TU WIEN Informatics

Property Inference Attacks with Fully-Connected Neural Networks in a multi-class setting

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieurin

im Rahmen des Studiums

Wirtschaftsinformatik

eingereicht von

Denise Berthold, BSc.
Matrikelnummer 01626568

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Andreas Rauber
Mitwirkung: Univ.Lektor Mag.rer.soc.oec. Dipl.-Ing. Rudolf Mayer

Wien, 31. März 2025

Denise Berthold	Andreas Rauber

TU WIEN Informatics

Property Inference Attacks with Fully-Connected Neural Networks in a multi-class setting

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieurin

in

Business Informatics

by

Denise Berthold, BSc.
Registration Number 01626568

to the Faculty of Informatics

at the TU Wien

Advisor: Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Andreas Rauber
Assistance: Univ.Lektor Mag.rer.soc.oec. Dipl.-Ing. Rudolf Mayer

Vienna, March 31, 2025

Denise Berthold	Andreas Rauber

Erklärung zur Verfassung der Arbeit

Denise Berthold, BSc.

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.
Ich erkläre weiters, dass ich mich generativer KI-Tools lediglich als Hilfsmittel bedient habe und in der vorliegenden Arbeit mein gestalterischer Einfluss überwiegt. Im Anhang „Übersicht verwendeter Hilfsmittel“ habe ich alle generativen KI-Tools gelistet, die verwendet wurden, und angegeben, wo und wie sie verwendet wurden. Für Textpassagen, die ohne substantielle Änderungen übernommen wurden, haben ich jeweils die von mir formulierten Eingaben (Prompts) und die verwendete IT- Anwendung mit ihrem Produktnamen und Versionsnummer/Datum angegeben.

Wien, 31. März 2025

Denise Berthold

Danksagung

An dieser Stelle möchte ich meiner Familie meinen tiefsten Dank aussprechen. Meinen Eltern, die mich stets unterstützt und mir den Rücken gestärkt haben – ohne ihre Hilfe und Ermutigung wäre dieser Weg nicht möglich gewesen. Meiner Schwester, die immer ein offenes Ohr für mich hatte und mich mit ihrer positiven Art motiviert hat. Und meinem Freund, der mir in stressigen Zeiten zur Seite stand und mich immer wieder daran erinnert hat, an mich selbst zu glauben. Eure Unterstützung bedeutet mir unendlich viel – diese Arbeit ist auch euer Verdienst.

Acknowledgements

At this point, I would like to express my deepest gratitude to my family. To my parents, who have always supported me and had my back – without their help and encouragement, this journey would not have been possible. To my sister, who was always there to listen and motivated me with her positive attitude. And to my boyfriend, who stood by my side during stressful times and constantly reminded me to believe in myself. Your support means the world to me – this work is also your achievement.

Kurzfassung

Maschinelle Lernmodelle (ML) werden zunehmend in verschiedenen Bereichen eingesetzt und erfordern oft große, repräsentative Datensätze für eine effektive Erstellung. Die Weitergabe oder Kommerzialisierung vortrainierter Modelle birgt jedoch Datenschutzrisiken, insbesondere durch sogenannte "Property Inference Attacks". Diese Angriffe zielen darauf ab, globale Merkmale des Trainingsdatensatzes zu extrahieren, ohne auf einzelne Trainingsinstanzen zugreifen zu müssen. Während sich bisherige Forschung weitgehend auf binäre Property Inference Angriffe konzentriert, erweitert diese Arbeit den Angriff auf einen Multi-Klassen-Ansatz.

Diese Arbeit untersucht die Anfälligkeit von fully-connected, feed-forward neuronalen Netzwerken gegenüber Multi-Klassen-Property-Inference-Angriffen. Dabei wird ein meta-maschinelles Lernmodell, das ursprünglich für binäre Angriffe entwickelt wurde, auf den Multi-Klassen-Fall angepasst. Zudem wird überprüft, ob eine hierarchische Klassifikationsstrategie die Angriffseffektivität verbessern kann. Darüber hinaus wird das Risiko von Property Inference Angriffe im Kontext des sequentiellen föderierten Lernens analysiert, in dem mehrere Teilnehmer gemeinsam ein Modell trainieren, ohne ihre Daten direkt zu teilen.

Die Experimente untersuchen die Effektivität von Multi-Klassen-Property-Inference-Angriffen in Abhängigkeit von der Klassengranularität und verschiedenen Loss-Funktionen. Die Ergebnisse verdeutlichen die Zusammenhänge zwischen Vorhersagegenauigkeit und Granularität und unterstreichen die Notwendigkeit robuster Datenschutzmaßnahmen, insbesondere im föderierten Lernen, um Property Inference Attacks zu verhindern und sensible Daten zu schützen.

Abstract

Machine learning (ML) models are increasingly used in various domains, often requiring large, representative datasets for effective training. Consequently, pre-trained models are frequently shared or commercialized, raising concerns regarding potential privacy breaches. Among these threats are property inference attacks, which aim to extract global characteristics of the training dataset without accessing individual records. While prior research has largely focused on binary property inference attacks, this thesis extends the attack to a multi-class setting, allowing a more detailed analysis of dataset characteristics.

This work investigates the vulnerability of fully connected, feed-forward neural networks to multi-class property inference attacks. The research explores the effectiveness of these attacks by adapting a meta-machine learning approach, originally designed for binary property inference, into a multi-class setting. Furthermore, it examines whether a hierarchical classification strategy can improve attack performance. The study also considers property inference risks in sequential federated learning scenarios, where multiple participants collaboratively train a model while maintaining data privacy.

The conducted experiments assess the accuracy of multi-class property inference attacks across varying class granularities and different loss functions. The findings provide insights into the trade-off between granularity and attack accuracy, highlighting risks associated with sharing machine learning models. The results emphasize the need for stronger privacy-preserving techniques, particularly in federated learning settings, to mitigate property inference attacks and safeguard sensitive data.

Contents

Kurzfassung **xi**

Abstract **xiii**

Contents **xv**

1 Introduction **1**
1.1 Motivation . . . 1
1.2 Problem definition . . . 2
1.3 Research questions . . . 3
1.4 Methodology . . . 4
1.5 Structure of the Thesis . . . 5

2 Background **7**
2.1 Machine Learning . . . 7
2.2 Neural Networks . . . 8
2.3 Model Performance Evaluation Metrics . . . 14
2.4 Sequential federated learning . . . 16
2.5 Property Inference Attack . . . 17
2.6 Deep Sets . . . 19

3 Methodology **21**
3.1 Data set . . . 22
3.2 Model description . . . 23
3.3 Data splitting . . . 30
3.4 Hierarchical approach . . . 34
3.5 Performance evaluation . . . 36

4 Results **39**
4.1 Multi-class attack . . . 39
4.2 Hierarchical approach . . . 55
4.3 Sequential federated learning . . . 66

5 Conclusion **85**

5.1 Contributions 85
5.2 Research questions 86
5.3 Future Work 90

Overview of Generative AI Tools Used **91**

List of Figures **93**

List of Tables **95**

Acronyms **97**

Bibliography **99**

CHAPTER 1

Introduction

This chapter provides an overview on the thesis and motivates the necessity to conduct research in the subject area of property inference attacks. The research questions are formulated, followed by a detailed description of the approach and methodology used to address these questions.

1.1 Motivation

Machine learning (ML) is increasingly used in a wide range of applications such as healthcare, finance, entertainment or autonomous systems. It is used to make predictions, automate decision-making, and get insights from complex data. An important reason for its increased usage are improvements in effectiveness made possible by neural network models. These have shown great capabilities in learning from different training data, and are able to perform a wide range of tasks. However, these models need large, realistic and representative training data to reach good results. This is one reason for the emergence of online markets where pre-trained machine learning models are shared and sometimes sold [AWS12].

Sharing (and/or commercialising) pre-trained models has advantages, such as quicker use in different application areas, but potentially also entails drawbacks. One example is that often, data sets used to train shared models contain sensitive data. As a result, concerns about a potential leakage of private information arise. For instance, models trained on medical records, financial transactions, or personal user data are at risk of leaking private information about the individuals whose data was used for training. Adversaries may try to misuse models to infer such sensitive information, potentially leading to violations of data protection regulations and other serious consequences.

Therefore, it is vital to investigate the vulnerabilities of machine learning models. Different types of attacks aiming to leak private information from machine learning models have

been studied. This thesis focuses on an attack known as *property inference* attack. A property inference attack aims to find out the global properties of the training data. For instance, such an attack could reveal whether the dataset has an under-representation of specific groups, such as having significantly more women than men or excluding dark-skinned individuals entirely. Beyond demographics, property inference attacks could be applied in competitive contexts, such as determining which sales channel a competitor relies on most. The acquired knowledge could be used for malicious purposes, such as discrimination or targeted exploitation, which underlines the need for robust defences against such attacks.

In this research, we want to go beyond current studies, which have mainly focused on the binary case of the property inference attack (e.g., whether there are more men than women). For example, Ganju et al. [GWY+18] investigate binary property inference attacks on fully connected neural networks. However, the binary attack has only moderate power, and extending the property inference attack to the multi-class case is important for understanding the full scope of the privacy risks associated with shared machine learning models.

The danger of property inference attacks is also present in the federated learning setting where multiple participants collaboratively train a model, each using their local data. If it is possible to derive information about the data of other participants, this can have disastrous consequences, such as the exploitation of marketing strategies or the violation of user privacy by exposing sensitive details that can be exploited, undermining trust, and potentially breaching legal protections.

1.2 Problem definition

The problem addressed in this thesis is the vulnerability of machine learning models, specifically fully connected, feed-forward neural networks, to property inference attacks in a multi-class setting. Property inference attacks are a type of privacy attack in which an adversary aims to detect sensitive properties about the entire dataset used to train a machine learning model. For instance, in a model using demographic data in the training process, an attacker might want to know whether the training data under-represents a certain ethnic group or gender. In contrast to model inversion attacks [FJR15], which aim to reconstruct training samples, or membership inference attacks [SSSS17], which determine whether particular data points were part of the training set, property inference attacks derive global information about the training dataset, instead of instance-specific information.

One consequence of property inference attacks is that repeated executions targeting different properties can lead to privacy breaches and legal violations. For instance, if an attacker performs the attack multiple times on various properties and gains increasingly accurate predictions about the dataset's characteristics, it might be possible to trace back to the source of the data. Such capabilities could lead to violations of data protection laws

like the EU's General Data Protection Regulation (GDPR) or the California Consumer Privacy Act (CCPA), highlighting the urgency of addressing these vulnerabilities.

Existing research has laid the groundwork for understanding these attacks in binary contexts, i.e., the property of interest has two possible states (e.g., "more men than women" or "more women than men"). However, these earlier works often assumed that the data used to conduct the attack has a data distributions highly similar to that of the target model, which is a very strong assumption on the attacker's capabilities. In practice, an attacker may not always have access to data that closely mirrors the target model's distribution, making such assumptions unrealistic in many real-world scenarios. Another disadvantage is that the information gain is quite small in the binary case. This thesis extends the attack to a multi-class setting, where the property of interest can have multiple categories or levels. Rather than simply inferring whether a dataset has more of one category than another, the attack in a multi-class setting enables more accurate predictions about the proportion of the training set that contains certain features.

The challenge lies in developing methods that can infer these properties without direct access to the training data while dealing with an increased difficulty in learning accurate patterns. Fully connected neural networks, which are frequently used for solving machine learning tasks due to their flexibility and effectiveness in a wide range of tasks, will be the primary focus for conducting these attacks.

The outcome of this research will provide a comprehensive understanding of the risks associated with property inference attacks in the multi-class setting and also explore the risks in a federated learning setting. The aim is to determine the accuracy with which the attack can be carried out, to be able to estimate the risk of sharing models with sensitive training data.

1.3 Research questions

The problem definitions leads to the following research questions:

RQ 1.1: What is the trade-off between the resulting accuracy of the property inference attack and an increase in it's forecasting granularity? For conducting the property inference attack the binary meta-machine learning model [GWY+18], which is trained on model parameters of different machine learning models (i.e. shadow models) is transformed into a multi-class model.

RQ 1.2: Can a hierarchical approach improve the performance of the attack in the multi-class setting, compared to a straightforward multi-class approach? For the hierarchical approach, the classification task is broken down into a series of simpler decisions using different meta-machine learning models.

RQ 2: How accurately can the property inference attack be conducted in a sequential federated learning setting by the second participant with access to the data set structure and model?

1.4 Methodology

The objective of this master thesis is to evaluate the risks of property inference attacks in different contexts. To be able to answer the research questions, it is important to apply a research method [KK16] that guides the steps within the thesis. After reading about different data mining processes in [SQ14] it was decided to use the Cross-Industry Standard Process for Data Mining (CRISP-DM) [WH00] approach. It is used to build an attack model (a form of meta-model) that should predict whether the training data of the input model contains a specific property or not.

Before creating the model, a literature review was performed to identify related research that has already been carried out in this area.

1.4.1 Literature Review

As stated in [OS10] there are different kinds of literature reviews. In this thesis, [GWY+18] serves as a starting point for the literature in use. Based on their cited works, further literature is explored to incorporate within this thesis. Therefore an overview of similar papers is achieved, being less precise than a systematic literature review [KC07].

1.4.2 Experiment Design and Evaluation

The biggest part of the master thesis consists of building a meta-model-based attack to infer specific properties in the training data, e.g. whether there are more male than female persons. For this purpose, the CRISP-DM [WH00] framework is used. CRISP-DM provides a process model as a framework for carrying out data mining projects.

The CRISP-DM Methodology consists of six phases.

- **Business Understanding:** The first phase is concerned with understanding why it is important to evaluate property inference attacks. This was performed in Section 1.2
- **Data Understanding:** The selection of the dataset to use is part of this phase. The criteria for the datasets are discussed in Section 3.1.
- **Data Preparation:** The training of the shadow models, which are then further used to train the meta-model, is part of this phase. In addition, preparation of the parameters from these models to become a correct input needs to be done.
- **Modeling:** The design of the meta-model is based on the one used in [GWY+18]. Our exact process is reported in Section 3.2.2. There are five different loss functions implemented, as detailed in Section 3.2.3.
- **Evaluation:** The evaluation consists of the evaluation of the meta-model and, therefore, of the successful execution of the property inference attack. More details on the evaluation methods are given in Section 3.5.

- **Deployment:** As the experiments are part of a master thesis, no deployment in a real-world setting is conducted. The process of conducting the attack is written down and the associated findings are analysed.

1.5 Structure of the Thesis

The remainder of this thesis is organised as follows.

Chapter 2 provides the necessary background information to get a basic understanding of the concepts needed for conducting the property inference attack. The chapter starts with an introduction to machine learning and neural networks, focusing on fully connected, feed-forward neural networks and their training processes. Also, the evaluation metrics accuracy and Mean squared error (MSE) are explained. The chapter also covers fundamental concepts of federated learning and the property inference attack. The last section describes the use of Deep Sets [ZKR+17] which are essential for creating meta-models.

Chapter 3 outlines the methodology used to conduct the experiments. It starts with a description of the selection criteria and the eventually chosen data sets. Also, the pre-processing steps are discussed. The next part is a detailed description of the auxiliary models which are necessary for conducting the property inference attack. Also the meta-model used for the attack is described. The meta-model is trained using five different loss approaches which are discussed. Finally, the different data-splitting approaches used to simulate the federated learning scenario are described. Also the hierarchical approach, which is designed to improve the result of the multi-class model, is discussed.

In Chapter 4, the results and findings of the experiments are discussed. The results for the different class granularity namely two, four, five, ten and 20 classes for the five different loss functions are discussed. These results are discussed for the basic case, the extension to the hierarchical approach and in the federated learning setting.

Finally, Chapter 5 summarizes the key findings and implications of the research. It discusses how the extension to multi-class settings provides a more nuanced understanding of property inference risks thereby answering the research questions. The chapter also outlines limitations and proposes potential avenues for future research, including exploring other types of machine learning models and enhancing privacy-preserving techniques in federated learning settings.

CHAPTER 2

Background

This chapter provides the reader with an overview of important concepts necessary to understand the subsequent work. First, we discuss the basic concepts of ML, followed by an explanation of neural networks and the type of neural network used in the thesis. After that, we focus on the property inference attack. We discuss the main idea and how it is performed. To understand some aspects of carrying out the attack, a basic introduction to DeepSets is provided as the last part of the background section. In Table 2.1, the notation used in the following sections is described.

Table 2.1: Notations of neural networks

Notation	**Description**
$X(x_1, x_2, ..., x_n)$	Input values of an ML-Model
y	True label (ground truth)
y'	Predicted label (model output)
f	ML-Model
a	activation value
$W(w_1, w_2, ..., w_n)$	Model weights
b	Model biases
K	Number of output classes
N	Batch size
α	learning rate

2.1 Machine Learning

Machine learning is a subfield of Artificial Intelligence (AI) and focuses on developing algorithms and models that allow machines to learn from data and make predictions based on learning. One goal of ML is to create systems that can automatically improve

their performance over time without having to be explicitly programmed for each specific problem. As Tom Mitchell defines it, "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E." [Mit97, p. 2]

There are different types of problems that can be solved by ML. According to [ANK18], the most common problems of ML are:

- **Classification Problem:** The output of a classification problem consists of a fixed number of categories. The goal is to assign one (or more) of those categories to an unlabelled data sample. One example of this problem is to assign a species like a cat, a dog, or a horse to a picture of an animal. The number of these categories is fixed before training the model and can be chosen arbitrarily. If there are only two output classes, the problem is called binary classification, whereas if there are more than two classes, it is called a multi-class classification problem.
- **Regression Problem:** In contrast to a classification problem, a regression problem predicts continuous output. Regression problems are typically employed for tasks that involve inquiries such as 'how much' or 'how many'.

In machine learning, parameters and hyperparameters are key concepts that influence how a model is trained and how it performs:

- **Parameters:** Parameters are the internal values or variables that a machine learning model learns during the training process. In some algorithms, these values are adjusted iteratively to optimize the model's performance and minimize the error in the training data.
- **Hyperparameters:** Hyperparameters are settings or configurations chosen before training a model. They are not learned from the data but are manually set or tuned to optimize the model's training process and resulting performance. Hyperparameters control how the learning process operates.

This thesis addresses a classification problem, and neural networks, which are described in the following, are used to solve that task.

2.2 Neural Networks

Neural networks are computational models inspired by the human brain. The main components of neural networks are interconnected nodes that are organized in layers. The layers include an input layer, one or more hidden layers, and an output layer.

One of the well-understood types of neural networks, which is yet still capable of universal function approximation, is the Multilayer Perceptron (MLP). We first describe the basic building block, the (single-layer) Perceptron.

2.2.1 Perceptron

The Perceptron algorithm was proposed in 1958 by Frank Rosenblatt [Ros58]. Its goal is to distinguish between two classes using a linear separation. It consists of an input layer and a corresponding output layer. The input to the Perceptron algorithm consists of continuous values $\mathbf{x}$, and the size of the input layer matches the size of the input data. In the first step, the input to the Perceptron is weighted using weights $\mathbf{w}$ and summed, before a bias b is added. The result of these operations is denoted as activation value a. After that, this activation value is put through a threshold activation function, with a threshold θ to determine the output.

$$a = \sum_{i=1}^{n} w_i \cdot x_i + b \tag{2.1}$$

$$y' = f(\mathbf{x}) = \begin{cases} 1 & a \geq \theta \\ 0 & \text{otherwise} \end{cases} \tag{2.2}$$

For the learning process, the weights are first initialized randomly and then adjusted based on the errors made in the classification on a training data set. This is done by comparing the correct label y with the predicted label y' on that training set and then updating the weights $\mathbf{w'}$ accordingly to minimize that error. For this, the learning rate α determines the magnitude of the weight updates. The adaption of the weights is done until the error is minimized, in the best case the prediction is correct ($y' = y$) for all samples.

$$\mathbf{w'} = \mathbf{w} + \alpha(y - y')\mathbf{x} \tag{2.3}$$

The Perceptron is a simple model and, therefore, has some limitations. One key limitation is that it can only converge to a stable state when the data is linearly separable. If the data is not linearly separable, the algorithm will fail to converge, as it continuously adjusts the weights without finding a satisfactory solution. This restricts its applicability to more complex datasets where class boundaries are nonlinear. In its basic form, the Perceptron can only be used for binary classifications.

2.2.2 Multilayer Perceptron

A more complex form of the Perceptron algorithm is the MLP. The following description of the MLP is based on [HN92].

The MLP is a fully connected, feed-forward neural network. It is fully connected because the neural network consists of multiple layers with interconnected nodes, which are

also called neurons. These interconnected nodes are, in principle, Perceptrons, though they usually utilize different activation functions compared to the original Perceptron algorithm. Feed-forward means that the information flows in one direction and that the connections between the different neurons thus do not form a cycle.

An MLP usually consists of one input layer, one or more hidden layers, and an output layer. The hidden layers are called as such as they can not be observed from outside. In other words, nodes of the hidden layer are located between the input and output layers. Usually, each layer consists of more than one neuron. As an MLP is fully connected, each neuron is connected to each neuron in the previous and subsequent layer. As mentioned previously, the size of the input layer is consistent with the size of the input data, whereas the size of the hidden layers is part of the hyperparameter tuning and is denoted as the number of neurons in each layer. The output layer produces the predicted output of the model y'.

The weights and biases assigned to each neuron are the so-called model parameters, and finding optimal values for them is part of the training of the MLP model. The number of weights in a specific layer is the product of the number of neurons in the previous layer and the number of neurons in that layer. The number of biases in a layer is equal to the number of neurons in that layer (since each neuron most commonly has exactly one bias term). The number of model parameters is, therefore, dependent on the number of layers and neurons in each layer. An exemplary neural network with two hidden layers is shown in Figure 2.1.

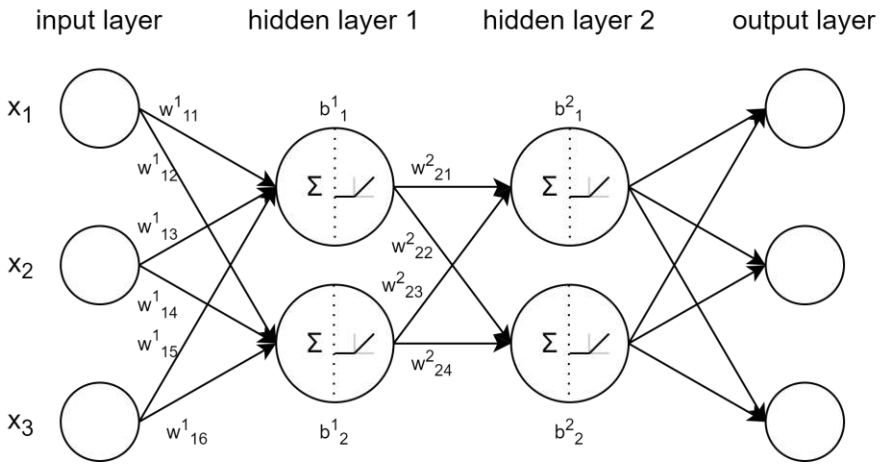


Figure 2.1: Neural network (own work)

The task of each neuron in an MLP is to process the outputs of all neurons in the previous layer. The neuron takes these outputs as inputs and calculates a weighted sum of these with the associated weights and bias. The next step is to pass the result through an activation function to produce an output, which in turn serves as input for the next layer.

Training of Multilayer Perceptrons

The goal of the training process is to find the optimal weight and bias values so that the model predicts accurately on unseen data.

The first step of training a MLP is to initialize the model parameters, e.g., with random values.

The first training iteration then starts by feeding the first input sample x_1 into the MLP, for which a forward pass is executed. A forward pass is defined as the process of computing an output for a given input. This means that the input vector $textbf{x}_1$ is fed into the input layer and each neuron computes the weighted sum of its inputs and its bias. Then the activation function (see Section 2.2.2) of the neuron is triggered, and the output serves as input for the neurons of the next layer. This is done until the last layer is reached, and then the model's output y' is produced [HN92].

In the next step of the MLP, the error between the ground truth y and the predicted output y' is calculated using a loss function. The distinction between error and loss lies in their roles: the error represents the difference between the predicted and actual values, while the loss function quantifies the impact or negative consequence of that error. In other words, the loss function transforms the error into a single value that guides the optimization process during training.

One of the most known loss functions is the negative log-likelihood loss function [Con23c], calculated as

$$l(X, y) = \sum_{n=1}^{N} = \frac{1}{\sum_{n=1}^{N} w_{y_n}} l_n, \tag{2.4}$$

where X is the input of the neural network, y is the true label, N indicates the batch size and w is the weight. l_n is calculated as follows: $l_b = w_{y_n} x_{n,y_n}$

In this work, other loss functions are also used, which will be described in detail in Section 3.2.3.

To optimize the weights and biases of the model, the computed loss of the loss function is processed backwards. This means that it is propagated from the final layer toward the input layer; this process is called backpropagation. In this step, the weights and biases are updated to better represent the currently presented sample, using an optimization function. The most common function to update the weights and biases is a variation of the Gradient Descent [Lem12].

The goal of Gradient Descent (GD) is to gradually minimize a loss function (f.e. Equation (2.4)) calculated on the data set X with the corresponding labels Y. This is done until the values reach a global minimum. The gradient is calculated for each parameter of the model, and the parameters are then updated according to the gradient by a predefined magnitude, which is called the learning rate. This process is carried out for all parameters, beginning at the last layer and moving toward the first layer. There are

three different variations of the GD differing in the number of samples used to update the parameters of the model.

- **(Batch) Gradient Descent:** The whole input dataset is processed before the loss is calculated and the model parameters are updated. This implies a precise direction for optimization but leads to a slow calculation.
- **Stochastic Gradient Descent** (SGD): The update is performed after each data sample making it more computationally efficient. One disadvantage of using Stochastic Gradient Descent (SGD) is convergence accuracy: due to the random sampling of individual training examples, it may be possible that the algorithm may not converge to the true minimum.
- **Mini-batch Stochastic Gradient Descent:** Here a mini-batch of samples is used to make a single optimization step. The batch size usually equals a power of two (e.g. 8, 16, 32). This is done to align with the memory storage of computers. The mini-batch SGD combines the advantages of the two methods discussed above, making it faster and more likely to converge correctly.

In this thesis, the Adam optimizer [KB15], an adaption of SGD, is used. One difference between SGD and Adam is that Adam adapts individual learning rates for each parameter. A benefit of the Adam optimizer is that it converges faster. This is because it uses momentum, meaning that it takes into account the exponentially weighted averages of the gradients, which leads to a faster convergence of the algorithm.

During the training process, the input data is generally presented more than once to the neural network. When the entire training data set is processed once, this is called an epoch. Thus, the number of epochs is defined by the frequency with which the neural network works through the entire training data set.

The batch size, number of layers, nodes, and epochs are hyperparameters of a neural network. To optimize the model for specific tasks, these parameters can be chosen differently, and an important part of training successful models is to find the optimal hyperparameters. There are different methods to find the optimal hyperparameters of a neural network ranging from untargeted trial and error to methods like Grid Search or automated hyperparameter tuning such as Bayesian optimization methods [Moc89].

Activation functions

In order to introduce non-linearity to the neural network, several common activation functions are available. Without non-linear activation functions, all neurons would behave linearly, and the MLP could be reduced to a simple Perceptron [MP17], limiting its ability to model complex patterns. For this reason, activation functions are typically chosen to be non-linear. Additionally, activation functions must be numerically differentiable to enable the backpropagation process, as differentiation is essential for optimizing weights and biases. Popular activation functions include [SSA20]:

- Threshold activation function, as with the Perceptron

$$f(x) = \begin{cases} 1 & x \geq \theta \\ 0 & \text{otherwise} \end{cases} \tag{2.5}$$

- Linear function

$$f(x) = kx, \text{ for some constant } k \in \mathbb{R} \tag{2.6}$$

- Sigmoid function

$$f(x) = \frac{1}{1 + e^{-x}} \tag{2.7}$$

- Tanh function

$$f(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{2.8}$$

- Rectified Linear Unit (ReLU) funciton

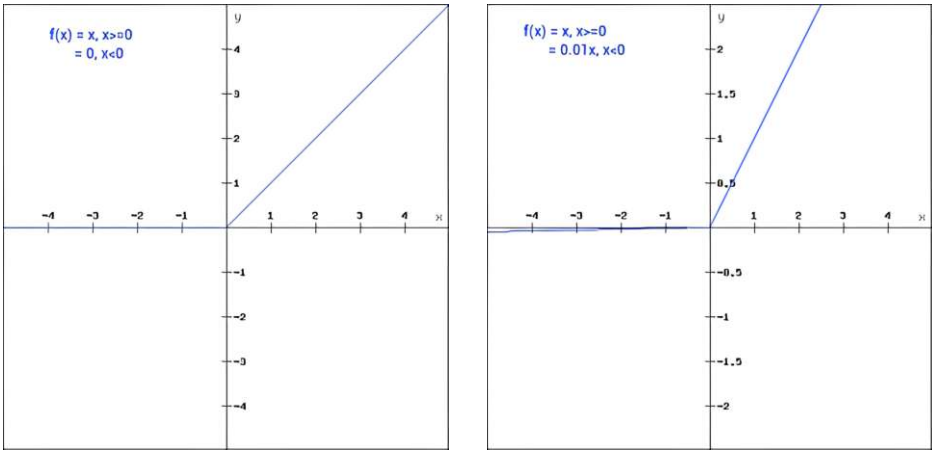
$$f(x) = max(0, x) \tag{2.9}$$

- Leaky Rectified Linear Unit (Leaky ReLU) function

$$f(x) = max(0.1x, x) \tag{2.10}$$

In this thesis, the Leaky ReLU activation function is used, as it was also employed in previous work on this topic, and we want to enable consistency and comparability with prior research. Leaky ReLU is an improved version of the ReLU function, which is widely used in deep neural networks. It is composed of a linear(identity) function and a constant function: the output of ReLU is the input if the input is positive, otherwise, the output will be zero [SSA20]; this behaviour is illustrated in Figure 2.2a.

One problem with the ReLU function is that neurons might have a weight equal to zero and, therefore, do not contribute to the learning process. The Leaky ReLU function tries to alleviate this problem. Instead of setting the output to zero for non-positive inputs,



(a) ReLU activation function (b) Leaky ReLU activation function

Figure 2.2: ReLU and Leaky ReLU activation function [SSA20]

it is defined as an extremely small linear component of x [SSA20]. The Leaky ReLU is shown in Figure 2.2b.

The output layers in the models of this thesis use a different activation function, namely the softmax activation function. The goal of softmax, a generalized version of the logit function, is to scale the output of the last hidden layer into probabilities. These probabilities inform about how likely a class is in the case of a classification problem. The output of a softmax is, therefore, a vector of the size of the number of all possible outcomes, containing the probabilities for each possible outcome; the vector thus sums up to 1. The softmax function S is defined as

$$S(z)_i = \frac{\exp(z_i)}{\sum_{j=1}^{n} \exp(z_j)}, \tag{2.11}$$

where n is the number of possible outcomes, exp() denotes the standard exponential function, z is the input vector of the softmax function and z_i is the i-th element of the input vector.

2.3 Model Performance Evaluation Metrics

After training a machine learning model, e.g. a neural network, its performance must be measured. This is done by comparing the ground truth of the data with the predictions of the model.

2.3.1 Evaluation of classification models

There are four different types of outcomes for classification models that could occur:

- **True positive (TP)**: A correct prediction in the form that the model predicts that an input does belong to one class and it actually does belong to the class
- **True negative (TN)**: A correct prediction in the form that the model predicts that an input doesn't belong to a class and it actually doesn't belong to a class
- **False positive (FP)**: A false prediction in the form that the model predicts that an input does belong to a class and it actually doesn't belong to that class
- **False negative (FN)**: A false prediction in the form that the model predicts that an input doesn't belong to a class and it actually does belong to that class

These four outcomes can be displayed in a confusion matrix. A picture of a confusion matrix in the binary case can be seen in Figure 2.3. The confusion matrix can also be extended to the multi-class case.

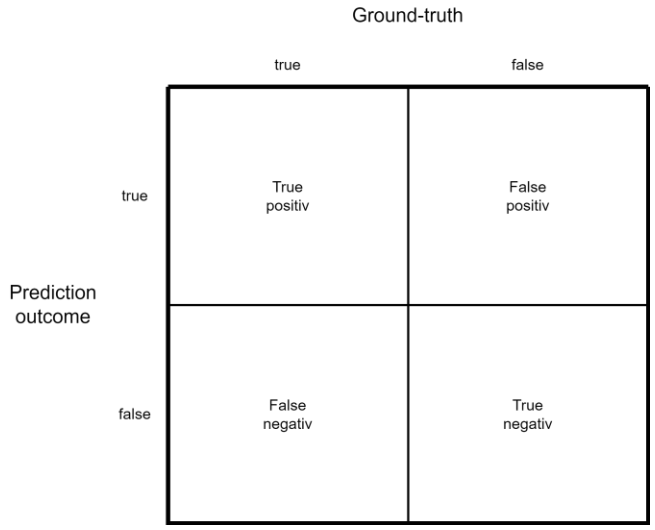


Figure 2.3: Confusion matrix for the binary case (own work)

There are different metrics used to assess how well a neural network works based on the four above outcomes. Among others, three popular metrics for classification models are accuracy, precision, and recall.

Accuracy defines the proportion of correctly classified samples over the total number of samples

$$\text{accuracy} = \frac{\text{correct predictions}}{\text{all predictions}} = \frac{TP + TN}{\#\text{ number of samples}} \tag{2.12}$$

Precision defines the proportion of correctly positive classified samples over all samples predicted to belong to a specific class

$$\text{precision} = \frac{TP}{TP + FP} \tag{2.13}$$

Recall defines the proportion of correctly positive classified samples over all samples actually belonging to the specific class

$$\text{recall} = \frac{TP}{TP + FN} \tag{2.14}$$

This thesis uses only accuracy to measure performance. This is done because in [GWY+18] it is stated that the precision and recall of the experiments are very similar to the accuracy. Therefore, it was decided that accuracy is sufficient to report.

2.3.2 Evaluation of regression models

A major difference between the evaluation of classification and regression models is that with classification models, the only concern is generally whether or not a prediction is correct. Regression models, on the other hand, predict in a continuous range, and thus it is also relevant to determine how far the model is from the correct value.

In this thesis, we also measure how far off the inferred information is from the real value, and thus, one regression evaluation method, the Mean squared Error (MSE), is used. As we deal with a specific case of classification, namely ordinal classification, as we will detail in Section 3.5, a regression evaluation method is well suited and required for this specific setting.

Mean squared error (MSE) defines the average of the squared differences between the predicted output and the ground truth

$$\text{MSE}(y, y') = \frac{1}{\#\text{ number of samples}} \sum (y - y')^2 \tag{2.15}$$

2.4 Sequential federated learning

Federated learning is a term first introduced by McMahan et al. [MMR+17] for a collaborative machine-learning effort. The goal is to keep data collected and stored at various participants decentralized, meaning it is not shared between the participants or a central server, while achieving a model of similar effectiveness as if the data was indeed shared and the model was learned from this centralized data. This approach is especially useful when data privacy is a concern or when it's difficult to centralize the data due to its size or distribution.

This thesis focuses on sequential federated learning, meaning that instead of training multiple models on different data at the same time a single model is initially trained on one client. After training the model for a specific time or if a previously identified criterion is met, the trained model is sent to the next participant for the next training round. This participant then further trains the model with their local data. This process is then repeated until the desired performance is achieved. The process is described in more detail in the next paragraph.

As a first step a global model is initialized at a single device. This can be a model pre-trained on a large dataset or a randomly initialized model. The model is then trained on the individual data set of the first participant before it is sent to the next participant. This participant further trains the model with their data (which is only known by this participant). This is repeated for all participants. If the last device is reached the model can be sent back to the first participant for the next round of training. This iterative process continues until the desired performance is reached or another convergence criterion is met.

2.5 Property Inference Attack

In this thesis, we investigate an attack that can be performed against ML models, where the attacker wants to infer some information on the training data of a model. The idea behind property inference attacks is that ML models with a similar structure and trained on similar data sets will exhibit similar behaviour, i.e. similar structures or patterns are recognizable in their parameters. The attacker wants to recognize these patterns in the target model and reveal some information about the training data set, which the model should not reveal. The first work on property inference attacks was introduced by Ateniese et al. [AMS+15]. They focused on property inference attacks against Hidden Markov models [BP66] and Support Vector Machines [Bur98] and carried out successful attacks in the binary case.

In [GWY+18], the property inference attack was first successfully conducted against fully connected neural networks. The approach presented in [GWY+18] was limited to binary statements (e.g. a property being smaller or larger than a fixed value). As a result, it provides only limited informative value to the attacker; we thus extend the approach to the multi-class case in this work, leading to a more precise statement about global properties that occur in the training data set of the target model, i.e. allowing more fine-grained prediction on the value of the target property.

To carry out a property inference attack, the attacker usually needs to train another ML model whose task is to recognize patterns in the target model, which is the model the attacker seeks to compromise. This secondary model, referred to as the meta-model, or attack model, learns to identify these patterns. To train the meta-model, a technique known as "shadow training" [AMS+15, SSSS17] is used. Its idea is to train multiple models that mimic ("shadow") several variants of the target model, each with different global properties; these shadow models then serve as training data (input) for the meta-model.

This attack strategy is depicted in Figure 2.4. The shadow models are trained using data sets that either contain or don't contain the property the attacker wants to exploit. If the attacker e.g. wants to infer the percentage of women in the training data set, he/she has to train an equal number of shadow models for each percentage class. If the attacker wants to check whether more than 50 % of the observations in the training dataset are women, then he/she must train one-half of the shadow models with training data containing more than 50 % women, and the second half of the shadow models must be trained with training data sets containing less than 50 % women. The attacker can obtain these data sets via sampling of a larger data set or by obtaining more data. Note that these data sets used for training the shadow models should have the same structure as the one used for the target model.

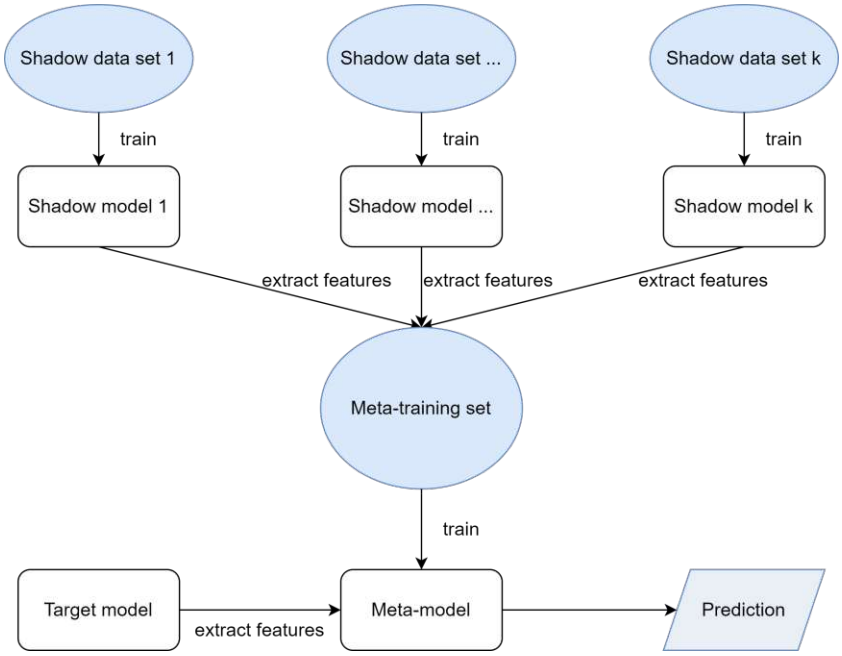


Figure 2.4: Workflow of the property inference attack

It is important to reduce the number of unknowns during the training of the shadow models to understand the impact of each aspect, i.e., the attacker has to try to create a training environment that is as similar as possible to the training environment of the target model. This thesis focuses on this type of attack in a white-box setting. This means the attacker has full knowledge of the architecture and parameters of the target model, and the shadow models are designed to replicate the same architecture. In order to obtain meaningful parameters in the shadow models, they have to be trained to reach a reasonably good performance on their original classification task. Otherwise, the model parameters might be adapted too little for their task to be able to recognize patterns.

After training the shadow models, the learned parameters, i.e. the weights and biases of these shadow models, serve as input data (features) for the meta-model. The parameters

of a single shadow model are one observation in this data set, referred to as X. The corresponding label y then denotes the to-be-revealed information, e.g., whether or not the model contains more than 50 % women in the training set.

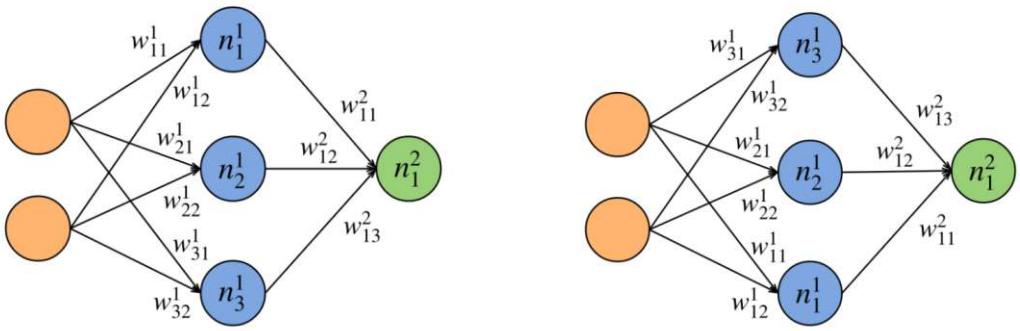
After training the meta-model, the attacker has to use the parameters of the target model (i.e. the weights and biases) and feed them into the meta-model. The resulting prediction indicates whether the target model was trained with a training data set containing more than 50 % women or not.

In [GWY+18], different approaches to extract the model parameters from the shadow and target models are developed, and the problem of permutation-equivalent neural networks is addressed, as described below.

A simple explanation of a permutation-invariant function is shown in Equation (2.16) for a function f:

$$\text{f(a,b,c)} = \text{f(a,c,b)} = \text{f(b,a,c)} \tag{2.16}$$

In Figure 2.5 two permutation-equivalent neural networks are shown. In the field of neural networks, permutation equivalence implies that one neural network f_1 is distinguishable from another neural network f_2 simply by a permutation of the nodes in the hidden layers.



(a) Neural network f_1[GWY+18] (b) Neural network f_2[GWY+18]

Figure 2.5: Two permutation equivalent neural networks

The result of their paper was that using a set-based representation of the model features to train the meta-model leads to the most successful property inference attack. This thesis extends the approach of [GWY+18] to the multi-class case.

2.6 Deep Sets

The idea behind deep sets was proposed in [ZKR+17], where the authors propose to represent neural networks as sets, instead of simple ordered lists of elements. One

difference between sets and ordered lists is that sets do not allow duplicates. A second difference, which is relevant to this work, is that a set is an unordered collection. Deep sets receive sets as input and achieve permutation invariance of the inputs due to their special architecture.

According to [ZKR+17], a function that operates on an input X is a valid set function, i.e. invariant to the order of the elements in X, if it can be decomposed into the following form:

$$\rho(\sum_{x \in X} \phi(x)) \tag{2.17}$$

In order to learn such a function using neural networks, the functions ϕ and ρ are parameterized by two separate neural networks. The network of the ϕ function is responsible for processing the set elements independently and the network ρ processes the summed output of the ϕ network. One can imagine this process as a map-reduce process [DG08].

Due to the fact that ϕ only processes each element of the set instead of the whole set and ρ only processes the sum of the ϕ network outputs, the architecture deals with fewer parameters. This fact makes Deep Sets highly computationally efficient and simpler to train compared to other architectures, such as Janossy pooling [MSRR18] or sorting [BTBD20].

CHAPTER 3

Methodology

The focus of this chapter is on the methodology used to evaluate the property inference attack. In the first part, the US Census data set [UC 00] is presented. The data set was selected meeting some prespecified criteria. In the second part, the US Census data set is described in more detail. The data set is compiled using survey data from 1994-1995, and its goal is to predict whether an individual earns more than $ 50,000 annually. This chapter also describes the necessary data preparation steps including removing irrelevant columns, encoding categorical features, and transforming binary features.

We then provide a detailed description of the model architecture for the shadow and target models. As the attack is conducted in a white-box environment, the same architecture is used for the shadow and target models. The next section explains the architecture of the meta-model where the Deep Set approach is used to handle the problem of permutation invariance of neurons in the shadow and target models.

We evaluate five different loss functions for training the meta-model. These loss functions are differentiated into one loss function not using ordinal classification, one interval-based loss function and three ordinal regression loss methods.

The next section focuses on different approaches for splitting the data to distinguish them into training and test data. The first part describes the data splitting approach also used in [GWY+18], while the second approach simulates a federated learning setting.

The hierarchical approach is discussed next. The hierarchical approach is an attempt to improve the results of the first data-splitting approach.

The last section discusses the performance evaluations used to evaluate the attack. Here, two methods are used, namely the accuracy and the MSE.

3.1 Data set

To ensure robust and reliable results, the data set selected for this thesis needed to meet several key criteria. These criteria are essential for the reproducibility, comparability, and overall integrity of the experiments conducted in this thesis.

- Availability: The data set should be publicly available without restrictions, allowing unrestricted access to ensure comparability with other studies and enabling reproducibility of the results. A widely accessible data set supports transparent research, allowing others to replicate and validate the findings.

- Size of the data set: Given that the models in this thesis are fully connected neural networks, a large amount of data is required to train models effectively. Additionally, this research involves training a significant number of shadow models, each with slight dataset variations, to closely emulate the target model's behaviour. A large data set is, therefore, necessary to support these variations and produce statistically meaningful results.

- Benchmarking: Benchmark results should be available for the training task of the shadow and target models. Having access to established benchmarks provides a reliable baseline to evaluate the performance of the models and determine whether they reach an optimal or near-optimal performance level. As already mentioned in Section 2.5, the shadow and target model must reach a reasonably good performance on the original classification task. To determine whether this is the case and to check if the performance of the original classification is (close to) optimal, the benchmarks are highly beneficial.

In the end, the same data set as already used by [GWY+18], namely the US Census data set (see Section 3.1.1), was used. The data set fulfils all the criteria mentioned above and allows for further comparison of the results.

3.1.1 Census-Income (KDD) Dataset

The Census-Income data set is a tabular data set conducted by the U.S. Census Bureau and is available, e.g. in the UCI Machine learning repository [UC 00]. The data is extracted from population surveys of 1994 and 1995 and its goal is to predict whether a person earns over $ 50.000. The data set consists of 42 demographic and employment-related variables (e.g. gender, race, education, material status, citizenship, ...) and has 299,285 instances overall. The data set contains categorical and integer features. A split into train and test sets is already done by the original publishers of the dataset and consists of a train set size of 199,523 instances and a test set size of 99,762 instances. This corresponds to a division into 2/3 and 1/3.

3.1.2 Data preparation

The data set was pre-processed prior to training the neural network. The first step was to remove the column "instance weight", as this column was not intended for use in classifiers. It only indicates the number of people in the population that each record represents due to stratified sampling [UC 00]. Next, the target variable was encoded using the LabelEncoder [ld07]. The third step was to one-hot-encode the 31 categorical features. The last step was to transform the binary-valued feature "sex" into 0 and 1. These preprocessing steps were applied consistently across both the training and test data sets, resulting in a data set with a total of 509 features.

3.2 Model description

3.2.1 Shadow and target model description

As mentioned in Section 2.5, the property inference attack is carried out in a white-box environment. This means that the attacker knows the architecture of the target model. In order to carry out the best possible attack, the shadow models the attack builds on use the same architecture as the target model. This is a realistic setting, as especially for the federated learning case it is clear that an attacker that participates in the learning knows the architecture. Additionally, even in other settings, users interacting with the model often have ways of discerning its structural details, making this assumption plausible in a range of real-world scenarios [TZJ+16].

In the remainder of this section, "models" refers to both shadow and target models.

To ensure that the results of the property inference attack are comparable the same model architecture as used in [GWY+18] was chosen. The models consist of three hidden layers with sizes 32, 16, and eight. These hidden layers are linear layers with a Leaky Rectified Linear Unit (Leaky ReLU) activation function. The output layer contains a softmax activation function. The weights are initialized using the "xavier_uniform" function of PyTorch [Con23a]. To update the weights and biases of the models the Adam optimizer [KB15] is used. The models use the negative log-likelihood loss function (see Equation (2.4)).

The original classification task of the Census US data set is to predict whether a person earns more than $ 50.000. This is also the goal of our models; therefore, the models are binary classification models.

3.2.2 Meta-model description

As input for the meta-model, the set-based representation approach of [GWY+18] is used. In this approach, the neural network is represented such that each layer is treated as an individual set of nodes. The major advantage of using the set-based representation is that all permutations of neurons in a layer have the same representation. In order to process these representations, the Deep Set approach [ZKR+17] is used.

The set-based representation of a neural network f with $|f|-1$ hidden layers and one output layer consists of $|f|$ different sets. The t^{th} set represents the layer h_t and consists of $|h_t|$ elements, which is equal to the number of neurons in the layer h_t.

As mentioned in Section 2.6, to implement a neural network that follows the Deep Set approach [ZKR+17], two different types of neural networks, namely ϕ and ρ, must be defined. To be more precise, it is necessary to define $|f|$ ϕ-type networks, i.e., one network for each layer. Since the ρ network is used to determine the final prediction and uses the summed results of the ϕ networks as input, only one ρ network is required. All ϕ and ρ are fully connected neural networks.

The exact workflow used to create the meta-model is shown in Figure 3.1 and follows the approach of [GWY+18]. The only difference is in the calculation of the final prediction – instead of a binary network, this work uses a multi-class ρ network.

The first step is to flatten the weights and biases of each node n_i^t for each layer h_t. These flattened weights and biases are then fed into the corresponding ϕ_t network to obtain the node-level representations N_i^t. This step is called node processing.

Figure 3.1 illustrates how the concatenated node-level representations are fed into the ϕ networks for each successive layer. Each ϕ_t network takes as input a flattened list of weights and biases for each node within its respective layer h_t. Additionally, for all ϕ_t networks except the first, the context N^{t-1} is included as an input, where

$$N^{t-1} = (N_1^{t-1}, N_2^{t-1}, ..., N_{|h_{t-1}|}^{t-1}) \tag{3.1}$$

This context is essential to capture how each node's function relates to its inputs, providing insight into the operational context of the node. For instance, while a node might compute a function such as the summation of its inputs, the broader context might differentiate between summing selected inputs versus summing the differences of selected inputs, which can impact the overall representation.

After the node processing step, the layer summation step is performed. This means that the node representations for each neuron in a layer h_t are summed to obtain the layer representation L_t.

The last step is to combine the $|f|$ layer representations and use them as the input of the ρ network. The output of the meta-model, which is predicted via the ρ network, is to determine to which class the target model belongs. As this thesis extends the approach of [GWY+18] to the multi-class case, it is not just determined if a target model contains a property or not, but it also tries to differentiate how often a property occurs in the target model. Figure 3.1 shows the binary case of the property inference attack. For this thesis, it is not just decided if P or $\overline{P}$ is true but a distinction is made between several classes.

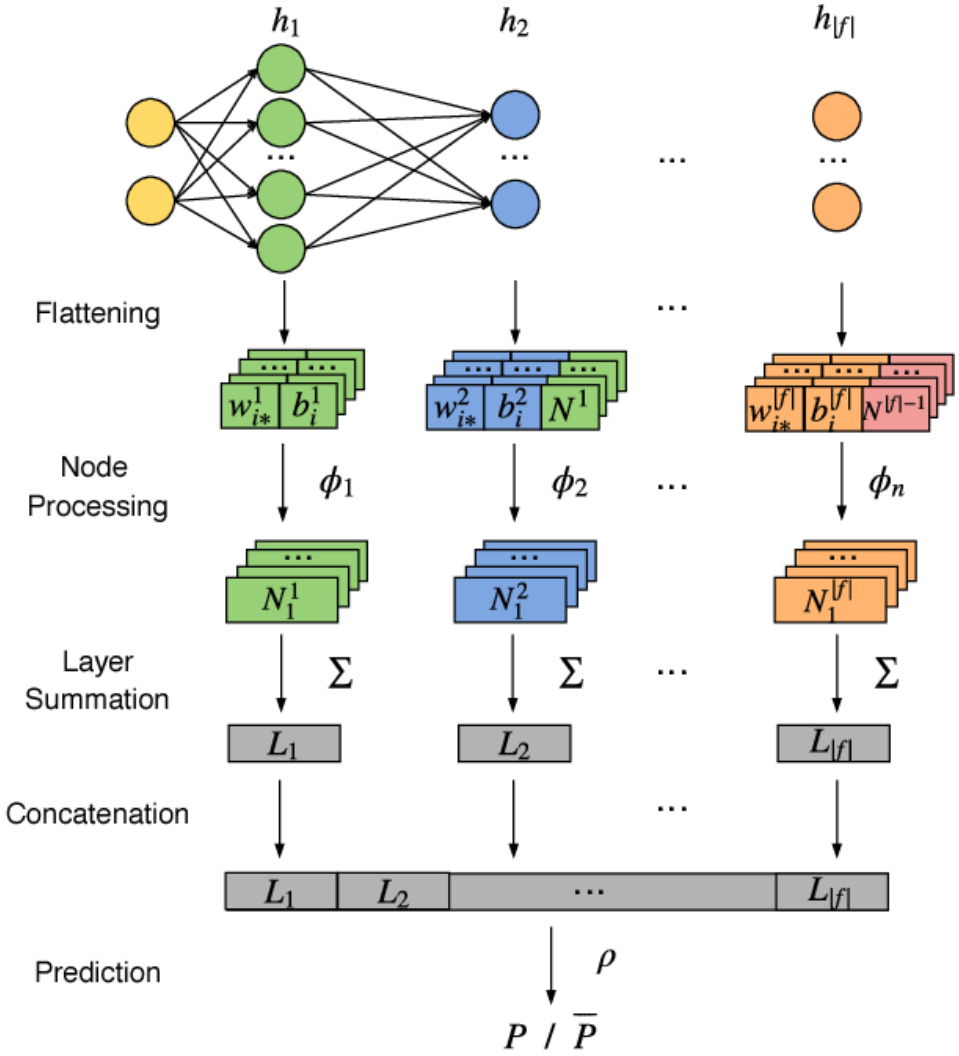


Figure 3.1: Workflow of the meta-model [GWY+18]

ϕ-Network

Since the architecture of the shadow and target models consists of four layers, the meta-model consequently has four ϕ networks. Each of these networks consists of one or two layers. All networks, except the one for the 4th layer of the shadow or target model, consist of one input and one output layer. The network for the 4th layer consists of only one layer. The output layer here is not the final output layer of the meta-model, but only the one that generates the results before summation. Therefore, no special activation function (like softmax) is used. All layers in all ϕ networks are linear layers with a Rectified Linear Unit (ReLU) activation function. The last layer in each network has an output size of eight.

Table 3.1: Percentages of women in a training set divided into 4 equal classes

Class 1	0-25 % women
Class 2	26-50 % women
Class 3	51-75 % women
Class 4	76-100 % women

ρ-Network

The ρ-network processes the concatenated summed outputs of the ϕ-networks. As each of the four networks produces an output vector of size eight, the rho network receives a 32-dimensional input. The output of the rho function depends on the loss function used to train the meta-model. In the following, we describe the different loss functions considered in this thesis.

3.2.3 Loss-functions of the meta-model

The goal of the meta-model is to correctly classify a target model based on specific properties. For instance, if the property in question is the percentage of women in the training dataset of the model, the attacker has to decide in how many classes the property should be divided. If the attacker decides to divide the percentages of women into four equal classes these classes would correspond to the percentages seen in Table 3.1. This classification enables the meta-model to categorize the target model's dataset according to predefined property ranges, allowing the attacker to infer detailed characteristics about the dataset used for training.

An important factor to consider when choosing loss functions is the severity of the classification errors. For example, misclassifying a model as being in class 1 even though its true label is class 4 is a more significant error than misclassifying it as class 1 when its true label would be class 2. This means that the meta-model is an ordinal classification problem, meaning an order of classes exists.

In this thesis, five different approaches to loss methods are tested. These methods are described in detail in the following sections.

Approach without ordinal classification

One approach was to not use an ordinal classification loss method but instead, the negative log-likelihood loss function [Con23c] was used. In this case, the activation function of the ρ-network was a softmax function.

Interval approach

The second approach follows the idea of transforming the classification problem into a regression problem. This transformation begins by redefining the class labels. Instead of using discrete labels such as 1, 2, 3,... the labels are recalculated as follows:

$$\frac{(\frac{1}{\text{Number\_of\_classes}})}{2} + (\frac{1}{\text{Number\_of\_classes}}) * \text{original\_class} \tag{3.2}$$

In general, this method produces labels that represent the midpoint of each class interval. This allows the model to treat each class as a continuous value. The values of a 4-class problem can be seen in Table 3.2.

Table 3.2: Labels for interval approach and 4-class problem

Class 1	0,125
Class 2	0,375
Class 3	0,625
Class 4	0,875

As an activation function for the ρ network the sigmoid function [Con23d] is applied. The loss method used to calculate the loss is the MSELoss [Con23b]. This loss is based on the MSE (see Equation (2.15)).

The last step is to transform the model results back into the original prediction classes. For this, the class ranges are calculated. In the case of 4 classes, the ranges can be seen in Table 3.3.

Table 3.3: Ranges for interval approach and 4-class problem

Class 1	0,00-0,25
Class 2	0,26-0,50
Class 3	0,51-0,75
Class 4	0,76-1,00

Simple ordinal regression approach

The Simple ordinal regression (SOR) approach follows the approach described in [JZP08]. The idea behind this SOR approach is to transform the classes (in this case again a 4-class problem) into the vectors seen in Table 3.4:

Table 3.4: Vectors for SOR approach and 4-class problem

Class 1	[1,0,0,0]
Class 2	[1,1,0,0]
Class 3	[1,1,1,0]
Class 4	[1,1,1,1]

In these vectors, the difference between the classes follows an ordinal scale so that the difference between classes reflects their relative order. For example, the difference between class 1 and class 4 is larger than the difference between class 1 and class 2.

The idea proposed in the paper is that the neural network's output is a 4-dimensional vector with a binary score for each label that predicts the transformation of the classes. A sigmoid function [Con23d] is used in the ρ network to produce values between 0 and 1 for each output. This implies that the output of the *rho* network consists of a 4-dimensional vector with values between 0 and 1.

To train the network an appropriate loss function is needed. The paper [JZP08] states that a squared error loss should be used. In this thesis, we thus use the MSELoss [Con23b].

The final step is to transform the predictions into class labels. This is achieved by applying a threshold function to the prediction vector, where any value above 0.5 is marked as True and the values below are marked as False. The resulting boolean vector is then processed by counting the True entries from left to right, which determines the final class label.

For example, consider a model prediction of [0,7800; 0,9600; 0,2300; 0,1000]. After applying the threshold (> 0.5) the vector is converted into [True, True, False, False]. Counting the True values from left to right yields a sum of 2, so the final predicted class is class 2. This approach leverages the ordinal structure of the predictions, ensuring that higher classes reflect a greater number of thresholds met.

CORAL framework approach

The SOR approach has one disadvantage: *rank inconsistency* occurs when the neural network does not enforce an ordering on the elements of the output vector, meaning it does not guarantee that the earlier elements are greater or equal to the last elements. One example of rank inconsistency is shown in Figure 3.2. The output vector of the probability that the age is greater than 50 is bigger than the probability that the age is greater than 40 which should not be possible. Each of these probability estimates is called a binary task as they calculate a probability between 0 and 1.

This problem is solved using the Consistent rank logits (CORAL) framework [CMR20]. CORAL addresses the rank inconsistency problem by including a weight-sharing constraint in the output layer. This constraint ensures that the K-1 binary tasks in the output layer share the same weight parameters while having independent bias units. As seen in Figure 3.2 the CORAL approach uses K-1 binary tasks rather than K, as lower thresholds automatically imply satisfaction of higher ones. In the graphic, the first probability decides if the age is greater than 10. If this is already not met, then any threshold for the age greater than 20 is also not met. Therefore, it is sufficient to have K-1 binary tasks.

The authors of [CMR20] provide a Python package called coral-pytorch, implementing a pre-build Coral Layer that enforces the weight-sharing constraint. Additionally, the package offers a special loss function that calculates the loss for the K-1 binary outputs of the Coral Layer against the true labels, utilizing weighted cross-entropy to maintain ordinal consistency throughout the classification process.

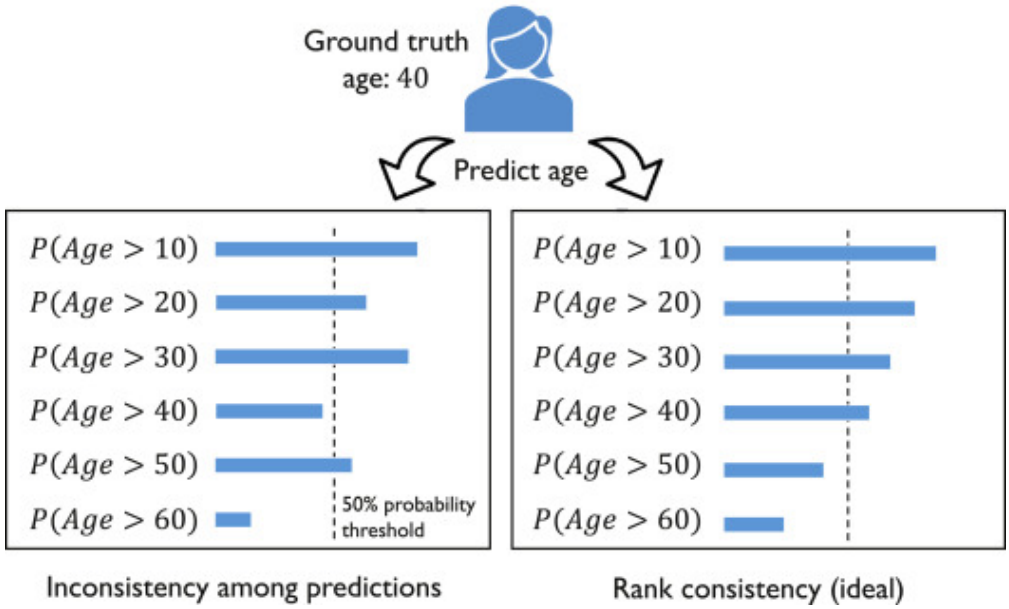


Figure 3.2: Example for rank inconsistency [CMR20]

CORN framework approach

The Rank-consistent ordinal regression (CORN) framework [SCR23] offers an improvement over the CORAL framework by also addressing the rank inconsistency problem but without the weight-sharing constraint in the output layer. According to the authors, removing this constraint enables the CORN framework to achieve even better performance in ordinal classification tasks.

The CORN framework introduces a new training procedure that maintains rank consistency through conditional training subsets and applies the chain rule of probability to ensure ordered predictions. The chain rule used in CORN is expressed as follows in Equation (3.3), where $y^{[i]}$ represents the true label, r_k denotes the k-th rank and $f_k(x^{[i]})$ indicates the conditional probability.

$$\hat{P}(y^{[i]} > r_k) = \prod_{j=1}^{k} f_j(x^{[i]}) \tag{3.3}$$

The Python package coral-pytorch, which provides necessary tools including a custom loss function tailored forCORN, is used. Unlike CORAL, CORN does not impose weight constraints in the output layer, making it flexible in layer selection - in this thesis, a linear layer is used. However, only K-1 learning tasks are used in the output layer.

The CORN specific loss function, detailed in Equation (3.4), calculates the loss for each conditional probability task. Here, $\mathbb{1}\{\cdot\}$ represents an indicator function that returns 1

if the inner condition is met and 0 otherwise, $f_j(y^{[i]})$ is the prediction for the j-th node in the output layer and $|S_j|$ represents the size of the conditional training subset S_j.

$$L(X, y) = -\frac{1}{\sum_{j=1}^{K-1} |S_j|} \sum_{j=1}^{K-1} \sum_{i=1}^{|S_j|} [log(f_j(x^{[i]})) * \mathbb{1}\{y^{[i]}\} + log(1 - f_j(x^{[i]})) * \mathbb{1}\{y^{[i]} \leq r_j\}] \tag{3.4}$$

This loss function enforces ordinal constraints by training each binary task independently, ensuring consistency in rank order without the need for weight-sharing, thus allowing CORN to perform well across a variety of ordinal classification tasks.

3.3 Data splitting

This thesis presents two different approaches for dividing the data and determining which models are trained on each data subset.

3.3.1 First data splitting approach

For the first research question, the same approach as in [GWY+18] was used to split the data and train the shadow and target models. This choice provides a foundation for comparability with prior research and highlights the effect of adapting the property-inference attack into a multi-class problem.

The process, shown in Figure 3.3, begins with the Census Income dataset, which contains 199,523 instances in total Section 3.1.2. To train each shadow model, a random subset of 11,971 instances is selected from the training data.

This number is based on the methodology of the second approach, where 30 per cent of the total training data is set aside for training shadow models, and each shadow model receives 20 per cent of this subset. For consistency, this same sample size (11,971 instances) is used in the first approach as well.

$$(199,523 * 0.3) * 0.2 = 11,971.38 \tag{3.5}$$

For each class, a total of 2,304 shadow models are trained to capture class-specific properties. In a four-class setting, that would yield a total of 9,216 shadow models.

To ensure shadow models perform well on their classification task, they are evaluated using the US Census test data. It is important to have a reasonably good performance on the classification tasks; therefore, only shadow models with an accuracy greater than 90 % are selected in the test set (see Section 2.5).

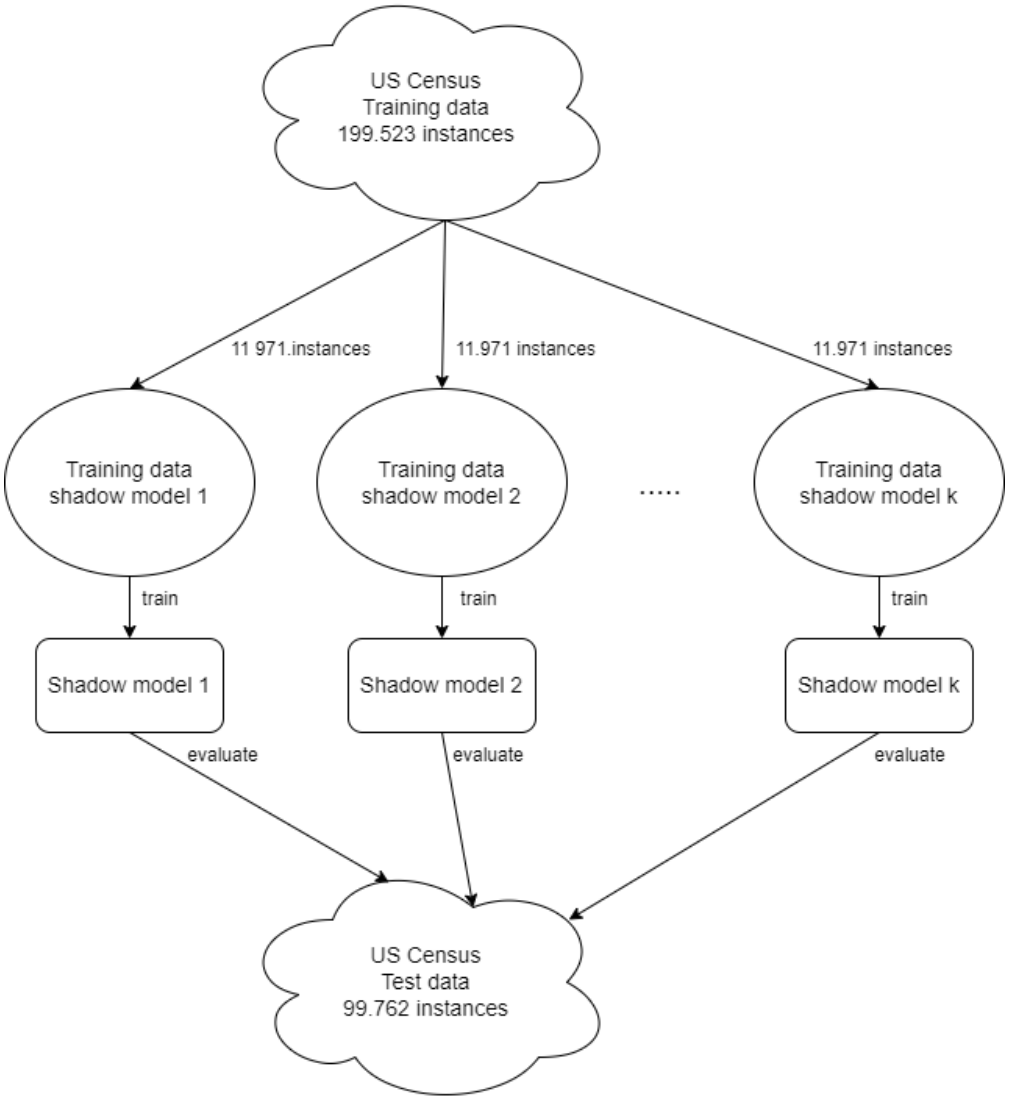


Figure 3.3: Split of data in the first approach

For the process of training and evaluating the meta-model the shadow models are split into training and target models. Each class contributes 2,048 shadow models for training the meta-model and 256 models for evaluation, with the latter serving as the target model. This process ensures a balanced and well-defined dataset for training the meta-model and accurately evaluating its performance in property inference tasks.

3.3.2 Second data splitting approach

The goal of the second data-splitting approach is to ensure that shadow and target models do not have access to overlapping data, thereby simulating a federated learning environment in which each participant trains on isolated datasets. This arrangement reflects real-world federated scenarios where each participant's data remains separate and unshared with others. To achieve this, the Census US training dataset is partitioned into four distinct parts, as illustrated in Figure 3.4.

Figure 3.4: Split of data in the second approach

In this setting:

- 10 % of the data (19,952 instances each) is reserved for evaluating both shadow and target models
- 30 % of the data (59,856 instances) is designated for training the shadow models, where the gender ratio is intentionally varied. This variation allows for the simulation of different demographic distributions between shadow and target datasets, ensuring that the attacker's shadow models do not always replicate the demographic proportions present in the target model's training data.
- The remaining 50 % of the Census US training data is allocated to the target models.

For consistency, each shadow model is trained on a random sample consisting of 20 % of the shadow training data, resulting in 11,971 training instances per shadow model. The target models are trained with the same number of instances, but the instances are taken exclusively from the data reserved for target model training.

The shadow and target models are evaluated using their respective test sets. As in the first approach, only models that achieve an accuracy above 90 % are selected, ensuring a baseline of model reliability and classification efficacy.

Different ratios for women and men

After removing the test instances, the remaining training dataset features an approximate 52/48 split of women to men, totalling 83,232 female and 76.387 male instances. To assess how variations in demographic ratios impact the performance of property inference attacks, two distinct gender ratios were tested between shadow models and target models, as shown in Table 3.5.

Table 3.5: Different train test splits

Data used for shadow models			**Data used for target models**		
# instances women	**# instances men**	**ratio**	**# instances women**	**# instances men**	**ratio**
29,928	29,928	50/50	53,304	46,459	53/47
41,899	17,956	70/30	41,333	58,431	41/59

The first spit, with nearly identical ratios, tests how the attack performs when there is minimal demographic difference between the attacker's and target model's training data. The similar distribution of women and men represents a federated learning scenario where participants' datasets align closely in the distribution of the exposed property, allowing the attacker to rely on similar patterns in the shadow models to infer properties about the target model.

In the second split, we introduce a more pronounced imbalance in the exposed property. Here, the shadow models are trained on data that contains 70 % women and 30 % men and the target models are trained on data that is 41 % women and 59 % men. This distinct ratio simulates a federated learning environment in which the attacker's shadow data exhibits a notably different structure than the target's training data. This is done to understand how effective property inference attacks remain under realistic federated learning conditions, where each participant's data may vary significantly in the property in question.

These differentiated configurations offer a view of how the distributions of the target property impact the success of the property inference attack. In federated learning, where participants' data often varies due to regional, social or organizational differences, this approach is crucial for the simulation of real-world scenarios. The experiment seeks to answer the research question by examining if and how an attacker can infer meaningful

information about the target model when the proportions in the training data differ. Lastly, it provides information on the reliability of shadow models in approximating target models in federated learning and helps identify conditions under which privacy risks are most pronounced when data distributions are unknown to the attacker but structurally varied.

Variations of second data splitting approach

In order to investigate the robustness and effectiveness of the second data-splitting approach, two further experiments were conducted.

1. **Extended training duration for shadow models:** Here, shadow models were trained for 10 epochs rather than the original single epoch. Again, only models achieving above 90 % accuracy were selected, maintaining the original accuracy threshold. Extended training aims to assess whether prolonged exposure to training data improves the shadow models' capacity to approximate the target model.
2. **Increased data variety for shadow model training:** The second attempt was to increase the data available for training shadow models from 30 % to 45 % of the total dataset. While the number of instances per shadow model remained constant, the expanded data allows for greater variability in the samples used to train each shadow model. This variation was implemented to determine whether increased dataset variety would enhance shadow model generalization and, in turn, the effectiveness of the property inference attack.

These modifications explore critical dimensions in the federated learning context, providing insights into how training duration and dataset diversity impact the effectiveness and precision of property inference attacks when the property in question differs across participants.

3.4 Hierarchical approach

In order to improve the results obtained in the first data-splitting approach (Section 3.3.1), a hierarchical approach was tested on the 20-class model. Instead of training a single attack model capable of distinguishing between 20 different classes, this method employs a staged classification process.

An overview of this hierarchical approach is shown in Figure 3.5. For this approach, initially, the 4-class model of the first data-splitting approach is used to categorize the target data into one of the four classes. Depending on the 4-class model's output, the data is then further distributed to one of several subsequent models, named hierarchical models. Each of these hierarchical models is tasked with a more specific classification within its assigned range of values. These hierarchical models then generate the final class predictions.

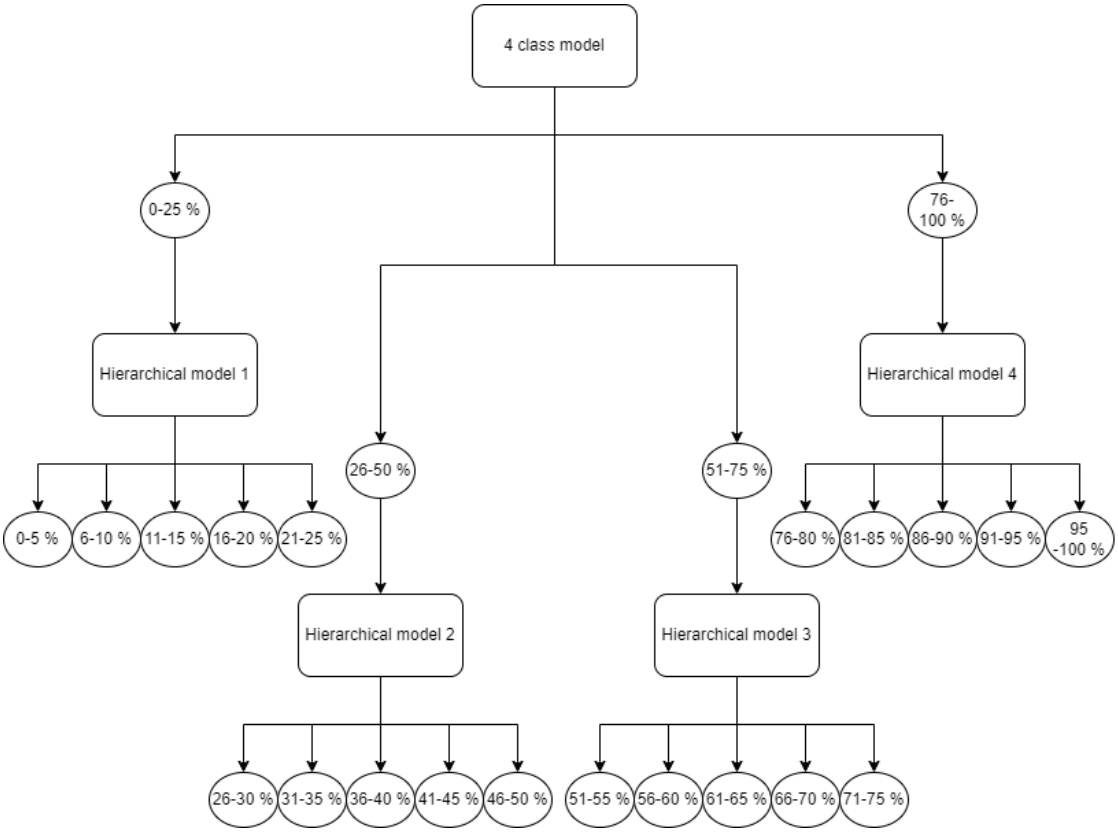


Figure 3.5: Hierarchical approach

Each hierarchical model is designed as a 5-class model that follows the same architecture as the original meta-model. Instead of training with the original 5-class distribution (e.g. 2,304 models per class, containing 0-20 %, 21-40 %, etc.) the hierarchical models are trained on data that is tailored to a specific range. In total 11,520 models were used to train each hierarchical model, distributed as 2,304 models per target class. The corresponding target classes for the different hierarchical models can be seen in Table 3.6.

Table 3.6: Data used for training the four hierarchical models

H. model 1	**H. model 2**	**H. model 3**	**H. model 4**
0-5 % of women	26-30 % of women	51-55 % of women	76-80 % of women
6-10 % of women	31-35 % of women	56-60 % of women	81-85 % of women
11-15 % of women	36-40 % of women	61-65 % of women	86-90 % of women
16-20 % of women	41-45 % of women	66-70 % of women	91-95 % of women
21-25 % of women	46-50 % of women	71-75 % of women	96-100 % of women

To train the models of the hierarchical approach five different loss methods were applied, consistent with those used in the other data-splitting approaches. For the evaluation of

this approach, a random test set is generated. By isolating the predictions within distinct ranges, this hierarchical structure aims to improve classification accuracy and reduce the errors inherent in a single 20-class model, which can improve the effectiveness of the overall property inference attack.

3.4.1 Test data used to evaluate the hierarchical approach

This test dataset used to evaluate the hierarchical approach consists of 2,304 models. These models share the same architecture and training as the shadow models from the first data-splitting approach (Section 3.3.1), with one distinction: the label for each model is the actual percentage of women used in its training data. Furthermore, these 2,304 models are generated using a random percentage of women ranging from 0 to 100 %. As a result the distribution into the 20 different classes is not uniform, reflecting the variability introduced by randomization.

This dataset is only used to evaluate the hierarchical approach, enabling assessment of its performance across different distributions. This setup ensures that the hierarchical models can be accurately evaluated on data that mirrors a realistic, non-uniform distribution. More information regarding the evaluation methods used is provided in the following section.

3.5 Performance evaluation

To evaluate the efficiency of the meta-model in performing property inference attacks, two different key metrics are used: accuracy and MSE. Each metric provides distinct insights into the model's performance, particularly when dealing with an ordinal classification problem.

1. **Accuracy:** Accuracy, as used in the work of [GWY+18], measures the percentage of correct predictions made by the meta-model. This metric allows for a direct comparison between the results of this thesis and previous studies, ensuring consistency and comparability in evaluating the property inference attack. While accuracy serves as a useful basis for evaluating model performance, it does not capture the nuance of ordinal misclassification, where the distance between predicted and actual classes can significantly affect the practical utility of the model.

2. **Mean Squared Error (MSE)**: To address the limitations of accuracy, MSE is also used. MSE evaluates the degree of error in the predictions by calculating the squared differences between the predicted and true class labels. This metric accounts not only for the occurrence of misclassification but also for the magnitude of each error, which is particularly valuable in ordinal classification tasks. For example, if a model incorrectly classifies a target in class 1 as class 2, this error is less severe than misclassifying it as class 5. The MSE metric captures this gradation

in error severity, making it a more meaningful measure for models where order matters.

By using both accuracy and MSE this evaluation approach provides a more comprehensive understanding of the meta-model's performance. While accuracy gives an overview of the model's basic predictive capability, MSE offers a deeper insight into how closely the model's predictions align with the actual ordinal structure of the data. This dual-metric evaluation ensures that the model's effectiveness is assessed in both general and nuanced terms, helping to determine whether the meta-model not only classifies correctly but also respects the ordinal structure.

CHAPTER 4

Results

This chapter presents the results for the various experiments conducted in this thesis to investigate property inference attacks on fully connected neural networks in a multi-class setting. In order to answer the research questions formulated in Section 1.3 the results of the different attack variants and loss functions are presented and analysed in detail in this chapter. This includes the results of extending the binary attack to a multi-class attack, the impact of the hierarchical approach to improve attack effectiveness and the findings from the experiments in sequential federated learning. The performance of the attacks is evaluated using metrics such as accuracy and MSE and visualized using confusion matrices to provide a comprehensive understanding of the capabilities and limitations of property inference attacks.

4.1 Multi-class attack

To answer the first research question, the attack proposed in [GWY+18] was extended to handle the multi-class case. To be specific, this work develops the attack for the 2, 4, 5, 10 and 20 class cases. As mentioned in Section 3.2.3, five different approaches for ordinal classification were implemented to assess the attack's performance across different classification strategies.

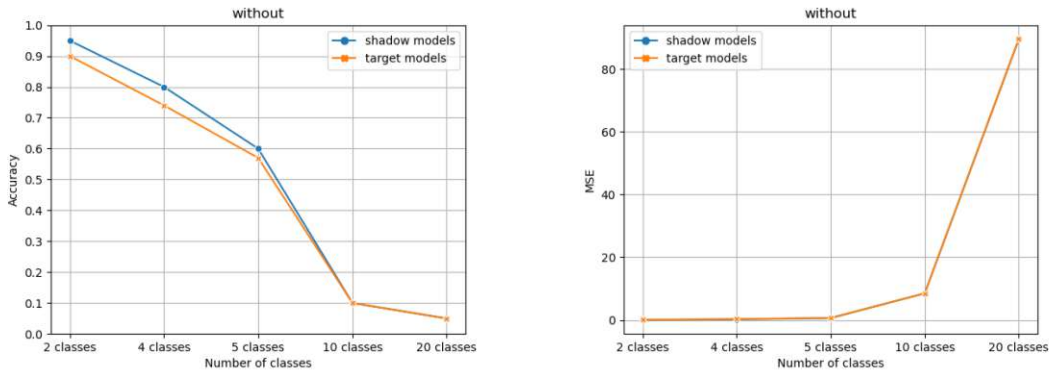
In the results section, confusion matrices are used to present the outcomes for each approach, where the labels refer to percentage groups of women used to train the corresponding shadow or target models. An overview of these percentage groups for the different class configurations is provided in Table 4.1. This setup ensures a clear representation of how accurately each model can differentiate between groups with varying gender compositions, offering a deeper understanding of the model's effectiveness in the multi-class property inference setting.

Table 4.1: Percentage ranges of women for the different class cases

2 classes	4 classes	5 classes	10 classes		20 classes	
0-50 %	0-25 %	0-20 %	0-10 %	51-60 %	0-5 %	51-55 %
51-100 %	26-50 %	21-40 %	11-20 %	61-70 %	6-10 %	56-60 %
	51-75 %	41-60 %	21-30 %	71-80 %	11-15 %	61-65 %
	76-100 %	61-80 %	31-40 %	81-90 %	16-20 %	66-70 %
		81-100 %	41-50 %	91-100 %	21-25 %	71-75 %
					26-30 %	81-85 %
					31-35 %	76-80 %
					36-40 %	86-90 %
					41-45 %	91-95 %
					46-50 %	96-100 %

4.1.1 Approach without ordinal classification

The first analysis focuses on the approach without ordinal classification. The performance metrics - accuracy and MSE - for the five different class configurations (2, 4, 5, 10 and 20 classes) can be seen in Figure 4.1.



(a) Accuracy of approach without ordinal classification

(b) MSE of approach without ordinal classification

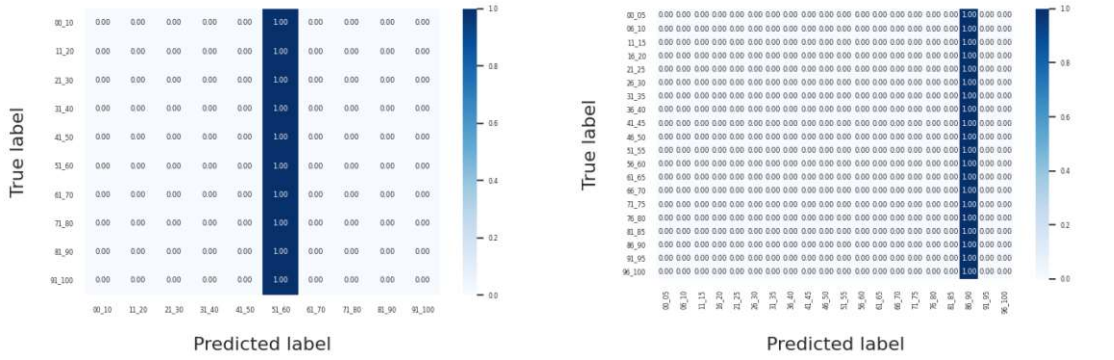
Figure 4.1: Accuracy and MSE for approach without ordinal classification

In Figure 4.1a the accuracy of the two-class case is notably high, achieving approximately 90 % for the target models and 95 % for the shadow models. As the number of classes increases, a linear decline in accuracy is observed for the 4- and 5-class cases, with accuracies of around 80 % and 60 %, respectively. However, as the number of classes reaches 10 and 20, accuracy declines sharply to just 10 % and 5 %, highlighting the limitations of this approach in handling more complex multi-class scenarios.

A similar trend is evident in the MSE that is shown in Figure 4.1b. For the 2-, 4- and 5-class cases MSE values remain low, indicating that predictions are relatively accurate

in these configurations. However, for the 10-class case, the MSE rises to around 10 and for the 20-class case even higher to around 90. This increase indicates that, as class granularity increases, the model struggles to predict accurate classes, resulting in a significant deviation from the true values.

Further insights into this performance are provided by the confusion matrices for the 10- and 20-class cases that can be seen in Figure 4.2. In both cases, it is clear that in these cases, this approach only predicts one class: the attack model for the 10-class case consistently predicts a class containing 51-60 % women and the model for the 20-class case only predicts the 86-90 % women class. This pattern indicates that without ordinal classification, the attack model lacks the ability to differentiate effectively when there is a large number of classes.



(a) Confusion matrix for target models in 10 class case.

(b) Confusion matrix for target models in 20 class case

Figure 4.2: Confusion matrices for 10 and 20 class case for the approach without

This issue starts to appear in the 5-class case, which can be seen in Figure 4.3. Here, however, it does not yet have such a strong effect on accuracy and MSE. In the 5-class scenario, the class representing 21-40 % women is no longer predicted by the attack model. According to the confusion matrix, this class is mostly misclassified as the 41- 60 % women class. For edge classes, that is, classes containing 0-20 % women and 81-100 % women, the model correctly classifies more than 80 % of the target models, suggesting that the attack model can accurately identify edge classes, but struggles with classes in the middle of the spectrum.

In contrast to the 5-class case, in the 4-class setting, all classes are predicted, yet a similar trend emerges: the attack model performs better for the edge classes, achieving around 85 % accuracy compared to approximately 60 % accuracy for the middle classes. This suggests that the model more accurately identifies target data with extreme compositions (e.g., very high or very low percentages of women) but struggles to distinguish between the middle groups that have a more similar feature distribution.

Overall, the findings indicate that the approach without ordinal classification struggles as the number of classes increases. This limitation becomes most pronounced in con-

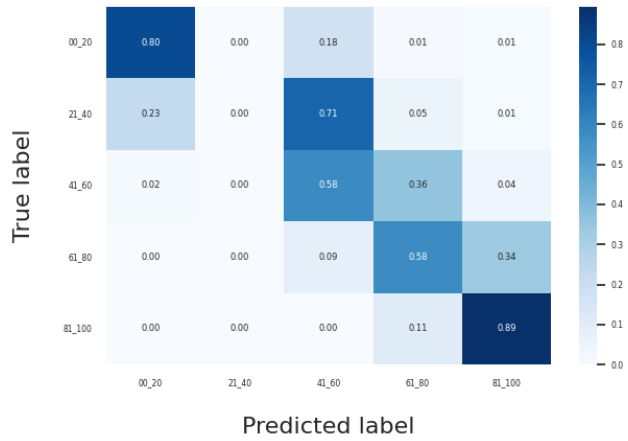


Figure 4.3: Confusion matrix for target models in 5 class case for approach without ordinal classification

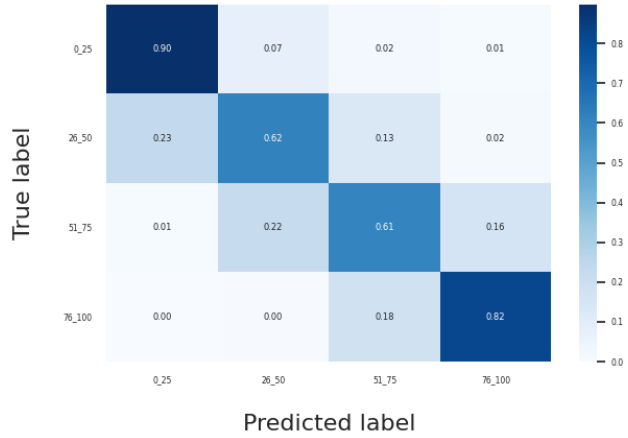


Figure 4.4: Confusion matrix for target models in 4 class case for approach without ordinal classification

figurations with higher class counts, where the model's accuracy and MSE deteriorate significantly. These findings highlight the importance of incorporating ordinal structure in scenarios with a large number of output classes, where the relationships between classes play a crucial role in improving predictive accuracy and minimizing error.

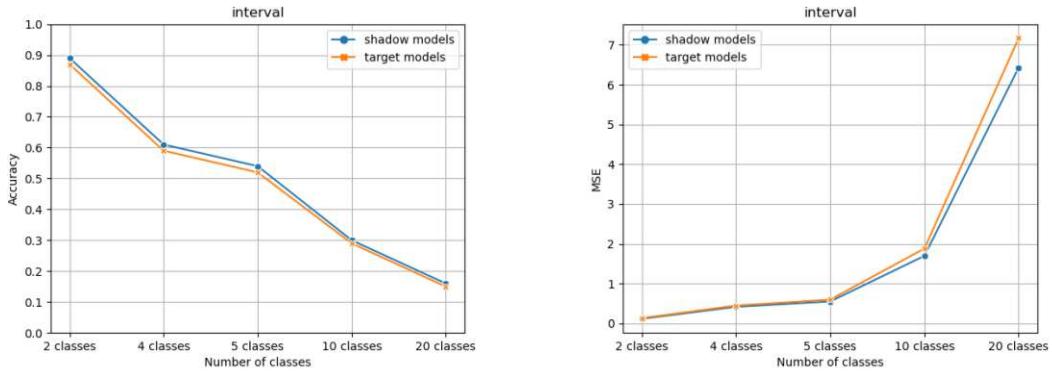
4.1.2 Interval approach

This section analyzes the results of the interval approach, evaluating the accuracy and MSE. Again, the 2, 4, 5, 10 and 20 class cases are examined; they are shown in Figure 4.5.

In Figure 4.5a, the accuracy trend shows a relatively steady decline as the number of classes increases. The 2-class case achieves an accuracy close to 90 %, which then drops to about 60 % for the 4-class case. Interestingly, there is only a slight decline in accuracy to just above 50 % for the 5-class case, demonstrating relatively stable performance despite the increase in complexity. However, in the 10-class case, accuracy decreases further to around 30 %, and in the 20-class case, accuracy falls to approximately 17 %. Throughout all class cases, accuracy remains consistent between shadow and target models.

Figure 4.5b shows the results of MSE. Here, it can be seen that the difference in the MSE between the shadow and target models becomes more pronounced as the number of classes increases. In the 2, 4 and 5 class cases, nearly no difference is visible, while for the 10-class case, a slight difference begins to emerge. This trend is even more apparent in the 20 class case, where the MSE difference between shadow and targe models is clearly visible.

Additionally, the MSE for the first three cases remains below 1, but it increases to around 2 for the 10-class case and rises further to approximately 7 for the 20-class case, indicating growing prediction errors with higher class counts.



(a) Accuracy of interval approach

(b) MSE of interval approach

Figure 4.5: Accuracy and MSE for interval approach

The confusion matrix for the 20 class case in Figure 4.6 reveals potential causes for lower accuracy. Namely, the edge classes - representing 0-5 % and 96-100 % women - are not predicted by the model. Also, the figure shows that many models near these edges are also misclassified. In the middle classes, however, the model generally predicts within a range of approximately seven neighbouring classes, which suggests that the ordinal nature of the interval approach is functioning as intended, with most predictions relatively close to the true class.

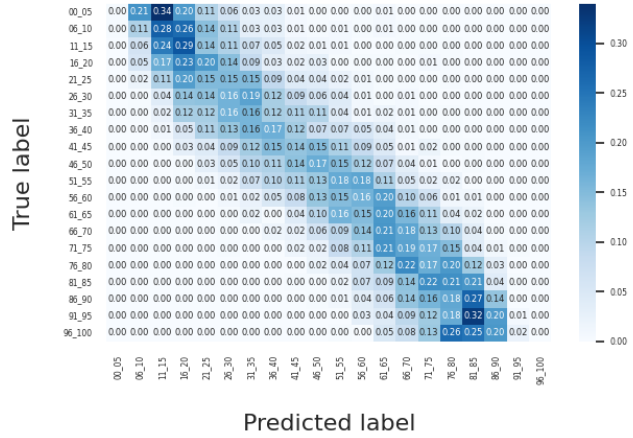
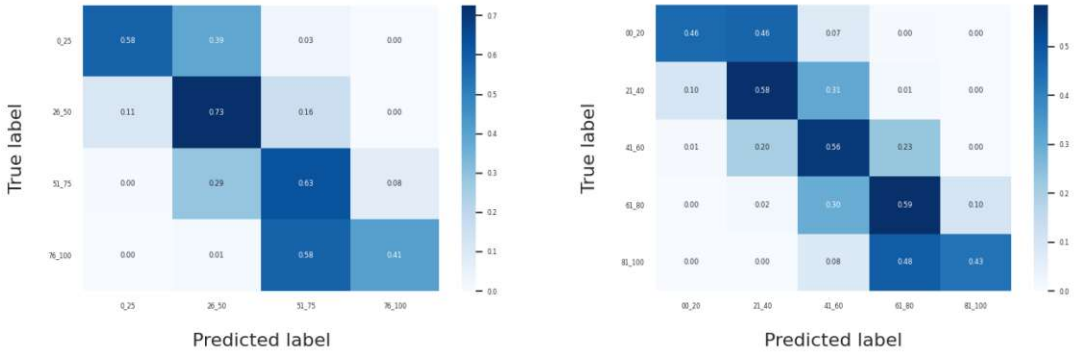


Figure 4.6: Confusion matrix for target models in 20 class case for interval approach

Figure 4.7 shows the confusion matrices of the interval approach for the 4 and 5 class cases. In both, one can see that the attack model predicts the groups quite well. A clear diagonal is recognizable, which indicates that most predictions are correctly classified or assigned to adjacent classes. Overall, it is quite interesting to see that the middle classes have a higher rate of correct predictions than the edge classes, which contrasts with the performance patterns seen in the other loss methods.



(a) Confusion matrix for target models in 4 class case.

(b) Confusion matrix for target models in 5 class case

Figure 4.7: Confusion matrices for 4 and 5 class case using the interval approach

In conclusion, the interval approach performs well across lower class counts, maintaining higher accuracy and lower MSE. However, as the number of classes increases, the model's performance decreases, especially for the edge classes in larger class configurations. This approach appears to be effective in capturing ordinal relationships in smaller to medium class structures but struggles with highly granular class distinctions without additional support for differentiating extreme values.

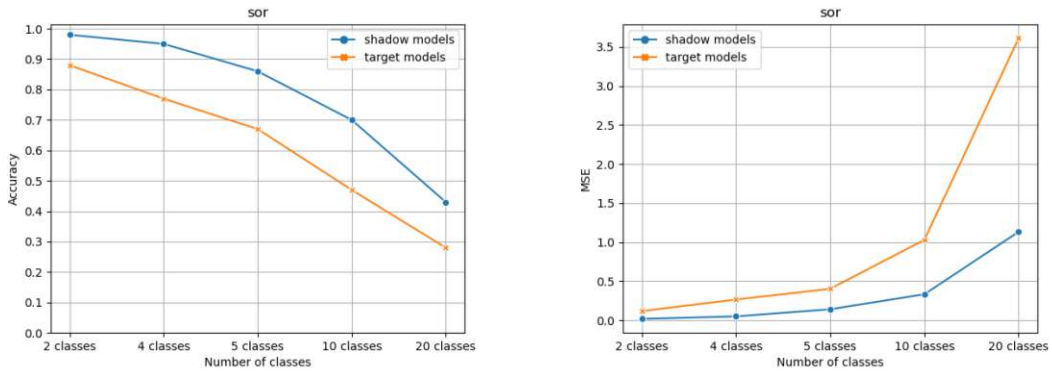
4.1.3 SOR approach

This section presents an analysis of the results of the SOR approach. The results for accuracy and MSE can be seen in Figure 4.8. As in previous approaches, we evaluate the 2, 4, 5, 10 and 20 class cases.

In Figure 4.8a, the accuracy trend shows a concave, non-linear decline rather than a linear trend seen in the interval approach Section 4.1.2. In addition, the accuracies of the shadow models and the target models show a notable difference, which is in contrast to the approaches analysed previously, suggesting possible over-fitting of the attack model on the shadow data.

The SOR approach reaches an impressive accuracy of nearly 100 % for the shadow models in the 2-class case. In the 4-class case, the accuracy remains high, only slightly decreasing to around 95 %. In the 5-class case, accuracy decreases further to around 87 %, dropping to 70 % and 42 % in the 10 and 20-class cases, respectively. In all cases, the target models consistently achieve accuracy rates 10-20 % lower than those for the shadow models.

The MSE results for the SOR approach, shown inFigure 4.8b, reveal a gap between shadow and target models that is growing with an increasing number of classes. In the 2-class case, both shadow and target model achieve a MSE below 0.2. In the 4-class case, the increase in MSE for the shadow models remains relatively low, whereas the one for the target models rises to around 0.25. In the 5-class case, the MSE of the shadow models remain below 0.2, while for the target models, it raised to slightly under 0.5. The 10-class case shows further divergences with an MSE of 0.4 for the shadow models, while the target models reach 1. The biggest change is recognizable in the 20-class case, with an MSE of 1.1. for the shadow models, whereas it climbs to 3.6 for the target models.



(a) Accuracy of SOR approach

(b) MSE of SOR approach

Figure 4.8: Accuracy and MSE for SOR approach

Due to the low MSE achieved by the target models in the 4-class case, further insights are provided by the confusion matrix shown in Figure 4.9. Here a strong diagonal pattern indicates that a lot of models are correctly classified: 84 % of the edge classes are correctly

classified, while for the two middle classes, approximately 70 % of models are correctly classified, indicating reliable performance in intermediate classes as well.

It is also recognizable in Figure 4.9 that most misclassifications occur between neighbouring classes, emphasizing that errors are minor and generally involve predictions that are close to the true class. Only in three cases, this is not the case:

1. 2 % of the models in the 0-25 % women class are misclassified into the 51-75 % women range.
2. 1 % of models from the 25 % women class are misclassified as being in the 76-100 % women range.
3. 1 % of models in the 26-50 % women class are classified into the 76-100 % women class.

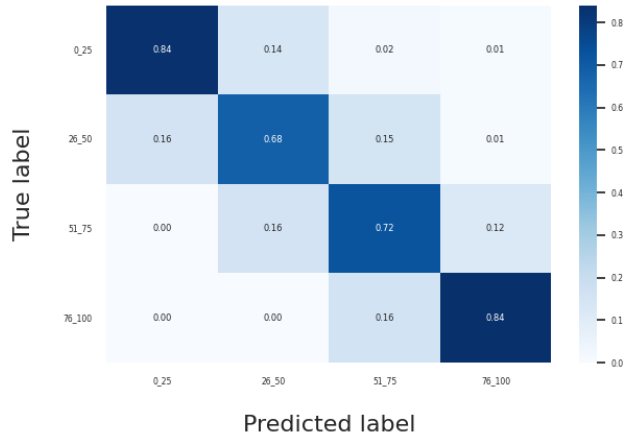


Figure 4.9: Confusion matrix for target models in 4 class case for SOR approach

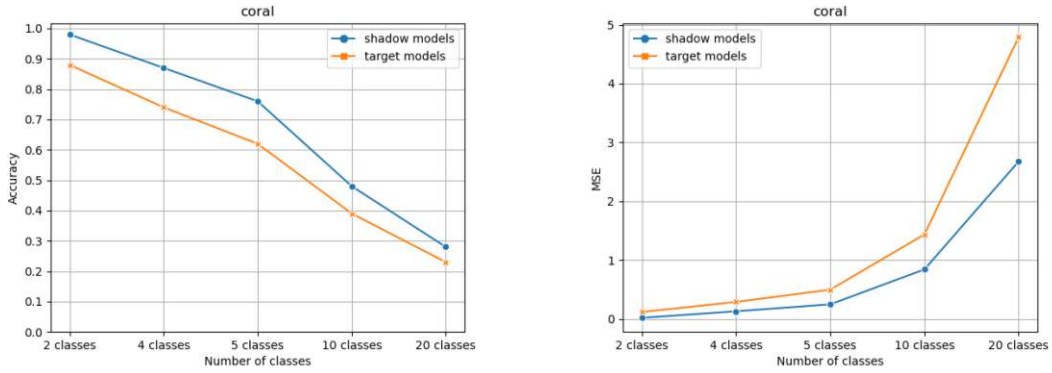
These results indicate that while the SOR approach performs well for low to moderate class cases, achieving high accuracy and low MSE, the model begins to struggle with finer-grained classification in the 10- and 20-class cases, particularly for target models. This trend suggests that the SOR approach is well-suited to maintaining ordinal structure in smaller class configurations but encounters limitations with higher class counts, where overfitting may play a role in the accuracy gap between shadow and target models.

4.1.4 CORAL approach

The results of accuracy and MSE for the CORAL approach can be seen in Figure 4.10a. Similar to the SOR approach as discussed in Section 4.1.3, the CORAL approach consistently performs better for shadow models than for target models, although the difference in performance is less pronounced.

In terms of accuracy, it can be said that the first three cases (2, 4 and 5 class cases) and the second three cases (5, 10 and 20 class) show a very linear behavior with a slight discontinuity (or kink) in between. The accuracy for the target models starts at around 89 % in the 2-class case gradually decreasing to around 74 % for the 4-class case and to around 61 % in the 5-class case. After the kink, accuracy decreases even faster to approximately 40 % in the 10-class case and to around 22 % in the 20-class case. The accuracy of the shadow models is constantly around 12 % better in the first three cases than the one for the target models. This gap narrows to around 9 % for the 10-class case and further to around 5 % in the 20-class case.

The measure MSE for the CORAL approach can be seen in Figure 4.10b. We can observe that the difference between the MSE of the target models and the shadow models is increasing with the number of classes. In the 20-class case, target models reach an MSE of nearly 5, while it remains around 2.6 for the shadow models. For the 10-class case, the MSE for the shadow models is around 0.9, but approximately 0.6 higher for the target models. In the 2, 4 and 5-class cases, MSE remains lower than 0.5 for both shadow and target models with a difference of less than 0.3 between them.



(a) Accuracy of coral approach

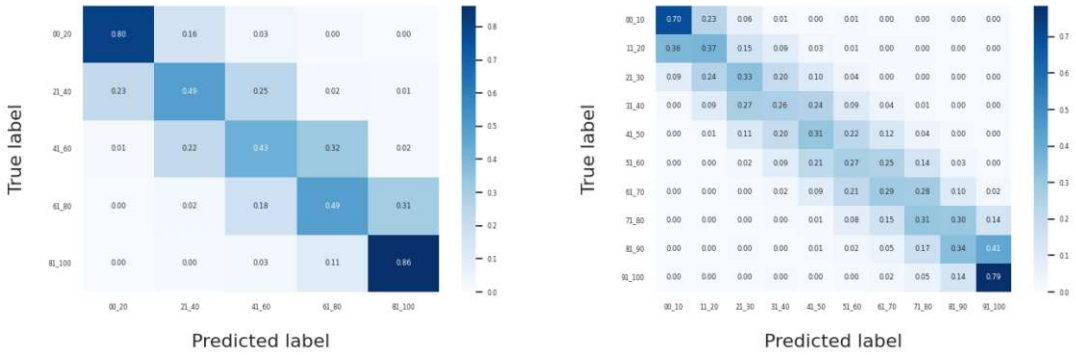
(b) MSE of coral approach

Figure 4.10: Accuracy and MSE for CORAL approach

For further analysis, the confusion matrices of the 5- and 10-class cases can be seen in Figure 4.11. These two confusion matrices aim to analyse the kink seen in Figure 4.10a in more detail. In Figure 4.11a, it can be seen that the edge classes, namely the classes representing 0-20 % women and 81-100 % women, are correctly predicted over 80 % of the time. Middle classes have lower accuracy with predictions in the 40-50 % range.

Misclassifications are typically off by one adjacent class, indicating that the ordinal structure of the CORAL approach is functioning correctly.

After analysing Figure 4.11b, similar patterns persist. Again, the edge classes have a retain a high accuracy of 70-80 %, while in middle classes a decrease in accuracy, with correct prediction rates between 26 % and 37 % can be seen. In difference to the confusion matrix in the 5-class case, wrong predictions are made into two adjacent classes, instead of just one. This explains the steeper drop in accuracy and rise in MSE. Additionally, nearly all classes show outliers where the predicted class differs from the true class by a gap of two classes.



(a) Confusion matrix for target models in 5 class case

(b) Confusion matrix for target models in 10 class case

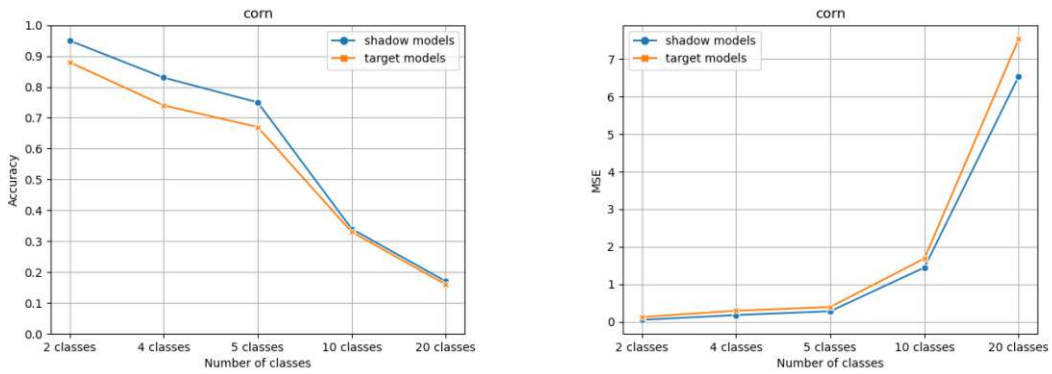
Figure 4.11: Confusion matrices for 5 and 10 class case using the CORAL approach

In summary, the CORAL approach maintains reasonable accuracy for lower class counts and effectively captures ordinal relationships, as evidenced by the concentration of misclassifications within neighbouring classes. However, as the number of classes increases, the model's precision diminishes, with accuracy dropping and MSE rising significantly, especially for target models. The CORAL approach is effective for limited class structures but faces challenges in maintaining accuracy and minimizing MSE with larger, more complex class distributions.

4.1.5 CORN approach

In Figure 4.12a the accuracy plot for the CORN approach is shown. One interesting observation is that the CORN approach shows similar trends to the CORAL approach (see Section 4.1.4) for the first 3 cases (2, 4, and 5 classes). However, for the 10-class case, the CORN approach shows a lower drop in accuracy, achieving around 32 %. Another interesting fact is that the accuracy of the shadow and target models is nearly identical in the 10- and 20-class cases, with the 20-class accuracy dropping even further to around 18 %.

Figure 4.12b shows the MSE results for the CORN approach. Here, the results for the shadow and target models are almost identical in all cases except the 20-class case, where the MSE difference is approximately 1. In all other cases, the MSE of the target models is slightly higher than the one for the shadow models. For the 2, 4 and 5 class cases MSE remains below 0.4, increasing to around 1.5 in the 10-class case and around 7 in the 20-class case. These trends indicate that the CORN approach effectively minimizes errors in lower-class cases but struggles to maintain accuracy in more granular configurations.



(a) Accuracy of corn approach

(b) MSE of corn approach

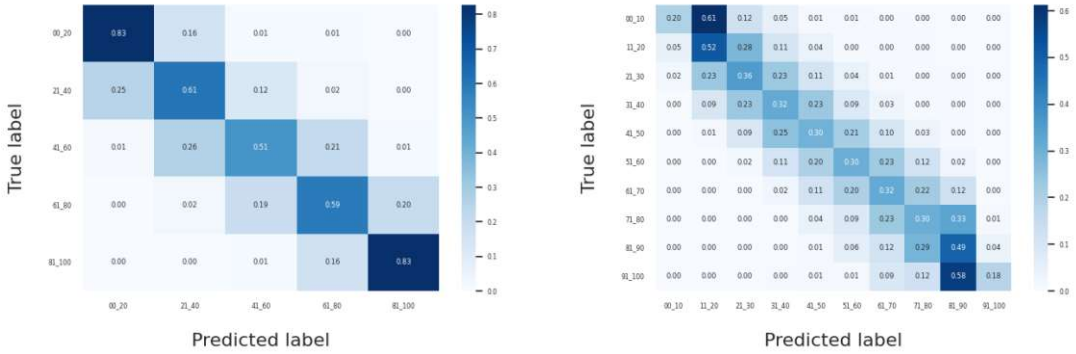
Figure 4.12: Accuracy and MSE for CORN approach

For further analysis, we analyse the 5- and 10-class confusion matrices shown in Figure 4.13. These are the same cases as previously analysed for the CORAL approach.

Figure 4.13a shows a result similar to the same confusion matrix for the CORAL approach (Figure 4.11a). Again, the edge classes, representing 0-20 % and 81-100 % women, achieve a correct classification rate of over 80 %. The accuracy of the middle classes is between 51 % and 61 %, with misclassifications typically involving neighbouring classes. The cases where there is more than one class difference between the correct class and the predicted class occur at a maximum rate of 2 % per class, demonstrating that the CORN approach largely maintains the ordinal structure, resulting in accurate predictions for closely adjacent classes.

In Figure 4.13b the confusion matrix for the 10-class CORN approach can be seen. An

interesting shift is, that unlike the 5-class case, the edge classes are no longer the most accurately predicted. Instead, in both edge classes, the model frequently predicts the neighbouring classes (approximately 60 % of the time). The correct edge classes are predicted only in a maximum of 20 % of cases. The highest accuracy rate was observed by the second class, with 52 %, closely followed by the ninth class with a correct prediction rate of 49 %. The remaining middle classes achieve an accuracy of around 30 %. In almost all classes, the misclassification occurs within two adjacent classes in either direction of the correct class.



(a) Confusion matrix for target models in 5 class case

(b) Confusion matrix for target models in 10 class case

Figure 4.13: Confusion matrices for 5 and 10 class case using the CORN approach

Overall, the CORN approach effectively maintains ordinal relationships in lower class cases, similar to CORAL, but struggles with increased complexity in the 10- and 20-class settings. The approach performs well in predicting classes close to the correct ones but exhibits difficulty in accurately classifying the extremes in higher-class cases. The nearly identical accuracy and MSE scores for shadow and target models in the 10- and 20-class cases suggest that CORN is less vulnerable to overfitting but requires further refinement to improve performance in high-class configurations.

4.1.6 Summary

This section compares the results of the five different approaches used to carry out the attack. To simplify the comparison, accuracy and MSE results for each approach are combined into single plots in Figure 4.14 and Figure 4.15. The plots are based on target data, as this is most relevant for evaluating the success of the property inference attack.

In Figure 4.14, the accuracy of the five approaches across all class cases is displayed. Notably, the SOR approach achieves the highest accuracy result in every class case except the 2-class setting, where all five approaches perform similarly, with accuracies around 90 %.

The worst results for the 2, 4 and 5-class cases are obtained from the interval approach. Especially in the 4-class case, there is a particularly large gap of approximately 15 % compared to the other four approaches, which all achieve around 77 % accuracy. As discussed in Section 4.1.1 the approach without ordinal classification predicts only one class in the 10 and 20-class cases. Therefore, it is not surprising that this approach results in a notably poor performance for these configurations.

In the 5-class case, the accuracies range from 50 % to 70 %. The SOR and the CORN approaches achieve the best accuracy at around 67 %. The CORAL approach shows slightly worse results for this case, with an accuracy of around 62 %. The approach without ordinal classification follows with 57 % while, as already mentioned, the interval approach shows the lowest accuracy at about 52 %.

In the 10-class case, the accuracy is wider spread. The highest result is achieved by the SOR approach with an accuracy of around 47 %. The second best accuracy in this class case is an accuracy of 39 % reached with the CORAL approach. The decrease of accuracy in the CORN approach between the 5 and 10 class cases already described in Section 4.1.5 is also visible here. This results in the third-best accuracy result for the CORN approach with around 33 % of accuracy. The interval approach reaches a result of approximately 29 %, and the approach without ordinal classification remains the lowest, consistent with previous findings in Section 4.1.1.

In the 20-class case the ranking of approaches remains consistent with the 10-class case. The top results are reached by the SOR approach with an accuracy of approximately 28 %. The CORAL approach achieves a slightly lower accuracy of around 23 %, while the CORN and interval approach both reach approximately 15 % accuracy; the approach without ordinal classification performs worst at 5 %.

Figure 4.15 illustrates the the MSE for the five approaches across different class cases. Here, one can see that the MSE for the approach without ordinal classification is the lowest for the 2-class case. The other four approaches achieve fairly similar results.

In the 4-class case, it is recognizable that the worst result is reached with the interval approach while the best results are achieved with the SOR approach. The other three approaches achieve a result similar to that of the SOR approach. The overall MSE is higher than in the 2-class case.

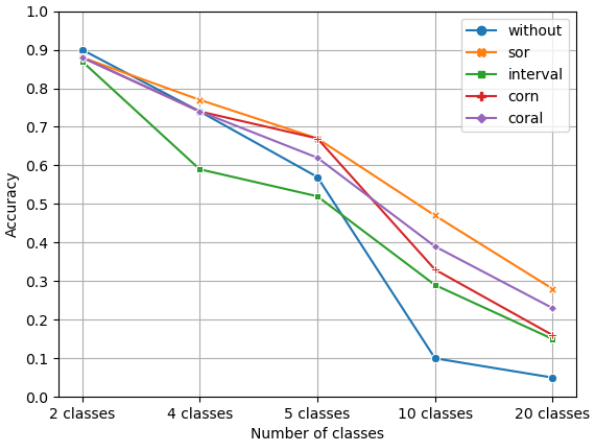


Figure 4.14: Accuracy for all five approaches for the different class cases

A higher increase in MSE is observed between the 4- and 5-class cases compared to the increase between the 2- and 4-class cases. In the 5-class case, the MSE for the interval approach again has the highest MSE, while the CORN approach achieves the lowest.

In the 10-class case, the issues with the approach without ordinal classification that were previously discussed in Section 4.1.1 are evident in the result of the MSE. The MSE is significantly higher than that for the other approaches. The best result is achieved with the SOR, followed by CORAL, CORN and the interval approach.

The 20-class case shows a similar result to the 10-class case, with the interval and the CORN approaches showing nearly identical MSE. Once again the approach without ordinal classification has the highest MSE, while the best result is reached with the SOR approach. For all approaches, the MSE increases in comparison to the 10-class cases.

As expected, across all five approaches performance decreases as the number of classes increases. The 2-class case yields strong results for all approaches, demonstrating that simpler configurations are more manageable for property inference attacks.

The results for the 4- and 5-class cases remain viable, with the SOR approach yielding the highest accuracy at 77 % and a MSE of 0.27 in the 4-class case. These results surpass random guessing significantly. The same can be said for the 5-class cases, even though accuracy is lower at 67 % and the MSE is higher at 0.39 for the best approach.

The 10- and 20-class cases demonstrate the limits of the approaches, especially of those that do not utilize ordinal classification. Although the SOR approach achieves the best performance (47 % and 28 % accuracy, respectively, with MSE values of 1.03 and 3.61), these results indicate that inference accuracy decreases as class complexity increases. This trend underscores the necessity of robust ordinal classification techniques when attempting attacks on models with high-class counts.

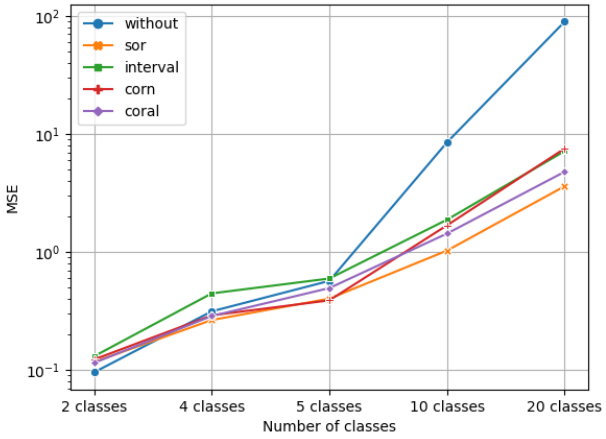


Figure 4.15: MSE for all five approaches for the different class cases (y-axis has a logarithmic scale)

Detailed accuracy and MSE values for all five approaches and class cases can be seen in Table 4.2. These findings underscore the importance of choosing an appropriate approach based on class complexity, with the SOR approach demonstrating clear advantages for more challenging class configurations.

Table 4.2: All results for RQ 1.1

			NLLLoss	sor	interval	corn	coral
2 classes	accuracy	shadow	95%	98%	89%	95%	98%
		target	90%	88%	87%	88%	88%
	mse	shadow	0.051	0.0183	0.1133	0.0479	0.0164
		target	0.0957	0.1172	0.1309	0.123	0.1152
4 classes	accuracy	shadow	80%	95%	61%	83%	87%
		target	74%	77%	59%	74%	74%
	mse	shadow	0.2212	0.0496	0.4158	0.1764	0.1281
		target	0.3145	0.2656	0.4463	0.291	0.2871
5 classes	accuracy	shadow	60%	86%	54%	75%	76%
		target	57%	67%	52%	67%	62%
	mse	shadow	0.617	0.141	0.5506	0.2771	0.2457
		target	0.6547	0.4031	0.5969	0.3898	0.4953
10 classes	accuracy	training	10%	70%	30%	34%	48%
		target	10%	47%	29%	33%	39%
	mse	shadow	8.5	0.3333	1.6965	1.4438	0.8399
		target	8.5	1.0289	1.8793	1.6844	1.4344
20 classes	accuracy	shadow	5%	43%	16%	17%	28%
		target	5%	28%	15%	16%	23%
	mse	shadow	89.5	1.1325	6.4291	6.5396	2.6759
		target	89.5	3.6107	7.1701	7.534	4.7895

4.2 Hierarchical approach

This section presents the results obtained using the hierarchical approach (cf. Section 3.4). The hierarchical approach was tested across the five different loss functions, and the results for each approach are compared in the following sections.

Each subsection includes two confusion matrices to compare the results of the hierarchical approach with those of the 20-class model for the respective loss function. The test data used for these comparisons is described in Section 3.4.1.

In the confusion matrix for the hierarchical approach, four boxed sections are highlighted as a visual aid. Ideally, all prediction rates outside these boxes should have values of 0, indicating that the 4-class meta-model has correctly assigned all test set models to the appropriate hierarchical models. Within each box, the optimal structure would show 100 % correct predictions along the diagonal, with no predictions (0 %) in the off-diagonal cells. Such a pattern would confirm that both the 4-class meta-model and the subsequent hierarchical models are functioning flawlessly.

We conclude this section with an overall summary of the hierarchical approach's performance across all five loss functions and compare them to the non-hierarchical 20-class models.

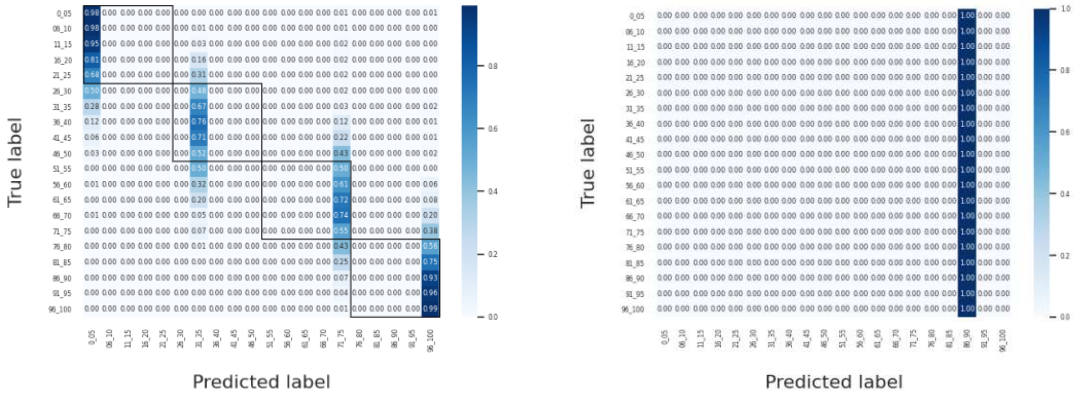
4.2.1 Approach without ordinal classification

Figure 4.16 compares the results of the hierarchical approach with those of the 20-class model for the loss approach without ordinal classification. As observed in Figure 4.16b, and consistent with the findings in Section 4.1.1, the 20-class model fails to differentiate between classes, predicting all models to belong to a single class, namely the class with 86-90 % of women.

In contrast, Figure 4.16a presents the results for the hierarchical approach. Unlike the 20-class model, the hierarchical approach successfully distributes the models across four classes. The performance of the initial 4-class meta-model is particularly noteworthy, as it correctly assigns the majority of models into their respective class boxes. This leads to the conclusion that the 4-class meta-model is capable of effectively distinguishing between the classes.

However, the subsequent 5-class models, designed to provide more granular predictions within each group, do not perform as intended. This is recognizable because each 5-class model predicts only a single class within its respective group. Specifically, the first 5-class model predicts only the first class, the second model predicts only the second class, and both the third and fourth models predict only the fifth class. This limitation highlights a lack of differentiation in the finer-grained predictions.

The hierarchical approach also exhibits a familiar trend: models that have a very low (0-15 %) or very high percentage (86-100 %) of women are correctly classified over 90 % of the time by the 4-class model. In contrast, predictions for models in the middle



(a) Confusion matrix for hierarchical approach

(b) Confusion matrix 20 class model with test data

Figure 4.16: Results of hierarchical approach with approach without ordinal classification

classes (16-85 %) are less accurate. For example, models with a true label of 31-35 % of women are correctly classified into the second class 67 % of the time, but 28 % are misclassified into the first class, 3 % into the third class, and 2 % into the fourth class. This has already been concluded in Figure 4.4. The edge models are well predicted by the 4-class model, whereas the prediction is not quite as good for the middle models.

Despite the inability of the 5-class models, in which the models are fed for further predictions, the hierarchical approach achieves broader categorization by assigning models to four distinct groups. This result demonstrates that the hierarchical structure provides better differentiation than a flat 20-class model, even if its finer-grained predictions are limited. This highlights the hierarchical approach's potential for improving classification outcomes in scenarios where traditional models struggle.

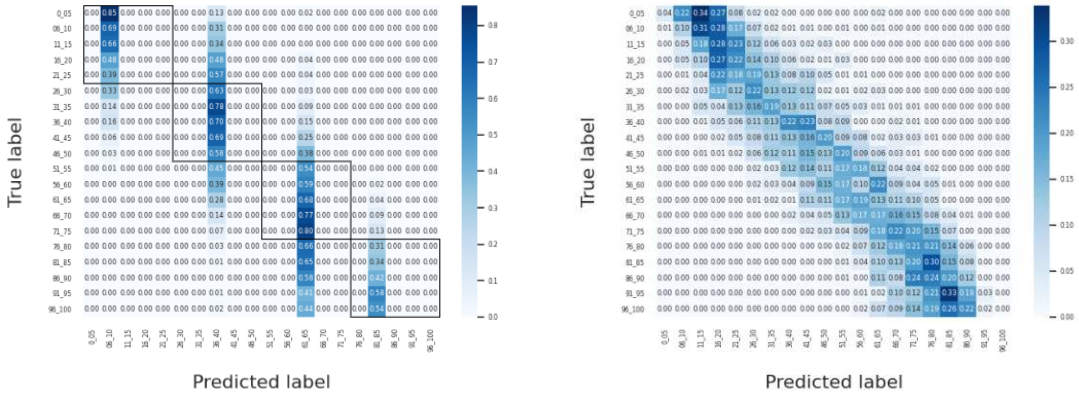
4.2.2 Interval approach

In Figure 4.17 the results of the interval approach can be seen. From Figure 4.17b, a pattern similar to Figure 4.6 is notable. Specifically, the edge classes perform poorly: the last class (96-100 % women) is not predicted at all, while the first class, namely 0-5 % women and the 19th class (91-95 % women) are predicted very infrequently. As in the previous case (see Section 4.1.2), the predictions of the middle classes are distributed in a range of about seven neighbouring classes. This means that the attack model predicts, in addition to the correct class, neighbouring classes of the correct model class. For our cases, this is preferable to distributing predictions over the entire range and indicates that the ordinal classification approach works.

In contrast, the hierarchical approach seems to struggle, as shown in Figure 4.17a. While the initial 4-class model correctly distributes data into the four hierarchical models, the subsequent 5-class models perform poorly. As in the approach without ordinal classification, each of these models only predicts one class. The first and fourth models predict the second class, and the second and third models each predict the third class. This lack of granularity renders the hierarchical approach ineffective for detailed classification.

Additionally, there are many errors in the initial distribution by the 4-class meta-model. For instance, a lot of models that should belong to the first hierarchical model (0-25 % women) are incorrectly assigned to the second model (26-50 % women). The same applies to models that originally belong to the fourth hierarchical model (76-100 %), but are misclassified into the third hierarchical model (51-75 %). This behavior is consistent with the earlier findings in Section 4.1.2, where the 4-class model has a better performance for the middle classes than for edge classes.

The interval approach shows, therefore, a mixed performance. While the ordinal classification framework demonstrates its strength by constraining predictions to a narrow range of neighbouring classes in the 20-class model, the hierarchical approach fails to use this advantage. The inability of the 4-class meta-model to correctly distribute data to the hierarchical models and the subsequent failure of the 5-class models to provide detailed predictions undermines the hierarchical method's overall effectiveness.



(a) Confusion matrix for hierarchical approach

(b) Confusion matrix 20 class model with test data

Figure 4.17: Results of hierarchical approach for interval approach

4.2.3 SOR approach

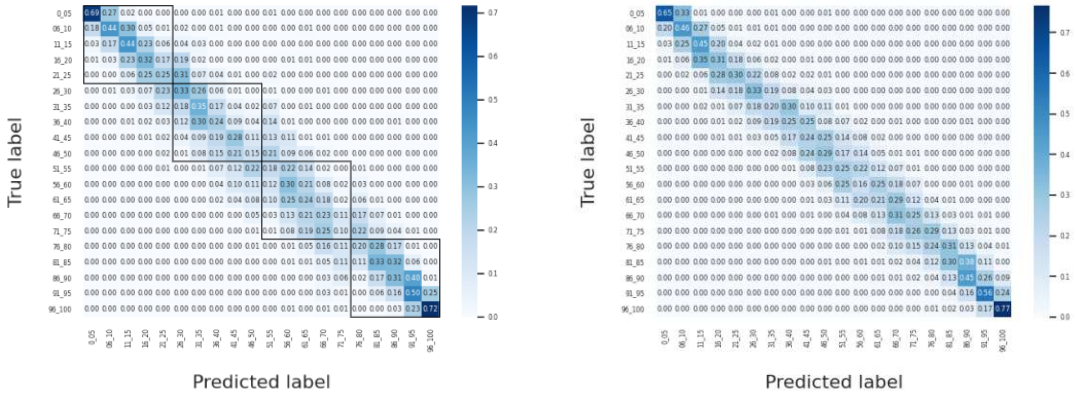
Figure 4.18 shows the results of the hierarchical approach using the SOR loss function compared to the results of the 20-class model. In Figure 4.18b the confusion matrix for the 20-class model can be seen. A clear diagonal is visible with some dispersion around the correct classes. Notably the dispersion is greater in the middle classes compared to the edge classes. For edge classes, the dispersion spans two to three classes, whereas, for the middle classes, it increases to around six classes.

In contrast to the previous results, all classes are predicted in the 20-class model for the SOR method. The highest correct prediction rates are achieved in the edge classes. Specifically, the class with 0-5 % of women has a correct prediction rate of 65 %, while the class with 96-100 % of women reaches an even higher correct prediction rate of 77 %. Conversely, the middle classes achieve lower correct prediction rates, ranging between 20 and 30 %.

The results for the hierarchical approach can be seen in Figure 4.18a. What is noticeable at first glance is that the confusion matrix for the hierarchical approach exhibits a much more uniform distribution compared to previous approaches. As already seen in Figure 4.9, the 4-class model effectively distributes the models into the appropriate hierarchical models. Especially the edge classes maintain high correct prediction rates, similar to the 20-class model. In the middle classes, however, slight misclassifications are noticeable. For instance, the predicted class of 51-55 % women (belonging to the third hierarchical model) includes models that actually belong to the second hierarchical model.

The results for the first and fourth hierarchical models (representing the lowest and highest percentage classes) are very similar to those obtained with the 20-class model, especially for the classes between 0 and 20 % of women and between 86 and 100 % of women. It can also be emphasized here that edge classes have a correct prediction rate of about 70 %, whereas middle classes remain more challenging to predict resulting in correct prediction rates ranging between 10 and 50 %.

The SOR approach shows strong results, with the hierarchical model delivering comparable performance to the 20-class model for edge classes while introducing slightly better uniformity in predictions overall. However, both approaches face challenges with the middle classes, underscoring the need for further optimization in handling finer-grained demographic distributions.



(a) Confusion matrix for hierarchical approach (b) Confusion matrix 20 class model with test data

Figure 4.18: Results of hierarchical approach with SOR approach

4.2.4 CORAL approach

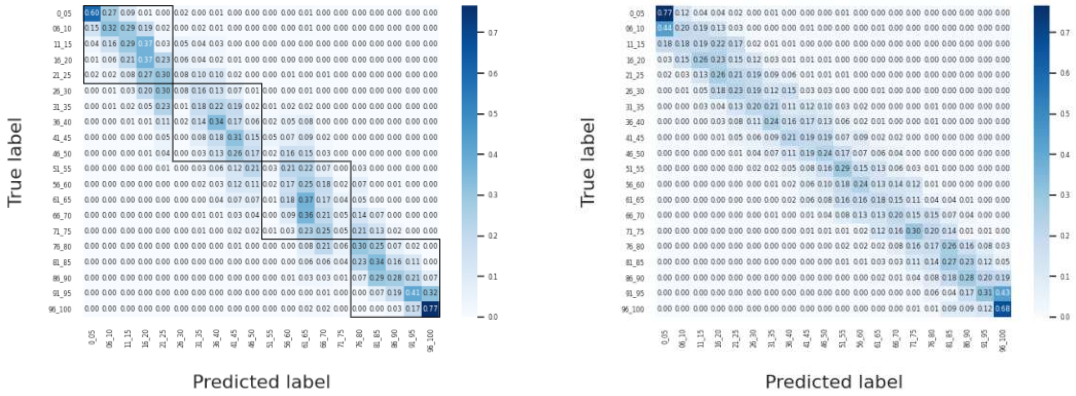
The performance of the CORAL approach is evaluated by comparing the hierarchical model results with those of the 20-class model, as illustrated in Figure 4.19. In Figure 4.19b one can see that the 20-class model preforms relatively well for edge classes. The first class (0-5 % women) achieves a correct prediction rate of 77 %, while the last class (96-100 %) is correctly predicted 68 % of the time.

For the middle classes, a clear diagonal is present in the confusion matrix, indicating that the model is making reasonable predictions for these classes. However, the range of predicted classes is relatively wide, spanning approximately seven classes. This broader range reduces the precision of the model and the correct prediction rates for middle classes are lower, ranging between 16-31 %.

For the hierarchical approach, the results can be seen in Figure 4.19a. A key observation is that the first and fourth hierarchical models perform well, evenly distributing predictions across their five respective classes. Especially for the classes in the first hierarchical model, the correct prediction rate is noticeably higher than that of the 20-class model, improving from approximately 20 % to 30 %. Similarly, the fourth hierarchical model also achieves a consistent distribution across its classes, suggesting that the hierarchical approach effectively handles edge categories.

In contrast, the second and third hierarchical models exhibit less even prediction distributions. In the second hierarchical model (26-50 % women) the first class (26-30 % women) is rarely predicted, and in the third hierarchical model (51-75 % women) the same behaviour is observed for the first (51-55 % of women) and fifth class (71-75 % of women). Despite these issues, the other classes within these models achieve correct prediction rates ranging from 17 to 37 %.

The CORAL approach benefits from the hierarchical model's ability to improve predictions for edge classes, with noticeable gains in correct prediction rates compared to the 20-class model. However, middle classes remain problematic, with uneven prediction distributions and under-represented classes in specific hierarchical models. These results highlight the potential of hierarchical modeling to improve performance in certain scenarios but also underscore the limitations, particularly for middle-class predictions.



(a) Confusion matrix for hierarchical approach (b) Confusion matrix 20 class model with test data

Figure 4.19: Results of hierarchical approach with CORAL approach

4.2.5 CORN approach

This section presents the results for the hierarchical CORN approach and compares them with the 20-class model using the test set for evaluation. The confusion matrices for both methods are displayed in Figure 4.20. From the confusion matrix for the 20-class model (Figure 4.20b), we can notice that neither the first edge class (0-5 % women) nor the last edge class (96-100 % women) is predicted by the model. The second class (6-10 % women) is frequently predicted, while the second to last class (91-95 % women) is rarely predicted.

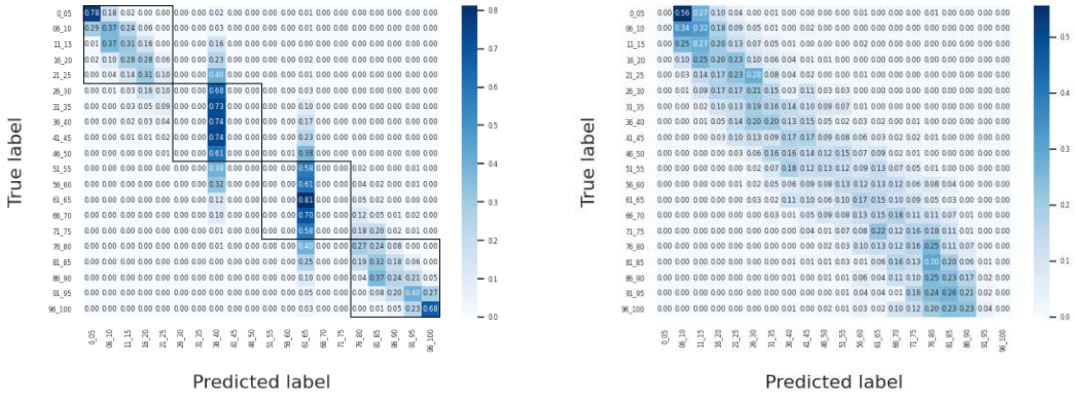
For the middle classes, a wide range of predictions per class can be seen. This is especially the case for models with 26-75 % of women, indicating difficulty in accurately distinguishing these classes. The diagonal of the confusion matrix is slightly shifted downward, revealing a bias where the model is more likely to predict higher percentages of women than are actually present in the data.

Figure 4.20a shows the results for the hierarchical approach. We can observe that the meta-model successfully distributes data into the four classes. When the correct class is not predicted, the misclassification is usually to a neighbouring class. The first hierarchical model performs particularly well, especially for the edge class (0–5 % women), with a correct prediction rate of nearly 80 %. For the remaining classes, the correct prediction rates decrease as they approach the middle, ranging from 28 % to 37 %. Notably, the fifth class (21–25 % women) is rarely predicted.

In the fourth model, all classes are equally predicted. We can also see that the result of the edge class with 96-100 % women is the best with 68 % of correct predictions. The remaining four classes show correct prediction rates between 24 and 40 %, without any clear increasing or decreasing trend.

Similar to the interval approach (Section 4.2.2), the second and third hierarchical models exhibit significant limitations. Both models predict only the middle class, failing to distinguish among the other classes, which reduces their effectiveness.

The hierarchical CORN approach demonstrates both strengths and weaknesses. While it improves prediction rates for edge classes and maintains ordinal consistency through the meta-model, its effectiveness is undermined by the poor performance of the second and third hierarchical models.



(a) Confusion matrix for hierarchical approach (b) Confusion matrix 20 class model with test data

Figure 4.20: Results of hierarchical approach with CORN approach

4.2.6 Summary

The summarized results of the hierarchical approach, compared to the 20-class models, are shown in Figure 4.21. The figure presents the two metrics accuracy and MSE; a higher accuracy and a lower MSE indicate better performance.

In Figure 4.21a one can see that the best accuracy is reached for the SOR loss approach with the 20-class model, which achieves an accuracy of approximately 35 %. Interestingly, the hierarchical approach with the SOR loss method achieves nearly the same accuracy, around 33 %. This means that with this loss method, it makes no big difference if the hierarchical approach is used or the 20-class model.

The second-best result is obtained with the CORAL method, where the hierarchical approach slightly outperforms the 20-class model, with accuracies of around 30 % versus 28 %.

For the CORN method, the hierarchical approach shows a significant advantage. It achieves an accuracy of around 27 %, which is nearly double the accuracy of the 20-class model, which performs at approximately 16 %.

For the interval approach, the 20-class model achieves a better result. The difference between the two approaches is less significant than for the CORN approach. The accuracy of the 20-class model is 15 %, while the accuracy of the hierarchical approach reaches 12 %.

The worst performance is seen with the approach that does not use ordinal classification. In this case, the hierarchical approach achieves significantly better results achieving an accuracy of 15 % compared to only 5 % for the 20-class model. This is explained by the fact that the 20-class model predicts only one class, whereas the hierarchical approach has the ability to divide predictions into four distinct classes.

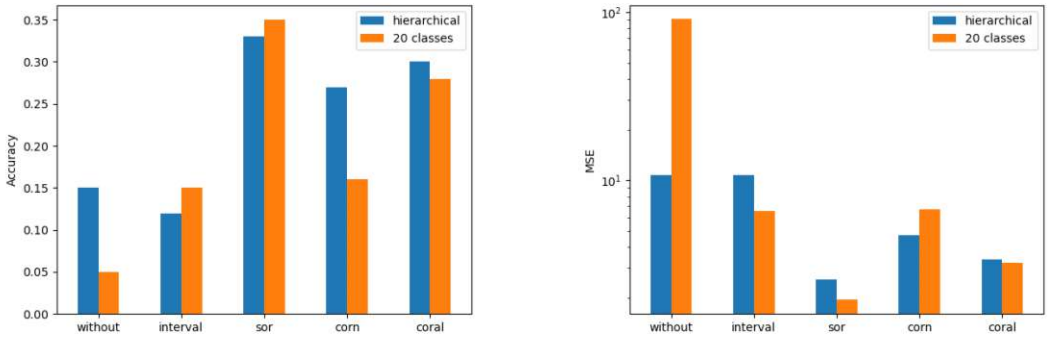
Figure 4.21b shows the evaluation results based on MSE. Consistent with the accuracy results, the SOR approach achieves the best results, with a MSE of 1.9 for the 20-class model and 2.6 for the hierarchical approach.

The CORAL method shows a very similar MSE for both approaches. More precisely, the difference between both approaches is 0.02. The MSE of the hierarchical approach is 3.4, while the MSE for the 20-class model is 3.2.

As already seen during the analysis of the accuracy plot, the CORN method reaches the third-best results. The MSE for the hierarchical approach is 4.7. The MSE for the 20-class model stands at 6.7.

For the interval method, the better result is reached by the 20-class model. This is the same behaviour as already seen for the accuracy. The MSE for the 20-class model is 6.6, while for the hierarchical approach, it is even higher at 10.7.

Again, the worst MSE is observed in the approach without ordinal classification. The hierarchical approach achieves a MSE of 10.7, which is comparable to the interval



(a) Accuracy for different methods for hierarchical approach and 20 classes model

(b) MSE for different methods for hierarchical approach and 20 classes model, y-axis is log scaled

Figure 4.21: Summarized results of the hierarchical approach

approach. Since the 20-class model only predicts a single class, the MSE is exceptionally high at 90.8.

In summary, it can be said that the SOR loss method demonstrates the best overall performance across both evaluation metrics. This is followed by the CORAL method, the CORN method, and the interval method. The approach without ordinal classification yields the worst result.

When comparing the hierarchical approach with the 20-class model, the latter performs better for the SOR and interval methods. Conversely, the hierarchical approach outperforms the 20-class model for the CORN method and the approach without ordinal classification. The results for the CORAL method vary depending on the evaluation metric: the hierarchical approach yields better accuracy, while the 20-class model achieves slightly lower MSE.

4.3 Sequential federated learning

This section summarizes the results of the experiments with sequential federated learning. These experiments were conducted using both the 2-class and 4-class models. As mentioned in Section 3.3.2, this experiment was carried out for two different ratios of men/women in the data set used to randomly select training instances for the shadow models.

In the baseline experiments, the models had varying degrees of success in accurately classifying the data depending on class size and demographic ratio. The results provide an initial indication of how sequential federated learning affects the performance of property inference attacks.

To improve these results, two distinct modifications to the baseline approach were tested. First, the shadow models were trained for 10 epochs instead of a single epoch. This extension aimed to improve the generalization capability of the shadow models and reduce the potential noise introduced by under-trained models. Second, the dataset size used for randomly selecting training instances for the shadow models was increased. By expanding the variety of data available for training, this adjustment was made in an attempt to create more robust shadow models with greater representation of variability.

4.3.1 50 / 50 split of women/men

The results of the experiments using a 50 / 50 split of women and men can be seen in Table 4.3. These experiments were conducted for the five different loss methods and the evaluation focused on the 2-class and 4- class cases.

To begin, the analysis focuses on the results from the original experimental setting, where shadow models were trained for only one epoch. This baseline provides insights into the performance of property inference attacks under controlled and evenly distributed demographic conditions. The accuracy and MSE results reflect the models' ability to infer the correct property classes when the demographic split between men and women is balanced.

Original experiment

The results of the original experiment, using a 50/50 split of women and men and training shadow models for one epoch, reveal notable differences in performance across the five loss methods.

For the 2-class case, the accuracy results demonstrate strong performance across all methods. The best accuracy, 87 % is achieved using the CORAL approach and the approach without ordinal classification. The other methods fall slightly behind, with the interval approach achieving the lowest accuracy at 83 %. The CORN approach and the SOR approach perform similarly and reach an accuracy of 85 %.

The MSE provides more insights. Here, one can see that the CORAL approach yields a slightly better result (0,1270) than the approach without ordinal classification (0.1328). The interval method again lags behind, with the highest MSE at 0.1660. The other two approaches fall in between with MSE values of 0,1465 for the SOR approach and 0.1543 for the CORN approach.

The accuracy of the 4-class model is more varied ranging between 48 % and 64 %. In contrast to the 2-class model, the best result is achieved by the SOR and CORAL approaches, both reaching 64 % accuracy. The approach without ordinal classification follows closely with an accuracy of 62 %, while the CORN approach achieves 60 %. The interval method again delivers the lowest accuracy at only 48 %.

In terms of MSE, the SOR approach stands out with the lowest value of 0.3896, making it the most effective method for this class case. The approach without ordinal classification

performs slightly worse, with a MSE of 0.4316. The CORAL and CORN methods achieve middling results, while the interval approach performs the worst with a MSE of 0.5928.

These results highlight key trends: while the CORAL approach and the approach without ordinal classification excel in simpler (2-class) cases, the SOR approach shows its strength in more complex (4-class) scenarios. The interval method consistently underperforms, suggesting it may not be well-suited for property inference attacks in this setting. Overall, the experiment confirms that both accuracy and MSE deteriorate as the number of classes increases, underscoring the greater challenge of multi-class property inference attacks.

Increased training epochs for the shadow models

After increasing the number of training epochs from one epoch to ten epochs, the results unexpectedly worsened. This contradicts the original assumption that training shadow models for more epochs would improve the attack's effectiveness. These results are again shown in Table 4.3.

For the 2-class case, the accuracy decreased by an average of 15 %, ranging from 62 % for the interval approach to 76 % for the approach without ordinal classification. The remaining three approaches - CORN, CORAL and SOR - reach an accuracy within this range.

The same result can be seen for the MSE values, showing a noticeable increase. The approach without ordinal classification achieved the lowest MSE, 0.2422, while the interval approach reached the highest MSE, 0.3789. Among the remaining methods, the CORN approach performed relatively well, with a MSE of 0.2637, followed by the SOR approach with 0.2930. The CORAL approach performed slightly worse, reaching a MSE of 0.3379.

A similar result is observable in the 4-class model. Here the accuracy also declines by an average of 15 %, ranging from 36 % to 48 %. The SOR approach achieves the best accuracy (48 %) and is followed by the CORAL approach (47 %). The CORN approach reached 45 % and the approach without ordinal classification attained 44 %. As with the 2-class case, the interval approach performed the worst, with an accuracy of 36 %.

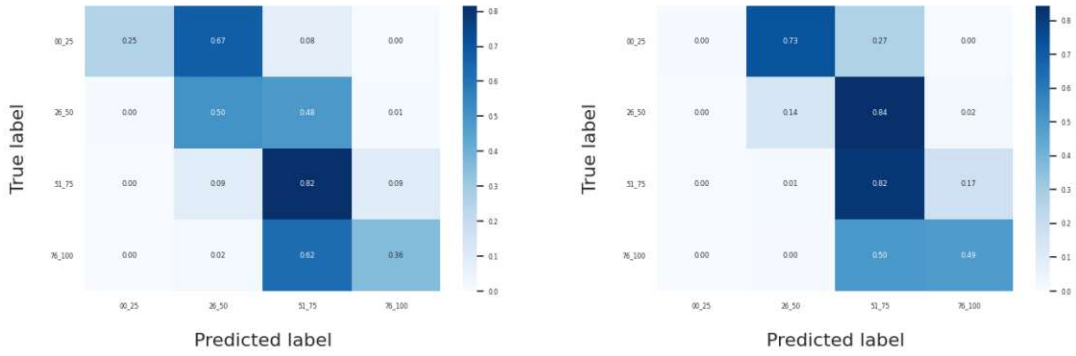
The MSE values for the 4-class model also worsened, increasing across all approaches. The SOR approach once again yielded the best result, with a MSE of 0.5908, followed by the CORAL approach (0.5986). The approach without ordinal classification was slightly worse, with a MSE of 0.6309, and the CORN trailed behind with 0.6768. The MSE of the interval approach is much higher with 0.8564.

Figure 4.22 and Figure 4.23 present the confusion matrices for the best (SOR) and the worst (interval) performing loss approaches in the 4-class case.

In Figure 4.22a, which depicts the interval approach with one training epoch, all classes are predicted. The first class (0-25 % women) achieves a correct prediction rate of only 25 %, while 67 % of these models are misclassified into the second class. Similarly, only 50 % of the second class (26-50 % women) are predicted correctly, with most of the remaining

models assigned to the third class. The third class has a higher correct prediction rate of 82 %. For the fourth class (76-100 % women) 62 % are wrongly classified to belong to the third class and 36 % of the models correctly classified.

Figure 4.22b (interval approach with 10 training epochs) reveals a worse performance. The first class, namely the one containing 0-25 % of women, is not predicted at all with most of the models misclassified into the second or third class. Similarly, 84 % of the models in the second class are misclassified as belonging to the third class. The third class shows a high correct prediction rate of 82 %, while only 50 % of the fourth class models are correctly classified. The other 50 % are misclassified as belonging to the third class. In general, the confusion matrix indicates that the model will most likely predict the third class, which explains the poor result in accuracy and MSE.



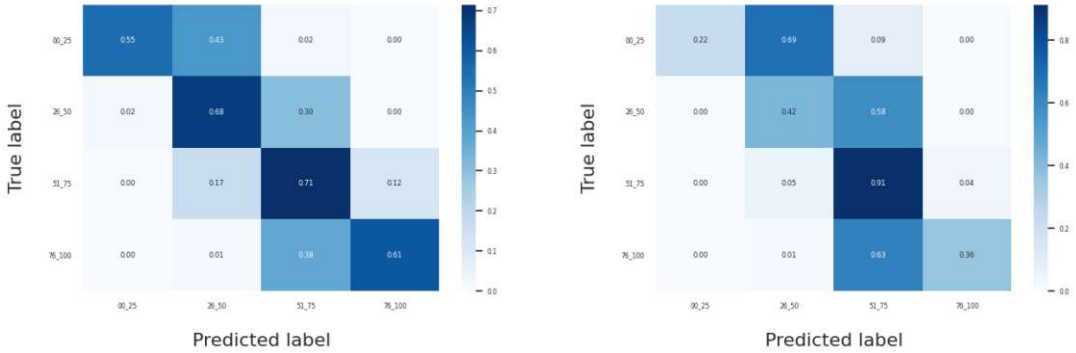
(a) 4-class model with interval approach - models trained for 1 epoch

(b) 4-class model with interval approach - models trained for 10 epochs

Figure 4.22: Results of 50/50 women/men ration - 4 class model for the interval approach

For the SOR approach, Figure 4.23a shows a much clearer diagonal. For all classes, the majority of the models are correctly classified. For the first class, the correct prediction is at 55 % with 43 % misclassified into the second class. The second class achieves 68 % correct predictions, while 30 % of the models are misclassified belonging to the third class. The third class reaches the best result with a correct prediction slightly above 70 %. 17 % of the models are predicted to belong to the second class and the remaining 12 % are predicted to be the fourth class. For the fourth class, 61 % are predicted correctly, and 38 % are misclassified as belonging to the third class.

In Figure 4.23b (10 epochs for the SOR approach) a shift is visible towards predicting the third class, similar to the interval approach. In the first class, 22 % of the models are correctly predicted, with 69 % of the models misclassified into the second class. In the second class, 42 % of the models are correctly predicted, with the majority (58 %) misclassified as the third class. The third class maintains a high correct prediction rate of 91 %, while 63 % of the fourth class models are misclassified as the third class. This leaves only 36 % correctly classified.



(a) 4-class model with SOR approach - models trained for 1 epoch

(b) 4-class model with SOR approach - models trained for 10 epochs

Figure 4.23: Results of 50/50 women/men ration - 4 class model for the SOR approach

The worsening performance after increasing training epochs can be attributed to overfitting. Evidence for this is shown in Figure 4.24, which presents the confusion matrix for the shadow models used to train the attack model. Here, nearly 100 % of the models are correctly classified across all classes, confirming that the shadow models have overfitted to their training data. This overfitting hinders the attack model's ability to generalize to unseen data, resulting in poorer performance.

In conclusion, while increasing the training epochs was expected to improve the results, the experiment highlights the risk of overfitting. These findings suggest that training shadow models for a limited number of epochs may yield better attack performance by preserving their ability to generalize to the target model.

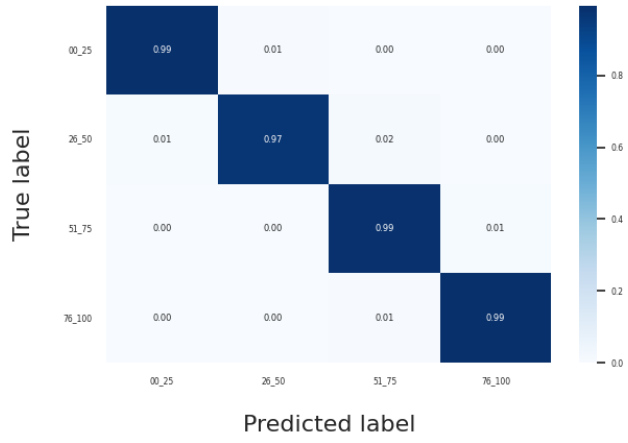


Figure 4.24: 4-class model with SOR approach - models trained for 10 epochs

Enlarging the training dataset for the shadow models

A further strategy involves increasing the size of the dataset from which the training instances for shadow models are randomly selected. This enhancement introduces greater diversity into the shadow models, making them more representative and capable of generalizing across a wider range of scenarios. These results are shown in Table 4.3 in rows labelled with "more training data".

This measure leads to a decrease in accuracy in the 2-class model with shadow models trained for 1 epoch and the approach without ordinal classification. All other accuracy results for this setting improve by an average of 3 %. The best accuracy is reached by the CORAL approach. It should also be noted that the interval method achieves an accuracy of 89%, a result that is also achieved with the CORAL approach. The method that does not employ ordinal classification yields an accuracy of 81%, which is 6% lower than the result achieved with the reduced data set size. The SOR approach achieves an accuracy of 88 %, and the CORN approach reaches 86 %.

The MSE results mirror the trends observed with accuracy. The approach without ordinal classification shows a worsening MSE compared to the smaller dataset, while all other methods achieve improved MSE values. The best MSE is reached by the CORAL approach with 0.1055 closely followed by the interval approach with a value of 0.1094. The other approaches reach a MSE value between 0.1172 for the SOR approach and 0.1855 for the approach without ordinal classification.

Analysing the result for the 2-class model with shadow models trained for 10 epochs leads to the following conclusion: The results highlight the recurring issue of overfitting, although some improvements are seen compared to smaller datasets. Quantitatively speaking, this results in an average increase of accuracy by 9 %. The range of accuracy in this experiment ranges from 74 % for the CORN approach to 80 % for the SOR approach.

The results are confirmed by MSE. Here, the worst result is reached for the CORN approach with 0.2559. The best result is achieved by the SOR approach with a MSE of 0.1992. However, these results are worse than those from the models trained for 1 epoch, suggesting that limiting training epochs may help prevent overfitting.

In the 4-class case, the same behaviour occurs. The attack with shadow models trained for a single epoch and with a larger training data set leads to a more pronounced improvement than the one with a smaller training data set. In the 4-class model, the accuracy increases by an average of 5 %, compared to 3 % in the 2-class model. The CORAL approach achieves the best accuracy at 70 %, followed by the SOR approach producing a result of 69 %. Both the method without ordinal classification and the CORN approach manage to reach an accuracy of 64 %. The interval approach performs the worst with an accuracy of 57 %.

The MSE results align with the accuracy improvements. The CORAL approach achieves the best MSE at 0.3057, followed by the SOR approach at 0.3145. The CORN approach slightly outperforms the method without ordinal classification, with MSE values of 0.3926

and 0.4004, respectively. The interval approach, as expected, delivers the worst MSE at 0.4434. In all cases, the MSE values are lower than those achieved with the smaller dataset, demonstrating the benefit of using a larger, more diverse dataset.

For the 4-class model trained with shadow models for 10 epochs, the same trends as the 2-class case are observed. Accuracy decreases compared to models trained for 1 epoch but increases compared to the smaller training dataset. The CORAL approach achieves the highest accuracy at 68 %, followed closely by the CORN approach at 62 %. The approach without ordinal classification and the SOR approach reach 58 % and 54 % respectively, while the interval approach again delivers the lowest result at 52 %.

The experiment confirms that increasing the size of the dataset from which shadow models are trained improves the results when models are limited to 1 epoch of training. This improvement is evident in both accuracy and MSE values, with the largest benefits seen in the 4-class case.

However, training shadow models for 10 epochs, even with a larger dataset, exacerbates overfitting. This reduces the generalization of the attack model, as indicated by higher MSE values and lower accuracy compared to models trained for 1 epoch.

Overall, the findings emphasize the importance of using a larger, diverse dataset to improve the shadow model's capacity to generalize across different scenarios. Moreover, the results strongly suggest that limiting the number of training epochs can mitigate overfitting and enhance the attack's performance. Balancing these factors—dataset diversity and training duration—is critical for designing effective property inference attacks.

Table 4.3: Results 50 / 50 slit of women and men

		50/50 split				
		NLLLoss	**sor**	**interval**	**corn**	**coral**
2 classes 01 epochs	accuracy	87%	85%	83%	85%	87%
	mse	0,1328	0,1465	0,1660	0,1543	0,1270
2 classes 10 epochs	accuracy	76%	71%	62%	74%	66%
	mse	0,2422	0,2930	0,3789	0,2637	0,3379
2 classes 01 epochs more training data	accuracy	81%	88%	89%	86%	89%
	mse	0,1855	0,1172	0,1094	0,1426	0,1055
2 classes 10 epochs more training data	accuracy	79%	80%	76%	74%	79%
	mse	0,207	0,1992	0,2422	0,2559	0,2109
4 classes 01 epochs	accuracy	62%	64%	48%	60%	64%
	mse	0,4316	0,3896	0,5928	0,4717	0,4443
4 classes 10 epochs	accuracy	44%	48%	36%	45%	47%
	mse	0,6309	0,5908	0,8564	0,6768	0,5986
4 classes 01 epochs more training data	accuracy	64%	69%	57%	64%	70%
	mse	0,4004	0,3145	0,4434	0,3926	0,3057
4 classes 10 epochs more training data	accuracy	58%	54%	52%	62%	68%
	mse	0,4219	0,4883	0,4961	0,4023	0,3213

4.3.2 70 / 30 split of women/men

The outcomes of the experiments that employ the 70 / 30 division of women to men are presented in Table 4.4. This setup, as described in Section 3.3.2, ensures that the datasets used to randomly select training instances for the shadow models and the target models have differing demographic distributions. Specifically, the dataset for the shadow models maintains a 70/30 ratio of women to men, while the dataset for the target models reflects a 41/59 ratio. This variation simulates realistic scenarios where an attacker's data distribution may not precisely match the distribution of the target model's training data. The experiments were conducted using the five distinct loss methods, allowing for a comprehensive analysis of the impact of these demographic differences on the success of property inference attacks.

Original experiment

The initial results for this approach focus on the 2-class model, using a smaller dataset for sampling the training instances for the shadow models, trained over a single epoch. The resulting accuracy for the target models ranges from 68 % for the interval approach to 86 % for the approach without ordinal classification. The CORAL approach yields an accuracy of 80 %, while the SOR and CORN approaches fall in between, achieving 78 % and 73 %, respectively.

The MSE provides further insight into the performance of these approaches. The approach without ordinal classification again performs best, with a low MSE of 0.1426, followed closely by the CORAL approach at 0.1973. The MSE for the SOR approach is approximately 0.03 higher, namely 0.2246. The CORN approach has a MSE of 0.2656 and the MSE for the interval approach is at 0.3164.

For the 4-class model, the results highlight a different distribution of performance. The SOR and CORN approaches achieve the highest accuracy, both at 55 %, while the interval approach records the worst performance, with an accuracy of only 42 %. The CORAL and the approach without ordinal classification fall between these extremes, achieving 46 % and 51 % accuracy, respectively.

According to the values of MSE, the CORN approach is the best working. The MSE for this approach is 0.5635, marginally outperforming the SOR approach at 0.5723. The MSE for the worst working approach according to accuracy yields a MSE of 0.7148 which is a better result than the MSE of the CORAL approach with 0.8359. The approach without ordinal classification reaches a MSE of 0.6260.

To better understand the performance of the CORAL approach, particularly its high MSE, the confusion matrix for this scenario is presented in Figure 4.25. It reveals that the predictions for the first three classes are slightly shifted, contributing to the elevated MSE. For the first class, only 25 % of the models are correctly classified, while 52 % are misclassified as belonging to the second class and 22 % are assigned to the third class. This indicates one reason for the high MSE because the meta-model is usually one class

off and rarely two. Similar trends are observed in the second class, where only 19 % of the models are correctly predicted. 67 % of the models are predicted to belong to class three and 14 % of the models are predicted to belong to class four. For the third class models are predicted to belong to the third class (48 %) or to belong to the fourth class 52 %. Finally, the fourth class exhibits the best performance with 96 % of the models correctly classified. Only 4 % are predicted to belong to the third class.

These findings suggest that the higher MSE of the CORAL approach is primarily due to a significant proportion of models in the first two classes being misclassified by two classes, further emphasizing the challenges in accurately predicting target data distributions in a federated learning environment with imbalanced datasets.

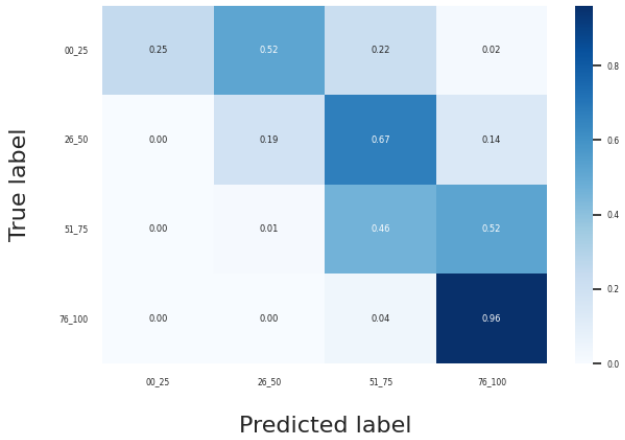


Figure 4.25: 4-class model with CORAL approach - models trained for 1 epoch

Increased training epochs for the shadow models

For the 70/30 data split, extending the number of training epochs for the shadow models to 10 resulted in a similar trend as observed in Section 4.3.1, namely a decrease in accuracy and an increase in MSE. The results are shown in Table 4.4.

The accuracy for the two-class model ranges between 51 % and 54 % for the five loss approaches. The best accuracy was achieved by the CORAL approach, closely followed by the CORN and SOR approaches with an accuracy of 53 %. The approach without ordinal classification achieved a slightly lower accuracy of 52 % while the interval approach produced the lowest accuracy at 51 %.

In addition, the MSE values for the two-class model, which used shadow models trained for 10 epochs, range narrowly from 0.4648 to 0.4902. The best MSE is reached for the CORAL approach which is consistent with the accuracy results. Based on MSE, it can be seen that the CORN approach is slightly better than the SOR approach with a MSE of 0.4707 as opposed to 0.4727. The approach without ordinal classification reaches a MSE of 0.4824 and the interval approach achieves the highest MSE value.

The accuracy for the 4-class model with shadow models trained for 10 epochs is, on average, 14 % lower than the results for the model with shadow models only trained for 1 epoch. The resulting accuracies range between 34 and 37 % and are, therefore, significantly lower than in previous experiments. The worst performing method was the SOR approach followed by the interval and CORN approaches with an accuracy of 35 %. The approach without ordinal classification achieves the best result (37 %), and the model using the CORAL approach achieves an accuracy of 36 %.

Similar results can be observed for MSE. The values in all cases are above 1, which was not the case for the previous results. The highest value was achieved by the interval approach, namely 1.4150 closely followed by the SOR approach with 1.3057. The best result came from the CORAL approach, which yielded a MSE of 1.1436; the approach without ordinal classification has an MSE of 1.2207, while it is 1.25 for the CORN approach.

The increase in training epochs for shadow models again leads to overfitting to the training task, which in turn leads to worse performance for both the 2-class and 4-class models. Accuracy and MSE results consistently highlight this limitation. These results underscore the importance of carefully selecting training epochs to avoid overfitting and maintain the generalization of shadow models in property inference attacks.

Enlarging the training dataset for the shadow models

Enlarging the data set to achieve a greater diversity of shadow models leads to different conclusions in the experiments conducted with a 70/30 split. The results are shown in Table 4.4.

In the case of the 2-class models with shadow models trained for 1 epoch, the results of the accuracy increase in nearly all approaches. Only the approach without ordinal classification attains an accuracy close to 10 % below the initial result, which is 75%. The other four approaches result in an accuracy between 72 and 83 %. The result for the approach without ordinal classification lies therefore within this range and as already stated in Section 4.3.2 was higher than that of the other approaches in the original setting. The best result in the experiment with more training data is achieved by the CORAL approach closely followed by the SOR approach with an accuracy of 82 %. The other two approaches reach an accuracy lower than or equal to 75 %.

The MSE for this experiment lies between 0.1660 and 0.2812. The CORAL approach achieves the best result, as can already be seen with accuracy. The SOR approach has a MSE of 0.1797. The other three approaches have values greater than 0.2. More precisely, the CORN approach reaches a value of 0.2461, the approach without ordinal classification 0.2480 and the worst MSE is reached for the interval approach, namely 0.2812.

In the 2-class case with shadow models trained for 10 epochs and a larger data set to sample from, the results are quite similar to the one with the smaller set. The accuracy varies by approximately 1 %. The worst result is reached for the interval approach with

an accuracy of 50 %. The approach without ordinal classification scores slightly better, with an accuracy of 51 %. The SOR and the CORN approach both reach an accuracy of 54 % and the CORAL approach achieves the best result with 56 %.

The MSE confirms these results. The interval approach has the worst value with 0.4980 followed by the approach without ordinal classification with 0,4863. The CORN approach is slightly worse than the SOR approach – they achieve a MSE of 0.4648 and 0.4609, respectively. The best MSE is reached for the CORAL approach with 0.4375.

In the 4-class model trained with shadow models with 1 epoch of training, the accuracy increases in nearly all cases. The only approach with a decrease to the original experiment is the CORN approach, with a marginal decrease of 1 % down to 54 %. In the other approaches, this measure increases the accuracy by an average of 8 %. The best working approach is the SOR approach with an accuracy of 64 %. In contrast to 55 % in the original experiment (Section 4.3.2) this is an increase of 9 %. The second best approach is the one without ordinal classification. Here, the resulting accuracy is 62 %. The accuracy of the CORAL and CORN approach is at 57 % and 54 %. The interval approach reaches an accuracy of only 44 % and is, therefore, much worse than the other approaches. In addition, the increase to the original experiment in this case is not as significant with only an increase of 2 %.

The improvement is also measurable in MSE. Enlarging the data set from which the training data are sampled reduces MSE from an average of 0.6625 to an average of 0.4900. The best working approach is the SOR approach that achieves a MSE of 0.3887 closely followed by the approach without ordinal classification with a MSE of 0.3955. The MSE of the CORAL approach is approximately 0.1 higher by 0.4990. The CORN approach reaches a MSE of 0.5352 and the interval approach is the worst working one with a MSE of 0.6318.

In the 4-class models with a larger set to sample the training data for the shadow models and the shadow models trained for 10 epochs, it is not possible to discern a distinct pattern. Two approaches, namely the approach without ordinal classification and the SOR approach, achieve slightly better results in contrast to the experiment with a smaller data set size. The resulting accuracies are 39 % for the approach without ordinal classification and 36 % for the SOR approach. The accuracy for both the interval and the CORN approach remains unchanged at 35%. The accuracy of the CORAL approach decreases by 6 % to a result of 30 %.

The MSE for this experiment ranges from 1.0312 for the approach without ordinal classification to 1.2656 for the CORN approach. The CORAL approach has a MSE of 1.2598. The SOR approach achieves the second-best result with 1.0977 and the interval approach achieves 1.1973.

Enlarging the dataset for shadow model training generally improves the results, particularly for shadow models trained for one epoch. The increased diversity of training instances enables shadow models to capture a broader range of data distributions, leading to better attack performance.

For shadow models trained for 10 epochs, overfitting issues continue to hinder performance. This is evident in the limited improvements in accuracy and the increased MSE values. These results emphasize the importance of balancing training diversity with appropriate training durations to optimize shadow model generalization for property inference attacks.

Table 4.4: Results 70 / 30 slit of women and men

		70/30 split				
		NLLLoss	**sor**	**interval**	**corn**	**coral**
2 classes 01 epochs	accuracy	86%	78%	68%	73%	80%
	mse	0,1426	0,2246	0,3164	0,2656	0,1973
2 classes 10 epochs	accuracy	52%	53%	51%	53%	54%
	mse	0,4824	0,4727	0,4902	0,4707	0,4648
2 classes 01 epochs more training data	accuracy	75%	82%	72%	75%	83%
	mse	0,2480	0,1797	0,2812	0,2461	0,1660
2 classes 10 epochs more training data	accuracy	51%	54%	50%	54%	56%
	mse	0,4863	0,4609	0,4980	0,4648	0,4375
4 classes 1 epoch	accuracy	51%	55%	42%	55%	46%
	mse	0,6260	0,5723	0,7148	0,5635	0,8359
4 classes 10 epochs	accuracy	37%	34%	35%	35%	36%
	mse	1,2207	1,3057	1,4150	1,2500	1,1436
4 classes 01 epochs more training data	accuracy	62%	64%	44%	54%	57%
	mse	0,3955	0,3887	0,6318	0,5352	0,4990
4 classes 10 epochs more training data	accuracy	39%	36%	35%	35%	30%
	mse	1,0312	1,0977	1,1973	1,2656	1,2598

4.3.3 Comparison of the different women/men splits

To compare the two women/men splits, Figure 4.26 presents the top results - the best of the five loss methods, evaluated using the MSE - for the different approaches.

Figure 4.26a shows the accuracy results. For the 2 class model with shadow models trained for 1 epoch the 50/50 split achieves a very high accuracy close to 85 %. The value of the 70/30 split is slightly lower. For the 2 class model with shadow models trained for 10 epochs, the 50/50 split achieves a moderate accuracy, around 70 %, and the 70/30 split has a much worse result, around 50 %. The result for the 2 class models trained with a more varied data set is very similar, although the general trend is that the accuracy is slightly higher.

The 4-class model with shadow models trained for 1 epoch achieves a moderate accuracy of around 60 % for the 50/50 split. The accuracy of the 70/30 split is slightly lower, namely around 55 %. For the same model with shadow models trained for 10 epochs, the result for the 50/50 split is around 50 % and the accuracy for the 70/30 split is even lower, around 40 %. Again, the increase in training data variety leads to an increase in accuracy. The accuracy of the 50/50 split is around 70 % and the 70/30 split is around 55 % for models with shadow models trained for 1 epoch. The increase in epoch numbers leads to a decrease in accuracy. The 50/50 split achieves around 60 % and the 70/30 split around 50 %.

In general one can say that the 50/50 split achieves higher accuracy in contrast to the 70/30 split around all configurations. In addition, models with a more varied data set tend to have a higher accuracy than those without it. It is again visible that accuracy tends to decrease as the number of epochs increases. This is especially the case for the 4 class models.

Figure 4.26b shows the different MSE result for all approaches. The best-performing loss method is used for the creation of the plot.

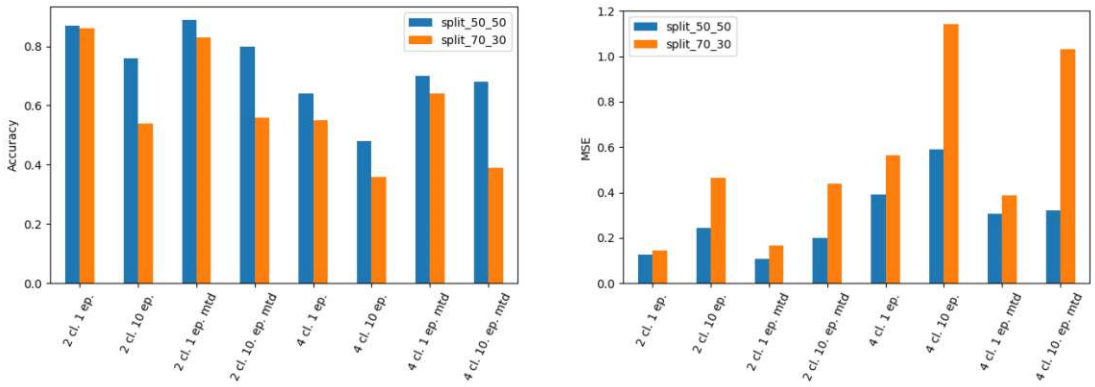
Overall it can be said that higher accuracy configurations generally show lower MSE values. For both splits, it is noticeable that configurations with lower accuracy in Figure 4.26a tend to show higher MSE values. It can also be observed that the 50/50 split consistently outperforms the 70/30 split.

Figure 4.26a confirms the assumptions made with accuracy that the approaches with higher variety in the shadow models generally show better performance than those with a lower variety. It is also confirmed that increasing the number of epochs (10 vs 1) tends to increase MSE, especially in the 4-class configurations.

In general, one can see that approaches with 2 classes tend to have lower MSE values and higher accuracy compared to those with 4 classes, which indicates that a model with fewer classes performs better, which is as expected.

Overall, the bar charts in Figure 4.26, offering a complementary perspectives on model performance, show that the patterns in accuracy and MSE are consistent with expectations.

High accuracy is associated with low MSE, and setups that excel in one metric generally excel in the other as well.



(a) Accuracy for the different splits and approaches

(b) MSE for the different splits and approaches

Figure 4.26: Most effective approaches for the different splits and approaches

4.3.4 Comparison of the different epochs

To determine the number of epochs in which performance decreases, an experiment was conducted on the 4-class model using a more diverse training data set. The attack was carried out with shadow models trained for 1, 3, 5, 8, and 10 epochs. In Figure 4.27 the results of accuracy and MSE for the different numbers of epochs and the 50/50 split are shown in two line plots.

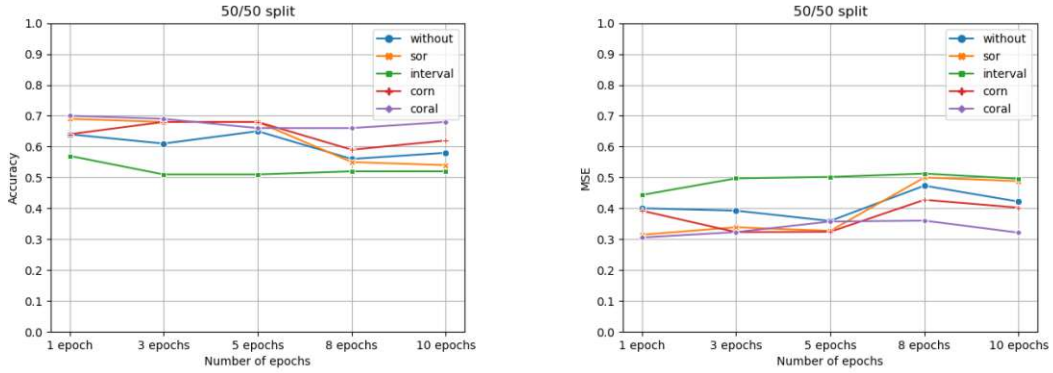
Figure 4.27a measures the accuracy of the different loss methods. The method without ordinal classification starts with an accuracy of around 60 % at 1 epoch, showing a slight fluctuation before ending at approximately 55 % after 10 epochs. The SOR approach begins with an accuracy slightly below 70 %, experiences a decline over time, and concludes slightly above 50 %. The interval approach starts at approximately 55 %, and declines to just above 50 % between 1 and 3 epochs. It then remains relatively stable. The CORAL approach starts with the highest initial accuracy, just below 70 %, decreases slightly at first and increases slightly again between 8 and 10 epochs. It then ends at approximately 68 %. The CORN approach starts around 65 %, slightly increases at first, and then decreases rapidly between 5 and 8 epochs and ends just above 60 %.

In Figure 4.27b the resulting MSE values are shown for the different number of training epochs for the different loss approaches. The approach without ordinal classification starts with an MSE of around 0.4, fluctuates slightly, and ends just above 0.4. The SOR approach begins at approximately 0.3, exhibits a significant rise between 5 and 8 epochs and stays stable between 8 and 10 epochs. It then ends just below 0.5. The interval approach starts at around 0.45, increases gradually over the epochs, and ends at about 0.5. The CORN approach begins at around 0.4, shows some fluctuations, and ends at

approximately 0.4. The CORAL approach starts at approximately 0.3, fluctuates slightly, and ends slightly above 0.3.

From Figure 4.27a, it is evident that the CORAL approach maintains the highest accuracy throughout the epochs, making it the best-performing method in terms of accuracy. The interval approach performs the worst, consistently showing the lowest accuracy among the five methods. The highest decrease can be seen with the SOR approach, which is especially visible between 5 and 8 epochs. Also in terms of MSE, the CORAL approach shows the lowest MSE in most epochs, indicating the best performance for this measure. The interval approach, on the other hand, has the highest MSE throughout, which confirms the results of the accuracy graph.

In summary, the CORAL approach is the best in terms of accuracy and MSE. The interval approach performs poorly in both metrics, consistently showing the lowest accuracy and the highest MSE. This was also the case in the previous experiments. Overall in the 50/50 split, there is no obvious pattern indicating after how many training epochs the performance declines. However, it is evident that between 5 and 8 epochs, the performance of the approach without ordinal classification, the SOR approach, and the CORN approach degrades.



(a) Accuracy for the different epochs for the 4 class model with more training data

(b) MSE for the different epochs for the 4 class model with more training data

Figure 4.27: Result for different epochs with a 50/50 split

In Figure 4.28a and Figure 4.28b the results of the same experiment for the 70/30 split are shown. Again the shadow models were trained for either 1, 3, 5, 8 or 10 epochs.

Figure 4.28a depicts the accuracy. The method without ordinal classification has an accuracy slightly above 60 % for the shadow models trained with one epoch. It then decreases to around 45 % for 3 epochs and remains unchanged for the 5 epochs and decreases even further to around 35 % for 8 epochs. Then it increases slightly to approximately 40 % for 10 epochs. The SOR approach has the highest accuracy for 1 epoch with a value of almost 65 %. Then it drastically decreases to around 35 % for 5 epochs and increases slightly to just under 40 %. The interval approach begins

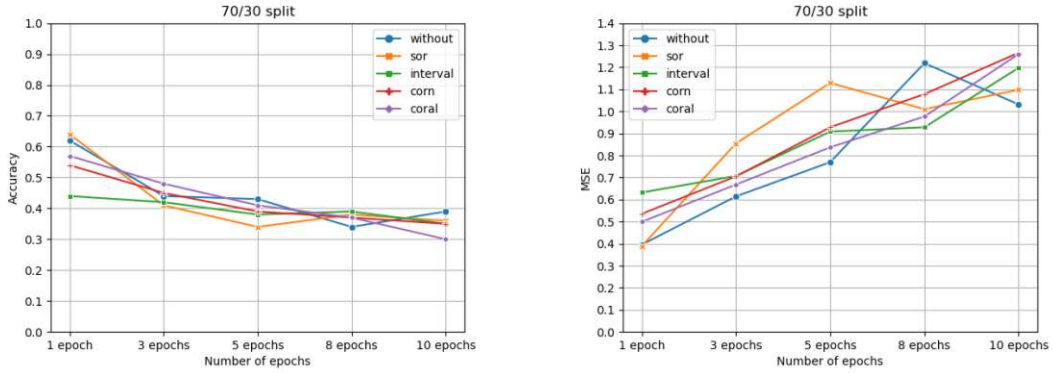
with an accuracy of around 45 %, showing a more steady performance with minimal decrease, ultimately ending just below 40 %. The CORN approach starts with an accuracy of around 55 % and then gradually decreases to 35 % for 10 epochs. The CORAL approach also shows a gradual decrease from 55 % for 1 epoch to the lowest accuracy of approximately 30 % for the 10 epochs.

Figure 4.28b shows the MSE results. Here the approach without ordinal classification starts with a value of approximately 0.4 and increases to 1.2 at 8 epochs. It then increases to around 1 at 10 epochs. The SOR approach also has a value of 0.4 for 1 epoch and then increases drastically to 0.85 at 3 epochs. It increases even higher at 5 epochs and then decreases to 1.1 at 10 epochs. The interval approach starts with the highest MSE value of around 0.65 and then increases gradually to 0.9 at 5 epochs. It stays the same at 8 epochs and increases to a value of 1.2 at 10 epochs. The CORN and CORAL approaches have the most gradual increase. The CORN approach starts with a value of around 0.55 at 1 epoch and ends at approximately 1.25 at 10 epochs; the CORAL approach scores slightly better for the lower number of epochs starting with a value of around 0.5 at 1 epoch and ending with 1.25 at 10 epochs.

From the accuracies depicted in Figure 4.28a), it is clear that all methods experience a decrease in accuracy as the number of epochs increases. Initially, the SOR approach shows the highest accuracy, followed closely by the approach without ordinal classification. When the shadow models are trained for 10 epochs, the approach without ordinal classification and the SOR approach still exhibit relatively better accuracy, despite both experiencing declines over time.

In terms of MSE, the SOR approach has the lowest value at 1 epoch and then shows the highest MSE value at 3 and 5 epochs, indicating the poorest performance in terms of error minimization. The approach without ordinal classification shows the lowest MSE values at 1 to 5 epochs. It then increases drastically to the worst working method at 8 epochs.

In general, the performance of the five methods shows a consistent decrease in accuracy and an increase in MSE as the number of epochs increases. Although no method shows a pronounced improvement over the others in both metrics, the CORAL approach seems to balance accuracy and MSE better than the others, making it the most stable performer over the given epochs.



(a) Accuracy for the different epochs for the 4 class model with more training data

(b) MSE for the different epochs for the 4 class model with more training data

Figure 4.28: Result for different epochs with a 70/30 split

CHAPTER 5

Conclusion

This thesis focused on investigating privacy vulnerabilities in fully connected neural networks, especially focusing on property inference attacks in a multi-class setting. The goal was to extend the current research, which was limited to binary property inference attacks, to more fine-grained, multi-class property inference attacks. Additionally, the federated setting was explored, as this is a learning paradigm that is increasingly used, especially when the training data is confidential. With the various experiments conducted this thesis contributes to a deeper understanding of the risks that machine learning models face in terms of data privacy and security.

5.1 Contributions

This work makes several key contributions to the understanding and analysis of privacy risks caused by property inference attacks.

- **Multi-class property inference attack:** The main contribution of this thesis is the successful extension of property inference attacks from the binary to a multi-class setting. In contrast to the binary case, in which an attacker can only infer whether a specific property exists in the dataset (e.g. "Does the training data set contain more than 70 % men?"), the multi-class attack can distinguish between multiple classes (e.g. "Were there between 0-25 %, 26-50 %, 51-75 % or 76-100 % of male instances in the training data set of the model?"). This extension is important information for the attacker, because a much more detailed understanding of the underlying data is received.
- **Hierarchical attack approach:** To solve the problem of increasing difficulty of correctly predicting the value range with a higher granularity, i.e. with a number of classes, a hierarchical approach was introduced in this thesis. Instead of directly

training an attack model distinguishing between a large number of classes, the attack is broken down into a series of smaller classification tasks. This allows each classification task to be responsible for distinguishing between fewer classes. This approach proved to be effective in maintaining accuracy and MSE even as the number of classes increased.

- **Evaluation of loss functions:** In order to improve accuracy and MSE various loss functions, including ordinal and non-ordinal regression approaches, are used. These loss functions are critical in determining how effective the multi-class property inference attack can be performed. This thesis shows that ordinal regression approaches provide better results in most cases.
- **Impact on federated learning:** Another significant attribution is the exploration of property inference attacks in the context of federated learning. The goal of federated learning is to enhance privacy by training a collaborative model without sharing the local datasets. In this thesis, it is demonstrated that even in a decentralized environment, sensitive information can still be inferred. Our results indicate that property inference attacks are a serious threat in these settings, potentially revealing private characteristics.

5.2 Research questions

The following section directly addresses the research questions posed at the beginning of the thesis:

- **RQ 1.1: What is the trade-off between the resulting accuracy of the property inference attack and an increase in it's forecasting granularity? For conducting the property inference attack the binary meta-machine learning model [GWY+18], which is trained on model parameters of different machine learning models (i.e. shadow models) is transformed into a multi-class model.**

  The extension of a property inference attack from a binary to a multi-class problem introduces new challenges. As the number of classes increases the complexity of the problem grows because the attack must now infer a more granular level of detail from the model. This is significantly more difficult. The experiments were started with a binary classification task, and the number of classes was gradually increased to more complex settings, including 4, 5, 10 and 20 classes.

  The findings in this thesis align with the findings already discussed by [GWY+18]. The binary case, as expected, demonstrated a high degree of accuracy with performances above 90 %. This shows that the attack in the binary case is not only feasible but highly effective.

  However, the goal of this thesis was to extend the property inference attack to the multi-class setting where the complexity increases significantly. In our experiments,

extending the attack to the multi-class case resulted in a modest drop in performance. The accuracy is slightly reduced but remains at approximately 75 %, showing that the attack is working with a reasonable performance. This slight drop in accuracy is not surprising as the complexity of the attack, but along that also the inferred information, increases and the task of distinguishing between the different classes becomes more complex. Nevertheless, even with four classes, the attack remained potent, demonstrating that property inference attacks can be successfully scaled beyond binary classification to reveal more detailed information about the dataset.

Further extending the attack to a five-class setting results in a more significant decline in accuracy, with values dropping to around 67 %. While this reduction reflects the growing difficulty of the attack as the number of classes increases, it is important to note that an accuracy of 67 % is still well above random guessing. These results indicate that the attack can extract valuable insights about the underlying data, making it a viable method for more granular classifications.

When analysing the results of the 10-class model, the accuracy decreases more steeply, reaching 47 %. This drop may seem substantial, but it is crucial to emphasize that even at this level, the attack performs significantly better than random guessing, which would yield an accuracy of 10 %. This performance indicates that despite the attack's difficulty in navigating a larger number of classes, meaningful patterns can be inferred from the model.

Finally, the 20-class model presents the most challenging case. The attack accuracy drops to 28 %. While this represents the lowest performance observed in the study it is important to underscore that an accuracy of 28 % again remains well above random guessing (which would be an accuracy of 5 %). Even in this highly complex scenario the results confirm that the attack is capable of inferring sensitive information of a model with a certain degree of success. This finding indicates that property inference attacks, although they become less accurate with increasing class granularity are still effective in the multi-class case. The ability of the attack to maintain a non-trivial level of accuracy even when faced with 20 classes speaks to the vulnerabilities in neural network models and the potential for these attacks to extract detailed and sensitive information in various real-world settings.

In summary, while the accuracy of the property inference attacks naturally declines as the complexity of the classification task increases - from the binary case to the 20-class setting - the results demonstrate that these attacks remain effective. This reduction in accuracy aligns with the increasing difficulty of the task, yet the performance levels observed in all experiments exceed those of random guessing. This consistency across various class settings highlights the adaptability of property inference attacks to the multi-class case. Ultimately, the findings suggest that multi-class property inference attacks represent a persistent privacy risk, particularly for models trained with datasets containing sensitive information and that they can reveal significant information about the training data even as the complexity of the attack increases.

- **RQ 1.2: Can a hierarchical approach improve the performance of the attack in the multi-class setting, compared to a straightforward multi-class approach? For the hierarchical approach, the classification task is broken down into a series of simpler decisions using different meta-machine learning models.**

  Given the challenges encountered with increasing property granularity, a hierarchical approach to improve the performance of the attack was introduced in the thesis. The main idea behind this approach is to decompose the multi-class attack into a series of smaller tasks. Instead of attempting to infer all class distinctions in one step, the hierarchical approach breaks the attack down into a set of smaller attacks, each responsible for distinguishing between fewer classes.

  The hierarchical approach was implemented to improve the results of the 20-class model, as this was the most complex task to solve. The results revealed significant insights into how this approach behaves for the various loss methods.

  The SOR approach provided the best overall performance with both the 20-class model and the hierarchical approach. The accuracy is over 30 % in both cases and simultaneously, the MSE is low in both cases. This small performance gap indicates that for this method, there is no clear advantage to using the hierarchical approach as both approaches are effective in managing the attack.

  For the CORAL approach the hierarchical method slightly outperformed the 20 class model in terms of accuracy but shows a marginally higher MSE (3.4 in contrast to 3.2). This implies that both methods handle the ordinal nature of the classes well and it is not necessary to implement the hierarchical approach for this loss approach.

  The CORN approach displayed a big benefit from using the hierarchical approach. Here the hierarchical approach is far outperforming the 20 class model in both accuracy (27 % vs. 16 %) and MSE (4.7 vs. 6.7). This implies that the hierarchical approach is better suited for the attack using the CORN loss method.

  In contrast, the interval method favoured the 20 class models, which outperformed the hierarchical approach in both accuracy and MSE. This finding indicates that the strength of the interval method lies in handling the data as a whole rather than through a decomposed approach.

  The approach without ordinal regression revealed big differences between the two methods. As the 20-class model does not work at all for this approach (accuracy of 5 % and MSE of 90.8), the hierarchical approach provides a better classification by splitting the problem into manageable chunks.

  These results indicate that the SOR approach is a promising method for property inference attacks. Overall, the findings suggest that the choice between a hierarchical approach and the standard approach for property inference attacks largely depends on the loss function used. In most cases, however, the direct multi-class models perform just as well, if not better, than hierarchical methods. This implies that for

many use cases, the standard approach is sufficient, and the added complexity of hierarchical structures may not be necessary unless specific requirements related to the loss function arise.

- **RQ 2: How accurately can the property inference attack be conducted in a sequential federated learning setting by the second participant with access to the data set structure and model?**

  In a sequential federated learning setting, participants do not have direct access to each other's datasets, but they contribute to the training of a shared model by updating model parameters with their local data. The second participant, with access only to the structure of the dataset (but not the raw data itself), can leverage property inference attacks to infer sensitive information about the dataset used by the first participant.

  The experiments and findings of this thesis suggest that the accuracy and MSE of such an attack largely depends on several factors. When only the dataset structure is known the second participant can still infer certain global properties of the first participant's dataset.

  From the results discussed, accuracy tends to decrease and MSE tends to increase as the complexity of the attack increases. However, it has been demonstrated that even with limited information, the second participant can achieve reasonable levels of accuracy and MSE in inferring certain properties.

  In general, the experiments demonstrated that the performance of the property inference attack improves significantly when the models are trained with a larger volume of data. Specifically, increasing the training data led to an average 10 % boost in accuracy, highlighting the importance of dataset size in refining the attack's effectiveness. This improvement is likely due to the fact that larger datasets provide more comprehensive patterns for the models to learn, resulting in more reliable predictions when attempting to infer the properties of the target data.

  Another important finding was that the attack's success is highly dependent on the degree to which the shadow models are trained. It was observed that overfitting the shadow models to their specific training tasks negatively impacts the attack's accuracy. When shadow models are too narrowly optimized for the training data, they fail to generalize well, thereby reducing their ability to mimic the target model's behaviour. On the other hand, more generalized shadow models were better at approximating the target model, which in turn improved the attack's effectiveness. This suggests that ensuring the shadow models maintain a balance between being well-trained and not overfitted is crucial for conducting a successful attack.

  A third key finding was the impact of data distribution similarity between the shadow models and the target model. The attack consistently achieved better results when the training data used for sampling data for the shadow models closely mirrored the distribution of the target model's data. This is because when the

shadow models are trained on datasets that reflect similar characteristics they are more likely to produce outputs that align with the target model. This makes it easier for the attacker to infer properties of the target dataset.

Previous work has primarily assumed this idealized scenario, where the attacker has access to training data that closely matches the target model's distribution. However, this is a strong assumption that may not always hold in real-world settings. In contrast, our approach explicitly examines different data distributions and their impact on attack success. By doing so, we demonstrate that while distributional similarity enhances attack performance, attacks can still infer meaningful properties even when the distributions differ.

In conclusion, these findings underscore that the success of property inference attacks hinges on several key factors: the amount of training data, the generalization of shadow models, and the alignment between the data distributions of the shadow and target models. These factors significantly influence the accuracy and effectiveness of such attacks, depending on the attacker's capabilities and knowledge of the target model.

5.3 Future Work

The work presented in this thesis opens up several important avenues for future research. First, while we have demonstrated the feasibility of multi-class property inference attacks, further work is needed to refine and optimize these attacks. For example, exploring different architectures, datasets, and attack strategies could help identify new vulnerabilities in machine learning models.

Second, this thesis focused primarily on fully connected neural networks. Future work could extend the findings to other types of machine learning models, such as convolutional neural networks (CNNs) or recurrent neural networks (RNNs). Investigating whether property inference attacks can be successfully executed on these models will provide a more comprehensive understanding of the risks.

Third, future research should place a greater emphasis on federated learning, as this distributed paradigm is increasingly being adopted in privacy-sensitive fields like healthcare and finance. While federated learning enables collaborative model training without direct data sharing, this thesis has shown that privacy risks persist. Future work should focus on refining property inference attacks within federated settings and developing stronger defences specifically designed for these environments. Techniques such as differential privacy could be explored to mitigate these risks.

Overview of Generative AI Tools Used

In this work, I used generative AI tools as supporting instruments for text improvement and rewording. Below is a description of the tools employed, their areas of application, and how they were utilized.

ChatGPT (OpenAI) was used for text revision, rewording, and stylistic optimization. It helped improve readability and coherence and provided suggestions for alternative formulations.

- **Prompt:** You are a technical student currently writing your diploma thesis. Can you reformulate the given input accordingly.

DeepL was employed for translation and linguistic refinement. It assisted in translating and fine-tuning specific passages to ensure accuracy and fluency.

Writefull was utilized for grammar checking and stylistic corrections. It helped verify grammar, refine style, and improve scientific phrasing.

List of Figures

2.1 Neural network (own work) 10
2.2 ReLU and Leaky ReLU activation function [SSA20] 14
2.3 Confusion matrix for the binary case (own work) 15
2.4 Workflow of the property inference attack 18
2.5 Two permutation equivalent neural networks 19

3.1 Workflow of the meta-model [GWY+18] 25
3.2 Example for rank inconsistency [CMR20] 29
3.3 Split of data in the first approach 31
3.4 Split of data in the second approach 32
3.5 Hierarchical approach 35

4.1 Accuracy and MSE for approach without ordinal classification 40
4.2 Confusion matrices for 10 and 20 class case for the approach without . . 41
4.3 Confusion matrix for target models in 5 class case for approach without ordinal classification 42
4.4 Confusion matrix for target models in 4 class case for approach without ordinal classification 42
4.5 Accuracy and MSE for interval approach 43
4.6 Confusion matrix for target models in 20 class case for interval approach . 44
4.7 Confusion matrices for 4 and 5 class case using the interval approach . . . 44
4.8 Accuracy and MSE for SOR approach 45
4.9 Confusion matrix for target models in 4 class case for SOR approach . . . 46
4.10 Accuracy and MSE for CORAL approach 47
4.11 Confusion matrices for 5 and 10 class case using the CORAL approach . . 48
4.12 Accuracy and MSE for CORN approach 49
4.13 Confusion matrices for 5 and 10 class case using the CORN approach . . 50
4.14 Accuracy for all five approaches for the different class cases 52
4.15 MSE for all five approaches for the different class cases (y-axis has a logarithmic scale) 53
4.16 Results of hierarchical approach with approach without ordinal classification 56
4.17 Results of hierarchical approach for interval approach 58
4.18 Results of hierarchical approach with SOR approach 60
4.19 Results of hierarchical approach with CORAL approach 62

4.20 Results of hierarchical approach with CORN approach 64
4.21 Summarized results of the hierarchical approach 66
4.22 Results of 50/50 women/men ration - 4 class model for the interval approach 69
4.23 Results of 50/50 women/men ration - 4 class model for the SOR approach 70
4.24 4-class model with SOR approach - models trained for 10 epochs 70
4.25 4-class model with CORAL approach - models trained for 1 epoch 75
4.26 Most effective approaches for the different splits and approaches 80
4.27 Result for different epochs with a 50/50 split 81
4.28 Result for different epochs with a 70/30 split 83

List of Tables

2.1 Notations of neural networks 7

3.1 Percentages of women in a training set divided into 4 equal classes 26
3.2 Labels for interval approach and 4-class problem 27
3.3 Ranges for interval approach and 4-class problem 27
3.4 Vectors for SOR approach and 4-class problem 27
3.5 Different train test splits 33
3.6 Data used for training the four hierarchical models 35

4.1 Percentage ranges of women for the different class cases 40
4.2 All results for RQ 1.1 54
4.3 Results 50 / 50 slit of women and men 73
4.4 Results 70 / 30 slit of women and men 78

Acronyms

AI Artificial Intelligence. 7

CORAL Consistent rank logits. 28, 29, 47–52, 61, 62, 65–68, 71, 72, 74–77, 80–82, 88, 93, 94

CORN Rank-consistent ordinal regression. 29, 30, 49–52, 63–68, 71, 72, 74–77, 80–82, 88, 93, 94

GD Gradient Descent. 11, 12

Leaky ReLU Leaky Rectified Linear Unit. 13, 14, 23, 93

ML Machine learning. 1, 7, 8, 17

MLP Multilayer Perceptron. 8–12

MSE Mean squared error. 5, 16, 21, 27, 36, 37, 39–53, 65–69, 71, 72, 74–83, 86, 88, 89, 93

ReLU Rectified Linear Unit. 13, 14, 25, 93

SGD Stochastic Gradient Descent. 12

SOR Simple ordinal regression. 27, 28, 45–47, 51–53, 59, 60, 65–72, 74–77, 80–82, 88, 93–95

Bibliography

[AMS+15] Giuseppe Ateniese, Luigi V. Mancini, Angelo Spognardi, Antonio Villani, Domenico Vitali, and Giovanni Felici. Hacking smart machines with smarter ones: How to extract meaningful data from machine learning classifiers. *International Journal of Security and Networks*, 10(3):137, 2015.

[ANK18] Jafar Alzubi, Anand Nayyar, and Akshi Kumar. Machine Learning from Theory to Algorithms: An Overview. *Journal of Physics: Conference Series*, 1142:012012, November 2018.

[AWS12] Inc. Amazon Web Services. Pre-trained Machine Learning models in AWS Marketplace, 2012. https://aws.amazon.com/marketplace/solutions/machine-learning/pre-trained-models, last visited: 2025-01-07.

[BP66] Leonard E Baum and Ted Petrie. Statistical inference for probabilistic functions of finite state Markov chains. *The annals of mathematical statistics*, 37(6):1554–1563, 1966. Publisher: JSTOR.

[BTBD20] Mathieu Blondel, Olivier Teboul, Quentin Berthet, and Josip Djolonga. Fast differentiable sorting and ranking. In *International Conference on Machine Learning*, pages 950–959. PMLR, 2020.

[Bur98] Christopher JC Burges. A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery*, 2(2):121–167, 1998. Publisher: Springer.

[CMR20] Wenzhi Cao, Vahid Mirjalili, and Sebastian Raschka. Rank consistent ordinal regression for neural networks with application to age estimation. *Pattern Recognition Letters*, 140:325–331, December 2020.

[Con23a] PyTorch Contributors. torch.nn.init — PyTorch 2.1 documentation, 2023. https://pytorch.org/docs/stable/nn.init.html#torch.nn.init.xavier_uniform_. last visited: 2023-10-13.

[Con23b] PyTorch Contributors. torch.nn.MSELoss, 2023. https://pytorch.org/docs/stable/generated/torch.nn.MSELoss.html, last visited: 2023-11-19.

[Con23c] PyTorch Contributors. torch.nn.NLLLoss, 2023. https://pytorch.org/docs/stable/generated/torch.nn.NLLLoss.html, last visited: 2023-11-19.

[Con23d] PyTorch Contributors. torch.nn.Sigmoid, 2023. https://pytorch.org/docs/stable/generated/torch.nn.Sigmoid.html, last visited: 2023-11-19.

[DG08] Jeffrey Dean and Sanjay Ghemawat. MapReduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113, January 2008.

[FJR15] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pages 1322–1333, Denver Colorado USA, October 2015. ACM.

[GWY+18] Karan Ganju, Qi Wang, Wei Yang, Carl A. Gunter, and Nikita Borisov. Property Inference Attacks on Fully Connected Neural Networks using Permutation Invariant Representations. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pages 619–633, Toronto Canada, October 2018. ACM.

[HN92] Robert Hecht-Nielsen. Theory of the backpropagation neural network. In *Neural Networks for Perception*, pages 65–93. Academic Press, 1992.

[JZP08] Jianlin Cheng, Zheng Wang, and Gianluca Pollastri. A neural network approach to ordinal regression. In *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, pages 1279–1284, Hong Kong, China, June 2008. IEEE.

[KB15] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015)*, pages 1–15, 2015.

[KC07] Barbara Kitchenham and Stuart Charters. Guidelines for performing systematic literature reviews in software engineering. Technical report, EBSE Technical Report EBSE-2007-01, 2007.

[KK16] Mohamed Al Kilani and Volodymyr Kobziev. An Overview of Research Methodology in Information System (IS). *OALib*, 03(11):1–9, 2016.

[ld07] Scikit learn developers. sklearn.preprocessing.LabelEncoder, 2007. https://scikit-learn/stable/modules/generated/sklearn.preprocessing.LabelEncoder.html. last visited: 2023-10-12.

[Lem12] Claude Lemaréchal. Cauchy and the gradient method. *Doc Math Extra*, 251(254):10, 2012.

[Mit97] Tom M. Mitchell. *Machine Learning*. McGraw-Hill series in computer science. McGraw-Hill, New York, 1997.

[MMR+17] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-Efficient Learning of Deep Networks from Decentralized Data. In Aarti Singh and Jerry Zhu, editors, *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pages 1273–1282. PMLR, April 2017.

[Moc89] Jonas Mockus. *Bayesian approach to global optimization: theory and applications*. Mathematics and its applications. Soviet series. Kluwer Academic, Dordrecht ; Boston, 1989.

[MP17] Marvin Minsky and Seymour A. Papert. *Perceptrons: An Introduction to Computational Geometry*. The MIT Press, September 2017.

[MSRR18] Ryan L. Murphy, Balasubramaniam Srinivasan, Vinayak Rao, and Bruno Ribeiro. Janossy Pooling: Learning Deep Permutation-Invariant Functions for Variable-Size Inputs. 2018. Publisher: arXiv Version Number: 3.

[OS10] Chitu Okoli and Kira Schabram. A Guide to Conducting a Systematic Literature Review of Information Systems Research. *SSRN Electronic Journal*, 2010.

[Ros58] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408, 1958.

[SCR23] Xintong Shi, Wenzhi Cao, and Sebastian Raschka. Deep neural networks for rank-consistent ordinal regression based on conditional probabilities. *Pattern Analysis and Applications*, 26(3):941–955, August 2023.

[SQ14] Umair Shafique and Haseeb Qaiser. A Comparative Study of Data Mining Process Models (KDD, CRISP-DM and SEMMA). *International Journal of Innovation and Scientific Research*, 12(1):217–222, 2014.

[SSA20] Siddharth Sharma, Simone Sharma, and Anidhya Athaiya. Activation functions in neural networks. *International Journal of Engineering Applied Sciences and Technology*, 4(12):310–316, 2020.

[SSSS17] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership Inference Attacks Against Machine Learning Models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 3–18, San Jose, CA, USA, May 2017. IEEE.

[TZJ+16] Florian Tramèr, Fan Zhang, Ari Juels, Michael K. Reiter, and Thomas Ristenpart. Stealing machine learning models via prediction APIs. In

Proceedings of the 25th USENIX Conference on Security Symposium, SEC'16, pages 601–618, USA, 2016. USENIX Association. event-place: Austin, TX, USA.

[UC 00] UC Irvine Machine Learning Repository. Census-Income (KDD), 2000. https://doi.org/10.24432/C5N30T. https://archive.ics.uci.edu/dataset/117. last visited: 2023-10-12.

[WH00] Rüdiger Wirth and Jochen Hipp. CRISP-DM: Towards a standard process model for data mining. In *Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining*, volume 1, pages 29–39. Manchester, 2000.

[ZKR+17] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabás Póczos, Ruslan R. Salakhutdinov, and Alexander J. Smola. Deep Sets. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 3394–3404, 2017.