weights of a neural network are contained within the hidden layers of the network [13].

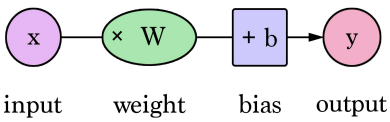

Figure 2 - Neural Network node. Picture from [13]

3.2.2. Neural Networks activation functions

An activation function is a mathematical function which decides if a neuron should be activated or not. This means that it will decide whether the neuron's input information is important in the process of prediction or not. One of the most used mathematical functions used in the NNs are the ReLU and Sigmoid functions.

Sigmoid or also known as Logistic activation function takes any real value as input and outputs values in the range of 0 to 1. The larger the input (more positive), the closer the output value will be to 1, whereas the smaller the input (more negative) will be closer to 0.

ReLU stands for Rectified Linear Unit. The main purpose is that the ReLU function does not activate all the neurons at the same time. The neurons will only be deactivated if the output of the linear transformation is less than 0.

Each layer could be represented as a different mathematical function, resulting in different outputs for the model classification. In a neural network, the activation function is responsible for transforming the summed weighted input from the node into the activation of the node or output for that input. The ReLU has become the default activation function for many types of neural networks because a model that uses it is easier to train and often achieves better performance.

3.2.3. Neural Networks accuracy metrics

In order to evaluate the performance of a NN, it is necessary to compare their predictions versus the actual values. During this work the main performance metric used was the F1 Score. The F1-score combines the precision and recall of a classifier into a single metric by taking their harmonic mean. It is primarily used to compare the performance of two classifiers.

The F1-score of a classification model is calculated as follows:

$$F1\ score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

Equation 1 - F1 Score Formula

Another commonly used accuracy metric is the confusion matrix. A confusion matrix is a table that is used to define the performance of a classification algorithm. he confusion matrix represents the correctly classified True Positive values, False Positive values in the relevant class while it should be in another class, and False Negative values in another class while it should be in the relevant class and the correctly classified True Negative values in the other class. [37].

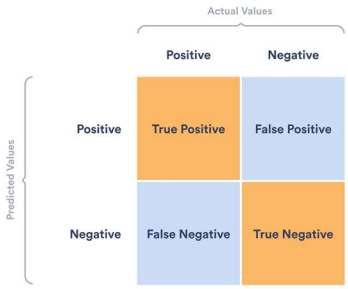

Figure 3 - Confusion Matrix components. Picture from [37]

It is important to mention that there are more metrics to measure the level of assertiveness presented by a classification model. However, the objective of this work was to validate that the performance of the FL training is similar to the conventional methodologies, for this reason, the selected metric needs to be very clear about tendencies of misclassification and percentages of correct classification.

Kolmogorov-Smirnov chart measures performance of classification models. More accurately, K-S is a measure of the degree of separation between the positive and negative distributions. The K-S is 100, if the scores partition the population into two separate groups in which one group contains all the positives and the other all the negatives.

On the other hand, If the model cannot differentiate between positives and negatives, then it is as if the model selects cases randomly from the population. The K-S would be 0. In most classification models the K-S will fall between 0 and 100, and that the higher the value the better the model is at separating the positive from negative cases.

3.3. Federated Learning

Federated Learning is simply the decentralized form of Machine Learning. In ML, usually trains data that was aggregated from several edge devices like mobile phones, laptops, etc. and is brought together to a centralized server. Then grabs this data training itself and finally predicts results for new data generated.

In FL there are two major components: participants and coordinators. The participants are the data owners and can perform local model training and updates. In different scenarios, the participants are made up of different devices, the vehicles on the Internet of Vehicles (IoV), or the smart devices on the Internet of Things (IoT) [10]. In addition, participants usually possess at least two characteristics:

1) Each participant has a certain level of hardware performance, including computation power, communication, and storage.
2) Are independent of one another and located in a wide geographic area.

The coordinators are central servers that orchestrate the actions and transfer algorithms between the participants. They receive information about how many participants are connected and instructs them how many participants to accept, based on which FL tasks are scheduled. It is important to mention that it can be more than one coordinator in FL, meaning that the coordinator can oversee groups of participants. As show in figure 4 FL is an iterative process between participants and coordinators.

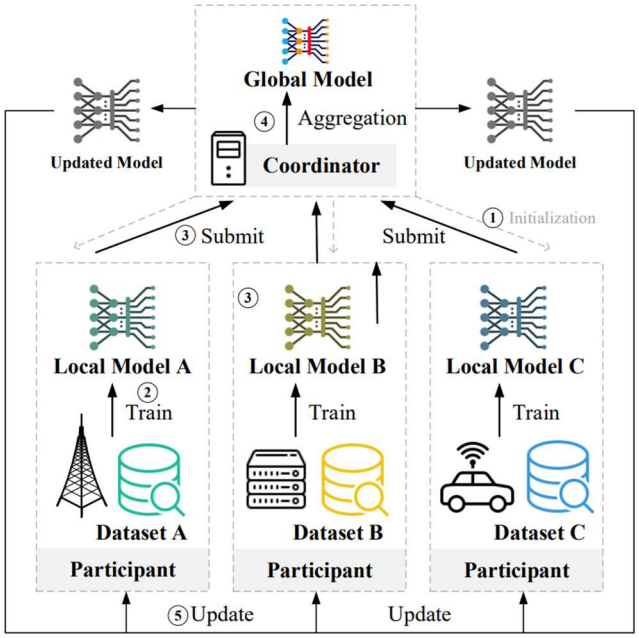

Figure 4 - Federated Learning schema. Picture from [10]

3.3.1. Categorization of Federated Learning

FL can be classified into three categories, namely, horizontal federated learning, vertical federated learning, and federated transfer learning. Each category is used for different applications depending on the context. A FL Horizontal approach was used for this work.

3.3.1.1. Horizontal Federated Learning

Horizontal federated learning, or sample-based federated learning, is introduced in the scenarios that data sets share the same feature space but different in samples. For example, two regional banks may have very different user groups from their respective regions, and the intersection set of their users is very small. However, their business is very similar, so the feature spaces are the same [39].

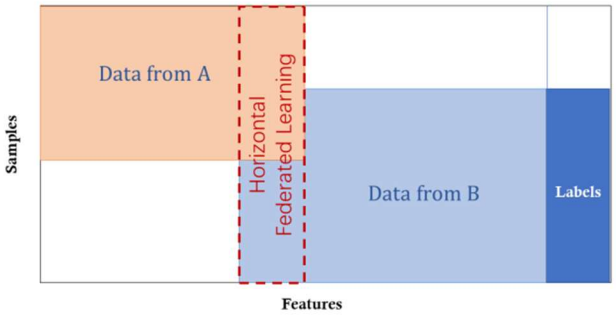

Figure 5 - Horizontal Federated Learning. Image from [39]

3.3.1.2. Vertical Federated Learning

Vertical federated learning or feature-based federated learning is applicable to the cases that two data sets share the same sample ID space but differ in feature space. For example, consider two different companies in the same city, one is a bank, and the other is an e-commerce company. Their user sets are likely to contain most of the residents of the area, so the intersection of their user space is large. However, since the bank records the user's revenue and expenditure behavior and credit rating, and the e-commerce retains the user's browsing and purchasing history, their feature spaces are very different. Suppose that we want both parties to have a prediction model for product purchase based on user and product information [39].

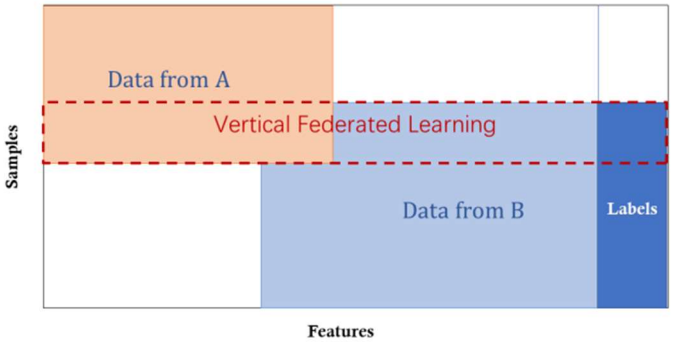

Figure 6 - Vertical Federated Learning. Image from [39]

3.3.1.3. Federated Transfer Learning

Federated Transfer Learning applies to the scenarios that the two data sets differ not only in samples but also in feature space. Consider two institutions, one is a bank located in China, and the other is an e-commerce company located in the United States. Due to geographical restrictions, the user groups of the two institutions have a small intersection. On the other hand, due to the different businesses, only a small portion of the feature space from both parties overlaps [15].

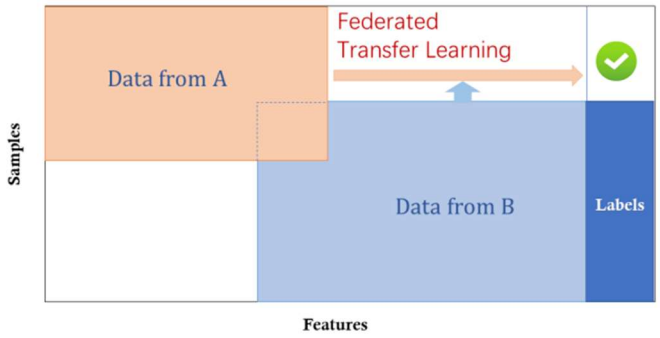

Figure 7 - Federated Transfer Learning. Image from [39]

3.3.2. Federated Learning model training

FL enhances model training by reaching greater amounts of data in distributed locations and on edge devices, at the point of generation and consumption. This approach can provide a significant untapped reservoir of data that greatly expands the available dataset.

This can be particularly useful in situations where so much data is generated at the edge that data transfer to a central location would be prohibitive, such as data that is generated by self-driving vehicles. It is also useful in situations where data privacy is highly regulated, such as the healthcare industry.

In the following sections, some of the most important training methodologies are described, to set a context about the implemented techniques of this work. Also, as part of the scope, a brief introduction to sentiment analysis using human words from reviews is provided.

As it was mentioned before the main goal for a neural network (NN) training is to determine the NN parameters, these parameters are calculated based on data, the following FL model trainings will discuss how it is possible to obtain the NN parameters but without sharing the end user data, only sharing the previous weights and parameters.

3.3.2.1. Federated Average (FedAvg)

Federated average (FedAvg) is a communication efficient algorithm for the distributed training with an enormous number of clients. In FedAvg, clients keep their data locally for privacy protection; a central parameter server is used to communicate between clients. This central server distributes the parameters to each client and collects the updated parameters from clients [11].

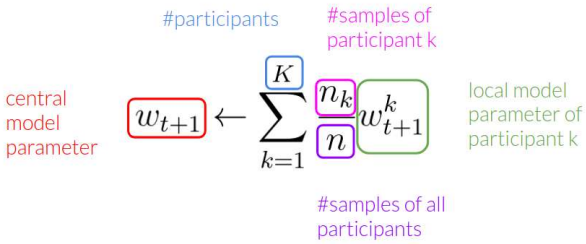

Figure 8 - FedAvg formula. Picture from [11]

The figure summarizes the algorithm behind the FedAvg method, which is a simple average between all the participant client's NN weights. It can be represented as a vector of numbers with K values, those values are averaged and the result is the new weight for the next communication round.

3.3.2.2. Federated Model Distillation (FedMD)

FedMD is an algorithm that enables federated learning for independently designed models. This framework is based on knowledge distillation and is tested to work on various tasks and datasets. It can also be applied to tasks involving NLP and reinforcement learning. It is also possible to extend extreme cases of heterogeneity involving large discrepancies in the amounts of data, in model capacities and very different local tasks. Heterogeneous federated learning will be an essential tool in future in a broad spectrum of business facing applications of deep learning [12]. The main idea of FedMD can be represented in figure 9.

In the figure there is a summary of all the rounds performed during the FedMD training process, showing which dataset are using and the moment where the best evaluated model transfers their weights.

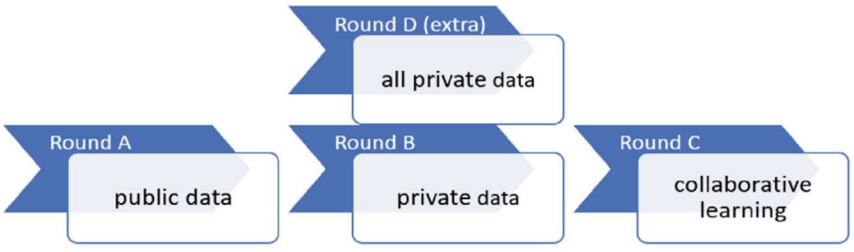

Figure 9 - FedMD Implementation phases

3.3.2.3. Federated Loss Adaptive Boosting (LoAdaBoost)

LoAdaBoost FedAvg it is based on cross entropy loss to adaptively boost the training process on those clients appearing to be weak learners. Since in this study the data labels (Sentiment associated to each IMDB review) were either 0 or 1 (Negative or positive) binary cross-entropy loss was adopted as the error measure of model-fitting and calculated as:

$$-\sum_{i=1}^{N}[y_i \log f(x_i) + (1-y_i)\log(1-f(x_i))]$$

Equation 2 - LoAdaBoost optimization function

Where N is the total number of examples, xi was the input feature vector, y was the prediction label, and f was the federated learning model. The objective function of each client model under LoAdaBoost learning was to minimize equation 2 which measured goodness-of-fit: the lower the loss was, the better a model was fitted [13].

A summary of the iterative process can be observed in the figure below, which basically describes how every client shares their weight then averages it with the initial model and shares it to the next client.

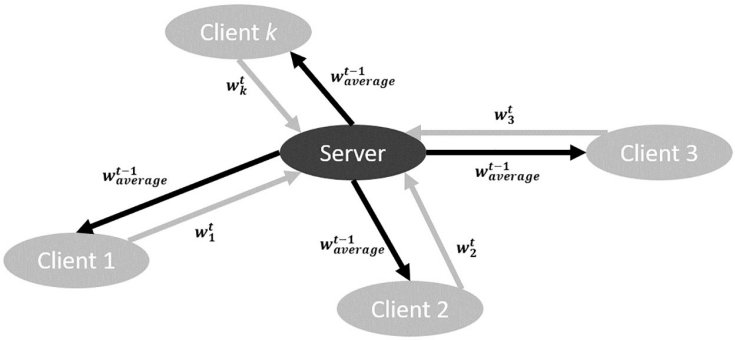

Figure 10 - LoAdaBoost phases. Picture from [13]

3.4. Sentiment Analysis

Sentiment analysis (or opinion mining) is a natural language processing (NLP) technique used to determine whether data is positive, negative, or neutral. Sentiment analysis is often performed on textual data to help businesses monitor brand and product sentiment in customer feedback and understand customer needs.

Sentiment analysis focuses on the polarity of a text (positive, negative, neutral) but it also goes beyond polarity to detect specific feelings and emotions (angry, happy, sad, etc.), urgency (urgent, not urgent) and even intentions (interested v. not interested).

Depending on how you want to interpret customer feedback and queries, you can define and tailor your categories to meet your sentiment analysis needs [14].

4. METHODOLOGY DEVELOPMENT

Summary: *This chapter presents in detail the methodological development that includes the preparation of the used dataset, exploratory analysis, and feature engineering for the implemented Machine Learning model. A summary of the work related to all the implementation of the Federated Learning in the dataset is included as well.*

4.1. Dataset description

The Internet Movie Database (IMDB) is an online database containing information and statistics about movies, TV shows and video games as well as actors, directors, and other film industry professionals. The IMBD data set used for this work contains reviews of movies made by users in English, each review is also categorized as positive or negative (1 or 0 respectively).

This is a dataset for binary sentiment classification contains 40,000 entries of highly polar movie reviews. The original data is available for download to anyone with a Kaggle account via this link. The dataset has 2 columns named as "text" and "label". The label column is our prediction variable. Most of the neural network's algorithms does not support missing values, hence we must take care of any lack of information.

	text	label
0	I grew up (b. 1965) watching and loving the Th...	0
1	When I put this movie in my DVD player, and sa...	0
2	Why do people who do not know what a particula...	0
3	Even though I have great interest in Biblical ...	0
4	Im a die hard Dads Army fan and nothing will e...	1

Figure 11 - First rows of dataset used

4.2. Exploratory Analysis

The primary objective of exploratory data analysis is to uncover the underlying structure and determine the trends, patterns, and relationships among them. Therefore, performing this analysis allows to detect errors, missing values, and the general structure of the data.

The dataset does not have any missing values and it has a balanced distribution of positive (49.95%) and negative reviews (50.05%). Now, after stating that the dataset is clean and balance a more in-depth overview of the column containing the reviews will be performed.

Since this is a text observation, Natural Language Processing (NLP) algorithms will be used to describe the reviews. In the following plot, the most frequent words are punctuation signs and stop words

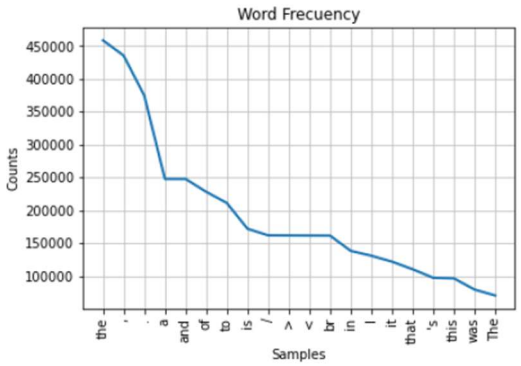

Figure 12 - Histogram of words without preprocessing

The average word count per review is 279 words, including the largest review with more than 2,900 terms. A word cloud plot representation of all the reviews it is also included. This kind of plot allows a better understanding of the written critic's context, it is important to note the presence of contractions and word abbreviations.

count	40000.000000
mean	279.689525
std	207.647439
min	8.000000
25%	151.000000
50%	209.000000
75%	341.000000
max	2911.000000

Figure 13 – Word cloud and basic statistics of original dataset

After reviewing the results, a data preprocessing is needed to eliminate the stop words and punctuation signs. According to NLP methodologies, better results could be obtained if the dataset is without words that does not apport meaning to the context (stop words), punctuation signs and standardizing common contractions.

4.3. Data Preprocessing

As it was previously mentioned in the exploratory analysis, the data is in good conditions and it can be used as a Neural Network input, nevertheless, it is necessary to perform a complete NLP process to eliminate stop words, contractions, and punctuation signs. This will help to minimize complexity of the calculations and improve overall performance of the neural network.

Stop words are those words that do not provide any useful information to decide in which category a text should be classified. This may be either because they do not have any meaning (prepositions, conjunctions) or because they are too frequent in the classification context. Starting by replacing contractions, eliminating punctuation signs, and lowercasing all the words and then proceed to eliminate all the stop words of each review. After all the NLP is executed, a word histogram plot and word cloud are presented once more.

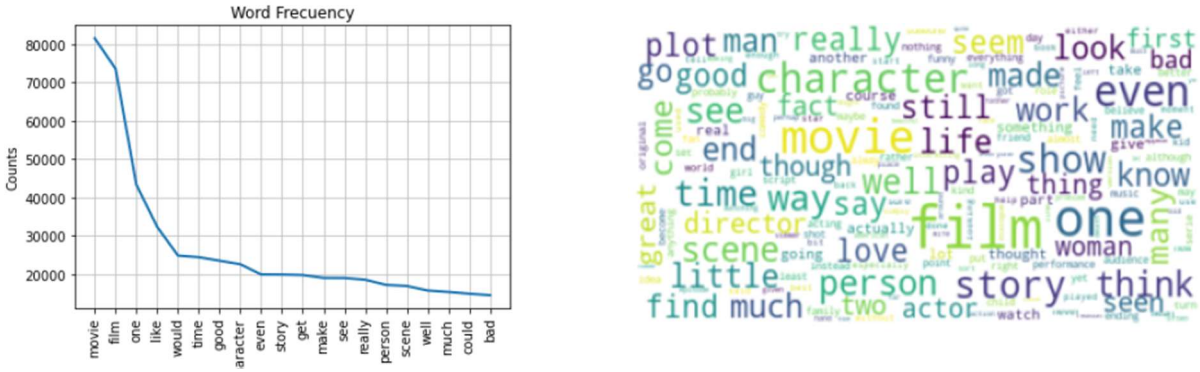

Figure 14 - Histogram and word cloud after preprocessing

With the previous output, it is evident a cleaner database with fewer words in the reviews. After the punctuation signs and stop words removal, the review still maintains the same context and message. After the replacement, the review average of 118 is words and the larger review is 1,425 terms.

count	40000.000000
mean	118.236750
std	89.012316
min	3.000000
25%	63.000000
50%	88.000000
75%	144.000000
max	1425.000000

Figure 15 - Basic statistics of the dataset after preprocessing

4.4. Descriptive Analysis

Descriptive statistics is a means of describing features of a data set by generating summaries about data samples. It's often depicted as a summary of data shown that explains the contents of data. After the cleaning process, the reviews are without any unnecessary word, it is important to understand each type of review (positive and negative) and describe the distribution of their verbiage. All the NLP was used with the NLTK python library.

In the case of subletting only the positive reviews, the average words per review is 120 with a maximum review length of 1,425 words. Here is a histogram of word frequency and their respectively word cloud.

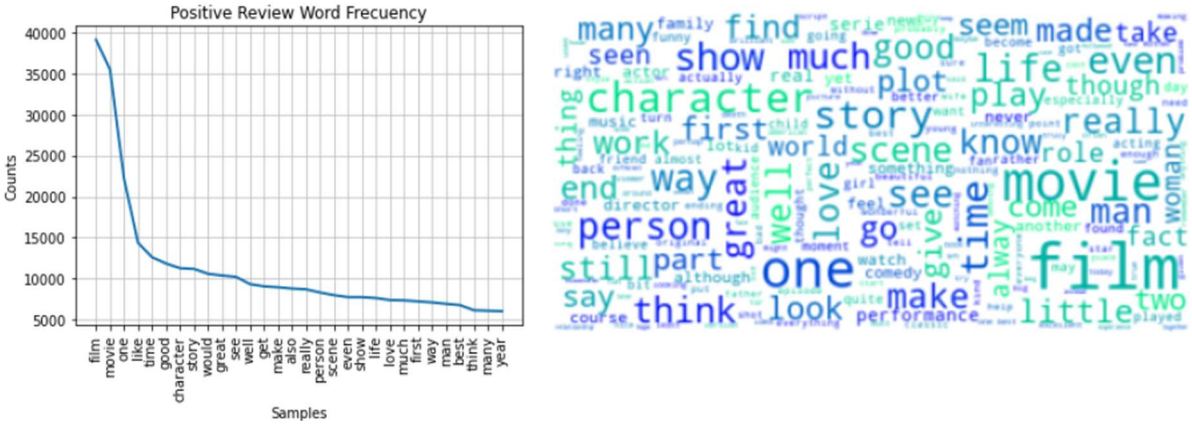

Figure 16 - Histogram and word cloud of positive reviews

Regarding the reviews categorized as negative, the average words per review is 116 with a maximum review length of 806 words. Here is a histogram of word frequency and their respectively word cloud.

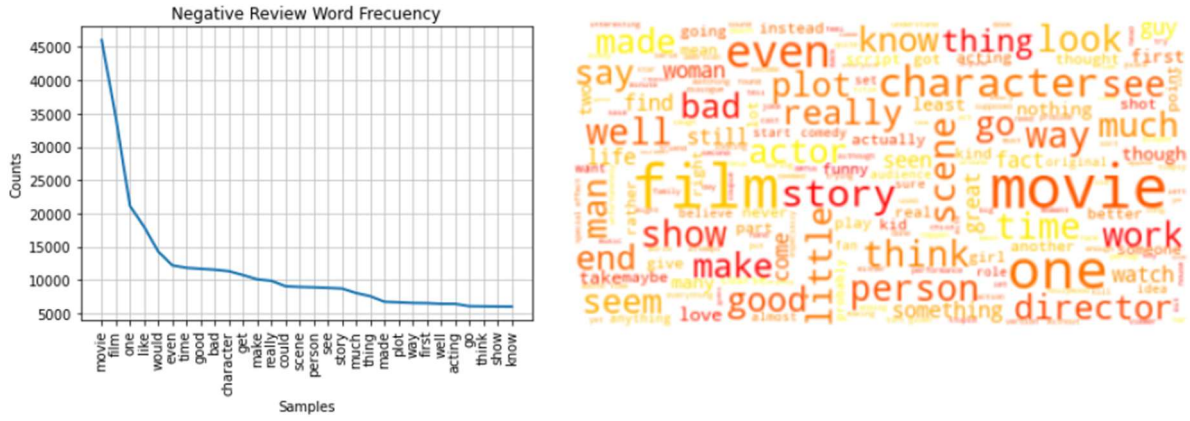

Figure 17 - Histogram and word cloud of negative reviews

After the representation of both types of reviews, the positive reviews show words commonly associated with positive feedback, such as "great", "good", "much", "love". Nevertheless, in the case of negative feedback, positive words are also present, which can indicate comparison to examples of what the user thinks is a good movie, but no many negative words.

4.5. Feature Engineering

Since the main objective of this work is to compare the performance of a federated learning algorithm versus a standard centralized neural network, it is necessary to develop a traditional neural network and measure their performance under controlled conditions. First, split the dataset into training and testing parts. Considering the nature of the data, separate into 80% train and 20% test. Also, it is important to keep the original positive/negative reviews ratio in both sets, training and testing.

Each input of a neural network needs to be numerical with a fixed dimension. For the purpose of this work, and since it is performed a sentiment analysis, each review will be a numerical matrix. The criteria to convert the text to a numerical matrix for each input of the neural network is the following:

- Every review will be represented as a row of a matrix with fixed number of columns.
- Each word will be numerically ranked according to their frequency into the whole review's dataset. It was previously detected that the data contains 111,019 different words.
- Based on the average review length, only 110,000 unique words will be considered.

- Each word will be ranked from 1 (more frequent) to 110,000 (less frequent).
- All the words that are not considered under the 110,000 more frequent will be represented as a “OOV” token (Out Of Vocabulary).
- Since the reviews have an average of 118 words and 75% of the reviews are under 144 words, the maximum review length will be 160 words. All the reviews that are over 160 words will be truncated.
- The embedding (how many dimensions represent a word) will be a vector of 16 dimensions. In the case that a review contains less than 160 words, the remaining spaces will be filled with zeros after the review ends.

The main neural network design will be performed using the python library Keras. As a first step it is necessary to tokenize all the sentences using the Keras tokenizer, get a dictionary of word frequency and a list showing the index of each word according to their frequency in the whole review's vocabulary. After the conversion from text to numerical values, a padding will be executed, truncating each review to our maximum length previously defined. Finally, here is a visualization of one review as an input for the neural network.

```
[    0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0   138    24   486   632  5208     9   383  1737
  3661   715   822   364    13    25   180   219 27110  4876   129 39374
    14    19   238    15   818    25   180   219   438   148    33   606
    34   252   114  4137   290   330   114   474   839  3240 39375  1282
   155    39   188     9  2553  5980   155  1970     6  1101   608    30
 10207 39376   179  1389  1737   759   207 16373   467   976     6  1101
   608   140   185  1737]
```

Figure 18 - Example of the Neural Network input

The image shows a vector of our maximum length which represents each word of the review. Every word of the review was changed to a number, representing the frequency of that word in our Vocabulary. The zeros represent the blank spaces.

4.6. Neural Network definition

Once the feature engineering is done, the next step is designing the layers of the neural network. With the previously defined values of dimensions, an embedding layer of 110,000 words, 16 dimensions and an input length of 160 words will be created. A 2-layer neural network with 6 neurons in the hidden layer is selected for this implementation using an activation function of ReLu and Sigmoid, respectively. No more hyperparameters of the neural network were adjusted.

Once the input parameters of the model are complete, here are the results of plotting the progress of the accuracy and loss functions in every epoch.

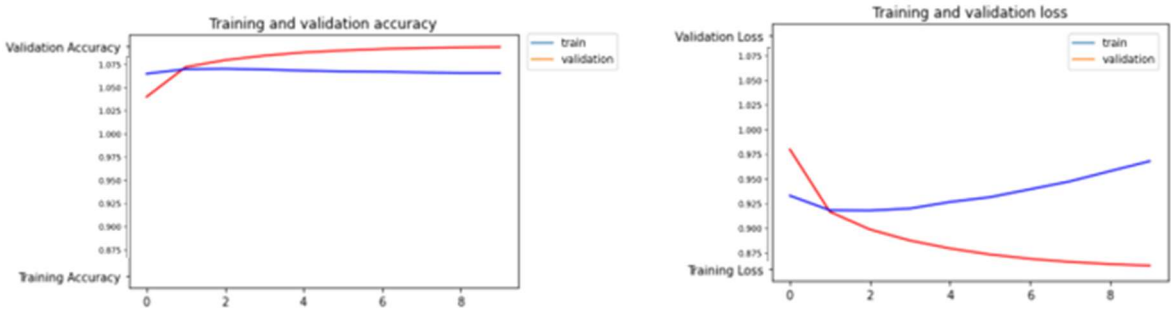

Figure 19 - Neural Network metrics

Due the nature of the analysis, it is important to visualize how the neural network is classifying the reviews and the type of predictions that it is presenting. For that reason, it is important the implementation of a confusion matrix and a F1 Score as a visual representation for summarizing the performance of the classification algorithm.

4.7. Federated Model Training

As it was previously discussed, the objective of this work is to simulate multiple federated learning training techniques and compare them with the recently developed Neural Network. For the construction of the federated learning algorithm, it is necessary to simulate a dataset from a client, for that reason, a function who randomly selects reviews from all the datasets and assign it to a client label as a python dictionary will be created.

To compare the federated learning algorithm under the same conditions, identical training and testing data as the centralized model will be used. The main difference is that this time the data will be split into 10 clients, simulating separated and private data. Also, one more client will be created which will serve as the initial benchmark model.

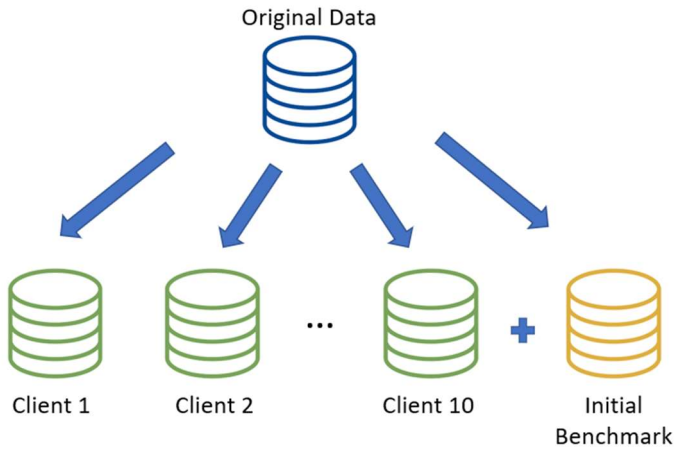

Figure 20 - Federated Learning structure setup

All done up to this point was pretty much standard ML pipeline. Of course, except for the data partitioning or client creation. Now it is time to define the federated communication process. In order to calculate the proportion of a client's local training data with the overall training data held by all clients, it is necessary to create a function who weights the participation of each client according to an arbitrary value (it could be the same for each client). Each client will be expected to indicate the number of data points they trained with while updating the server with new parameters after each local training step.

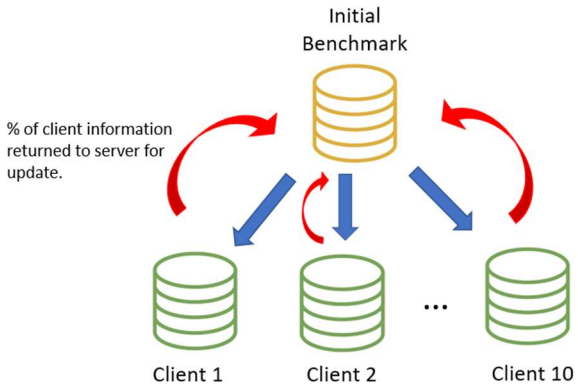

Figure 21 - Federated Learning setup communication

Now, it is time to design a function that will scale the local model weight with the weight scaling factor obtained previously. The input of this function will be the client model weights obtained after the training process. To complete the average weighting process, it is necessary to combine all the client data and sum each weight accordingly.

4.7.1. Federated Average Model (FedAvg)

Federated averaging is a communication efficient algorithm for the distributed training where each client communicates the resulting weights of a locally trained model and submits an average of all the client's weights. With the new averaged weights, the central model is updated.

An example of this communication of each customer could be observed in the image below:

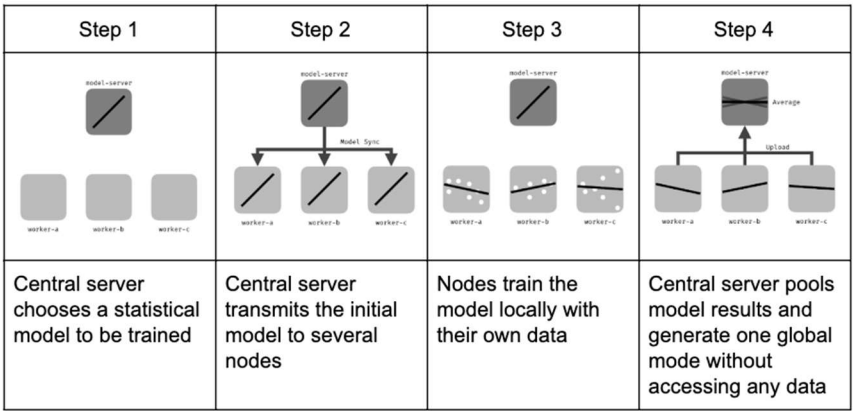

Figure 22 - FedAvg Implementation steps. Picture from [56]

First, as it is stated in the chapter 3, for the federated averaging, an initial model is required, this model will help us as a benchmark, will be passed to each client. Then each client executes their own NN, it is important to mention that all the neural networks across clients and benchmark are the same (number of inputs, layers, etc.) the only difference is the dataset used to determine the NN weights.

After all the clients executed their NN, only the weights are saved in a variable, which represents the process of sharing the information to the central server. Once all the weights information is safely stored, a common average of each client is made, resulting in the new set of weights which will be passed to the client for recalculation. In total, a set of 8 epochs of this process were executed.

4.7.2. Federated Learning Model Distillation (FedMD)

Now have a different approach of the Fed Avg model. The idea is to test 2 models, one using a public dataset (same for all the clients) and other using their own private dataset. After each client trains their model using private data locally, test the model with the public dataset communicating the results to the central server. Once all the clients have submitted their results, only the weights that meet predefined requirements will be selected, in this case, the average of the worst F1 score with the public initial model. The reason behind this idea is to ensure a more prepared model using new data that the centralized model is not able to predict.

The implementation could be summarized in the following 4 iterative phases:

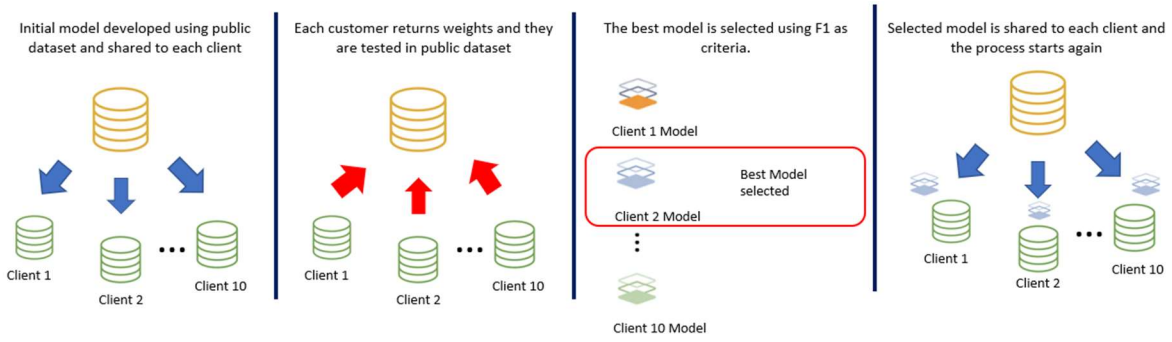

Figure 23 - FedMD Implementation steps

4.7.3. Federated LoAdaBoost (FedLoAdaBoost)

As we previously mentioned, the sharing weights between the client and the server plays a major role during the FL training process. Another approximation is to train each client and retrieve weights to the server, but every communication round, the weights will be updated for the next client. This implementation consists of share the weights of the first costumer with the next one and that customer to the next in line. In this case, the weights will be communicated to the server and the server will decide the weights of the model to train locally. At the end, the last client will get weights optimized with all the previous clients.

The implementation could be summarized in the following 4 iterative phases:

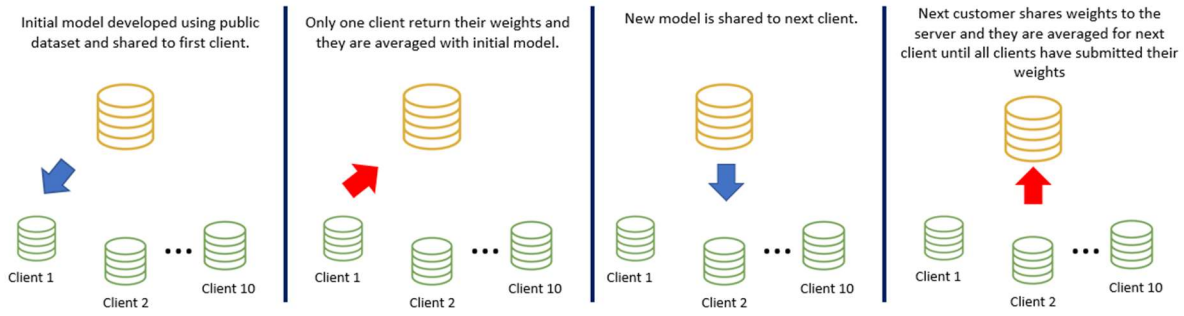

Figure 24 - FedLoAdaBoost Implementation steps

5. RESULTS AND DISCUSSION

Summary: *This chapter presents the results obtained from the development of this work, discussion of the implemented Federated Learning training methodologies and their comparison versus traditional Machine Learning training.*

5.1. Results

As previously mentioned, sentiment analysis is a process in which the response to a review or comment is categorized by a human feeling; there can be many categories (anger, sadness, happiness, etc.); however, in this work we chose to classify reviews in a binary manner, indicating whether they are positive or negative. To follow up on these responses and qualify the predictive capacity of the models used, the confusion matrix was used as a way to summarize on the results and the F1 Score as an indicator of model success.

Centralized Model:
It was the model used as a benchmark and to have a frame of reference regarding the new training systems. It presented an F1 Score of 0.887 with no marked tendency to one type of error (False-Negative or True-Negative).

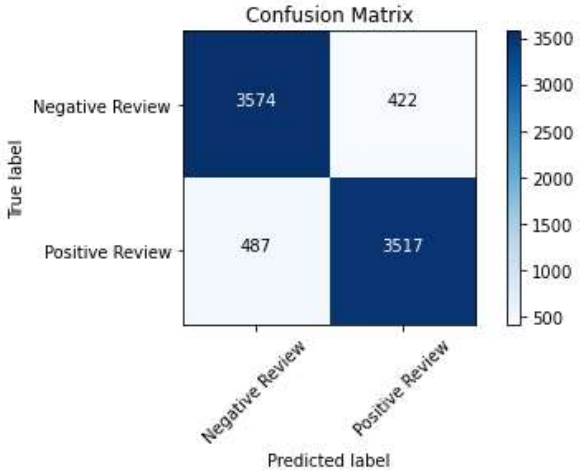

Figure 25 - Confusion Matrix Centralized Model

FedAvg Model:
According to the literature, it is the most used model within the FL world, since its operations are simple, and it does not require considerable loads of communication between client and server. It presented an F1 Score of 0.836 with a tendency to cause False-Positive error.

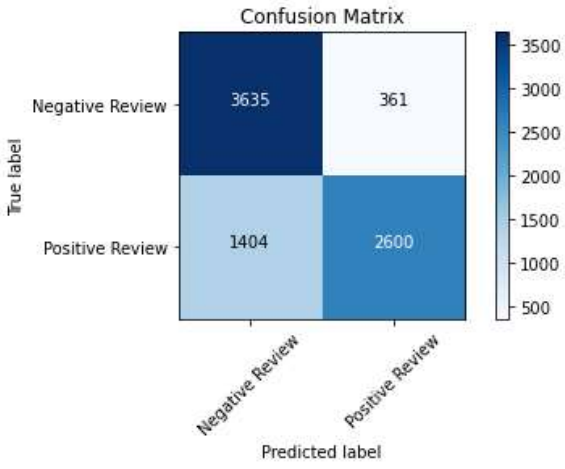

Figure 26 - Confusion Matrix FedAvg Model

FedMD Model

One of the advantages of this model is that it can learn from very diverse situations where data can be very scarce since it allows working with less information without sacrificing robustness. It presented an F1 Score of 0.6903 with a strong tendency towards False-Positive errors.

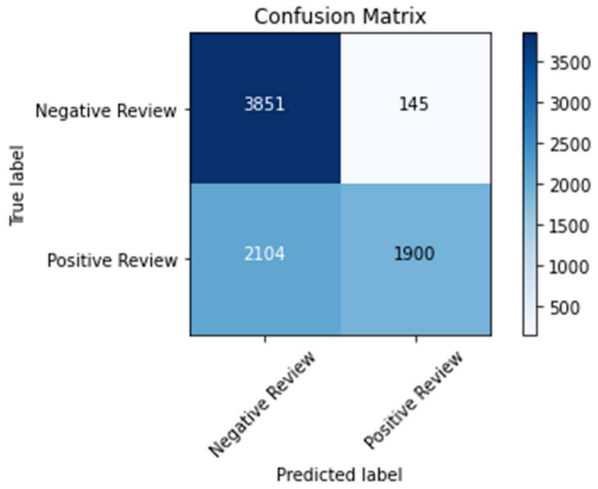

Figure 27 - Confusion Matrix FedMD Model

FedLoADABoost Model

According to the literature, this model allows a better performance when there is a balanced database and the amount of data is very diverse, since it tends to cause overfitting if all the data are similar. It presented an F1 Score of 0.891, surpassing even that obtained by a neural network trained in a conventional way, it is worth mentioning that it does not tend to make any type of error either.

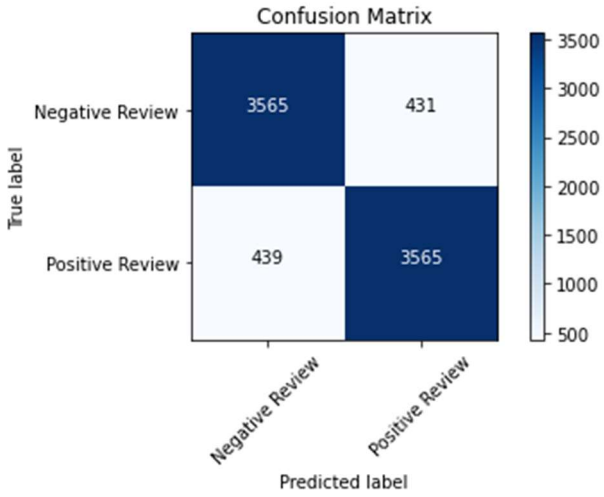

Figure 28 - Confusion Matrix FedLoADABoost Model

5.2. Discussion

As part of the results obtained, it can be observed that the models trained by federated learning offer a fairly affordable alternative to make predictions in the context of sentiment analysis, it should be noted that neural network models focused on sentiment analysis are among the most complex that exist, in this iteration, all models presented similar results, however, the FedLoADABoost model was the one that obtained a better performance both in F1 and in terms of balanced results observed through its confusion matrix.

Training Method	F1-Score
Centralized	0.887
FedAvg	0.836
FedMD	0.693
FedLoADABoost	0.891

Table 1 - F1 Scores obtained

An important issue to mention is that the implemented work was performed using iid data, however, this situation is not so common during real applications. One of the main areas of FL research is dealing with non-iid data, since this type of data has proven to have greater difficulty in the FL process.

It should be noted that all the results and assumptions made were considering a dataset of reviews with balanced responses (positive and negative). It is proposed as future work to analyze the effects of having unbalanced datasets and how it affects the performance of the training methods, since this may result into an overfitting towards the most predominant class.

Another point to discuss is the misuse that can be given to this communication system, although it does not directly store or treat customer information, it is necessary to make a connection to the system, so it is possible to be a victim of a cyber-attack. It could be executed as a parameter data of the model, which leaves exposed to the place where the customer information resides.

FL promises to be a competitive advantage with great utility for users with sensitive information, however, it is open to discussion how the intellectual owner of the algorithm can continue to have control over the intellectual property, since the product is constantly sent to devices out of reach of the administrator, resulting in possible theft of intellectual property.

6. CONCLUSSIONS

Summary: *This chapter presents the conclusions and future work related to schemes based on Federated Learning for decentralized training in Machine Learning.*

6.1. *Conclusions*

During this work, three types of training using FL (FedAvg, LoAdaBoost and FedMD) were analyzed, using a 2-layer neural network with 6 neurons in the hidden layer as a benchmark model. The objective was to test if the above-mentioned methods were similar in performance compared against centralized training methods.

In order to set the experiment environment, as a main evaluation metric the F1 Score was used, after the implementation of all the FL models can conclude that the performance of all the models trained with FL were similar compared to the benchmark model. Some aspects to consider is that the FedMD scored 20.6% lower than the benchmark model, concluding that this is the worst performing, on the other hand, it was noticeable that the FedLoAdaBoost model even exceeded the benchmark model by 0.4%, leading to a higher performance with the benefits of a FL environment that were previously discussed.

The proposed FL training methodologies used for this work proved to be effective, which assent the results obtained by similar works of applied FL with sentiment analysis, concluding that it is a viable option to use the proposed FL training methods as an alternative. It is worth noting that the used dataset of movie reviews from the IMBD, could be easily compared to highly sensitive information from a company (Customer reviews, workers complaints, doctor diagnostics, etc.) which indeed could be analyzed to obtain important highlights, but the information management could be complicated to share.

The main difference between the analyzed FL training methodologies was the way of sharing and adjusting the neural network weights, after the implementation, concluding that the simple averaging is a good methodology to include information from multiple clients, but the best results were obtained by the LoAda Boost methodology, leading to the conclusion that sharing weights information and averaging it more often leads to better performance in the model.

It is also worth noting that the FL code implementation and mathematical background could be easily proved and executed, however, since on the main key points of FL are the security and privacy of the information, it is necessary to prioritize the data security and communication of the model weights, similar to what the actual FL libraries are focusing.

6.2. *Future Work*

FL is a relatively new branch of Machine Learning, and it is constantly growing at unprecedented rates. The conclusion of this work is aligned with the theory previously discussed and explores the basic functionality of FL under the sentiment analysis context. As a new investigation line would be important to analyze if the FL methodology is optimal in another context such as Time Series or Regression Models.

During this implementation, communications between the server and clients were shown to be quite efficient, since they were simulated on a local computer (where lag is virtually nonexistent) however, the communication and response time factor plays a very important role in a real environment, there is extensive literature to minimize the effects of it. It is recommended to consider for upcoming projects

As future work, it would be worth exploring a comparison under the same FL methods but now changing the neural network architecture, changing the number of layers, number of neurons, and activation function, to detect if the FL training techniques deliver consistent performance even when the model presents different variations. Also, it would be interesting to perform analogies of models but now using specialized FL libraries, Tensor Flow Federated, for example.

BIBLIOGRAPHY

[1] C.-E. L. o. D. N. f. D. Data, "H. Brendan McMahan," *International Conference on Artificial Intelligence and Statistics (AISTATS),* 2016.

[2] L. Huang, "LoAdaBoost: loss-based AdaBoost federated machine learning with reduced computational complexity on IID and non-IID intensive care data," 2020.

[3] P. Tsankova, "Sentiment detection with FedMD: Federated Learning via Model Distillation," *Varna Free University Chernorizets Hrabar,* 2020.

[4] S. EK, "A Federated Learning Aggregation Algorithm for Pervasive Computing: Evaluation and Comparison," *Pervasive Computing and Communications (PerCom 2021),* 2021.

[5] INAI, "Regulation of the federal law on the protection of personal data in the possesion of individuals," *US Goverment Translation,* 2011.

[6] Official Journal of the European Union, "General Data Protection Regulation," pp. 32, 83, 2016.

[7] K. S. S. W. F. G. Y. G. Nguyen Truong, "Privacy Preservation in Federated Learning: An insightful survey from the GDPR Perspective," *Elsevier,* p. 1, 2011.

[8] Y. Guo, "Privacy preservation in federated learning: An insightful survey from the GDPR perspective," *TC 11 Briefing Papers,* p. 1, 2021.

[9] R. O. Ogundokun, "A Review on Federated Learning and Machine Learning Approaches: Categorization, Application Areas, and Blockchain Technology," *Information,* p. 1, 2022.

[10] Q. Li and Z. Wen, "A survey on federated learning systems: Vision, hype and reality for data privacy and protection," *IEEE Trans. Knowl. Data Eng,* pp. 1-20, 2021.

[11] R. M. T. O. F. B. Mingqing Chen, " Federated learning of out-of-vocabulary words," *ArXiv,* 2019.

[12] A. Hard, K. Rao, R. Mathews, S. Ramaswamy, F. Beaufays, S. Augenstein, H. Eichner, C. Kiddon and D. Ramage, "Federated learning for mobile keyboard prediction," *arXiv,* 2018.

[13] T. Yang, G. Andrew, H. Eichner, H. Sun, W. Li, N. Kong, D. Ramage and F. Beaufays, " Applied federated learning: Improving google keyboard query suggestions," *arXiv,* 2018.

[14] S. Ramaswamy, R. Mathews, K. Rao and F. Beaufays, " Federated learning for emoji prediction in a mobile key-board," *arXiv,* 2019.

[15] X. Wang, Y. Han, C. Wang, Q. Zhao, X. Chen and M. Chen, "In-Edge AI: Intelligentizing Mobile Edge Computing, Caching and Communication by Federated Learning," *IEEE Netw,* 2019.

[16] Y. Qian, L. Hu, J. Chen, X. Guan, M. Hassan and A. Alelaiwi, "Privacy-aware service placement for mobile edge computing via federated learning," *Inf Sci,* 2019.

[17] G. Szegedi, P. Kiss and T. Horváth, "Evolutionary Federated Learning on EEG-data," *ITAT,* pp. 71-78, 2019.

[18] R. Antunes, "Federated Learning for Healthcare: Systematic Review and Architecture Proposal," *ACM Trans. Intell. Syst. Technol,* pp. 1-13, 2022.

[19] T. S. Brisimi, "Federated learning of predictive models from federated electronic health records," *J. Med. Inform,* pp. 59-62, 2018.

[20] X. Li, "On the Convergence of FedAvg on Non-IID Data," *Arxiv,* pp. 1-2, 2019.

[21] D. Liu, D. Dligach and T. Miller, "Two-stage Federated Phenotyping and Patient Representation Learning," *Proc. Conf. Assoc. Comput Linguist,* 2019.

[22] X. Han, H. Yu and H. Gu, "Visual Inspection with Federated Learning," in *International Conference on Image Analysis and Recognition*, Cham, Switzerland, 2019.

[23] B. Liu, L. Wang, M. Liu and C.-Z. Xu, "Federated Imitation Learning: A Novel Framework for Cloud Robotic Systems With Heterogeneous Sensor Data," *IEEE Robot Autom Lett,* 2020.

[24] F. J. OrestesAppel, "A hybrid approach to the sentiment analysis problem at the sentence level," *Knowledge-Based Systems,* 2016.

[25] D. W. J. Li, "FedMD: Heterogenous Federated Learning via Model Distillation," *arVix,* 2019.

[26] K. J. G. D. A. R. a. A. T. Lisha Li, "Hyperband: A novel bandit-based approach to hyperparameter optimization," *The Journal of Machine Learning Research,* 2017.

[27] D. C. R. T. P. K. S.-H. L. R. M. P. Steven R Young, "Optimizing deep learning hyper-parameters through an evolutionary algorithm," *Workshop on Machine Learning in High-Performance Computing Environments,* 2015.

[28] S. T.-P. P. I. F. a. A. G. W. Jian Wu, "Practical multi-fidelity bayesian optimization for hyperparameter tuning," *Uncertainty in Artificial Intelligence,* 2020.

[29] M. Al-Shedivat, J. Gillenwater and E. Xing, "FEDERATED LEARNING VIA POSTERIOR AVERAGING: A NEW PERSPECTIVE AND PRACTICAL ALGORITHMS," *ICLR,* 2021.

[30] B. McMahan, E. Moore, D. Ramage and S. Hampson, "Communication-efficient learning of deep networks from decentralized data. In Artificial Intelligence," *PMLR,* p. 1273–1282, 2017.

[31] S. Ek, F. Portet, P. Lalanda and G. Vega, "A Federated Learning Aggregation Algorithm for Pervasive Computing: Evaluation and Comparison," *HAL archives-ouvertes,* 2021.

[32] A. L. Samuel, "Some studies in machine learning using the game of checkers," p. 1, 1959.

[33] D. Weedmark, "Machine Learning Model Training: What It Is and Why It's Important," DOMINO, November 2021. [Online]. Available: https://www.dominodatalab.com/blog/what-is-machine-learning-model-training#:~:text=Training%20a%20machine%20learning%20(ML,from%20which%20it%20can%20learn.. [Accessed September 2022].

[34] IBM Cloud Education, "Neural Networks," IBM Cloud Education, August 2020. [Online]. Available: https://www.ibm.com/cloud/learn/neural-networks. [Accessed September 2022].

[35] upGrad, "Neural Network: Architecture, Components & Top Algorithms," upGrad, 22 September 2022. [Online]. Available: https://www.upgrad.com/blog/neural-network-architecture-components-algorithms/. [Accessed 1 November 2022].

[36] J. Alammar, "A Visual and Interactive Guide to the Basics of Neural Networks," The Illustrated Transformer, 2018. [Online]. Available: http://jalammar.github.io/visual-interactive-guide-basics-neural-networks/. [Accessed September 2022].

[37] F. Demir, "Deep autoencoder-based automated brain tumor detection from MRI data," *Artificial Intelligence-Based Brain-Computer Interface,* 2022.

[38] J. Qi, "Federated Reinforcement Learning: Techniques, Applications, and Open Challenges," *Arvix,* 2021.

[39] Q. YANG, "Federated Machine Learning: Concept and Applications," *CCS Concepts,* 2019.

[40] T. Sun, "Decentralized Federated Averaging," *JOURNAL OF LATEX CLASS FILES,* vol. 14, p. 1, 2015.

[41] D. Li, "FedMD: Heterogenous Federated Learning via Model Distillation," *Center for Fundamental Laws of Nature,* p. 5, 2019.

[42] L. Huang, "LoAdaBoost: Loss-based AdaBoost federated machine learning with reduced computational complexity on IID and non-IID intensive care data," *PLOS ONE,* p. 5, 2020.

[43] "Sentiment Analysis: A Definitive Guide," Monkey Learn, [Online]. Available: https://monkeylearn.com/sentiment-analysis/.

[44] R. O. Ogundokun, "A Review on Federated Learning and Machine Learning Approaches: Categorization, Application Areas, andBlockchain Technology," *MDPI Open Access Journals,* p. 1, 2022.

[45] J. Qi, "Federated Reinforcement Learning: Techniques Applications, and Open Challenges," *Arvix,* 2021.

[46] Z. Du, "Federated Learning for Vehicular Internet of Things: Recent Advances and Open Issues," *IEEE Computer Graphics and Applications,* p. 3, 2020.

[47] Q. Yang, "Federated machine learning: Concept and applications," *ACM Trans. Intell. Syst,* pp. 1-19, 2019.

[48] P. Kairouz, "Advances and Open Problems in Federated Learning," *arXiv,* 2019.

[49] Q. Li, "A survey on federated learning systems: Vision, hype and reality for data privacy and protection," *IEEE Trans. Knowl. Data Eng,* pp. 1-20, 2021.

[50] D. Nguyen, " Federated Learning Meets Blockchain in Edge Computing: Opportunities and Challenges," *IEEE Internet Things,* p. 8, 2021.

[51] V. Mothukuri, " A survey on security and privacy of federated learning," *Futur. Gener. Comput. Syst,* pp. 619-640, 2020.

[52] M. Ali, "Integration of blockchain and federated learning for Internet of Things: Recent advances and future challenges.," *Comput. Secur.,* p. 108, 2021.

[53] H. Lee, "Trends in blockchain and federated learning for data sharing in distributed platforms," in *International Conference on Ubiquitous and Future Networks (ICUFN)*, Barcelona, Spain, 2021.

[54] L. Khan, "Federated learning for internet of things: Recent advances, taxonomy, and open challenges," *IEEE Commun. Surv. Tutor,* p. 23, 2021.

[55] L. Li, "A survey on federated learning," in *IEEE 16th International Conference on Control & Automation (ICCA)*, Hokkaido, 2020.

[56] Wikipedia, "Federated learning," 2022. [Online]. Available: https://en.wikipedia.org/wiki/Federated_learning#:~:text=Federated%20averaging%20(FedAvg)%20is%20a,weights%20rather%20than%20the%20gradients.. [Accessed 9 November 2022].