FOUNDATION OF META-LEARNING FOR MULTI-DIMENSION CLASSIFICATION AND APPLICATION

by

FARID GHAREHMOHAMMADI

(Under the Direction of Hamid R. Arabnia)

ABSTRACT

Machine Learning (ML) has become an integral component of Artificial Intelligence (AI). The outputs of ML algorithms are explicable and resolve a variety of complex scientific and engineering issues. ML has been the subject of many research studies that tried to find the best way to solve a problem. The main goal of this dissertation is to look at typical problems with machine learning while it is learning. This research's primary objective is to address machine learning's limitations, such as a lack of training samples, inefficient use of time, and poor accuracy. In this dissertation, meta-learning has been the main way to make up for the lack of examples in the training process. Meta-learning (MTL) is the process of learning to learn, which provides a new path for scientists to generate a robust model. MTL offers various learning models and training procedures for the conventional classification step. Zero-shot learning (ZSL) is one of the most popular models. In ZSL, we learn to recognize unseen classes based on classes we've already seen. We begin by employing zero-shot learning to identify unseen animals and class categories. Then, we apply the solution to various data domains, ranging from computer vision to signal processing and from 2D images to 3D point cloud models. Lastly, we were inspired by zero-shot learning to come up with meta-semantic learning (Meta-SeL), an efficient classification algorithm that can recognize and classify 3D objects in 3D point cloud models.

INDEX WORDS: [Machine learning, Meta learning, Zero-shot learning,Hyper-Parameter Tuning, computer vision, Signal processing, 3D point cloud models]

FOUNDATION OF META-LEARNING FOR MULTI-DIMENSION CLASSIFICATION AND APPLICATION

by

FARID GHAREHMOHAMMADI

M.Sc. Tarbiat Modares University, Iran, 2013

A Dissertation Submitted to the Graduate Faculty of the
University of Georgia in Partial Fulfillment of the Requirements for the Degree.

DOCTOR OF PHILOSOPHY

ATHENS, GEORGIA

2022

FOUNDATION OF META-LEARNING FOR MULTI-DIMENSION CLASSIFICATION AND APPLICATION

by

FARID GHAREHMOHAMMADI

Major Professor:	Hamid R. Arabnia
Committee:	Thiab Taha Khaled Rasheed M. Hadi Amini

Electronic Version Approved:

Ron Walcott
Vice Provost for Graduate Education and
Dean of the Graduate School
The University of Georgia
August 2022

DEDICATION

I am dedicating this dissertation to three beloved angles of my life:

◇ **Prof. Arabnia**,

◇ **My Mom**,

◇ and **My Dad**.

ACKNOWLEDGMENTS

First and foremost, I would like to express my sincere appreciation to my incredible advisor, Professor Hamid R. Arabnia, for his unending support throughout my Ph.D. program, as well as for his guidance, advice, motivation, and vast knowledge. His counsel and assistance have allowed me to be the best version of myself. He assisted me in the classroom and beyond. Academically and personally, I could not have imagined a better advisor and mentor at the University of Georgia.

In addition to my advisor, I would like to thank the other members of my dissertation committee: Prof. Thiab Taha, Prof. Khaled Rasheed, and Prof. M. Hadi Amini, for their insightful comments and encouragement, as well as for the suggestions that prompted me to investigate further and narrow the scope of my research. I would also like to extend my deepest gratitude to Prof. Amini for his assistance with my research dissertation.

I'd like to thank my friends for the enlightening conversations, the late nights and early mornings we spent working together, and all the exciting activities and trips we've shared over the past four years. Special thanks to Farzan Shenavar Masouleh, who has been an excellent collaborator, coworker, and companion, and to Cheng Chen, who has been a devoted and helpful collaborator.

Last but not least, I would like to express my gratitude to my family: my mother and father, my brothers and sisters, for their unwavering mental and spiritual support throughout my Ph.D. studies and my life in general.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Implementing advanced artificial intelligence solutions is a crucial and unavoidable aspect of data analytics when working with real-world data. [25], [130]. Techniques and algorithms for machine learning (ML) are among the most prominent AI technologies. Scientists use machine learning to solve complex engineering and science problems. [24]. Numerous solutions, such as deep learning and evolutionary algorithms, have been proposed to address the issues that conventional ML algorithms cannot resolve. In contrast to conventional machine learning algorithms, which are extremely limited to static processing steps and specific data types and structures, these solutions take a different approach, employing data processing steps that mimic the processing capabilities of the human brain.

Human intelligence provides an influential learning process that uses its own and others' experiences to recognize certain patterns based on learned (observed) experiences [187].

The algorithm that allows computers to learn and store this ability is called Machine Learning.

According to Tom Mitchell's 1997 definition [132], machine learning is defined as a computer program M (Machine) that learns from experience E (Experience) with respect to task T and is evaluated by P (Performance), which is enhanced by learning from experience. E. ML has enabled computers to solve problems by learning from experience. ML works with standard datasets (experiences) when we have balanced classes and enough samples for each category. Additionally, it works best with a certain amount of data and associated features. Overall, machine learning provides a black box procedure in which we create a new model (logic or rules) based on data or experience.

1.1 Applications of machine learning

Machine learning has shown promising results in improving task performance and predicting class labels. Recently, computers have been programmed to perform a wide range of tasks that are as accurate as, if not more accurate than, humans citestupar2021importance. Machine learning-based systems are widely used all over the world and are recognized as one of the foundations of modern technology. Intelligent systems based on machine learning are programmed to offer users who are considering purchasing products on online markets or choosing which TV shows they would like to watch. Every query search on the internet uses a system that learns from the language of the query and customizes the results based on that.

Furthermore, many other machine learning-based systems are used in the majority of the high-tech industry: 1) Object recognition on and near the road for self-driving cars, drones, and humanoid robots during shipment delivery, 2) speech and speaker recognition techniques, 3) natural language processing techniques for chatbots and service-oriented robots that communicate with users 4) Face recognition and image classification for surveillance systems 5) Medical imaging for diabetic retinopathy detection on fundus images and tumor detection on X-ray images, among other studies.

1.2 Machine learning challenges

Universal AI algorithms provide an abundance of distinguishing characteristics for class labels. However, this leads us to the NP-hard problem known as the curse of dimensionality (CoD). With a large number of features, large datasets are produced, for which machine learning cannot produce the optimal model. This procedure is also time-consuming, rendering real-time processing in any optimization and real-time execution environment, such as a visual computer, impossible. Deep learning and evolutionary algorithms that prevent the limitations of machine learning could be viable solutions for this CoD. We examined previously developed evolutionary algorithms for enhancing real-time image processing and argue that they are the most promising solutions to the CoD problem **ch1 farid**, **ch2 farid**.

The lack of sufficient samples per class, and in some cases no samples at all due to the nature of certain objects, is a significant problem caused by online data and other AI data collection approaches.

Using machine learning algorithms, researchers can learn from both supervised and unsupervised data. The majority of the data collected would be offline and would not change over time.

Consequently, the overall behavior of future data is sufficiently ambiguous for us to process. It is not possible to learn the entire behavior using the traditional machine learning, evolutionary algorithms, and optimization algorithms discussed earliet [182].

This dissertation is primarily concerned with addressing the difficulty of the learning process when dealing with the absence of samples for a given category. We discuss an alternative solution, the Meta-Learning (MTL) algorithm, along with its applications and potential to solve the emerging issue.

1.3 Contributions of this dissertation

This dissertation's primary contribution is to provide a linear optimal function whose hyperparameters are tuned to achieve the most promising classification result for zero-shot classification that outperforms the most recent work. We present a comprehensive procedure for meta-Mapping, an enhanced semantic auto-encoder (OSAE).

Consequently, we propose Meta-sense, learning to sense, a new optimized semantic auto-encoder (SAE) based on sensing as opposed to visual images. We apply the algorithm to a human motion dataset to learn from observed categories in order to achieve high performance in a healthcare application for recognizing unseen human motion. This human motion dataset was captured by sensors, whereas the feature vectors represent smartphone signal outputs.

Finally, we present Meta semantic learning (Meta-SeL), a method for classifying 3D point cloud models that outperforms other current methods. Additionally, it makes efficient use of resources. In addition, Meta-SeL performs linear computations on CPUs rather than GPUs. As shown in figure 1.1, we initially proposed a tuned-ZSL for animal recognition [148] and another concept of Meta-sense in [144], applied to signal processing for human motion detection utilizing ZSL [149].

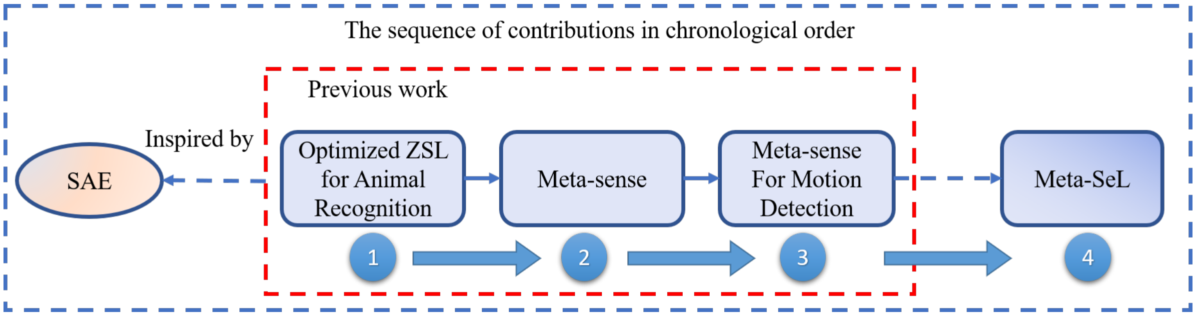


Figure 1.1: The outline of contributions: ZSL: zero shot learning, Meta-SeL: Meta- Semantic learning

Our prior work, which was motivated by the Sylvester equation, was utilized to identify the optimal learning projection function (textitW) from vector space to semantic space. The following are the contributions of Meta-SeL:

⋄ We design a novel framework, Meta-SeL, that is suitable for 3D point cloud models.

⋄ We demonstrate that the training and testing phases require only one *epoch* to produce a promising result. Meta-SeL has a high convergence rate, making it extremely time and cost-efficient.

⋄ We show how Meta-Sel, a linear projection-based learning approach, has a significant positive impact on point cloud classification, specifically shapeNet Core dataset classification.

⋄ We indicate that Meta-SeL is resistant to jittering noise and translation.

⋄ We demonstrate that Meta-SeL can predict each label of a 3D Model classification accurately and precisely.

⋄ We illustrate that Meta-SeL generates semantic information for each entry instantaneously.

1.4 Dissertation outline

In this section, we show a detailed flowchart of this dissertation in figure 1.2 and then package the flowchart steps into chapters which are shown in 1.3. Further, we discuss an abstract explanation for each chapter as follows:

♦ **Chapter 2: Introduction to Meta-learning:** We address one of the known ML limitations, a lack of samples,and investigate an advanced machine learning algorithm that uses Meta-Learning to optimally solve this problem (MTL). MTL provides three important extensions for new data and complex problems.

The first is few-shot learning (FSL), which utilizes k training sessions. One-shot learning (OSL) is a subset of few-shot learning in which each training class has only one opportunity. Last, but certainly not least, is zero-shot learning (ZSL). Although good work has been done with FSL and OSL, ZSL is a promising extension of meta-learning for situations in which researchers lack sufficient knowledge about unseen classes and insufficient data.

♦ **Chapter 3: Meta-learning for computer vision for unseen classes:** We investigate image recognition for unseen categories in a given dataset using limited training data. We employ a zero-shot learning (ZSL) algorithm to achieve this. We propose an optimized semantic auto-encoder (OSAE) and study the impact of parameter tuning on the performance of semantic auto-encoders (SAE). We continue to investigate the meta-learning parameter tuning problem, with a focus on

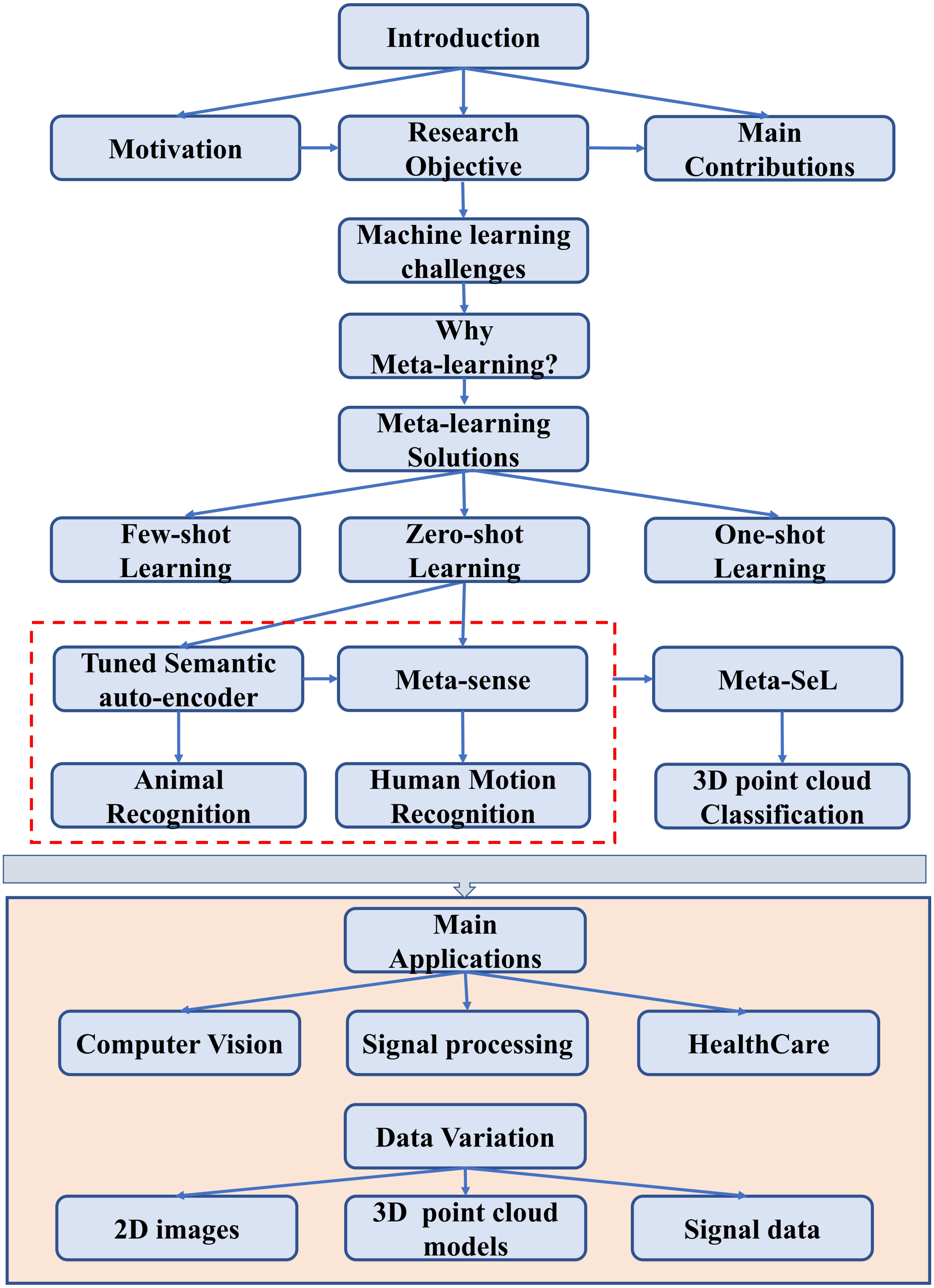


Figure 1.2: The progress of the work from Meta-sense to Meta-Semantic Learning (Meta-SeL)

zero-shot learning. We enhance tuned-SAE precision by combining multiple embedded parameters. We also discussed the benefits and drawbacks of parameter tuning and its application to image classification.

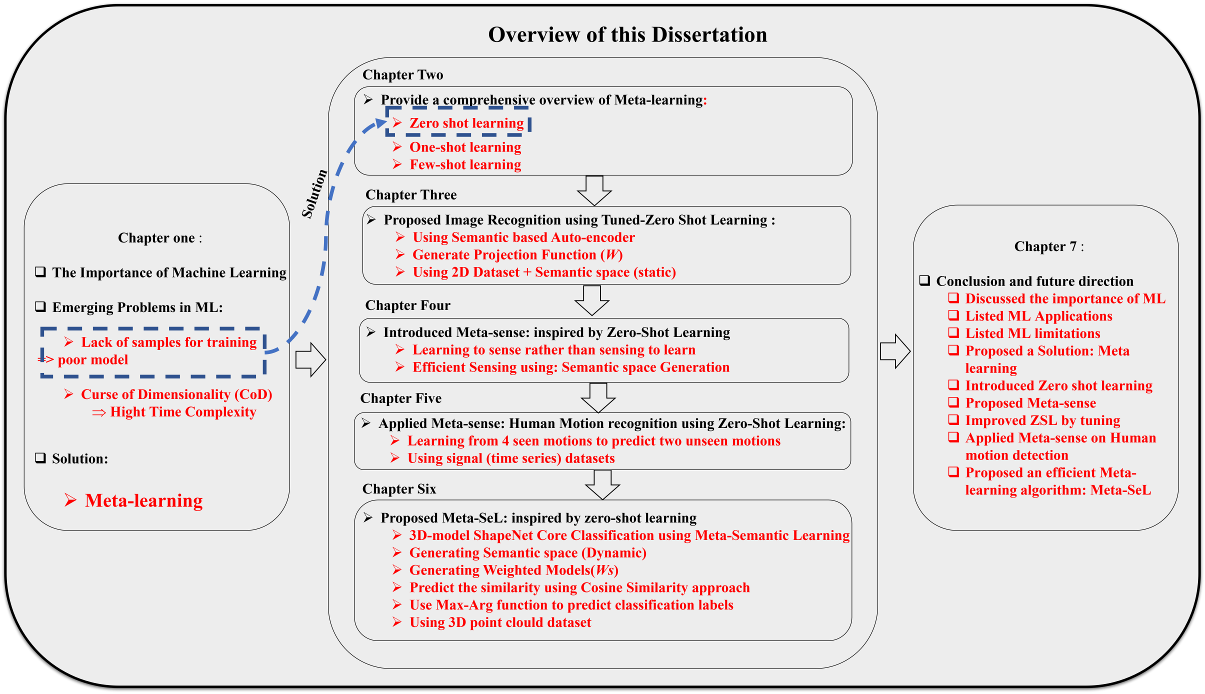


Figure 1.3: The general overview of this dissertation

♦ **Chapter 4: Meta-Sense Solution:** We investigate the potential use of meta-learning algorithms to enable device-free sensing. This is referred to as "meta-sensing," which is essentially learning to sense through the discovery of available information as opposed to deploying additional sensors. We are particularly interested in the application of *zero-shot learning*, a specific learning algorithm that does not require *priori* information.

♦ **Chapter 5: applied Meta-sense for human motion recognition in the application of healthcare:** We overcome the limitation of machine learning in the field of human action recognition by recognizing previously unseen classes. Unseen motion recognition addresses some of the emerging difficulties in computer vision applications, such as analyzing actions in security video with insufficient training data. Human behavior and action recognition in healthcare heavily rely on motion recognition. In this chapter, we propose using meta-sense, a novel action and motion recognition solution based on zero-shot learning to monitor the daily behavior of patients and maintain their medical records. To evaluate the efficacy of the proposed solution, we utilize a dataset from the

UCI machine learning repository. This dataset has allowed us to apply zero-shot learning to the recognition of human motion and action.

♦ **Chapter 6 : Meta semantic learning (Meta-SeL):** We investigate the semantic dimension of given 3D data points and create an effective classification algorithm called Meta-Semantic Learning (Meta-SeL). Meta-SeL is an integrated framework that provides time- and cost-efficient running performance by utilizing two input 3D local points (input 3D models and part-segmentation labels). In addition, it provides a precise projection model for a variety of 3D recognition tasks that produce promising results.

CHAPTER 2

INTRODUCTION TO META-LEARNING [1]

---

[1] This chapter is a reprint and extended version of the following paper, with the permission of publisher:
Farid Ghareh Mohammadi, M. Hadi Amini and Hamid R. Arabnia, An Introduction to Advanced Machine Learning: Meta-Learning Algorithms, Applications, and Promises ,Optimization, Learning, and Control for Interdependent Complex Networks. pp. 129-144, 2020.

2.1 Abstract

To address the curse of dimensionality (CoD) issue, the majority of optimization algorithms possessed by evolutionary algorithms, such as the Artificial Bee Colony, guarantee optimal performance on all data types, including supervised and time series. However, these algorithms are not intended to address emerging learning issues. A lack of sufficient samples per class is one of the major challenges posed by newly emerging data. Moreover, due to the proliferation and dispersion of data, traditional machine learning algorithms can not generate a robust model from limited distributed data. Despite using a strict model or embedded engine to train and predict test class labels, machine learning fails to predict unseen class labels. This chapter discusses these obstacles in depth. We come up with a solution by looking at the Meta-Learning (MTL) algorithm and how it can be used to solve new problems.

2.2 Introduction

In [138], [139], we investigated the theoretical and application aspects of optimization processes for solving large-scale problems and overcoming machine learning limitations. The majority of optimization algorithms presented in [138], [139] guarantee optimal performance of solutions for dimension (CoD) problems.

Algorithms for machine learning allow researchers to learn from supervised and unsupervised data. When the data collection phase is complete, we will have a vast amount of offline data that is ready for analysis and does not change over time. Therefore, the total behavior of future data is sufficiently imprecise for us to process it. Traditionally, it is not possible to teach an entire behavior using traditional machine learning techniques [182].

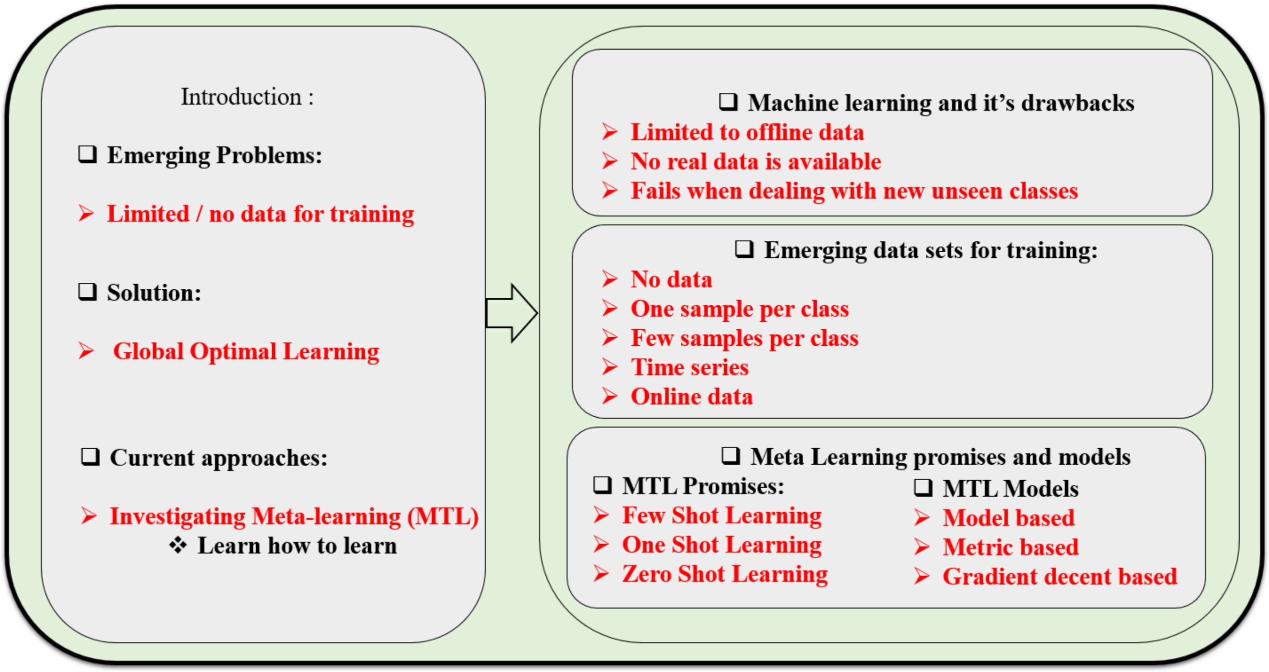


Figure 2.1: Overall structure of this study

The last decade has been devoted to the study of advanced research paradigms in an effort to solve the learning process. Researchers intend to learn by focusing attention on previous tasks or experiences. Meta-learning is one of the most promising educational paradigms (MTL). Prior research examined MTL techniques that learn to modify a function or learning rule [168], [182].Adaptation is a distinction between MTL and conventional machine learning. MTL is the process of learning how to learn. It uses past data to determine the parameters and learning process of a prior model, i.e., an algorithm. In contrast to conventional learning algorithms, MTL investigates how to choose the correct bias in a non-fixed manner [203]. MTL investigates a scenario in which a set of tasks ($\mathcal{T}_i$) are made available in advance. It cannot, however, handle problems in a sequential or dynamic manner.

Early research studies introduced sequential learning, in which tasks are revealed one after the other repeatedly. The goal of learning is to learn as autonomously as possible in order to achieve zero-shot learning with non-task-specific adaptation. We contend that neither environment is ideal for studying continuous lifelong learning. MTL addresses learning to learn but ignores the problem's sequential and non-stationary aspects. We motivate and present the MTL setting in this chapter, where the agent uses past experiences in a sequential setting to learn good priors while also adapting quickly to the current task at hand [29], [74].

2.3 Machine learning : challenges and drawbacks

To produce a high-performance model, machine learning requires two important prerequisites: (1) collecting enough data for a training dataset, and (2) identifying patterns in the given dataset. Machine learning produces poor results when there is no pattern. Furthermore, machine learning is effective when mathematics alone cannot solve a problem. Machine learning is no longer optimal if mathematics can learn the data and solve the problem. However, the same result would be obtained at an unnecessary cost. Machine learning is only required when mathematics alone cannot solve the problem.

Prior research on the learning process, regression, and optimization problems attempted to learn and categorize the behavior of input data in order to develop high-performance algorithms. Machine learning (ML) is commonly applied to both supervised and unsupervised problems. ML deploys different algorithms, such as online learning, multi-task learning and supervised algorithms, including rule-based [21], function-based [34], [85]. Some are used to transform data, such as dimension reduction for optimization, while others are used to construct classifiers, such as supervised algorithms, and still others are used for prediction, such as regression, etc. Machine learning still has small flaws when it comes to data that changes over time. This means that it can't accurately think about future and unseen classes to get a general idea and knowledge from data.

Traditionally, machine learning refers to a machine that learns from only input data and predicts new data according to the rules of the equation. $\mathcal{P}_i \times \mathcal{D} \longrightarrow \mathcal{M}$, where $\mathcal{P}_i$ stands for the specific supervised algorithm parameters, $\mathcal{D}$ represents the space of training data distribution, and $\mathcal{M}$ defines the space of generated models which will be applied on test data to evaluate the supervised algorithm performance.

Figure 2.2 depicts a number of machine learning approaches and algorithms that provide diverse applications for a variety of data types, including offline versus online data; labeled versus unlabeled data; multi-model data versus single model data; and multi-domain data versus single domain data. As demonstrated, machine learning has significant drawbacks in that it cannot process vast quantities of data simultaneously. In addition, each sample is treated as a new model, and each model is updated based on its predecessors. Therefore, the order of the input data is essential.

In addition, figure 2.2 depicts the relationship between traditional and advanced machine learning. In conventional machine learning, we must deal with offline data and ground truth in limited quantities. However, in the world of technology, where data growth has exploded and is generated everywhere technology exists, it is essential to comprehend the patterns and rules that govern entire data sets as well as the trend of generated data for a particular domain. In order to accomplish this, we must divide data into three major categories: time-series data, offline data, and online data. All of these classes are shown in three different ways: with supervised and unsupervised data, multi-model data, and multi-domain data.

Transfer learning and online learning provided by machine learning are compatible with learning from subsequent tasks and classes. Transfer learning is the process of transferring knowledge from one task to another and learning from non-randomness. In addition, meta learner (stacking) is a bootstrap algorithm that learns data by randomly sampling a given data set and generating new data sets. Before choosing a meta learner to figure out the class of the current instance, the meta learner uses multiple supervised algorithms.

We would like to propose meta-learning to solve preconditioned machine learning requirements. We intend to transition from traditional machine learning to meta-learning in order to overcome the challenges discussed in the following section. MTL can learn from a training dataset even when there are few or no classes of test datasets available.

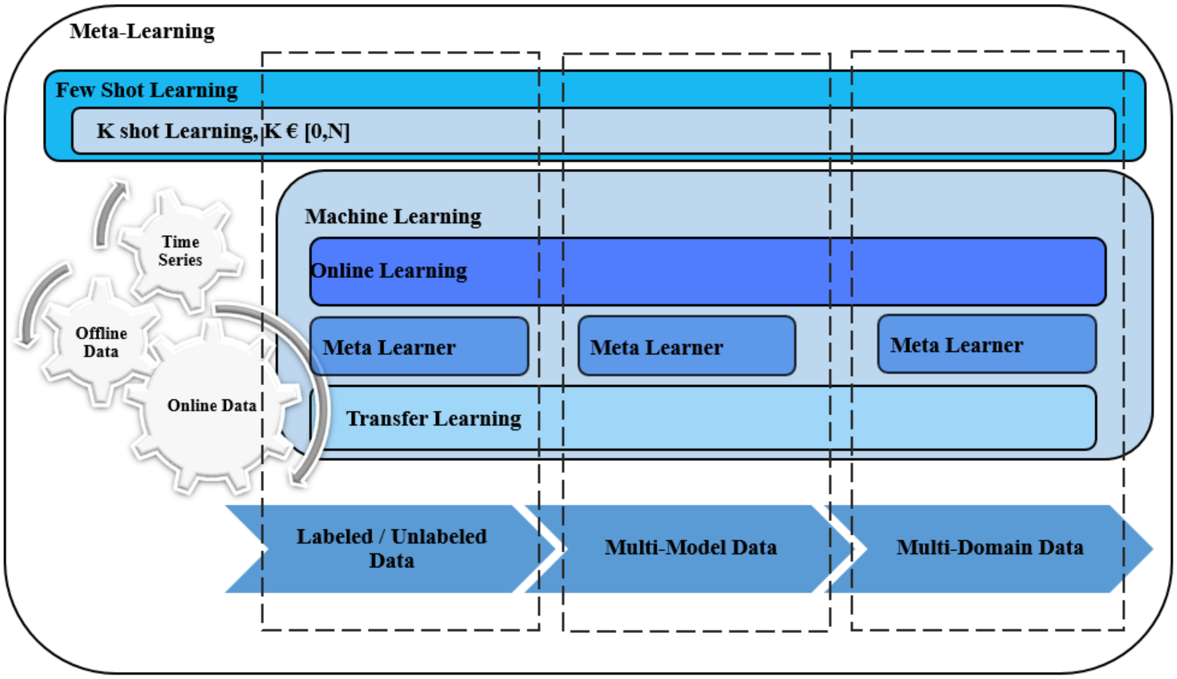


Figure 2.2: The general overview of Learning against emerging data

2.4 Why do we need Meta-learning?

Meta-learning, also known as learning to learn, has recently gained popularity in the artificial intelligence community. Meta-learning is extremely prevalent in nature, has a long history in cognitive science and psychology, and has been extensively studied in neuroscience. Meta-learning, or learning to learn the process from previously seen categories, is the ultimate key to creating a high-performance model, and this process is essential in artificial intelligence. Effectively transitioning from machine learning to meta-learning involves inferring an optimized function directly from data. This expedites the acquisition of sequentially new tasks based on previously observed categories. A traditional limitation of machine learning, namely machine learning's reliance on a training dataset, Meta-learning seeks two primary solutions: supervised and unsupervised classification. For starters, it allows us to train a high-performance model for n-way classification using only a few samples per class. This makes use of a meta-learner (i.e., the outer learner), which is then updated by a learner (i.e., the inner learner) with each iteration figure 2.3. Second, it assists us in learning from seen categories in order to recognize objects from unseen categories.

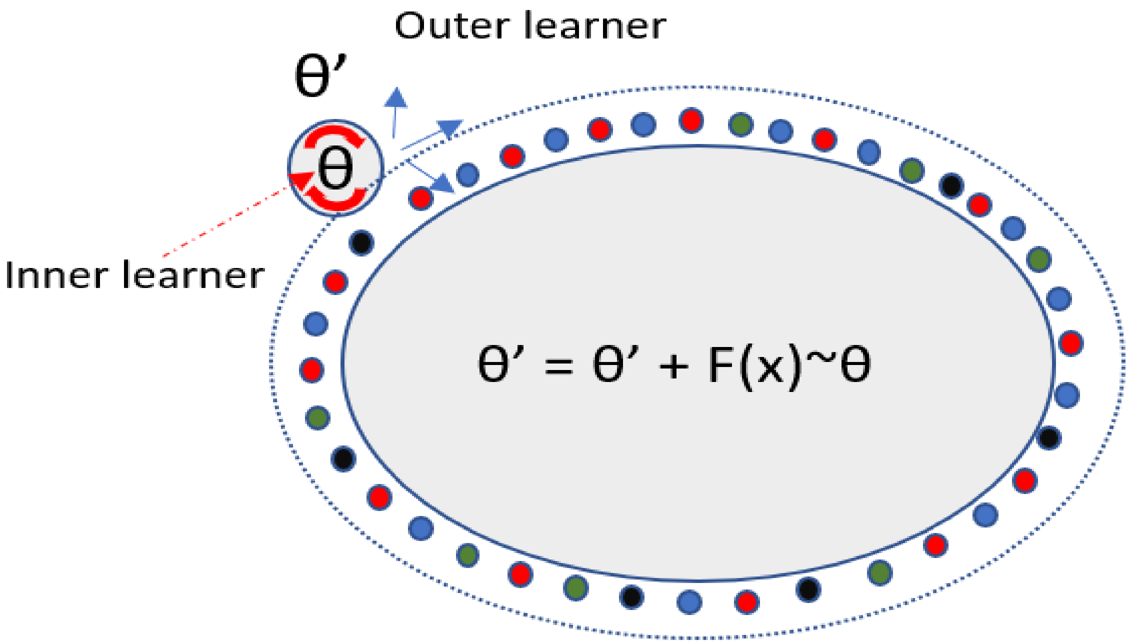


Figure 2.3: Meta learning: inner and outer learner

In supervised classification, the meta-learner is updated once through the generation of an efficient projection function, which yields promising results when evaluating unseen classes. Meta-learning resembles a task-based learning process by adding an additional step to prevent overfitting and enhance meta-learner performance in order to provide a generalized function. This chapter provides a comprehensive review of historical and contemporary meta-learning approaches. Meta-learning should be put first because traditional machine learning doesn’t have a core learning process based on a training dataset. It is noteworthy that Meta-learning has different meanings according to different resources.

The meta-learner seeks a parametric synaptic learning rule using learning algorithms that produce networks that can learn to perform challenging tasks. The hyper-parameters of the network have been fine-tuned to enable this meta-learner to complete classification tasks. This hyper-parameter tuning optimization (HPO) makes use of gradient descent and one of the most well-known evolutionary algorithms, the genetic algorithm (GA). Evolutionary algorithms have also been widely utilized for optimization. A genetic algorithm is motivated by natural biological evolution [23], [182]

Meta-learning is also used to explain machine learning algorithms such as voting, stacking, and ensemble learning algorithms such as random forest.

Meta-learning enables the development of a meta-learner that learns from a simple learner. Literally, MTL refers to a procedure in which the learning process occurs on two levels with distinct time scales. Following the construction of a simple learner, the meta-learner is trained iteratively.

2.5 Meta-learning algorithms

In the field of inductive reasoning, knowledge transfer refers to the process of learning from prior learning experiences. The most appealing application of knowledge transfer is meta-learning, which is the process of learning to ignore specific problem details in favor of seeking overall learning harmony. The process of learning to learn is a promising direction in advanced machine learning that has attracted empirical interest in recent years across a variety of disciplines, including human computer interaction, robotics, and, of course, computer vision. In addition, meta-learning is utilized in the reinforcement learning framework, in which the meta-learner gains knowledge through a series of experiences. Most meta-learning algorithms, on the other hand, work in a supervised learning framework. During the training phase, the learned model gets a projection estimation function that maps input data to targets [181] [183].

Meta-learning (MTL) is originally presented in [182] and [23]. The most recent research studies attempted to reimplement MTL after a ten-year hiatus. MTL is a machine learning system that acquires knowledge from diverse input data. MTL methods must acquire new tasks more rapidly by utilizing prior knowledge. MTL does not account for prior experience separately. MTL is an acronym for "learning to learn." MTL is a novel learning algorithm that generates new research challenges and questions. Transfer learning is an extension of one of the multitask learning algorithms. The MTL offers three different learning solutions as illustrated in Figure 2.6. Few shot learning (FSL), one-shot learning (OSL), and zero-shot learning. In comparison to conventional machine learning algorithms, FSL and OSL produce extremely precise results. However, they face a significant obstacle that prevents them from achieving optimal results. The limitations of ZSL have been overcome by employing domain semantic space, in which all information has been transformed into a new dimension that is aligned with the learning process, as depicted in figure 2.6.

2.5.1 Model-based MTL

Model-based MTL is based on a model and does not employ a conditional probabilistic method, making it the best fit for a fast learning model that updates its hyper-parameters so rapidly by training on a small number of examples. The process of updating their hyper-parameters is executed by an internal architecture or an external meta-learner. The goal of model-based MTL is to accelerate the learning process by having one neural network interact with multiple sequential neural networks. In other words, as shown in the figure 2.4, it attempts to learn a model for each label by pixel by pixel value. In other words, this model's algorithms try to train a recurrent model like the work presented [168], which proposed long short term memory (LSTM). Hochreiter and Schmidhuber introduced the LSTM theory for the first time in 1997. Model-based algorithms sequentially process the data set, analyzing each instance individually. Due to the fact that model-based algorithms use RNN for learning, they are the most effective models compared to other models[76]. Nagabandi *et al* [154] proposed online deep learning using MTL towards Continual adaptation for model-based reinforcement learning. Santoro *et al* [179] proposed memory-augmented neural network using MTL.

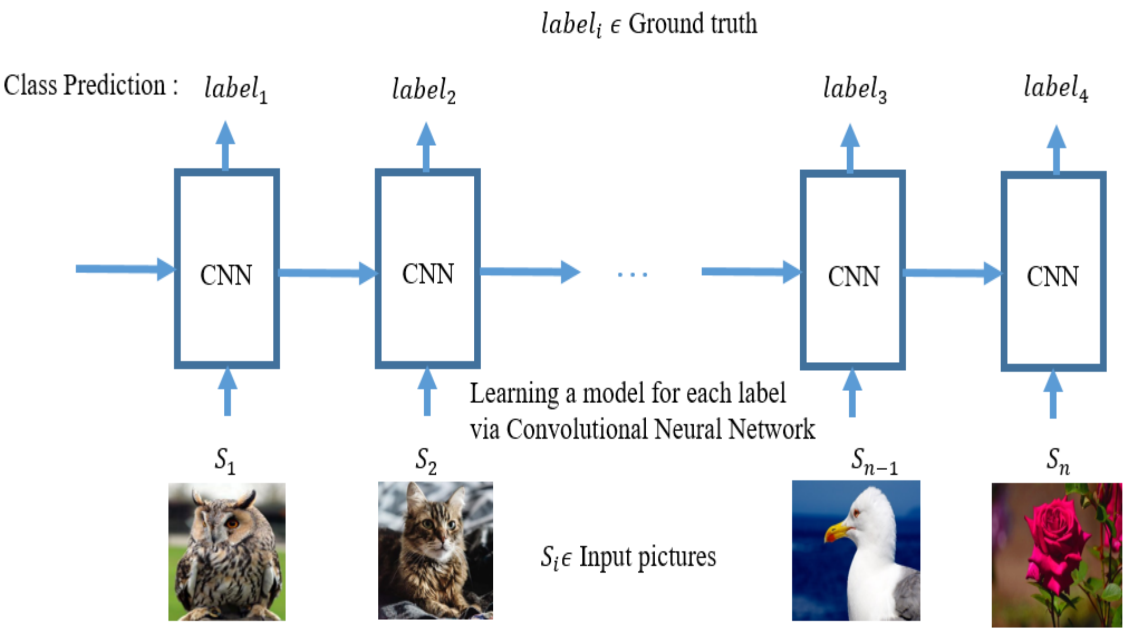


Figure 2.4: Model-based MTL

2.5.2 Metric-based learning

Metric-based learning takes advantage of metric space learning, which results in efficient data processing and is suitable for few-shot learning. Consider our objective to be image classification. As model-based learning attempts to learn each image pixel by pixel, which is time-consuming and laborious, metric-based learning circumvents this limitation by comparing two images provided to the network. The output for each input is a vector that compares the two vector states to determine if they are similar. The process of metric-based learning is shown in figure 2.5. One of the most common applications of metric-based learning is the Siamese network presented in [97]. Koch *et al* introduced a superior Siamese neural network (SNN) for one-shot learning. Comparing the input images using a twin or half-twin network is the concept underlying SNN. One of the inputs has already been computed, so all that remains is to compute the vector by traversing the layers of the second image. Then, SNN tries to figure out how far apart they are. If the result is small, the entities are similar; if not, they are different. Another application of metric-based learning is [204], where Vinyals *et al* proposed a matching network(MN) for one-shot learning.

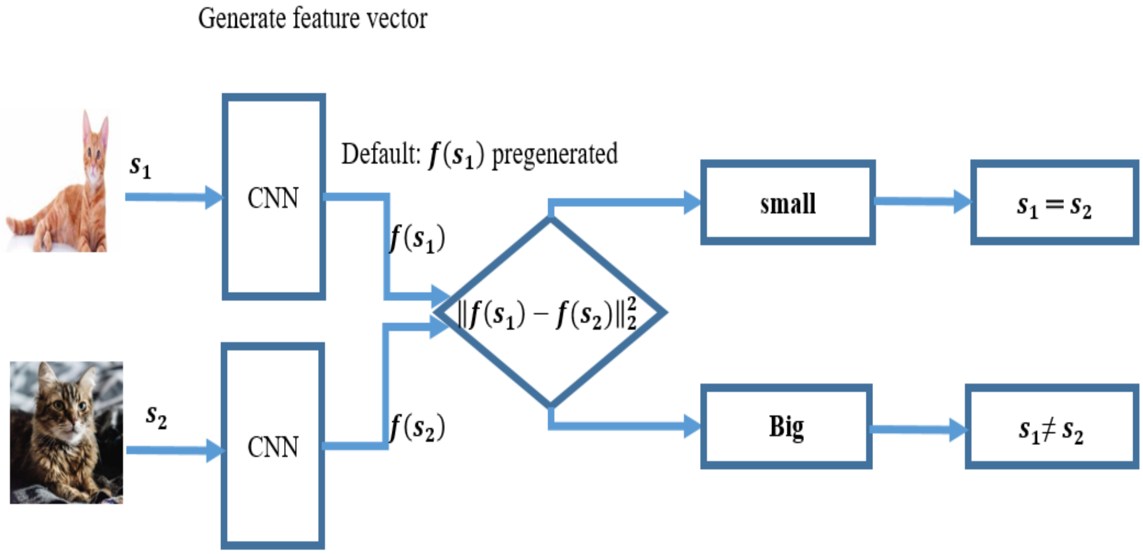


Figure 2.5: Metric based MTL

2.5.3 Gradient decent-based learning

This MTL model is also known as an optimization-based model for parameter (θ) tuning. The idea is to use stochastic gradient descent (SGD) and update the parameters to be a universal learner for each new given sample. Because it does not rely on a small number of samples, it may not converge to a local optimal.

Although the gradient-based learning model works well, it still has some drawbacks. Ravi and Larochelle [168] addressed these problems carefully and provided LSTM-based MTL to overcome those problems. Finn [54] presented MAML to improve the accuracy of LSTM-based MTL.

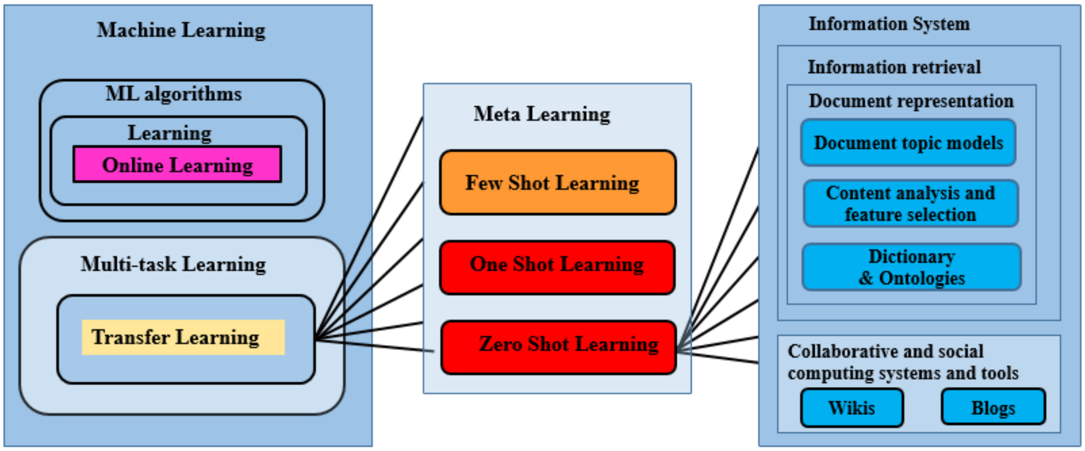


Figure 2.6: The relation among Machine Learning, Meta Learning, and Information system

Table 2.1: Abbreviation of words

Abb	Definition
CNN	Convolutional neural network
EGNN	Edge-labeling graph neural network
fSL	Few-shot learning
GSP	Goal-conditioned skill policy
LSL	Low-shot learning
MAML	Model-agnostic MTL
MANN	Memory-augmented neural network
MIL	Meta Imitation Learning
MTL	Meta-learning
ML	Machine learning
MN	Matching Network
OSL	One-shot learning
PN	Prototypical Network
RN	Relation network
RNN	Recurrent neural network
SAE	Semantic AutoEncoder
SNN	Siamese neural network
TL	Transfer learning
ZSL	Zero-shot learning
ZSL-FGVD	ZSL Fine-Grained Visual Descriptions
ZSL-H	Zero-shot learning by mitigating the Hubness problem
ZSL-KT	Zero-shot learning and knowledge transfer

Table 2.2: An overview of previous studies on Meta-learning

Paper	Meta-learning	Proposed method	Meta-learning Models	Conference / journal	Domain	Year
[54]	Few-shot learning	MAML	Gradient decent based	ICML	Image classification	2017
[19]	Few-shot learning	MAML++	Gradient decent based	ICRL	Image classification	2019
[56]	Few-shot learning	Probabilistic MAML	Gradient decent based	NIPS	Image classification	2018
[190]	Few-shot learning	PN	Metric based	NIPS	Image classification	2017
[237]	Few-shot learning	HML	Model based	arXiv	Image classification	2019
[197]	Few-shot learning	RN	Metric based	CVPR	Image classification	2018
[168]	Few-shot learning	LSTM-Metalearner	Gradient decent based	ICLR	Image classification	2017
[96]	Few-shot learning	EGNN	Model based	CVPR	Image classification	2019
[208]	Few-shot learning	LSL	Metric based	CVPR	Image classification	2019
[237]	One-shot learning	HML	Model based	arXiv	Image classification	2019
[179]	One-shot learning	MANN	Model based	ICML	Image classification	2016
[204]	One-shot learning	MN	Metric based	NIPS	Image classification	2016
[54]	One-shot learning	MAML	Gradient decent based	ICML	Image classification	2017
[57]	One-shot learning	MIL	Gradient decent based	CoRL	Visual imitation	2017
[97]	One-shot learning	MIL	Metric based	ICML	Image recognition	2015
[169]	Zero-shot learning	ZSL-FGVD	Metric based	CVPR	Image classification and retrieval	2016
[64]	Zero-shot learning	ZSL-H	Metric based	ICLR-workshop	Hubness problem	2015
[197]	Zero-shot learning	RN	Metric based	CVPR	Image classification	2018
[99]	Zero-shot learning	SAE	Metric based	CVPR	Image classification	2017
[197]	Zero-shot learning	ZSL-RN	Metric based	CVPR	Benchmark classification	2018
[136]	Zero-shot learning	Meta sensing	Metric based	ACM Buildsys	Device Free Human Sensing	2019
[41]	Zero-shot learning	ZSL-KT	Metric based	arXiv	Music classification	2019
[158]	Zero-shot learning	GSP	Metric based	CVPR-workshop	Visual imitation	2018
[157]	Zero-shot learning	GSP	Metric based	ICLR	Visual imitation	2018

2.6 Meta-learning solutions

Learning to learn is an advanced process that provides three solutions: few-shot learning(FSL), one-shot learning (OSL), and zero-shot learning(ZSL). Figure 2.7 presents a general view of each promise. we have three layers: input data, meta-training, and meta testing. Input data for FSL and OSL are the same type, particularly images for particular image classification aims. Further, ZSL becomes an independent learning MTL algorithm that evaluates input data based on domain semantic space and visual information of that domain.

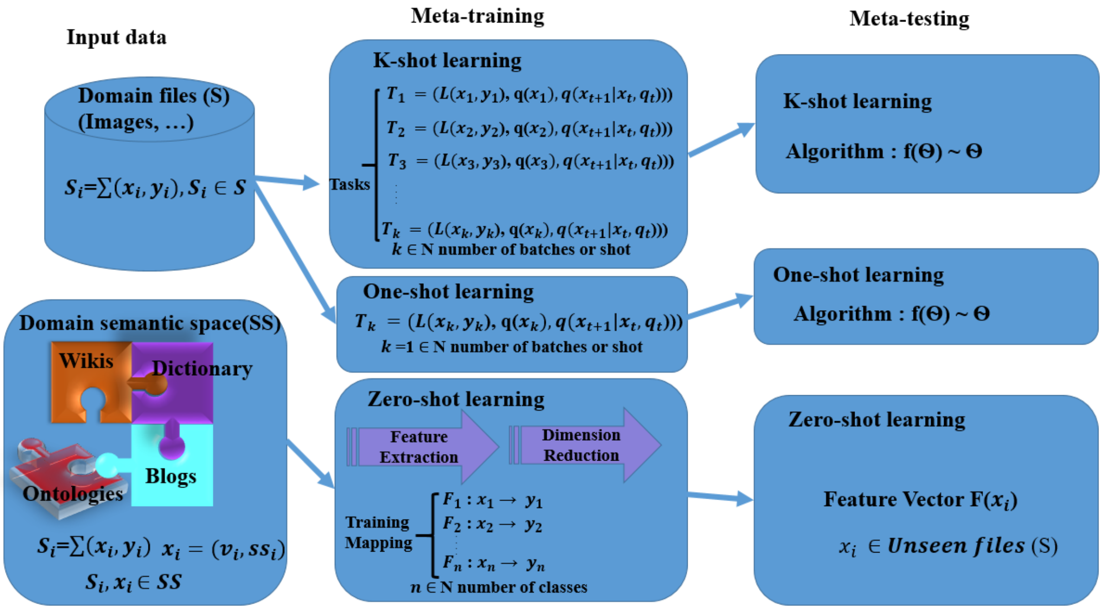


Figure 2.7: Structure of meta-learning models

FSL attempts to learn k-shot tasks in the second layer, implying that MTL is trained using k distinct shots in meta training datasets. Prior to the beginning of the learning process, K-shots were created. MTL is thus a type of bootstrap algorithm, but in the k-shot dataset, only K instances per class are available. The bootstrap algorithm, on the other hand, attempts to split given datasets at different rates while keeping the number of classes constant. MTL attempts to determine the loss of each shot using the loss function.

In addition, OSL attempts to learn the task using k=1-shot learning, meaning it has only one shot at a time. In other words, when bootstrapping is initiated, only one sample from each class is chosen for the training set. OSL is a subset of K-shot or few-shot learning, as should be noted. Both FSL and OSL employ the following equation [54].

$\mathcal{T}_i = \sum_{i=1}^{k}(\mathcal{L}(x_i, y_i), q(x_i), q(x_{t+1}|x_t, q_t))$

Furthermore, unlike FSL and OSL algorithms, ZSL attempts to work with domain semantic space rather than domain files such as images in the second layer. The goal is to find the best learned projection function from semantic to vector space. ZSL attempts to map extracted semantic features to a new space known as vector space.

Finally, the final layer is meta-testing, which is in charge of predicting and analyzing the given test data. The first two algorithms use f(θ) to predict seen data; however, ZSL attempts to solve the problem by mapping unseen data to the new vector space.

2.6.1 Few-shot learning

The first and most common MTL solution is few-shot learning (FSL). Few-shot classification is a supervised learning extension of MTL. Lake *et al* [106] addressed the limitations of traditional machine learning algorithms and proposed an algorithm for learning every concept from a few shots of that meta training dataset. MTL attempts to re-sample the given input dataset for the meta-training process, using only K samples per class. To put it another way, the meta-training process is completed by learning k shot from meta-training sets chosen by replacement. Despite outperforming traditional machine learning algorithms, few-shot learning faces an explanatory challenge known as task ambiguity. This issue arises when a small task, generated from a large input data set, learns through few-shot learning. After undertaking a new task that appears to be too ambiguous to establish a single model for that task that covers a large number of samples.

The majority of MTL algorithms leverage few-shot learning. FSL has decent important extension in which Finn *et al* proposed model-agnostic MTL (MAML) [54], which adapts to new tasks via gradient descent-based MTL. In [56], Finn *et al* re-sampled models for a new task using a model distribution. This chapter extends MAML to conduct a parameter distribution that is trained through different lower bound. In [56] Finn et al addressed the ambiguity problem by proposing probabilistic MAML.

The second important extension is Online learning which is the learning process of training data sequentially and continuously. The next one is online MTL. Finn *et al* [55] proposed online MTL based on the regret-based meta-learner. Kim *et al* [96] proposed EGNN, which applied a deep neural network on a certain model, edge-labeling graph. Zou and Feng [237] introduced a new type of MTL that works based on a hierarchical structure, called Hierarchical MTL(HML). HML overcomes previous MTL limitations which are limited to the tasks where training sets and identical output structure. HML enables MTL to optimize the adaptability of meta-models to tasks, that are similar. Figure2.8 provides a general view of few-shot learning, one-shot learning and 2-shot learning, and generalized k-shot learning.

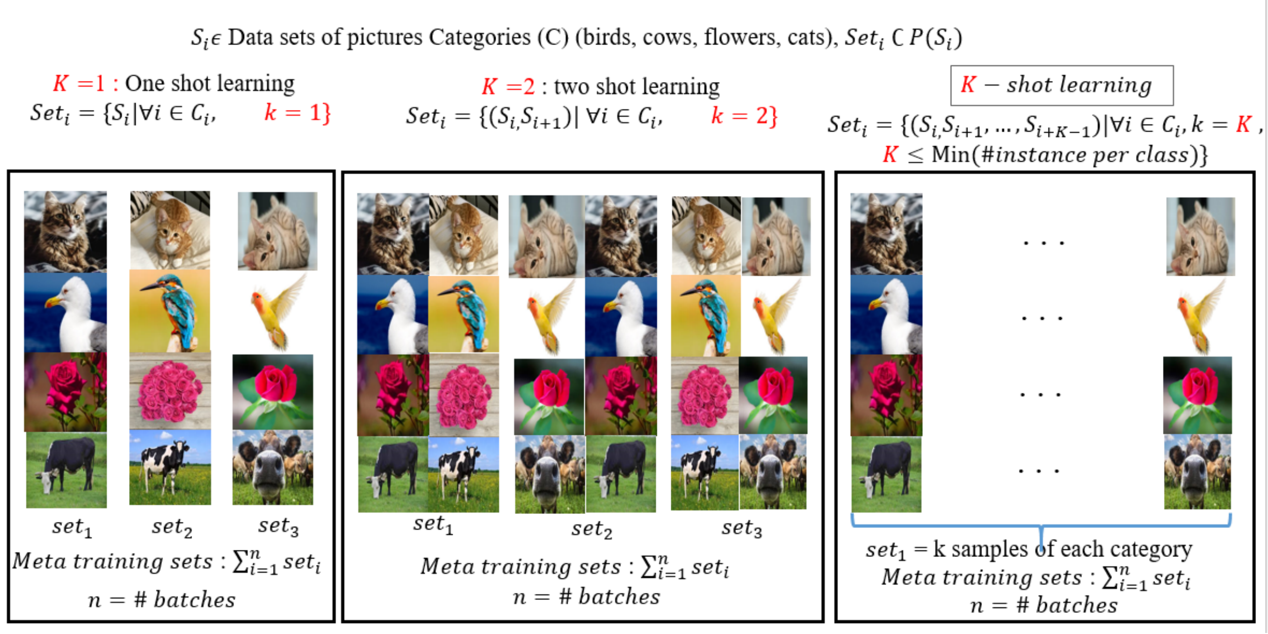


Figure 2.8: Few-shot learning structure

2.6.2 One shot learning

One-shot learning (OSL) is a critical challenge in deep neural network applications. OSL is a type of few-shot learning or k-shot learning in which k=1 shot is assigned to the meta-training section. In other words, when the algorithm begins training, it only uses one instance of each class at a time from different batches. Table 2.2 lists the research studies that have been conducted for one-shot learning such as matching networks [204] which is a metric-based MTL.

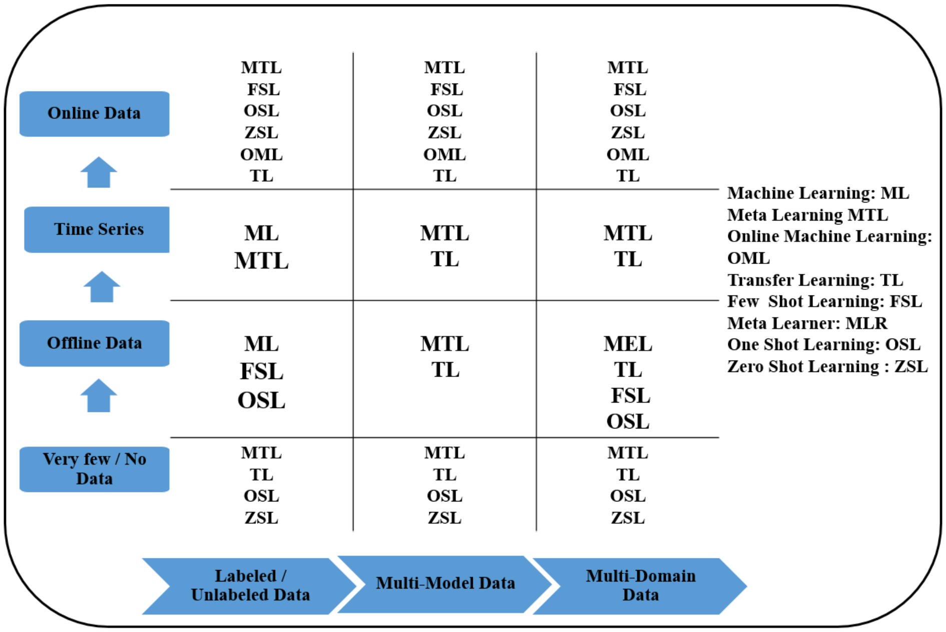


Figure 2.9: Machine Learning : ML, Meta-Learning: MTL, Online Machine Learning: OML, Transfer Learning: TL, Few Shot Learning: FSL, Meta Learner: MLR, One Shot Learning: OSL

2.6.3 Zero-shot learning

Zero-shot learning (ZSL) is a new machine learning paradigm that has recently been proposed [41], [169], [197] to outperform supervised learning algorithms by addressing their critical limitations, which work only with a fixed number of classes for each training and test dataset. Zero-shot learning is a technique for embedding domain-specific and semantic space that requires knowledge extracted from ontologies, wikis, dictionaries, and blogs. ZSL transcends the limitations of few-shot and one-shot learning and promises to outperform FSL and OSL for unknown class labels. The objective is to recognize and classify unseen samples of various classes in the absence of training data samples for those classes. ZSL makes this possible when you have accurate information about the problem's domain, classes, properties, and, most importantly, its functionality. Utilizing feature extraction, word2vec algorithms, and dimension reduction algorithms, the ZSL process is a technical journey from vector space to semantic space. The feature vector lists the things that each class that represents the input object has in common. Reed *et al* [169] applied neural language model to overcome supervised learning limitation. ZSL has been accomplished for visual recognition [169] , music classification [41], image classification [197], device-free human sensing [136], human motion recognition [149].

One of the recent efficient methods, semantic auto-encoder (SAE) that has been proposed by [99] using auto-encoders for ZSL where the goal is to regularize the learned model by mapping the image feature vectors to semantic space.

2.7 Discussion

Choosing the right type of data for machine learning algorithms is a critical but difficult task. According to [207], to ensure optimal decision-making, it is crucial to select the optimal algorithm for each specific problem. Researchers compared the results of interviews with domain experts to experimental results. It is essential to comprehend our current situation, the obstacles we face, and the data we currently possess. What is the relationship between emerging data and traditional and modern machine learning algorithms? Figure 2.9 shows the information that is needed to figure out which algorithms can be used with a certain type of data.

In figure 2.10, we have compiled a list of the best publications. According to our findings, few-shot learning is one of the most promising areas of supervised learning, whereas zero-shot learning excels at recognizing and classifying unseen class labels.

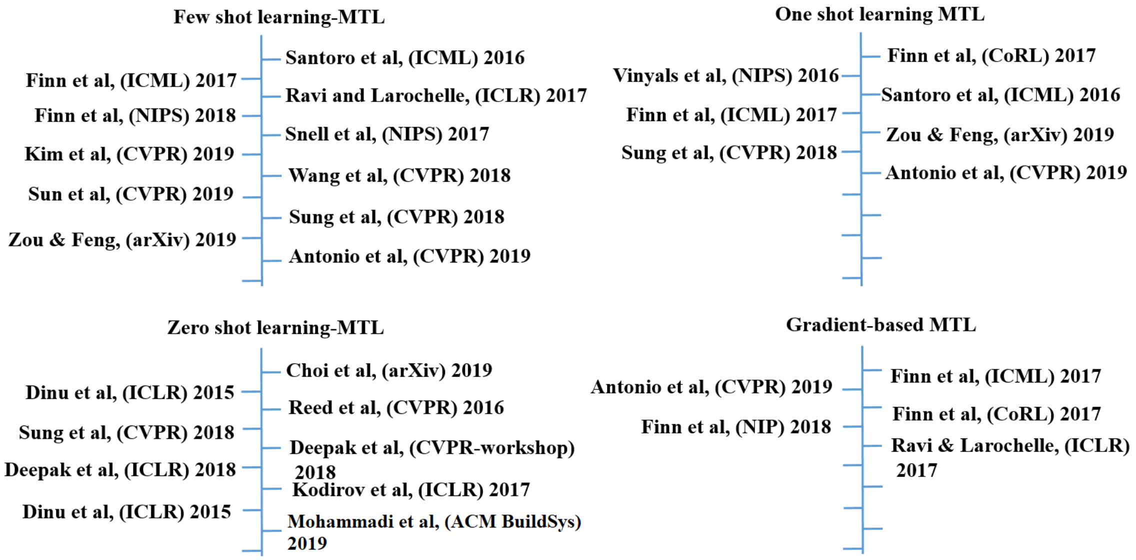


Figure 2.10: A brief studies over promises of Meta-learning

2.8 Conclusion

Optimization algorithms for offline data are nearly ubiquitous across all fields. The majority of research has identified the optimal solution to large-scale problems. Recent advances in AI technologies have made it possible for people to collect and access data from any internet-connected location. Therefore, it is essential to process continuous online data and implement a sophisticated learning algorithm to assist scientists in accurately predicting future data. In this chapter, this problem was looked at, and an advanced MTL-based machine learning algorithm was studied to find the best way to solve it.

Model-based, metric-based, and gradient-descent-based, also known as optimization methods, are three essential classifications that encompass MTL research studies. In addition, MTL has three essential extensions for emerging data and large-scale problems: few-shot learning, which is practically implemented on k-shots of training classes; and multi-shot learning, which is implemented on k-shots of training classes. The second extension is a form of few-shot learning in which each training class is allotted a single attempt. Last but certainly not least, is zero-shot learning. ZSL is a promising extension of meta-learning in which there are no examples of unseen classes in training sets, despite recent work using FSL and OSL. In the next chapter, we talk about the Optimized Semantic Auto-encoder (OSAE), which uses zero-shot learning to recognize unseen classes, such as animals.

CHAPTER 3

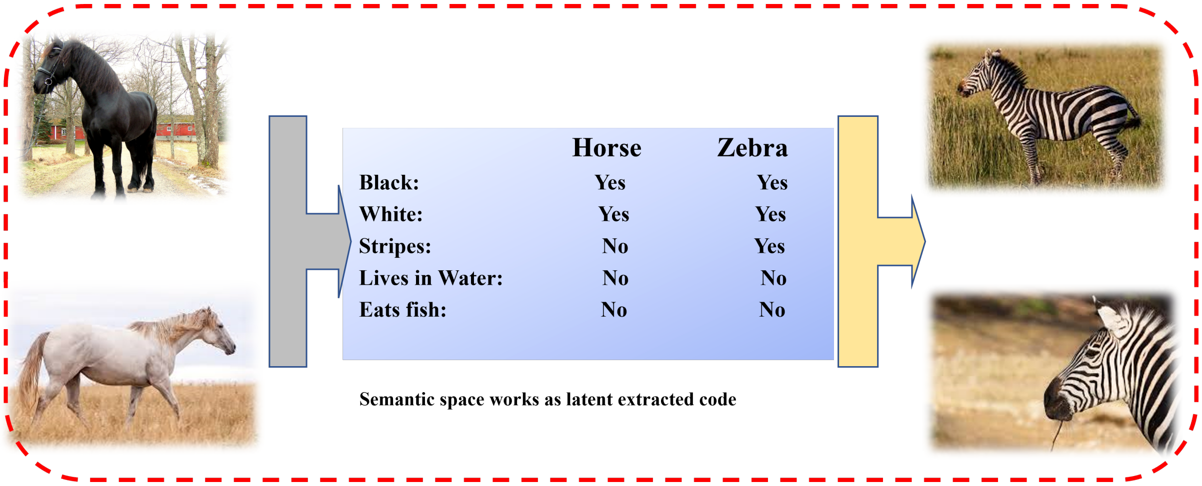
ON PARAMETER TUNING IN META-LEARNING FOR COMPUTER VISION[1]

---

[1] This chapter is a reprint and extended version of the following papers, with the permission of publishers:.
1) Farid Ghareh Mohammadi, M. Hadi Amini and Hamid R. Arabnia, On Parameter Tuning in Meta-learning for Computer Vision, IEEE Xplore 2019 International Conference on Computational Science and Computational Intelligence (CSCI), pp. 300-305, 2019.
2) Farid Ghareh Mohammadi, Farzan Shenavarmasouleh, Khaled Rasheed, Thiab Taha, M. Hadi Amini and Hamid R. Arabnia, The Application of Evolutionary and Nature-Inspired Algorithms in Data Science and Data Analytics, 2021, IEEE Xplore International Conference on Computational Science and Computational Intelligence (CSCI), pp. 3255-261, 2021

3.1 Abstract

In meta-learning (MTL), learning how to learn is crucial to achieving an optimal learning model. In this chapter, we investigate image recognition for unseen categories in a given dataset with limited training data. Using a zero-shot learning (ZSL) algorithm, we investigate the effect of hyper-parameter tuning on semantic auto-encoder performance to propose an optimized semantic auto-encoder (OSAE). By adjusting various embedded hyper-parameters, we improve OSAE's unseen class recognition accuracy. We also talk about the pros and cons of parameter tuning and how it can be used in image classification.



	Horse	Zebra
Black:	Yes	Yes
White:	Yes	Yes
Stripes:	No	Yes
Lives in Water:	No	No
Eats fish:	No	No

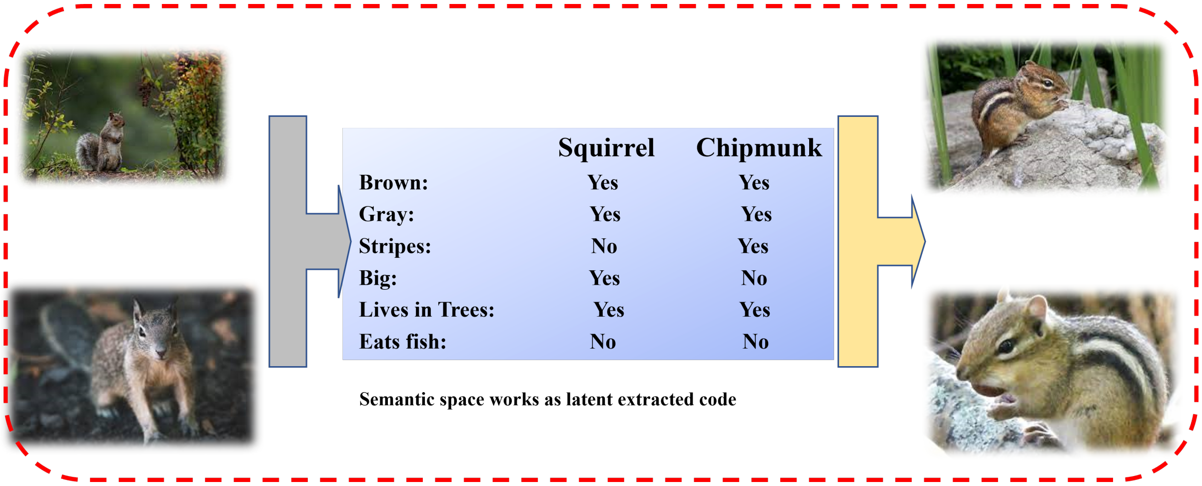
Figure 3.1: Seen and unseen classes examples during a training phase: Applying semantic space

3.2 Introduction

Motivation: Face recognition and automatic license plate recognition are two of the most important computer vision algorithms required to enable futuristic smart city features. Computer vision provides useful applications for the visually impaired, such as visual and infrared sensor-based obstacle detection. Deep learning is an advanced tool for learning in computer vision. It relies heavily on a training dataset to do well on a testing dataset [13]. Furthermore, auto-encoders have recently emerged as a difficult task in computer vision [211]. As a result, the focus of this chapter is on semantic auto-encoders. In the realm of artificial intelligence, the ability to efficiently learn new tasks by utilizing prior knowledge from related tasks is crucial. Meta-learning (MTL) is the most advanced machine learning algorithm because it learns from prior information as efficiently as possible. MTL first was presented as early as 1987 by Schmidhuber [182], and recent state-of-the-art studies [54] illustrate how MTL perfectly learns from a limited training dataset. In this chapter, we examine an image recognition task in which there were insufficient examples in the training dataset and we were required to classify images of previously unseen categories in the testing dataset. To get around this problem, we would choose zero-shot learning (ZSL) as the procedure for recognizing images that have never been seen. Figure 3.1 shows how an optimized semantic auto-encoder (OSAE) can tell that the zebra is a class that hasn't been seen by using the horse as a class that has been seen. In this example, we use four distinct semantic features: *black, white, stripes, lives in water*, and *eats fish*. Horses and zebras are both *black* and *white*, and they never share the *eat fish* or *live in water* features. The only feature that distinguishes these two is the *stripes* feature. This feature enables the learned projection function to comprehend animal characteristics, allowing it to map the visualized space of a horse to an appropriate semantic space and the semantic space back to a visualized space. In addition, this function can differentiate zebra as an unseen class from semantic space to visual space.

Furthermore, in figure 3.2 illustrates how we can generate a learned projection function for recognizing unseen chipmunks based on previously observed squirrel images. These two creatures share many similarities with horses and zebras. If we use the same semantic space for learning from horses and for recognizing zebras, then the model will almost certainly identify a chipmunk as a zebra and vice versa if we use the same semantic space for learning from horses and for recognizing zebras. To resolve this issue, we employ a number of semantic features. Instead of the *live in the water* feature, we use the *live in the trees* feature to distinguish these animals. Besides that, we use *gray* instead of white color, and Big feature is a new feature. So in general, we use five features as follows: *brown, gray, stripes, Big, lives in trees* and *eats fish*. It is clear that still *stripes* and *Big* features play the main features to distinguish these two animals.

Zero-shot learning, or attribute-based learning, is the process of learning to recognize unseen objects. ZSL concentrates on learning a new distribution of seen classes based on a meta description of the seen classes and seeks correlations with existing seen classes from a training dataset. This means that ZSL no longer needs to see examples of classes it hasn't seen before figuring out how well it can predict classes it hasn't seen. In recent years, ZSL [48] [230] [36] [54] [216] has been an active, challenging and hot research topic in advanced computer vision, machine learning, and medical data analysis [108]. Drug discovery [108], image classification [54] and meta-sense [146] are examples of such research studies. Furthermore, ZSL has penetrated other domains like human



	Squirrel	Chipmunk
Brown:	Yes	Yes
Gray:	Yes	Yes
Stripes:	No	Yes
Big:	Yes	No
Lives in Trees:	Yes	Yes
Eats fish:	No	No

Semantic space works as latent extracted code

Figure 3.2: Seen and unseen classes examples (Squirrel and Chipmunk) during a training phase: Applying semantic space

action recognition, networks [230], etc. In [230], researchers provided comprehensive information about ZSL applications in computer and mobile networks.

ZSL is an exciting MTL algorithm that simulates human cognitive activity. We discuss emerging ZSL issues here. First, semantic space representations, which are made up of attributes ('A') and word vectors ('W'), are important for understanding and learning from categories that have already been seen so that they can be applied to categories that have not yet been seen. However, they seem hard to use well [48]. To recognize new images, humans can only understand visual objects and attributes based on image explanations. However, not all the explanations produce distinguishable outcomes, and even traditional machine learning algorithms will not have promising performances. Second, ZSL provides a selection of mapping models for use with unseen classes. ZSL's ability to semantically represent objects and to learn a compatible visual-semantic or attribute-based function is its most important characteristic. When applied to unobserved classes, the risk of over-fitting increases proportionally to the complexity of the functions. A simple linear function, on the other hand, doesn't do a good job of classifying the classes that have already been seen and doesn't adapt well to the classes that haven't been seen yet.

Contribution: This chapter addresses the two emerging problems in ZSL and offers one promising solution. The primary contribution of this chapter is a linear optimal function with tuned parameters that achieves the most promising classification result that outperforms the current state of the art. We show the optimized semantic auto-encoder (OSAE), which was based on Kodirov's semantic auto-encoder (SAE)[99] presented in algorithm 1 and is a complete method for meta-mapping. OSAE aims to expand upon the work presented in algorithm 3 by enhancing the performance of unseen image detection via a hyper-parameter tuning mechanism.

Semantic Auto-Encoder (SAE) [99] is a sophisticated linear mapping auto-encoder that seeks the best mapping function ($mathscrW$) to recognize and classify unknown classes. Algorithm 1 depicts a thorough SAE procedure. In the first step of this chapter, we develop SAE correctly. In the second step, we optimize the algorithm by tuning a few embedded SAE parameters. It should be noted that, in addition to meta-learning for computer vision applications, parameter tuning is critical in ensuring the convergence of several algorithms.

Algorithm 1 Basic Semantic auto-encoder(SAE) by [99]

```
Require: A batch set of training input ($\mathscr{X}\mathscr{Y}$) and $training_{size}$
Ensure: The mapping matrix ($\mathscr{W}$) for zero-shot learning
  Begin Training
  for t=0 $\cdots training_{size}$ do
      Learn $\mathscr{W}$:$\mathscr{Y} \Leftarrow \mathscr{W}\ \mathscr{X}$
  end for
  End Training
  Begin Testing
  Compute acc. on Unseen Classes
  for t=0 $\cdots test_{size}$ do
      $\Delta = || Pred - Ground_T ||$
  end for
  End Testing
```

3.3 Related work

In this section, we examine related research on the application of zero-shot learning and present previous research on the evaluation of unseen classes. While the number of new zero-shot learning methods increases annually, not all methods are evaluated using the same criteria. This makes evaluating the methods particularly challenging.

We present meta-learning to overcome a limitation of traditional machine learning in which the learned model cannot predict testing data when the class has not yet been trained during the training phase. Meta-learning, specifically zero-shot learning, overcomes this limitation by identifying instances of previously untrained classes. This section discusses related ZSL research studies.

Table 3.1: Comparing related datasets (SS-D refers to dimension of semantic space dimension)

Datasets / **Features**	AwA	CUB	ImNet-2
Images	30,475	11,788	218,000
Total classes	50	200	1360
Seen classes	40	150	1000
Unseen classes	10	50	360
SS-D	85	312	1000

3.3.1 Generalized zero-Shot learning

Lampert *et al* [107] suggested ZSL as a method for attribute-based prediction in object recognition. They devised a classification technique known as direct attribute prediction (DAP), which learns from observed classes (y) and tests on unseen classes (y'). The authors say that using attributes makes it possible to transfer knowledge between classes seen in a training dataset and classes not seen in a testing dataset in a very accurate and cost-effective way. Romera and Torr *et al* [171] demonstrated a simple zero-shot learning (ESZSL) algorithm and compared it to DAP, a baseline algorithm. The compatibility of the learning algorithm was then enhanced by adding a regularization term.

Zhang and Saligrama [231] developed a brand-new ZSL based on semantic similarity embedding (SSE) and joint latent similarity embedding (JLSE). These researchers defined zero-shot recognition (ZSR) as a problem of binary classification. They looked at a framework that used dictionary learning to learn all of the model's parameters. This led to a class-free classifier.

Akata *et al* [6] demonstrated structured joint embedding, an image classification technique (SJE). Meanwhile, Changpinyo *et al* [31] developed a synthesized classifier (SYNC) method that uses combined semantic descriptions of $(A + W)$ to outperform DAP in image recognition.

From there, Bucher *et al* [28] came up with a method that took advantage of adding visual features to the attribute space. The method then learned a metric to reduce inconsistency by making the semantic embedding as flexible as possible. Shigeto *et al* [188] discovered that a least-squares regularized mapping function does not generate an optimal solution for the hubness problem. So, they came up with regression and CCA-based approaches to ZSL for calculating reverse regression, which involves putting class prototypes in the visual feature space. SAE [99] proposed by Kodirov *et al* a semantic auto-encoder to regularize the learned model by mapping the image feature to semantic space. Although Xina *et al* [217] proposed a feature generating framework for $any - shot$ learning called, f-VAEGAN-D2, there is a room for improvement. A lot of work have done for small datasets such as [99], however, just few methods are proposed for large datasets[61].

3.3.2 Hybrid embedding system

In generalized zero-shot learning (GZSL), the set of classes is divided into seen and unseen classes, with training relying on semantic properties of both seen and unseen classes and visual representations of only the seen classes, and testing relying on both seen and unseen visual representations. Currently, GZSL is addressed by learning a transformation from the visual to the semantic space, based on the assumption that class distributions in the visual and semantic spaces are similar. Such methods tend to convert unseen testing visual representations into semantic aspects of one of the seen classes rather than the semantic features of the correct unseen class, resulting in low GZSL classification accuracy.

Norouzi *et al* [156] proposed a hybrid image embedding system, referred to as convex combination of semantic embedding (ConSE), to deal with *n-way* image classification that lies in between independent classifier learning and compatibility learning frameworks. ConSE takes images and maps them into the semantic embedding space using a convex combination of the class label embedding vectors. Furthermore, Fu and Sigal [61] presented a learning algorithm called semi-supervised vocabulary-informed learning (SS-Voc).

Researchers [99], who provided new benchmarks using widely accepted evaluation methodologies on publicly available datasets, served as the beginning point for our literature review. These benchmarks make it possible to compare ZSL and GZSL approaches that have been proposed recently in a fair way. We use them to compare our results to those from the current state of the art in the field. We provide a high-level overview of the methodology presented in [215], and we invite readers to read that publication for more information on earlier research. The majority of ZSL and GZSL approaches compensate for the lack of visual representation of unseen classes by learning a visual-to-semantic mapping function [33].

The lack of visual training data for unseen classes biases the mapping between visual and semantic spaces toward the semantic properties of seen classes, especially for unseen test images, which is the fundamental drawback of the aforementioned approaches. This is a problem for GZSL because it reduces the classification accuracy of unseen classes. GAN models trained to create visual representations for the seen and unseen classes can be used to train a classifier for the seen and unseen classes, according to recent research [53].

On the other hand, unrestricted generation of synthetic visual representations for unseen classes permits the creation of synthetic samples that may deviate too far from the true distribution of visual representations, particularly for unseen classes. In the GAN literature, this is referred to as unpaired training because not all source samples (e.g., semantic features) have corresponding training target samples (e.g., visual features). This results in an optimization problem with few constraints. Researchers in [234] solved this problem by using a cycle consistency loss to push the representation from the target domain back to the source domain, therefore constricting the optimization problem.

3.3.3 Part-Based zero-shot learning

Part annotations were used in the early studies [89] to find discriminative part features for fine-grained ZSL. Part annotations, on the other hand, are expensive and labor-intensive. Attention techniques [222] have recently been applied to ZSL and GZSL [121] for capturing numerous semantic areas, which can enhance desirable information transfer, by pursuing automatic component discovery [219]. These approaches enhance ZSL performance significantly, but their gains on GZSL are insufficient, showing that they do not solve the domain bias problem.

3.3.4 Graph-bases zero-shot learning

Researchers use region-based relation reasoning in ZSL to model relationships between local picture regions. Researchers in [218] Researchers presented a method known as Region Graph Embedding Network (RGEN), which is trained using raw image data from the ground up. Constrained Part Attention (CPA) and Parts Relational Reasoning are the two branches of RGEN. The CPA branch, which is dependent on attention, generates the image regions. They modeled these regions as region graphs on which graph convolutions are used to perform part relation reasoning, resulting in the PRR branch. The model was trained with a transfer loss and a balancing loss. It compared similarities between classes and tried to get the most consistent responses for both seen and unseen responses.

3.4 The Sylvester equation

In 1884, J. Sylvester created one of the most well-known matrix equations, the Sylvester matrix equation. In addition, he was the first mathematician to demonstrate the term "Matrix". Matrix equations are used a lot in engineering and science, and they are a very important part of computational analysis [49]. We would like to expand on the definition of the Sylvester equation because it has become an integral part of computer vision solutions and signal processing and has been utilized in numerous engineering and scientific fields. This equation has been shown to work, and supervised and unsupervised optimization algorithms often use it to solve problems [44], [46]. Furthermore, the Sylvester equation plays an important role and serves as the dissertation's skeleton.

Following the use of the Sylvester equation to solve a distributed multi-agent network problem for convex optimization, Researchers [46] proposed a distributed continuous-time solution. They assumed that there was a unique solution to the equation and problem. So, scientists use the Sylvester equation to figure out the best way to solve the distributed computation problem.

The researchers [99] developed a semantic auto-encoder (SAE) that learned from the training image dataset (seen classes) and efficiently applied the Sylvester equation to estimate a projection function that could better adapt to the test image dataset (unseen classes). The equation was utilized in a solution for an auto-encoder in which both the encoder and decoder operate in linear time. Their symmetric approach yielded an effective solution to the problem of image recognition. The Sylvester equation works well and quickly for computer vision problems, which has been proven by more tests on many public data sets.

3.4.1 Sylvester input data for computer vision

We present a mathematical solution of Sylvester equation in Equation 3.1, where $A \in R^{n\times n}$, $B \in R^{k\times k}$, $C \in R^{n\times k}$, and our projection function or a learned model $\mathscr{W} \in R^{n\times k}$. The goal of This equation is to solve by estimating a projection function or learned model($\mathscr{W}$). The main input arguments: a matrix of feature vectors (X) and a matrix of semantic space(S) to estimate the optimal projection function ($\mathscr{W}$) where maps feature vector space to semantic space. The sylvester equation converts these two arguments into three variables of A, B and C as known matrices and generate one the unknown matrix ($\mathscr{W}$) denoted in equation 3.1.

$$A \times \mathscr{W} + \mathscr{W} \times B = C \tag{3.1}$$

Manipulating linear equations quickly and effectively is an important topic in research. There are two methods for solving linear algebraic systems: direct and iterative. Frequently, direct methods are employed to solve small linear equations. With direct methods, you can get an exact answer even if there are rounding errors [1], but if the condition number of the coefficient matrix is high, the accuracy of the answer is affected by rounding errors [58]. The condition number is the product of the norm and inverse norm of a matrix, and it is used to determine whether or not a matrix is ill-conditioned. The matrix is ill-conditioned as the condition number increases. Also, because of the way computers store information, direct methods often can't keep the sparsity of the coefficient matrix, need a lot of storage space and time to compute, and aren't very efficient [155].

When dimension m is very large, direct methods like the Gauss elimination method or the Cholesky decomposition method don't work. The use of a bandwidth minimization approach [63] to solve problems in the numerical calculation of large linear equations has been widely studied, with the major focus on how to compress storage [115], reduce computation time [86], improve computation stability, and so on. Therefore, it is crucial to develop an efficient model for large-scale problems. In this dissertation, we use a MATLAB function at link [2], and an efficient direct method like the Sylvester equation to solve these large-scale linear problems.

Sylvester variables definition

Let's assume we have input data of X and S, and then we compute three variables of A, B and C. When these three variables are initialized with the correct values, the Sylvester equation yields an optimal solution. For any given C, the Sylvester equation has only one solution if and only if the eigenvalues of matrices A and B are not the same [44].

◇ **Generating variable *A***

Let's have A deal with the semantic space. Thus, $A= SS^T$, where S^T is a transpose of S. If S is a square matrix, the resulting matrix would be symmetric. In real-data problems, however, semantic space dimension may prevent us from having a symmetric matrix. To make variable A appear symmetric, we calculate a multiplication of S and ST that produces a symmetric matrix.

◇ **Generating variable *B***

On the other hand, B is a matrix that deals with feature vectors(X). We need to make B as well as matrix$A$ to be a symmetric matrix. So, we compute $B= \lambda XX^T$. This time B takes one coefficient of λ.

---

[2] `https://www.mathworks.com/help/matlab/ref/sylvester.html#bt5pqdk-2`

⋄ **Generating variable C** We have generated matrices A and B, now we generate C with a condition that the dimensions of X and S being valid. Unlike A and B, C deals with both semantic space and feature vector space. Let's have C takes $(1 + \lambda) \times (S \times X^T)$, where S is the matrix of semantic space, X is the matrix of feature vectors and $(1 + \lambda)$ is the coefficient of C. When we generate these three variables, Sylvester equation is ready to compute the unknown variable $\mathscr{W}$.

Algorithm 2 Implementation of Meta-learning (MAML) [54]

```
Require: A batch set of input targets (I,Y) and $Batch_{size}$
Ensure: The best meta-learner $F(\theta')$ for few shot learning and an optimal mapping matrix ($\mathscr{W}$) to
    zero-shot learning
Begin
while work is not done do                     ⎫
    for t=0 $\cdots Batch_{size}$ do          ⎪  We train
        Learn from training batch set         ⎬  the learner
        Learn new ($\theta$)                  ⎪
        Update the ($F_\theta(X)$)            ⎭
    end for                                   ⎫
    $\theta'$=$\theta'$+$\nabla\mathscr{L}\mathscr{F}\theta\mathscr{X}$   ⎬  We optimize the meta-learner
    Update $\mathscr{F}\theta \frown \theta'$                             ⎭  until $\nabla(F_\theta(X)) \rightsquigarrow 0$.
end while
End
```

3.5 Meta-learning for computer vision: preliminaries and algorithms

This chapter's objective is to present an optimized semantic auto-encoder that can learn from seen classes (such as horses) and recognize unseen classes (animals like zebras). Figure 3.3 depicts how the encoder generates the optimal projection function ($\mathscr{W}$) and generalizes the model to recognize previously unseen classes.

3.6 Pre-processing

To apply meta-learning to the input dataset, the input data (feature vectors) and semantic space must be converted. Meta-learning, and zero-shot learning in particular, operates on a standard dataset with predetermined rules.

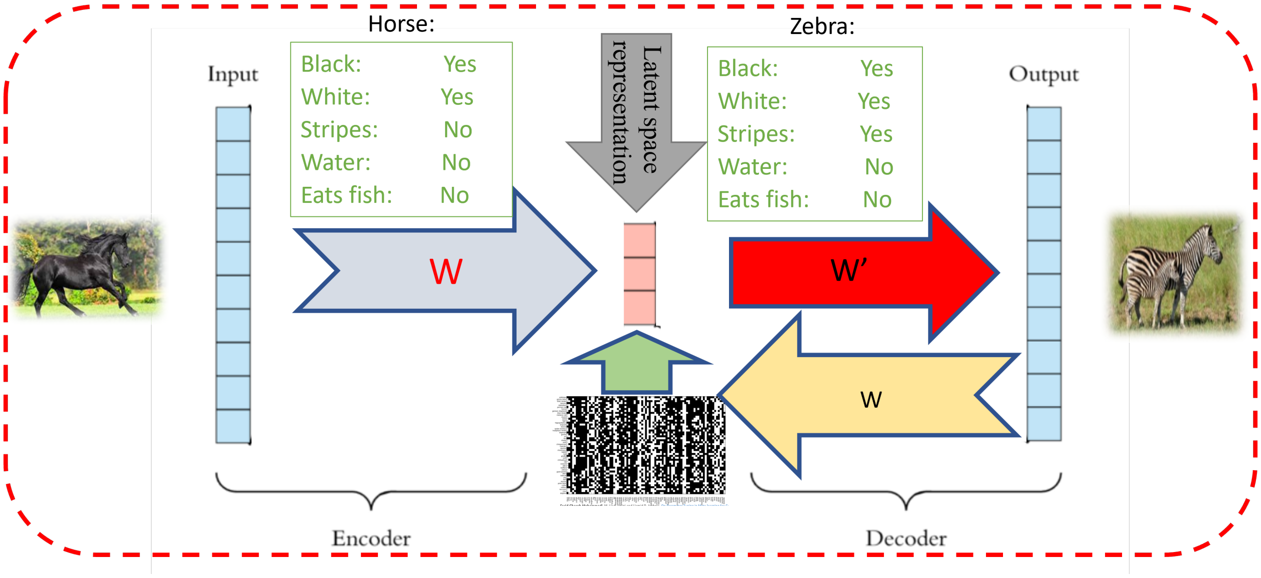


Figure 3.3: Optimized Semantic Auto-encoder (OSAE) framework

In order to accomplish this, we read images, flatten them, and store them in a matrix called X, which we refer to as feature vector space. In addition, we convert the semantic space to a new dimension, resulting in an output matrix that is sparse and filled with zeros. The values of each row show the label for a specific class that can be seen or not seen. Figure 3.4 depicts a semantic space for an animal with attributes. Each row clearly represents the animal's category, and the columns are the semantic features, with 0 indicating that an animal does not apply to a specific animal and 1 indicating that an animal does have this feature.

3.6.1 Preliminaries

We need machine learning and evolutionary algorithms to solve high dimensionality problems like the curse of dimensionality in order to generate a rich model from a large dimension dataset (CoD)[142] [134]. When there are few instances per class, however, machine learning cannot learn every sample. This issue is resolved by meta-learning, which facilitates advanced learning. Meta-learning offers three significant solutions for computer vision issues, specifically image classification and recognition: Few-shot learning (FSL), one-shot learning (OSL), and zero-shot learning (ZSL) are the three types of learning. During the training phase, FSL or *k*-shot learning collects

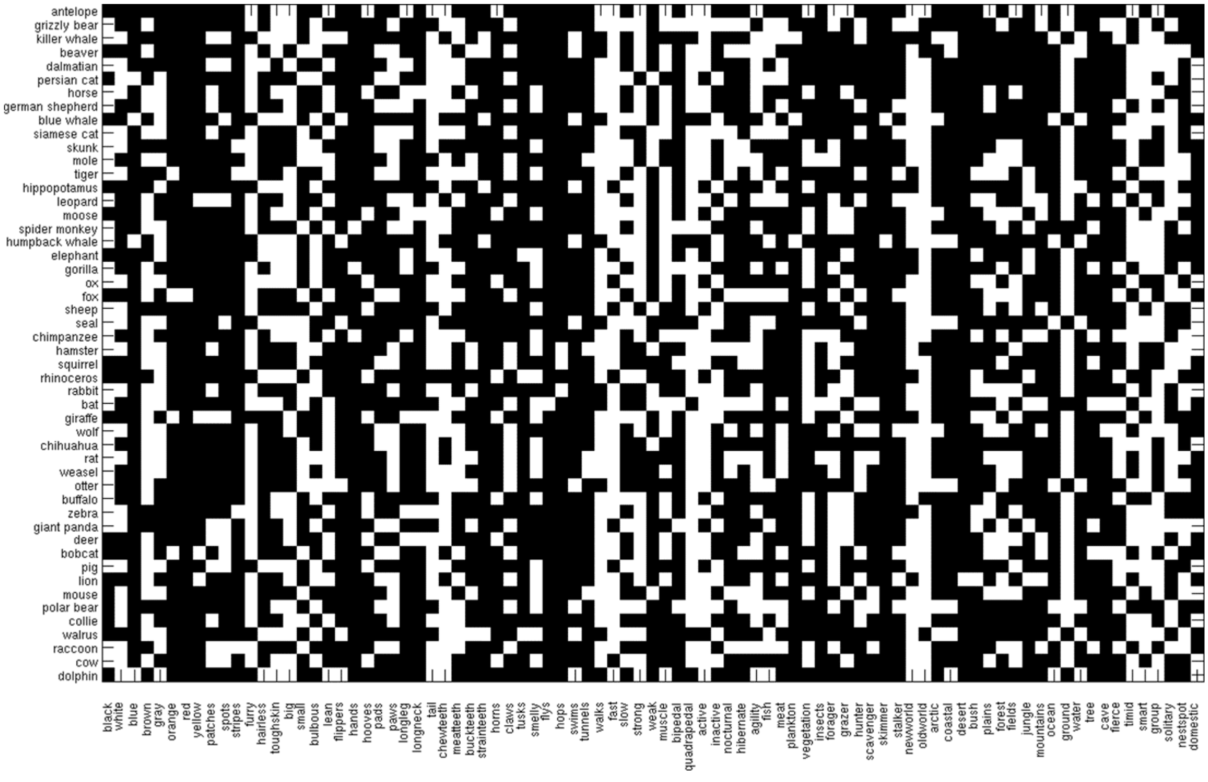


Figure 3.4: A sample semantic space for animal with associated attributes

k samples for each category by performing sequentially different tasks. Algorithm 2, a hybrid of FSL and OSL, presents semantic pseudocode of model agnostic meta-learning (MAML), as proposed by [54]. This algorithm added another step to show how MTL can get around the problems with traditional machine learning by using a meta-learner to use gradient descent optimization to improve the learner.

3.6.2 Preliminaries for zero-shot learning

Notations Let's suppose $(\mathscr{D})$ stands for a training dataset $(\mathscr{D})$=$(\mathscr{X}, \mathscr{Y})$ with the number of seen classes E=$\{1, 2, \cdots, n\}$. Consider $\mathscr{S}$ involves $\mathscr{d}$ dimensions of semantic space with required data, we map $\mathscr{X}$ into d-dimensional latent space with a mapping matrix $\mathscr{W}$. Then, we map this latent representation back to feature space $\hat{\mathscr{X}}$ using the transpose of $\mathscr{W}$, which is $\mathscr{W}^T$. In line six of algorithm 1, a well-known Sylvester equation is used to calculate an optimum mapping matrix using

$\mathscr{A}$, $\mathscr{B}$ and $\mathscr{C}$ which stands for SS^T, λXX^T and $(1+\lambda)SX^T$, respectively. We use this $\mathscr{W}$ to recognise unseen classes in a testing dataset, D^t=(X^t, Y^t), with unseen classes Z=$\{n+1, n+2, \cdots, m\}$.

3.7 Hyper-parameter optimization for optimized semantic auto-encoder

Machine Learning with Nature-Inspired Algorithms and Evolutionary Algorithms are two well-known and important areas of Artificial Intelligence (AI).

ML algorithms use data to improve algorithm performance automatically based on past experience with a variety of problems and to build a robust model. Nature-Inspired Optimization has recently gained popularity as a method for efficiently addressing such challenges in a timely manner. Examples of such algorithms are Genetic Algorithms (GAs), Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), and Artificial Bee Colony (ABC).

Due to the scale of the problem, hyper-parameter tuning in machine learning algorithms is an intensive computational process. Using computational intelligence algorithms to develop an effective strategy for hyper-parameter fine tuning is an intriguing technique. Artificial Bee Colony (ABC) optimization is a promising and effective technique for this purpose. But ABC may have a slower convergence rate or longer execution time in some cases because the first set of solutions is not good and the objective functions are expensive.

Due to its robust global search capabilities and small number of parameters, ABC has been utilized more frequently in the literature than other nature-inspired algorithms. Because of this, it has been used in a wide range of settings to solve difficult optimization problems.

3.7.1 Hyper-parameter optimization (HPO)

All machine learning algorithms utilize default hyper-parameter values. Hyper-parameters are parameters with constant values during algorithm execution. Nevertheless, there are some parameters that change as the algorithm executes. Consequently, it is essential to set the hyper-parameters at the beginning of each algorithm in order for it to converge more quickly and run optimally. There are some basic hyper-parameter tuning algorithms in machine learning, like grid search and random search, that need algorithms and the values of their hyper-parameters.

The goal of hyper-parameter optimization (HPO) is to build a machine learning (ML) model with the best set of hyper-parameters so that the objective function is optimized or accuracy is improved [153].

$$x = arg \min_{x \in S} g(x) \tag{3.2}$$

where $g(x)$ stands for the objective function that should be minimized, x is the optimal set of hyper-parameters, S is the search space, and $arg \min g(x)$ is the optimal set for which $g(x)$ reaches its optimum value.

Searching through a large number of hyper-parameter configurations, on the other hand, can be computationally expensive [153]. The majority of HPO problems are non-convex optimization problems with many local optimums. As a result, standard optimization procedures are rendered ineffective.

Automated ML is defined as tuning the hyper-parameters to achieve the highest accuracy performance from classifiers (AutoML). Grid search (GS) and random search (RS) are two common fundamental automated HPO techniques (RS). However, tuning the parameters of these two machine learning algorithms takes longer than anticipated. Some classifiers require approximately one year to be tuned. Thus, evolutionary/nature-inspired algorithms are required to optimize this procedure [138], [153].

To address this issue, more advanced algorithms, such as the Artificial Bee Colony, based on a combination of machine learning and evolutionary algorithms, have been proposed. ABC is optimal for HPO because it requires the setting and execution of few parameters. ABC uses three types of bees: worker bees, observer bees, and one scout bee [150].

According to the literature, classifiers such as SVM and MLP use evolutionary/nature-inspired algorithms such as the ABC algorithm to optimize their parameters [153], ABC has also been used to design and evolve hyper-parameters of convolutional neural networks (CNNs)[235]. Figure 3.6 depicts a detailed ABC process in which the algorithm selects a solution consisting of a set of hyper-parameter values, updates the solution using onlooker bees, and uses a scout bee to randomly select another solution that has not yet been discovered.

3.7.2 Optimized semantic auto-encoder (OSAE)

The majority of machine learning algorithms, including meta-learning, are constructed from a set of hyper-parameters whose values should be selected with care. This hyper-parameter configuration

Algorithm 3 Implementation of Optimized Semantic Auto-encoder (OSAE) using Hyper-parameter tuning

```
Require: A batch set of training input ($\mathscr{X}\mathscr{Y}$) and $training_{size}$
Ensure: The best mapping matrix ($\mathscr{W}$) for zero-shot learning

  Begin Training
  Tuning Hyper parameters
  for t=0 $\cdots$ $training_{size}$ do
      Learn $\mathscr{W}:\mathscr{Y} \Leftarrow \mathscr{W}\,\mathscr{X}$
      $Err_{dst.}=\|\,\mathscr{X} - \mathscr{W}\mathscr{W}'\,\mathscr{X}\,\|_F$
      Optimize ($\mathscr{W}$) :$\mathscr{W} = \frac{\mathscr{C}}{\mathscr{A}+\mathscr{B}}$
      Return $\mathscr{W}$ and $\mathscr{W}^T$
  end for
  End Training
  Begin Testing
  Compute acc. on Unseen Classes
  for t=0 $\cdots$ $test_{size}$ do
       $\Delta = \|\, Pred - Ground_T \,\|$

      Minimise $\Delta$
      if ($\Delta$ is minimum) then
          Performance+ = $\frac{1}{test_{size}}$
      end if
  end for
  End Testing
```

We tune Hyper parameters to generate optimal $\mathscr{W}$
where $Err_{Dst.} \rightsquigarrow 0$

We test
unseen classes
maximize the
performance of SAE

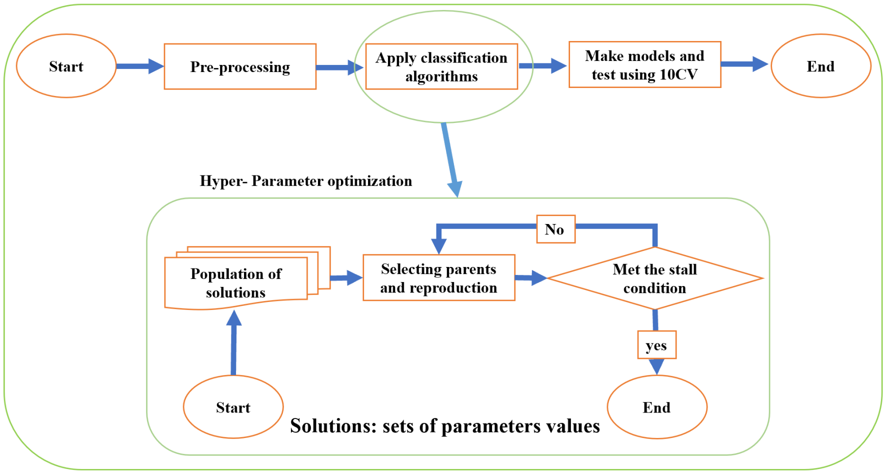


Figure 3.5: Hyper-paramter tuning using evolutionary algorithms

has a substantial effect on the performance of the algorithms. Several hyper-parameter optimization (HPO) methods are presented for supervised machine learning in order to determine the optimal hyper-parameter configurations. Due to the fact that our problem combines supervised and unsupervised algorithms, we must fine-tune the algorithms for optimal performance. Techniques for hyper-parameter tuning include grid or random search, combinations of evolutionary algorithms and machine learning, Bayesian optimization, Hyperband, and racing techniques. In this study, we use an artificial bee colony (ABC) to tune parameters. Figure 3.5 illustrates how hyper-parameter optimization occurs during the application of a supervised classification algorithm, in which an evolutionary algorithm selects technically appropriate sets of hyper-parameter values.

The basic semantic auto-encoder (SAE) presented in algorithm 1 has some hyper-parameters that are required to get tuned before running optimally. So, we presented an optimized solution in algorithm 3 in which we list the hyper-parameters and find the range of values that we set for each hyper-parameter. Table 3.2 displays an example value for each hyper-parameter. The tuning process is time-consuming. We reset all hyper-parameters and run the entire training phase to determine whether or not the performance meets the termination condition. We determine the highest possible precision for the termination condition. Figure 3.6 depicts an abstract method for adjusting the hyper-parameters of a classifier employing one of the evolutionary/nature-inspired algorithms. The algorithms are responsible for optimizing the hyper-parameters so that, when tested on a dataset, the classifier achieves the highest accuracy possible. Each solution in the population of food sources is a set of hyperparameter values for each classifier. At the end of the tuning process, the optimal values

Table 3.2: Hyper-parameters' values list

Parameter	Value
Lamda ($\mathscr{W}$)	[0, . . . ,10]
HITK	0, . . . ,10]
dist-Kernel	['minkowski', 'seuclidean', 'mahalanobis','euclidean', 'cityblock','cosine','jaccard','spearman','correlation','chebychev']
Sorting type	ascend, descend

for the hyper-parameters are written down, and the classifier makes changes to the configuration to get the best result.

Notably, we compute tuning parameters from scratch for each dataset, as the value of each hyper-parameter varies across datasets. Since we use three different datasets, we have to go through the process of tuning the hyper-parameters three times.

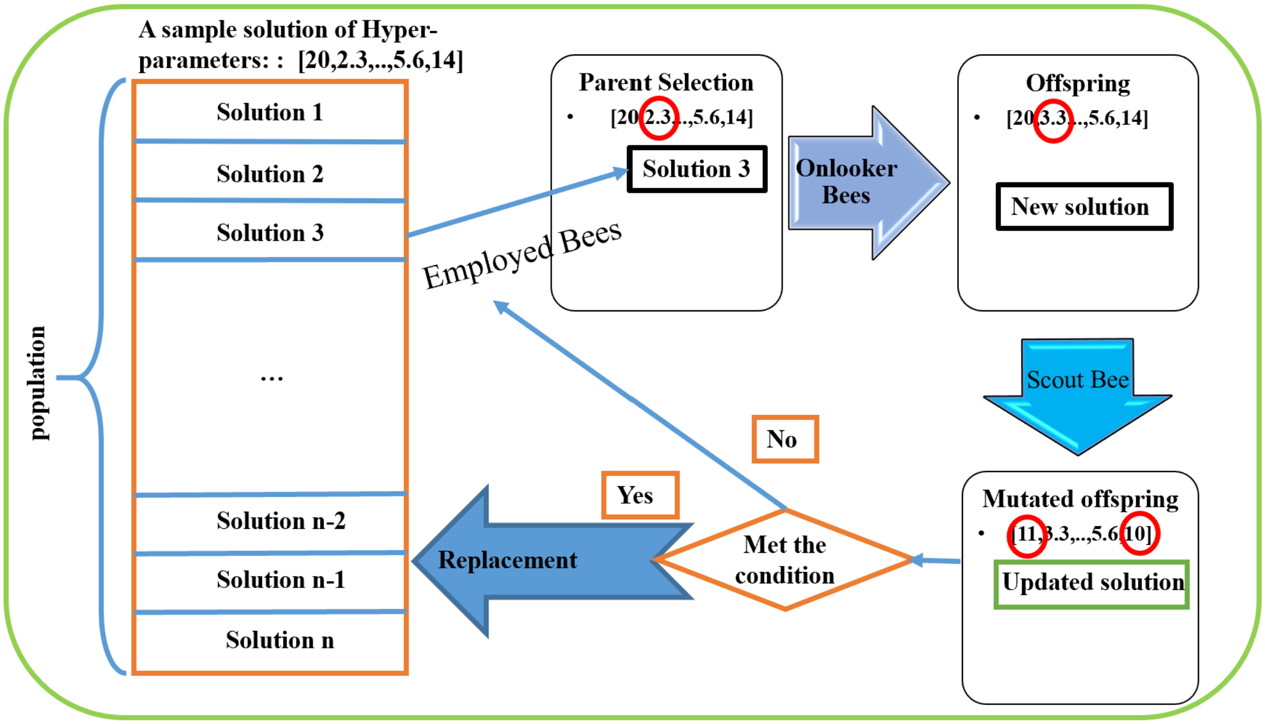


Figure 3.6: A general Schema of Hyper-Parameters tuning by ABC

Table 3.3: Comparing the related methods with our contribution for small datasets

Datasets methods	AwA	CUB
DAP	60.1	-
ESZSL	75.3	48.7
SSE	76.3	30.4
JLSE	80.5	41.8
SJE (A+W)	76.3	50.1
SynC	72.9	54.4
MLZSC	72.9	54.4
PRZSL	80.4	52.4
f-VAEGAN-D2 (IND)	71.1	61.0
f-VAEGAN-D2 (TRAN)	**89.8**	**71.1**
SS-Voc (A/W)	78.3/68.9	-
SAE (W)	84.7	61.4
$SAE(W^T)$	84.0	60.9
Our top three results of tuning		
$SAE(W)$−1st	74.6	**78.0**
$SAE(W^T)$−1st	**88.9**	**97.4**
$SAE(W)$−2nd	**91.0**	**89.9**
$SAE(W^T)$−2nd	**97.5**	**97.4**
$SAE(W)$−3rd	**99.7**	**94.1**
$SAE(W^T)$−3rd	**99.7**	**97.4**

3.8 Experimental result

We conduct extensive research to comprehend the value of a few SAE algorithm factors. SAE has only a few hyper-parameters $lambda$, HITK, dist-Kernel, and sorting type that are set differently for distinct datasets as displayed in table 3.2. We adjust the parameters using the following ranges and values: Dist includes a kernel algorithm to calculate the similarity between mapped unseen class instances and learned seen class instances, and the sorting mode occurs in either ascending or descending order.

Due to the high cost of examining all possible combinations, we present a set of results per HITK value for comparison. We find that one of the effective parameters, HITK, has a direct, positive, and sensible effect on the outcome of OSAE.

Table 3.4: Comparing the related methods with our contribution for a large dataset

Datasets **methods**	ImNet-2	ImNet-2
ConSE	15.5	15.5
SS-Voc	16.8	16.8
$SAE(W)$	26.3	26.3
$SAE(W^T)$	27.2	27.2
Our top solution :	λ= 5	λ= 6
$SAE(W)-$1st	**28.3**	**28.3**
$SAE(W^T)-$1st	**29.2**	**29.3**

3.8.1 Semantic Space

To accurately calculate the performance of OSAE, we describe the semantic space (SS) and its dimension (SS-D). We emphasize semantic space because it is the foundation of zero-shot learning. In the training phase of ZSL, specifically SAE, how inputs are described becomes crucial. In prior research endeavors, scientists utilized two distinct types of semantic space. The first property is (A), and the second is (W). We present all of our work in tables 3.3 and 3.4 mostly used attribute (A), except two research studies. SJE [6] worked with a combined semantic space (A+W) that yielded a result, 76.3 %, which was better than the basic attribute-based method (DAP). Moreover, SS-Voc[61] leveraged both (A/W) but not at the same time, and the performed results, 78.3/68.9%, illustrate that their approach had been well-computed. However, SAE [99] only used attribute-based semantic space to calculate the performance of recognizing unseen classes. It is noteworthy to say that, only word-vector('W') as SS has been used for large datasets in the compared work illustrated in table 3.4 [99].

3.8.2 Pre-defined Parameters

In [99], compared their proposed method, SAE, with more than 10 highly qualified methods using small datasets, which include AwA [107], CUB [205], $aP\&Y$ [52] and SUN [160]. The authors improved the accuracy of recognizing unseen images at least 6% in comparison with $SS-voc$ and at most 20% in comparison with basic attribute-based learning method, DAP. Further, the researchers used two large datasets: $ImNet-1$ [61] and $ImNet-2$ [61]. Their image recognition errors for large datasets are beyond 60%.

3.8.3 Effect of Parameter Tuning on Accuracy of Meta-learning for Computer Vision

To show how parameter tuning affects the accuracy of meta-learning for computer vision, we will first talk about the datasets, do an ablation study, define an evaluation metric, show state-of-the-art works, and compare them.

3.8.4 Dataset

We choose two small, but popular, and one large benchmark dataset for ZSL in this study: AwA (Animals with Attributes) [107] consists of more than 30,000 images with respect to 50 different classes of animals; CUB-200-2011 Birds (CUB) [205] consists of 218 instances, 1000 seen classes and 360 unseen classes; ImNet-2 [61] provides 1000 classes for seen classes and 360 classes for unseen classes, where seen and unseen classes are extracted from ILSVRC2012, ILSVRC2010, respectively. Table 3.1 illustrates details information of these datasets either training and testing.

3.8.5 Ablation Study

In this chapter, we collaborate with an optimized semantic auto-encoder (OSAE), a sophisticated supervised clustering algorithm designed for zero-shot learning. The primary strength of this chapter is tuning SAE, which is accomplished by analyzing output and modifying parameters. ZSL typically employs a complex projection function, whereas SAE uses a linear projection, according to algorithm 1.

3.8.6 Evaluation Metric

Given a similarity metric between objects, the evaluation metric is executed intrinsically, i.e., based on how close elements from one category or class are to each other, and how distant from elements in other categories. We compute the performance of the OSAE based on the loss function $|| Pred - Ground_T ||$, which is also presented in [109] for a metric learning function and supervised clustering.

3.8.7 Competitors

We compare our method, OSAE, with state-of-the-art method [217] and other work are compared in [99]. All compared research studies have used zero-shot learning (supervised learning) [232] [107] and semi-supervised learning[61].

3.8.8 Comparative Evaluation

We make the following observations according to the results in tables 3.3 and 3.4: (1) Our OSAE model obtained the highest results on both small and large datasets. (2) On the small datasets, the gap between $OSAE's$ results and the strongest competitors are varied due to different results of SAE. Note that our optimized model is a linear projection function, while most of the compared models use complex nonlinear projection functions, and some of them use more than one semantic space like SJE [6] and SS-voc [61]. (3) Although OSAE performed well on the large-scale dataset ($ImNet-2$), our model did not improve the performance by more than %7 due to limited number of semantic space dimension, but OSAE yields a promising result in comparison with other methods. (4) The performance of OSAE on the small datasets is far better than on the large dataset. (5) Last but not least, there is a high correlation between **HITK** value and OSAE's performance.

3.9 Discussion

All algorithms, and machine learning algorithms in particular, depend heavily on hyper-parameter tuning. It provides an all-encompassing environment for algorithms to learn from a training dataset, resulting in high performance on a testing dataset. However, machine learning algorithms may not perform well for unknown classes even with optimized parameters. In this chapter, we look at semantic auto-encoders (SAE) and make unsupervised semantic auto-encoders (OSAE) to improve unsupervised classification performance, which is one way to do zero-shot learning. The results in table 3.3 show that OSAE outperforms SAE and other related methods. To compare results in table 3.3 and table 3.4, we believe OSAE performs significantly better with small datasets than with large datasets. This chapter illustrates the significance of zero-shot learning tuning for image recognition. However, it does not perform well with large datasets. Table 3.4 shows that with different λ values we have different results for $SAE(W)$ and $SAE(W^T)$, such that $SAE(W^T)$ plays as a decoder and maps data from semantic space to feature space to compute performance of the work.

3.10 Conclusion

In the realm of machine learning, a superior learning model is essential. In contrast, recognizing unseen classes is a crucial issue in conventional machine learning. This chapter addresses this issue and investigates advanced learning processes that enable accurate prediction of unseen classes by learning from observed classes. We plan to focus on SAE as a semantic auto-encoder, which will allow us to discover an optimal mapping function between semantic space and feature space that also works for semantic space and classes that have not been observed. By adjusting the hyper-parameters of SAE, we propose an optimized semantic auto-encoder (OSAE) in this chapter. We investigated evolutionary algorithms for optimizing hyper-parameters. In the next chapter, we're going to combine zero-shot learning algorithms with signal processing to come up with Meta-sense.

CHAPTER 4

META-SENSE: PROMISES OF META-LEARNING FOR DEVICE-FREE HUMAN SENSING: LEARN TO SENSE [1]

---

[1] This chapter is a reprint and extended version of the following papers, with the permission of publishers:
1)Farid Ghareh Mohammadi, M. Hadi Amini, Promises of Meta-Learning for Device-Free Human Sensing: Learn to Sense, Proceedings of the 1st ACM International Workshop on Device-Free Human Sensing , pp. 44-47, 2019.
2) Farid Ghareh Mohammadi, Farzan Shenavarmasouleh, M. Hadi Amini, and Hamid R. Arabnia, Malware detection using artificial bee colony algorithm. In Adjunct Proceedings of the 2020 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2020 ACM International Symposium on Wearable Computers, pp.568-572 Mexico, Sep. 2020.
3) Farid Ghareh Mohammadi, Farzan Shenavarmasouleh, M. Hadi Amini, and Hamid R. Arabnia, Evolutionary algorithms and efficient data analytics for image processing IEEE The 2021 15th International Conference on Ubiquitous Information Management and Communication (IMCOM): Jan 3-5, pp. 1-8. IEEE, 2021

4.1 Abstract

This chapter explores the use of meta-learning algorithms to enable device-free sensing. This is referred to as "meta-sensing," which is learning to sense by discovering available information as opposed to deploying additional sensors. The application of zero-shot learning, a specific algorithm that does not require a priori information to learn, is of particular interest to us. This method seeks to learn how to learn, i.e., meta-sensing learns not only from available data but also how to learn over time without requiring additional sensing inputs. Meta-sensing learns and predicts from test data by transforming the data. It maps data and updates information, learns how to transform input data into a new feature vector space, and creates a new data cluster. In meta-sensing, the meta-learner, an agent that updates the learning model, is indispensable. When new data is received, the meta-learner updates its bias information using transformed data. The meta-learner learns by using ontology as a semantic space for all possible data as well as training dataset data.

4.2 Introduction

As technology advances, humans are able to collect and label more data for a variety of applications. Despite assisting researchers in categorizing spatiotemporal characteristics and activities, current methods may not be able to capture all the necessary details using direct data analytic methods due to incomplete or inaccurate data.

In addition, human sensing is ineffective at providing adequate data. Because of this, it is important to use human sensing without devices for data labeling and mining, learning from the little data that is available without adding more sensors, and learning to sense use over time.

In the majority of engineering optimization problems, proposed models attempt to analyze and classify input data so that humans can make the best decision. The use of machine learning (ML) to solve both supervised and unsupervised problems has gained popularity. Although machine learning outperforms previous algorithms, it has limitations when dealing with input data that varies over time. Data for ML techniques is either unavailable or limited to a few samples per category, which slows down the rate at which these algorithms learn. Figure 4.1 demonstrates that data-generating devices such as the Internet of Things (IoT) and sensors have proliferated, as has the need for a model capable of identifying unseen classes. In addition, a large number of features and data are generated, which prevents the learning model from effectively acquiring knowledge from the data. In addition, the curse of dimensionality (CoD) manifests, resulting in a low learning

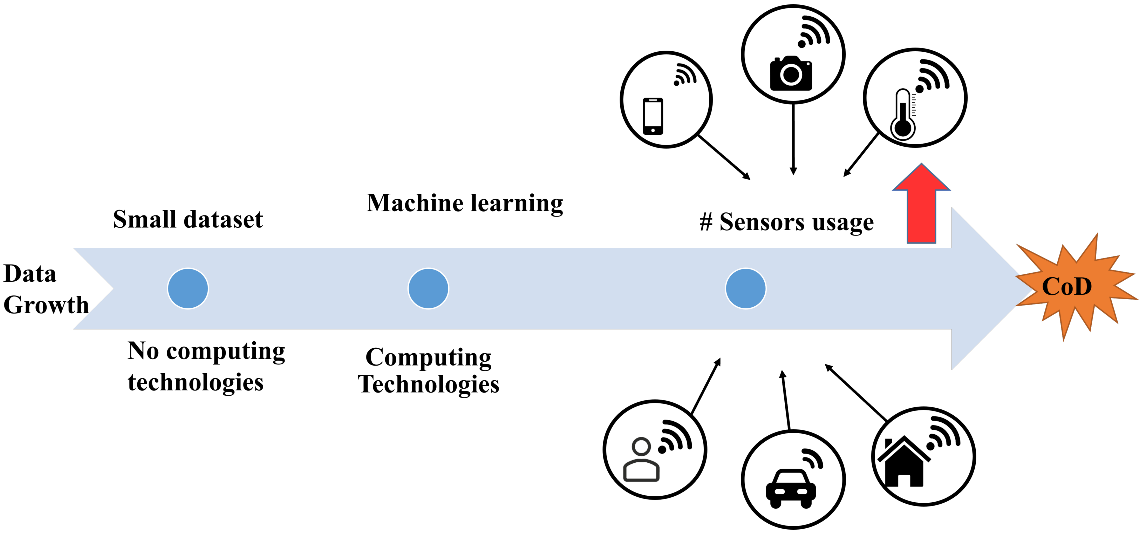


Figure 4.1: Data generation proliferation causes Curse of dimensionality (CoD)

rate. Feature selection optimization is one answer to this problem (FSO). Smartphones, for instance, are becoming more intelligent, while the high probability of exposure to malware and viruses has raised social community concerns. As a result, scientists collected a large number of features, which led to CoD problems with poor performance in predicting target classes. Using an Artificial Bee Colony, we proposed an optimized feature selection optimization solution (ABC). Algorithm 4 represents the complete pseudo-code for this process. Table 4.2 illustrates how ABC improves recall or sensitivity, a crucial metric for assessing the rate of malware detection. The slight decline in specificity has had no effect on smartphone security. Table 4.1 shows the formulas for calculating accuracy, specificity, and sensitivity.

Table 4.1: Evaluation criteria

Assessment Criteria	Formula
Accuracy (ACC)	$\frac{TP+TN}{TP+FP+TN+FN}$
Specificity	$\frac{TN}{TN+FP}$
Recall or sensitivity	$\frac{TP}{TP+FN}$

Furthermore, figure 4.2 shows some new problems with cooperative human-device sensing, such as poor sensing performance, no labels for online data, a slow learning rate, no performance guarantee for real-time data, overfitting, and incomplete learning.

Algorithm 4 Implementation of ABC algorithm for feature selection optimization inspired by [134]

```
Require: $S = \{x_0, x_1, x_2, ..., x_n\}$, $limit \geq 0$, $0 \leq$ lower Bound $\leq n/2$, lower Bound $\leq$
    upper Bound $\leq n$, $max_{iteration} \geq 0$, t=0 , $BestSolution = \emptyset$.
Ensure: Best Solution : $F = \{x_0, x_1, x_2, ..., x_m\}$ , m≤ n , $(\forall f_i \in F) \in S$ , $F_{length} \leq S_{length}$.

  Evaluate the whole food source (S) using SVM
                                              ▷ Step 1: Explore S using Employed bees
  for { dot=0 $\cdots$ $max_{iteration}$}
      Exploit the local foods to generate new food    ▷ Step 2: Exploiting by using Onlooker bees
      if limit is met then :
          {
          Explore new (unseen) food source to prevent from local optimum ▷ Step 3: Exploring by
  using Scout bee
        return {New Solution}
      end if
      Call fitness function to evaluate the Solution
      if any Solution obtained the best score then
            {Update the BestSolution}
      end if
  end for
```

Table 4.2: Comparing the related literature with our study

Criteria **Approach**	Recall (%)	Specificity(%)	Accuracy (%)
DroidFusion (J48) [226]	98.4	99.89	98.6
Proposed method (ABC+SVM)	**98.9**	**99.46**	**99.18**

Meta-learning appears promising as a solution to these problems, as it provides a consistent and evolving learning process. In particular, zero-shot learning (ZSL), an emerging extension of transfer learning, enables learning from available data without additional input data. In the majority of cases, machine learning algorithms are misled by insufficient training data that does not cover all categories. Consequently, there is no sample of a particular category within the training data. Notably, ZSL is domain-specific and learns new parameters for new domains. However, the zero-shot learning process is identical for other domains, but the input semantic space is entirely different. In this chapter, we talk about meta-sense, which is a new way to learn how to sense without a device using ZSL.

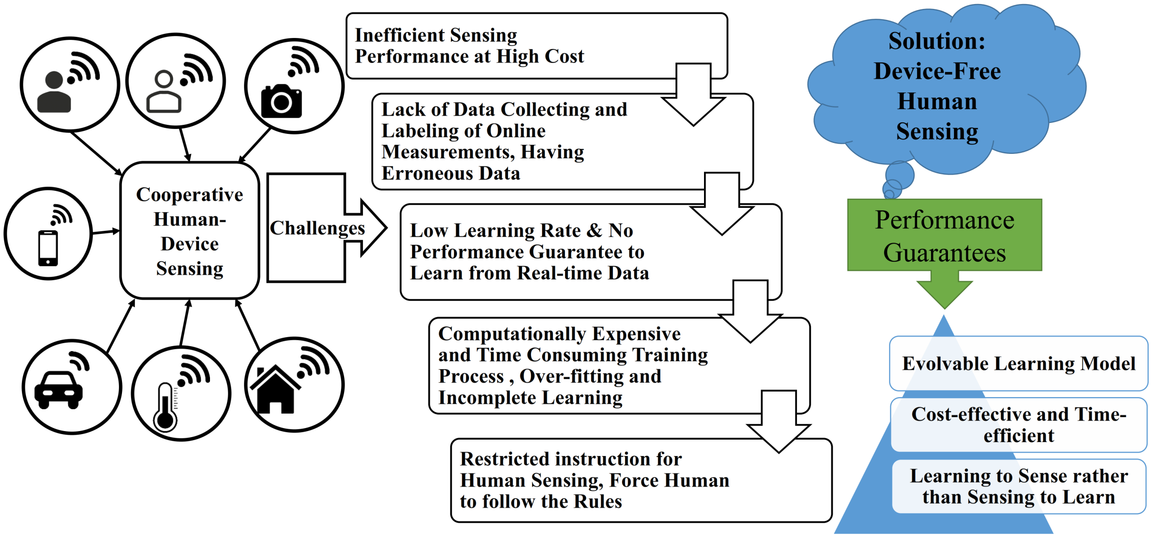


Figure 4.2: Challenges of human-device sensing

4.3 Solutions for solving curse of dimensionality

The curse of dimensionality (CoD) is regarded as an NP-hard problem [151]. To generate a machine learning-based model, it adds a lengthy overload to machine learning algorithms, rendering real-time processing in time-intensive fields such as visual computing impossible. In this section, we examine previously developed evolutionary algorithms for enhancing real-time data analysis through the use of image processing in certain applications, and we argue that they provide the most promising solutions to the CoD problem.

4.3.1 CoD in image classification: steganalysis

Focusing on a particular steganographic scheme, researchers are currently seeking a way to distinguish between cover images and stego images in a fraction of a second. However, when we do not know the scheme, steganalysis becomes challenging, especially in real-time processing. In order to address this issue, scientists employ a universal steganalysis that enables them to determine whether an image is a stego or a cover. This universal steganalysis must detect all steganographic schemes in order to be able to classify images by extracting key features. The greater the number of extracted features, the greater the likelihood of detecting steganographic schemes that allow us to accurately classify images. In contrast, the large number of features generates a high dimensional problem known as the curse of dimensionality (CoD). This is an important problem to solve because a detailed analysis of the features is time-consuming and difficult to perform in real time.To generate a machine learning-based model, it adds a lengthy overload to machine learning algorithms, rendering real-time processing in time-intensive fields such as visual computing impossible. In this section, we look at evolutionary algorithms that have already been made to improve real-time data analysis through the use of image processing in certain applications. We argue that these algorithms offer the best hope for solving the CoD problem.

This section aims to address this issue and present potential steganalysis solutions. Figure 4.3 depicts the problem's overall structure, the need for dimension reduction (feature selection), and promising solutions. Steganography is a sophisticated skill and method of communication that enables the transmission of secret messages concealed within a harmless multimedia file. In multimedia communications, images, audio, video, and even text files or Internet protocols are frequently used [141]. Stego multimedia refers to the cover multimedia that contains hidden messages. Steganography's goal is to make the stego multimedia and associated cover look as similar as possible. Otherwise, there is a high risk of detection.

Steganalysis is a difficult field to master because each steganography scheme must be identified using a unique set of algorithms. Researchers have developed steganalytic algorithms for specific steganography schemes. To achieve this, they first extract significant features using their preferred extraction techniques. These characteristics serve to differentiate between stego images and cover images.

In addition, the researchers propose a universal steganalysis to differentiate stego images from cover images regardless of the steganographic scheme used to embed the hidden messages. Numerous research studies have proposed various features, such as CC-C300 with 48600 features, to improve steganalysis performance. In some studies, the authors wanted to find out how useful it is to use high-dimensional features with ensemble classifiers by comparing them to some existing steganography algorithms. Another example is PHARM with 12600 features [78], where the authors made an important discovery that the pixels of a decompressed JPEG image are not shift-invariant and their statistical properties are heavily dependent on their position in the filtering (e.g. 8*8 grid). These are illustrations of the universal (blind) approach. However, in order to perform universal steganalysis such as those mentioned above, we must extract as many features as possible, which is also its main disadvantage. Because this raises the issue of the Curse of Dimensionality. CoD occurs when too much information is extracted or gathered.

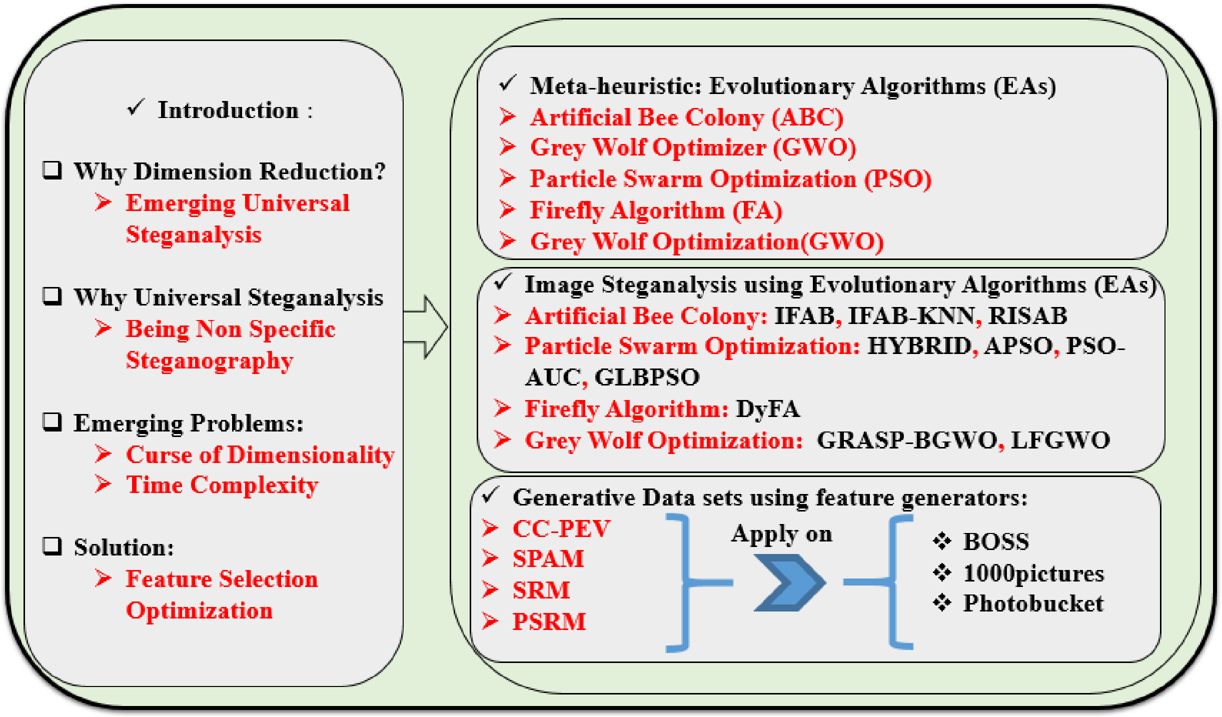


Figure 4.3: A general overview of this study

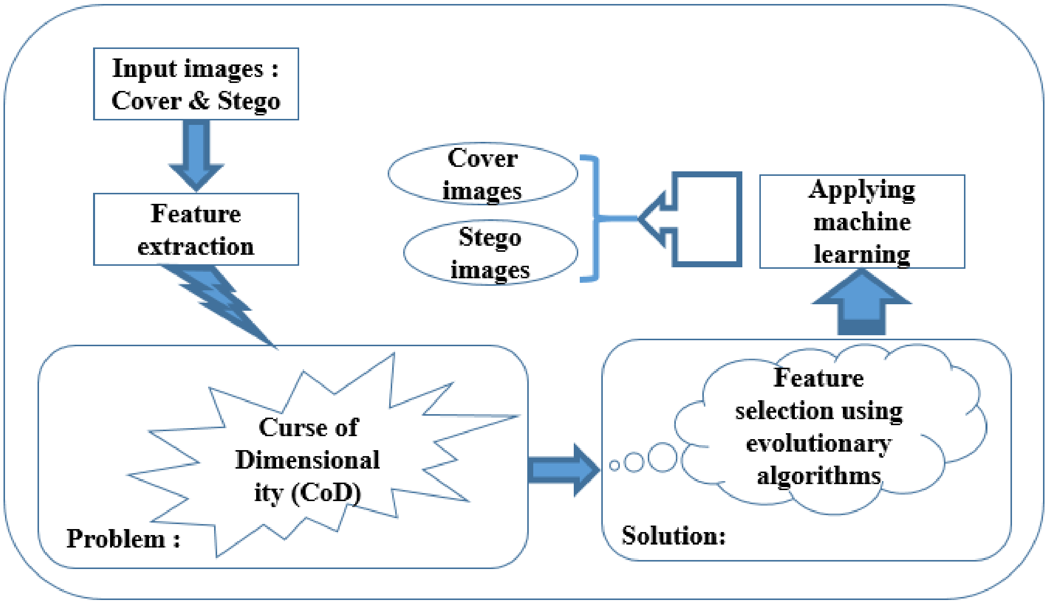


Figure 4.4: A general flowchart of image steganalysis using evolutionary algorithms [139] [138]

4.3.2 Image steganalysis using evolutionary algorithms

The process of steganography has been enhanced by researchers, particularly in the image domain. As a result of their research, they created a successful algorithm, which will be discussed in greater detail later. The combination of evolutionary algorithms and steganalysis is portrayed in figure 4.4 at a high level.

The objective of universal steganalysis is to find a model that performs optimally on both known and unknown steganography schemes, but doing so incurs the curse of dimensionality. Researchers attempted to solve the CoD issue by employing evolutionary algorithms. EAs aim to reduce the number of feature dimensions as much as possible while maintaining or enhancing performance. EAs always select the optimal feature dimension, resulting in performance that is significantly superior to that of conventional machine learning feature selection algorithms.

To determine the global maximum or minimum in a continuous or discrete space, scientists work on large-scale engineering and scientific projects. Due to the presence of multiple local maximums or minimums, traditional machine learning algorithms risk reaching local maximums or minimums instead of global maximums or minimums. CoD increases the probability of becoming entrapped in a local maximum or minimum. Therefore, researchers utilize EAs to seek global maximum or minimum solutions while ignoring local alternatives. Universal steganalysis requires evolutionary algorithms to resolve this problem in discrete space. EAs are used as a feature selection tool to reduce feature dimensions in discrete space in order to achieve the global maximum solution, i.e. the most relevant subsets of features that distinguish stego images from cover images with the highest accuracy.

Deep learning tries to enhance detection accuracy while considering high-dimensional images. For example, Boroumand *et al* [26] proposed SRNet, a novel architecture for convolutional neural networks (CNN) for steganalysis. Through deep learning, SRNet is unaffected by many newly introduced design elements and still achieves high performance. While deep learning improves steganalysis performance significantly, it still has time complexity issues. Evolutionary algorithms (EA) have also been proposed for this task and have been shown to improve steganalysis performance more than deep learning [138].

It is essential to develop efficient data analytics algorithms in order to address emerging engineering issues, such as those involving critical infrastructures. Evolutionary algorithms are potent computational algorithms that can employ a vast array of techniques to solve complex engineering and scientific problems. EA provides an environment in which steganalysis generates low time complexity. In steganalysis, evolutionary algorithms have been used for a variety of purposes, including feature selection, locating the most probable sub-images in the spatial domain, and more. We focus on algorithms that change over time, like the Artificial Bee Colony, which has been used to try to improve steganalysis. These studies [138] utilized feature selection techniques to solve the COD problem.

4.3.3 Image steganalysis using artificial bee colony

The concept of Karaboga's artificial bee colony (ABC) algorithm is more suitable for continuous and discrete optimization problems. The ABC algorithm is a potent optimization algorithm that simulates the behavior of honeybees searching for the best food source. ABC uses employed bees, onlooker bees, and one scout bee. ABC begins with employed bees calculating the goodness of each solution, followed by onlooker bees seeking the global maximum solution with respect to the best goodness via cross-over, which produces new offspring from selected high solutions in order to create a better solution. This procedure is repeated until no higher maximum goodness is discovered or the maximum number of crossover iterations is reached. The scout bee then leaves the vicinity of onlookers in search of global maximum goodness (solution). This procedure is repeated until the condition is satisfied or the maximum number of iterations is reached. Observer bees flee until the limit is reached. Afterward, one of the worker bees transforms into a scout. Each iteration of Scout involves mutation, the final stage of evolutionary algorithms.

To solve discrete engineering problems such as CoD, researchers have proposed a new customized version of ABC. There are plenty of customized ABC applications particularly for image processing, including but not limited to [135] [140]. Due to its simplified implementation, ABC is a widely utilized algorithm with adaptability, robustness, and the ability to exploit local solutions. Additionally, it can search the entire sample space for global solutions. ABC's primary benefits are its ability to solve any problem and its simplicity, resulting from the small number of tuning parameters. ABC appears to be significantly superior to other evolutionary algorithms such as the Genetic Algorithm (GA). ABC, on the other hand, has a low convergence rate during sequential processing and a slow execution rate for computing precise solutions. These are the reasons why ABC might not achieve the optimal solution.

4.3.4 Image steganalysis using Feature Selection based on ABC

The concept of dimension reduction utilizing evolutionary algorithms such as ABC was first presented in [133], and then a new steganalysis method dubbed IFAB was proposed in [135], which improves the performance of steganalysis in distinguishing stego images from cover images. The authors employed a wrapper-based feature selection technique by modifying the original ABC algorithm and incorporating it into the discrete ABC algorithm. ABC selects the subsets of features, and a classifier evaluates each subset of features generated by the algorithm. The feature selection procedure solves the CoD problem by using employed bees, onlooker bees, and a scout bee. This is how the IFAB feature selection algorithm works.

According to the table 4.3, these studies [133], [135] would be considered the first two work to apply the EAs on steganalysis. IFAB significantly minimizes the feature dimension and selects 80 features out of 686 SPAM (Subtractive Pixel Adjacency Matrix) attributes. IFAB also shrinks the CC-PEV (Cartesian Calibrated features by PEVny) feature dimension well by selecting 250 features out of 548. The Final accuracy of IFAB is 60.98% and 68.22% for SPAM and CC-PEV datasets, respectively.

4.3.5 Comparative studies

Mohammadi and Saniee Abadeh in [143] presented IFAB-KNN, an improved evolutionary algorithm approach for image steganalysis to improve IFAB. A wrapper-based feature selection

Table 4.3: Research study Comparison

Paper	Evol. alg.	Proposed method	FS type	Year
[135]	ABC	IFAB	Wrapper	2014
[143]	ABC	IFAB-KNN	Wrapper	2014
[140]	ABC	RISAB	Wrapper	2017
[39]	PSO	HYBRID	Wrapper &Filter	2016
[2]	PSO	APSO	Filter	2018
[174]	PSO	PSO-AUC	Filter	2016
[105]	PSO	GLBPSO	Wrapper	2017
[92]	PSO	MI-APSO	Wrapper*	2018
[40]	FA	DyFA	Wrapper	2018
[202]	GWO	GRASP-BGWO	Wrapper	2019
[159]	GWO	LFGWO	Wrapper	2019
[69]	PGO	GLCM-PGO	Wrapper*	2019

* goes to special kind of wrapper that select features by eliminating the non-informative features. The rest of research studies select subsets of features regularly

algorithm was also provided by IFAB-KNN. The authors incorporated K-Nearest Neighbor (KNN) into ABC to help it evaluate each subset of features more thoroughly. Specifically, KNN is an important fitness function for evaluating subset features in ABC. With the updated tuning parameters and the same number of chosen features, IFAB-KNN does better than IFAB.

Mohammadi and Sajedi in [140] proposed a new hybrid steganalysis approach called region-based Image Steganalysis Using Artificial Bee Colony (RISAB). RISAB enables ABC to search image space, specifically the spatial domain, for the sub-image carrying the hidden messages with the highest probability. Indeed, the likelihood of embedding messages in a sub-image is high if the sub-image has a greater amount of intensity and energy than the rest of the image. RISAB is the combination of IFAB and ABC applied to the entire image. The first big step and goal of their research is to look at the whole image to find a high-energy sub-image. They will do this by using a customized ABC algorithm to move through the image's spatial domain.

The features are extracted twice thereafter. They first extract features from the whole picture. Second, the same features are extracted from the sub-image of the previous phase. They utilize the same features as IFAB, which are the optimal subset of features for feature extractors, SPAM, and CC-PEV. After extracting the features from the entire image and the sub-image, researchers separately combine the datasets for both feature extractors. The final datasets for SPAM and CC-PEV contain 160 and 500 features, respectively. In either dataset, instances are represented by features that allow the classifier to train and find a high-performing model. Because the authors took into account both the whole image and its parts, their method is much better than IFAB.

Researchers used ABC on two different types of data: numerical data and spatial domain. Two distinct methods, IFAB and RISAB, were proposed. As a result of their dimension reduction properties, both significantly enhanced the performance of steganalysis while solving the CoD problem. It is also important to note that ABC is used to detect malware in cutting-edge research[150].

4.3.6 Image steganalysis using particle swarm optimization

Particle Swarm Optimization (PSO) [93] is an optimization method to solve non-linear optimization problems, presented by Kennedy and Eberhart. PSO is based on the behavior of flocks of birds or schools of fish. Several researchers proposed extending PSO to solve optimization problems, specifically steganalysis problems such as [39] [2]

Chhikara *et. al* [39] proposed HYBRID, a new hybrid strategy for steganography based on PSO. To address the computational complexity of image steganalysis, a hybrid filter and wrapper-based feature selection strategy was proposed. The researchers evaluated their strategy against a variety of steganography algorithms, such as NFS, PQ, Outguess, and Steghide. The customized PSO made it easier to tell the difference between stego images and cover images while also making the process faster. While testing the HYBRID algorithm against various steganography algorithms, the authors improved classification accuracy for SPAM and CC-PEV by up to 10

Using Area Under Curve (AUC) as a new measure for fitness function to evaluate the selected features, a few researchers proposed a novel approach to the problem of discovering the message embedded in covert multimedia. They considered another filter-based algorithm for feature selection in steganalysis. Authors in [2] He presented APSO, an adaptive inertia weight-based PSO (APSO). APSO is utilized for steganography with two primary phases. First, a feature selection process is used to reduce the dimensionality of features; then, a final model is developed to distinguish stego images from cover images using the subsets of features that were selected. APSO utilized a novel fitness function that computes Area Under Curve to evaluate the chosen feature subset (AUC). The authors used several classifiers, including SVM, DT, NB, and KNN, to evaluate the given method. When the hyper-plane encountered the greatest distance between support vectors of a given stego and cover classes, the SVM performed better than the competition. APSO-AUC reduced the number of features and chose the top 140 features from a pool of 686 SPAM attributes and the top 363 features from a pool of 548 features. The final accuracy of APSO using SVM is 82.62% for SPAM and 87.72% for CC-PEV datasets, respectively. Because the time complexity on large data sets has been reduced, these methods produce better results when compared to IFAB and RISAB.

Another version of PSO for steganalysis was proposed [105]. Global Local PSO is a new wrapper-based feature selection introduced by the authors (GLBPSO). They used neural networks to evaluate the GLBPSO-selected feature subsets. The GLBPSO algorithm enhanced the standard PSO by combining the best global and local PSOs. In [105], researchers take Chen's approach [32], and decrease its dimension by selecting the best feature subset. The prediction performance of GLBPSO provides no more than 7 percent improvement in comparison with the basic results, where performance is calculated based on all features. GLBPSO reduced features down to 282 features out of 486 features. Furthermore, Kaur and Singh [92] proposed a new feature selection leveraging mutual information and adaptive PSO (MI-APSO) using area under curve for image steganalysis. MI-APSO was also inspired by IFAB as a feature selection technique and further improved the performance of image steganalysis.

4.3.7 Image steganalysis using firefly algorithm

Yang introduced the Firefly algorithm (FA), which is inspired by the flashing behavior of the fireflies [225].FA uses its light to attract other objects, particularly mates. This algorithm was presented by Yang under the assumption that all fireflies are of the same gender. It implies that the probability of teammates being attacked is equivalent. In addition, the rate of attraction is proportional to the amount of light; the greater the amount of light, the greater the rate of attraction. Moreover, if there are no bright fireflies, mates will move towards any of the other fireflies. Researchers designed and developed FA to be compatible with steganalysis. The Firefly algorithm can assist in resolving ABC's convergence rate issue. This section examines the application of FA in steganalysis.

Chikara *et al*[40] proposed Dynamic firefly (DyFA), a modified implementation of the firefly algorithm for universal steganalysis. The primary function of DyFA, which reduces the computational complexity of universal steganalysis, is feature selection (FS). FA provides two crucial parameters, alpha and gamma, which enable FA to converge more quickly with each iteration. Tuning these parameters is essential for FA to accelerate the feature selection process. The researchers also employ a hybrid FA known as DyFA, which combines the filter method (t-test and regression) and the wrapper method for FS. The results reveal that DyFA reduces features by about $77 - 93$ percent of the original feature dimension, which improves the accuracy of distinguishing stego images from cover images by about $2 - 10$ percent. Like other discussed studies, the scientists use CC-PEV and SPAM features to DyFA to decrease their feature dimensions. Accuracy for SPAM has been enhanced by $9 - 15$ percent, and CCPEV shows an improvement of $10 - 13$ percent. The results reveal that DyFA outperforms IFAB. However, DyFA has still not been tuned to decrease time complexity.

4.3.8 Image steganalysis using grey wolf optimizer

Mirjalili *et al* [129] proposed the Grey Wolf Optimizer, a new algorithm based on the concept of grey wolf packs (GWO). GWO outperforms other evolutionary algorithms in the search for the solution of multidimensional nonlinear functions. The GWO algorithm imitates the grey wolf's leadership structure and hunting mechanism. Based on the behavior of different types of wolves, such as the alpha, beta, delta, and omega, the leadership hierarchy is simulated. The GWO, which was inspired by the grey wolf hunting procedure, consists of three primary phases. Exploration is the first step, followed by prey encirclement, and finally prey attack, which is exploitation. This process makes sure that scientists find the best solution for the whole world so they can solve optimization problems.

The GWO is a population-based algorithm that seeks the optimal solution through the collective behavior of wolves. GWO begins with the exploration of the search space and gradually exploits it to determine the local optimum. GWO offers the most crucial parameters for adjusting the step size with the "A" parameter and controlling convergence with the exploration and exploitation parameters. It is well-known that the GWO has a low computational cost. However, it still has limitations, such as a slow convergence rate and the occasional inability to escape local optima. Controlling the trade-off between exploration and exploitation, as "A" says, is the most important part of GWO.

Pathak *et al* [159] proposed levy flight-based grey wolf optimization as a new GWO for solving steganalysis using feature selection (LFGWO). LFGWO aims to find the most prominent features in feature space, overcoming the limitations of GWO. Random forest is one of the decision tree classifiers included in the fitness function. The main advantage of LFGWO over other evolutionary algorithms in the literature is its superior convergence precision. In addition, the performance of image steganalysis over selected features was evaluated using five distinct classifiers: SVM, LDA, RF, KNN, and ZeroR. It is important to note that the LDA is used here as a classifier. LDA is also employed as an algorithm for feature reduction [139]. The LFGWO extracted 84 out of 686 and 89 out of 1000 features from SPAM and AlexNet extracted features, respectively. Although LFGWO obtained better results than IFAB[135] and advanced IFAB, called IFAB-KNN [143], IFAB and

IFAB-KNN extracted fewer features, i.e. 80 out of 686 for SPAM. The results reveal that the 84 selected features were more relevant than those 80 features, particularly for SPAM.

Veena *et al* [202] devised an improved method for attacking the well-known steganography algorithm, least significant bit (LSB). The authors sought optimal characteristics using the hybrid technique Greedy Randomized Adaptive Search – Binary Grey Wolf Optimization (GRASP-BGWO). They were able to make the ensemble logistic regression classifier better at classifying while at the same time reducing the number of features.

Veena *et al* [202] presented five distinct spatial LSB algorithms, including LSB Replacement (LSBR), LSB Matching (LSBM), LSBM Revisited (LSBMR), Two-bit LSBR (LSBR2), and Modulo 5 LSBR (LSBRmod5). Using GRASP-BGWO, the authors discovered that the detection process is highly dependent on three crucial properties: training algorithms, payloads, and features. According to their research, GRASP-BGWO outperformed SRM (Spatial Rich Mode), PSRM (Projected SRM), and SPAM, even with low volume payload per pixel, according to their research. GRASP-BGWO enhanced performance by up to 13%. In addition to the given GRASP-BGWO and it's performance on LSBR, Shojae Chae-ikar and Ashmadi [30] introduced an original ensemble similarity weight (SW)-based image steganalysis. The ensemble SW steganalysis involves three fundamental steps. First, a SW analysis is performed, then an SVM classifier is implemented, and finally, a decision between stego and cover is made. The researchers compute the pixel and channel similarity weights of the specified object and produce PSW and CSW datasets. Then, the datasets are compared to their respective reference profiles. In the last step, the datasets made in the second step are used to figure out if the image is stego or cover.

4.3.9 Image steganalysis using Pine Growth Optimization

Pine Growth Optimizer (PGO) is inspired by pine trees' growth pattern. A pine is a type of tree that develops two branches at each developmental stage (time period or height). Initially, it has no branches (k = 0; k represents the number of branches on the tree). The tree then reaches a certain height and begins to form branches on two opposite sides (k = 1 for the first set of branches). At each stage, two new branches are created (k = 2, 3, etc.). PGO conducts the feature selection process based on the growth pattern of the pine. It is important to note that pine tree growth can be affected by a problem's requirements[69].

The primary benefits of PGO are as follows: 1) It enables any classifier to distinguish between stego and cover images. 2) Any classifier's steganalysis performance is enhanced by selecting the most pertinent features using PGO. 3) PGO chooses the subsets of features that are sensitive

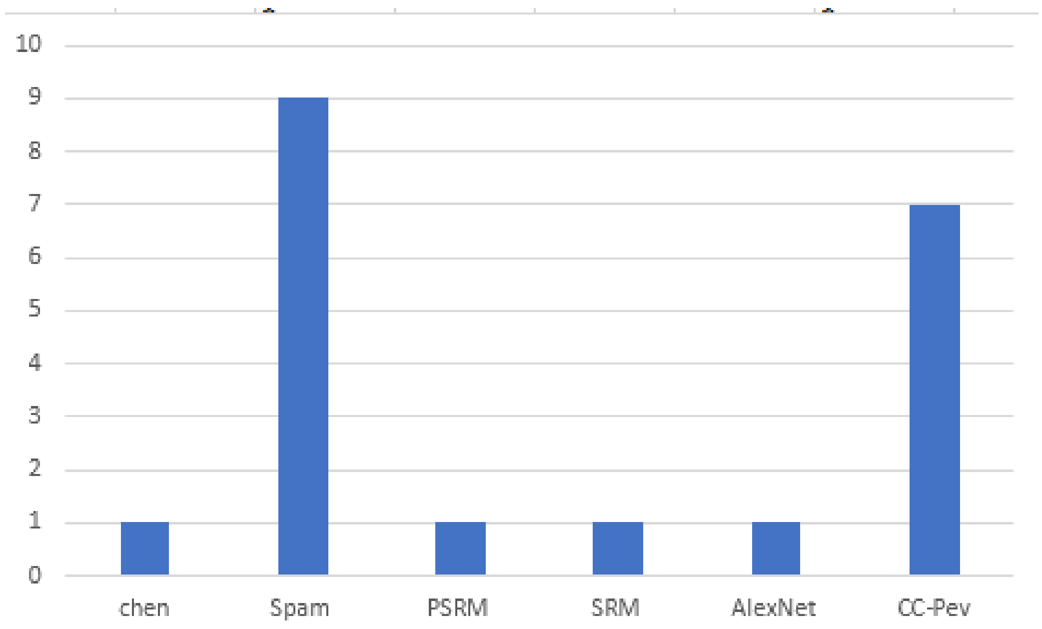


Figure 4.5: Frequency usage of Feature extractors

enough to function with the steganographic scheme. 4) According to this study, PGO also serves as a remedy for COD.

4.3.10 Generative dataset for steganalysis

The authors of the aforementioned research studies extracted features for classification purposes, specifically image steganalysis, from well-known raw image datasets, such as Breaking Out Steganography System (BOSS) (BOSSbase 1.01), 1000 Pictures, and Photobucket.

⋄ **SPAM** Researchers have adopted the most common feature extractors. One of them is proposed in a research study presented by Pevny [162] for Subtractive Pixel Adjacency Matrix (SPAM) feature extraction from the spatial domain of the digital images. SPAM has two clusters of features: one group derived from markov-horizontal and vertical and another extracted from markov-major and minor diagonal features. Merging these two groups brings the total number of SPAM features to 686.

⋄ **CC-PEV** Researchers propose CC-PEV features [161], from transforming the domain of the digital images. It is important to note that CC-PEV consists of 548 feature vectors, 274 of which come from the original image and the remaining 274 from the calibrated image [100].

⋄ **AlexNet** The third feature extractor found in the literature is AlexNet [103], which is a deep convolutional neural network that extracts the features directly from images.

⋄ **SRM and PSRM** There are still other state-of-the-art feature extractors that researchers take into account: Spatial Rich Model (SRM) [60], which provides 34,671 features; projected SRM (PSRM), which extracts 12,870 features [77]; and CHEN[32], which is the first common feature extractor that generates 486 features.

4.3.11 Algorithms efficiency

The training of a model using data for training plays a crucial role in machine learning. The trained model allows researchers to accurately predict test data. When the number of feature dimensions is excessively high, either the learning process is prolonged or the trained model performs poorly due to the CoD. Due to the aforementioned obstacles, the phases of data analytics require more time and are not compatible with visual computing solutions. In this study, a large number of studies have been analyzed to provide a trend in the world of image processing for real-time processing. All studies indicate, and it goes without saying, that when more resources and processors are available, the likelihood of real-time computing increases. When we looked at the above studies, we found a direct link between how well data analytics algorithms for visual computing work and how much hardware is in a system.

Due to the nondeterministic and heuristic nature of evolutionary algorithms, as well as their wide range of application-dependent tuning parameters, there is no simple evaluation metric that can be used to compare the performance of these algorithms to other existing methods. We think that future research should focus on a new way to measure how well evolutionary algorithms work.

4.3.12 Further optimization and efficient data analytic solutions for CoD

All of the scientific studies discussed in this chapter are based on an optimization procedure utilizing imaging applications and evolutionary algorithms. In their research, Mohammadi and Saniee Abadeh focused primarily on EAs and provided an exhaustive analysis of image steganalysis. Mohammadi *et al* [139] also investigated a large number of research studies to tackle high dimensional data using evolutionary algorithms and their applications on real-world problems. Further, Farahani *et al* [51] intended to address unsupervised domain adaptation and worked on domain adaptation groups from various prospectuses. Scientists tried to put together a list of the most important and influential research studies that had a big effect on big engineering and science problems [186].

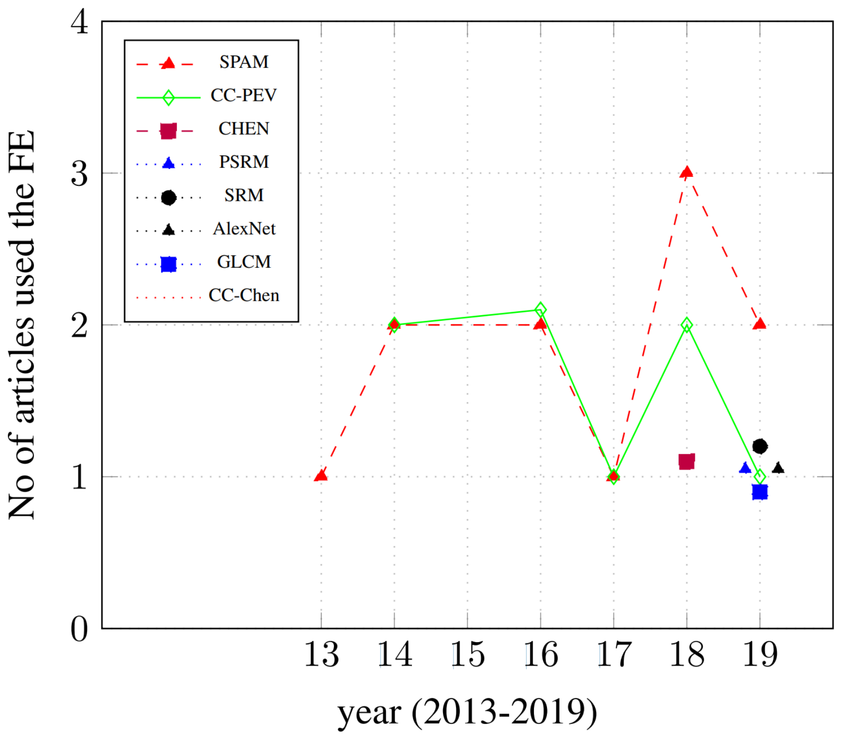


Figure 4.6: Feature extractors usage within 2013-2019

Additionally, Kumar *et al* [104] To address CoD, he presented a randomised smoothing technique. In addition, there is a vast selection of distributed optimization algorithms for complex engineering problems, such as image captioning [14], distributed methods for smart cities applications [12] and decentralized artificial intelligence for energy systems [82]. Decentralized/distributed large-scale problem-solving in the realm of decision-making is facilitated by these computationally effective techniques.

4.4 Promises of MTL: learn to learn

Prior research generated models from a training dataset using a simple learning mechanism. Applying the generated model to test data. The learned model is trained using the hyper-parameter $theta$, which is once-tuned and will be used in subsequent testing processes. To improve this method, however, new technology and human intelligence for learning in new environments are required. They must be able to learn both new data categories and a new data domain. Meta-learning, a more

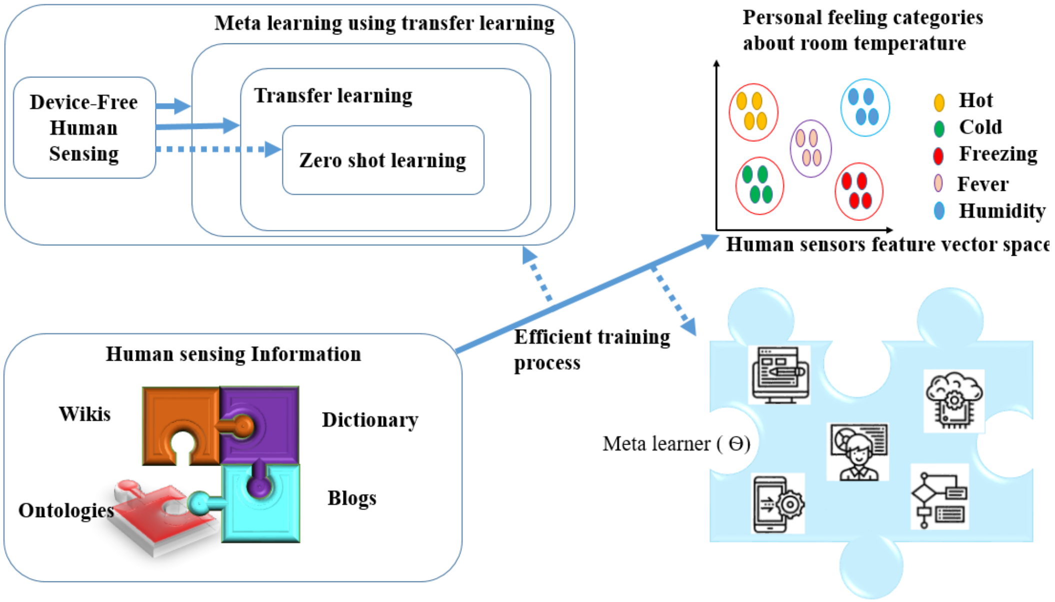


Figure 4.7: Proposed device-free sensing framework for zero shot learning in human sensors

flexible algorithm for learning that needs less data to train, is what is needed to solve this problem [55]. Instead of relying on the model, meta-learning (MTL) attempts to learn an algorithm with the parameter *theta*. It focuses on learning how to learn. It yields different results than conventional ML techniques.

4.4.1 (Meta)Transfer learning

: Transfer learning is the new machine learning paradigm that enables the machine to transfer the learned model to the next level and add new information to itself, as opposed to remaining static. The primary difference between transfer learning and machine learning is that transfer learning begins with a bias as opposed to a random starting point or with no prior data. This allows the algorithm for transfer learning to disregard outliers and incorrect data.

4.4.2 Zero-shot learning

The most important thing about ZSL is the ability to learn data without knowing anything about it before and to use data to learn and move forward [41], [191]. ZSL has an abundance of practical applications. Zero-shot learning, unlike conventional ML models, does not require additional sensors due to its evolving learning model, i.e. the learning model is updated in real time. In other words, once the initial training is complete, the model is able to predict newly added human sensor data. Using the method depicted in figure 4.8 to extract side information from a separate semantic space from labels, this is possible. This information can be gathered from direct human input, such as subjective room temperature perceptions, and used to fine-tune the meta-sensing model. The ZSL method is utilized to learn how to navigate through semantic space to the most important human sensor feature vector space. Once the journey has been finished through zero-shot learning and the semantic data has been carefully mapped into feature vector space, the model can start predicting labels it hasn't seen yet.

Numerous research studies on large-scale recognition problems, such as ImageNet Large Scale Visual Recognition Challenge (ILSVRC), which examines algorithms for performing large-scale object detection tasks, have been completed with success [177]. The objective of artificial intelligence and deep learning, particularly deep neural networks (DNNs), is to achieve human-level performance at scale. In this dataset [177], there are approximately 8.7 million distinct images, making object recognition nearly impossible. In addition, to achieve high performance in object recognition, DNN requires hundreds of images, but many of the classes are uncommon and difficult to locate. Because of this, it is not possible to get enough samples for some of the rare classes [233].

Utilizing a new learning mechanism known as zero-shot learning (ZSL) [122] is one strategy for overcoming this obstacle. An example of zero-shot learning is the process of recognizing unseen objects in the test process by extracting prior knowledge from seen classes during the training process. Suppose, for instance, that you have seen a horse before and have learned from your experience that there is a group of animals called zebras that resemble horses but have black and white stripes. Then, you would have no trouble distinguishing a horse from a zebra. Upon encountering these two animals. In order for a learned model to behave similarly to humans, it is referred to as a ZSL model, in which a subset of labeled seen classes is required during the training process to determine a projection function from seen classes to semantic space. To do this, ZSL needs to be able to transfer knowledge from classes that have been seen to classes that have not been seen.

4.4.3 Zero-shot learning vs few-shot learning

The majority of research studies [54], [56] address the problem of object recognition in a fully-supervised, few-shot learning environment. In this instance, all object categories that appear during the testing procedure are referred to as prior information, and examples from all object categories are available during the training procedure. For many real-world applications, such as image classification, where information about certain image categories is unavailable during the training process, the fully-supervised problem setting appears to be at odds with scientific reality. Consequently, researchers are seeking new learning tasks to solve the problem of object recognition using a zero-shot learning strategy. In a zero-shot learning environment, tasks and objects that haven't been seen before are tested without any prior knowledge. This makes zero-shot learning a convincing environment.

4.4.4 Zero-Shot learning: challenges and drawbacks

Semantic space plays the main role in zero-shot learning such that without reaching semantic space zero-shot learning fails to examine projected function from semantic space to unseen classes. Given semantic space provides different templates: one attribute space (A) [99], [147], one word vector space (W) [61], [191], one combination of both the same time (A+W) [6] and one manual attributes (M) [125].

word vector space (W):

For the generation of word vector space, researchers employ the word2vec [128] model, which is a well-known distributed word representation algorithm in natural language processing (NLP). Initially, scientists used word2vec to map both visible and invisible class names to semantic vectors. Then, they build a semantic-directed graph on top of the semantic vectors to show the direct connections between classes that can be seen and classes that can't be seen [113].

In addition, the domain gap between the seen classes in the training process and the unseen classes in the testing process is one of the challenging aspects of ZSL. In other words, the seen and unseen classes are extremely dissimilar. Therefore, using attribute features to define unseen classes, even for humans, is ineffective. So, the projection function that was made during training might not be good enough to project an unseen class sample to its associated class name in the semantic space so that it can be recognized [122]. Researchers have addressed the projection domain problem; a number of ZSL models [37], [206] use unlabeled unseen classes and transductive learning to mitigate the domain gap problem.

Figure 4.8: Semantic space generation process for new domain

4.5 Meta-Sensing: learning without additional sensor deployment

Currently, we are investigating our novel meta-sensing framework and algorithms as an advanced method of leveraging meta-learning algorithms, i.e. advanced transfer learning, for device-free sensing. Our novel ZSL-based framework for meta-sensing is depicted in figure 4.7 . The framework is comprised of three key elements: semantic space (ontology, wikis, dictionaries, and blogs); ZSL application; and meta learner. First, the algorithm attempts to extract the most pertinent features from the semantic space, i.e. ontology, which enables meta-sensing via the analysis of meta-training data. Then, without transformation, the most important features are extracted using a feature selection algorithm. Singular value decomposition is used to locate the most valuable eigenvalue and eigenvector. This adds a new dimension to a set of qualities. Additionally, it helps us locate the label space. Figure 4.8 provides the sequential process of semantic space generation. In the next section, we provide details of the semantic space generation process.

Notably, Meta-sense learns the label for each category, indicating that MTL has a static bias toward each category. Once a new category is recognized, the separate bias is updated to include the newly recognized category so that the performance rate for minority classes is not affected. The model stores the bias for each class separately.

4.6 Semantic space generation process

In this section, we describe the methods used to create a semantic space dataset for zero-shot learning. The process of feature extraction is first discussed, followed by feature selection. We'll end by talking about the step of reducing dimensions that will lead to a clean, useful semantic space.

4.6.1 Feature extraction using natural language processing (NLP)

Preparation is one of the most difficult aspects of computing machine learning and NLP models, requiring up to 80 percent of researchers’ time at the beginning of their studies. Text processing is a step in natural language processing (NLP) used to generate clean datasets and extract significant and meaningful features prior to feeding machine learning models. Extraction and selection of features are crucial preprocessing steps that are discussed in depth.

The extraction of features is crucial for creating a dense semantic space. Typically, natural language processing (NLP) tools, such as nltk, one of the most popular libraries that offers numerous functions to users, are used to extract semantically significant and meaningful information from text, blogs, and dictionaries. During text processing, the nltk provides an option for various languages, including English, to comprehend the roots of words.

4.6.2 Regular expressions (RegEx)

During text processing, it is crucial to understand our regular expression expectations for each class. When searching for a series of words, a combination of words and a number, or a specific number, we employ a specific search pattern. When searching for ’nlp’ or any meaningful words between ’j’ and ’q’, for instance, we use the search pattern ’[j-q]+’. In addition, we use ’[0-9]’ to search for specific digits, but ’[0-9]+’ to return a sequence of numbers, such as the year 2018. Moreover, we use ’[j-q0-9]+’ to search for a combination of words and numbers.

4.6.3 Punctuation

We need to be aware of punctuation like [!”&# ∗ @ <=> | ′.− :; (/)]. To remove punctuation from the input data, we utilize a *string* library that contains a punctuation function that includes all possible punctuation.

4.6.4 Tokenization, stop words and root finding

After removing punctuation, we must remove words that are irrelevant to our analysis, such as [and, I, you, at, she, he]. To achieve this, the text must first be tokenized and separated into individual words. To ensure that all words are treated equally, we lowercase them. Then, we employ a function 4.1 from nltk version that returns a list of stop words. Next, we look at each tokenized word to see if it’s on the list of stopwords. If it’s not on the list, we keep the words for further analysis.

$$Stopwords = nltk.corpus.stopwords.words('english') \tag{4.1}$$

Tokenization

To tokenize words, we must first find out what the root of the word is. This lets us figure out how often a word appears and use the most common word as an important feature for a certain category.

NLP offers numerous solutions for determining the root of a word, each with its own advantages and disadvantages. In this section, we provide two important solutions that *nltk* library offers to us. For stemming we use a function called *PorterStemmer* is shown in equation 4.2. Furthermore, for lemmitazing, we use *WordNetLemmatizer* is shown in equation 4.3.

$$Stemming = nltk.PorterStemmer() \tag{4.2}$$

$$Lemmitazing = nltk.WordNetLemmatizer() \tag{4.3}$$

Stemming

The first step is stemming, which is the process of taking each word and retaining its root. In other words, Stemming is the process of removing the ends of words, leaving only the roots. This solution is faster but less precise. Table 4.4 provides examples of various stemming root words. This demonstrates that stemming is executed algorithmically and disregards the semantic value of words.

Table 4.4: Stemming and Lemmitazing examples of finding root

Words	Stemming root	Lemmatizing root
Stemming/Stemmed	Stem	Stemming/Stemmed
Berries/berry	berri	Berries/berry
Go/goes	Go/goe	Go
flowers/flower	flower	flower
do/doing	do	do/doing

Lemmatizing

Lemmatizing is the process of forming groups of words with similar meanings based on context. This solution is effective because it performs semantic root-finding, although it is slower than stemming. Table 4.4 demonstrates that the outcome of lemmatization is meaningful and that it is highly probable that we can recognize nonsensical words using stopwords. However, Stemming solution provides no meaningful words, making it difficult to determine whether a word is meaningful or not.

4.6.5 Vectorizing

Vectorizing is the process of encoding text as integers to generate feature vectors that machine learning algorithms can comprehend. A feature vector is an n-dimensional vector that represents an object. In order to accomplish this, we employ various techniques, such as N-gram vectorizing and $TF - IDF$, a combination of term frequency (TF) and inverse document frequency weighting (IDF).

4.6.6 Feature Selection and dimension reduction

Even in natural language processing, feature engineering, which includes feature selection, has become a computationally time-consuming process. Numerous text data are being generated from diverse sources. Therefore, it is necessary to extract important and valuable features from these massive datasets.

The dataset is vectorized after creating a dataset of words and their associated values based on the vectorizing algorithm. It's time to start feature engineering and feature selection by identifying features that have strong correlations to the target feature.The objective of feature selection is to select one or more promising sets of features.

In certain instances, after feature selection, we are left with a large dataset and must convert it into a new dimension and new features. In order to solve the high dimensional problem, also known as the curse of dimensionality, we generate a new dataset with a new dimension using PCA and LDA (CoD).

4.7 Applications of semantic generation process

Machine learning has been utilized to address problems that cannot be resolved using conventional programming, i.e. programming using algorithms. Text recognition (OCR - Optical Character Recognition), shape, face, and motion should all be incorporated into the application of machine learning in computer vision. The most prevalent applications of machine learning in natural language processing (NLP) are spam email detection and text translation. In robotics, machine learning has made it easier for robots to move around in space and do tasks that should be done by their hands [195]. This chapter concludes with a discussion of an unusual ability of intelligent systems based on machine learning, such as Rasa and GPT-3. Despite the fact that Rasa leverages all the steps we discussed in this chapter with a pipeline, it does not guarantee a positive outcome because it depends on the input data you provide. However, GPT-3 has already learned from millions of archives and created a rich and advanced demonstration through tuning to provide a superior result than Rasa.

4.7.1 Rasa

Rasa is an open-source system for machine learning and voice-based discussions. Receive messages, participate in discussions, and interface with notification channels and APIs. Rasa utilizes four essential records, including space, stories, configuration, and normal dialect understanding (NLU) records. NLU could be a module for understanding common dialects. It consists of modules with loose coupling that combine a variety of normal dialect handling and machine learning libraries into a dependable API. Using Tokenizer, NLU will transform the approaching content into tokens for this preparation. Part of the discourse indicator: A part-of-speech tagger, or POS-tagger, groups words together and assigns a discourse tag to each word.

Rasa computes capacities using an activity record, and all anticipated activities will be activated similarly. From preprocessing to implementing the machine learning display, the configuration file contains a list of the sequential steps that must be taken. We are able to update this list based on the domain-specific and quantity of prepared-related data we possess.

4.7.2 GPT-3

In particular, in 2020, OpenAI's GPT-3 (Generative PreTraining Transformer 3) [2] created a machine-learning-based model that could write like a person on virtually any subject. GPT-3 is a deep neural network that has been trained on billions of English internet articles and can generate new articles.

GPT-3 provides a variety of services to aid in the study of NLP applications. A well-known example is a question-and-answer system, in which any question can be asked and the system will provide an answer. In addition, the technology can convert phrases into machine-understandable

---

[2] `https://beta.openai.com/`

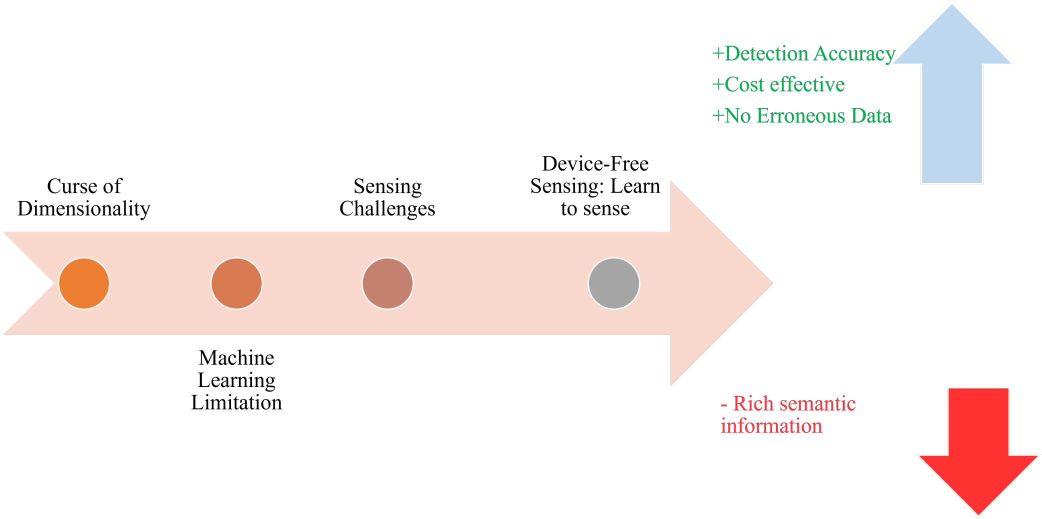


Figure 4.9: Meta-Sense using Zero shot learning: advantages and disadvantages

SQL queries. In addition, GPT-3 can remember all questions and responses and respond based on your current answers. Additionally, GPT-3 offers classification capabilities.

4.8 Discussion

This chapter presents the Meta-sense algorithm, which can detect unseen classes by learning from previously observed classes and the current environment. Meta-sense addresses sensing issues by offering a promising outcome with its evolving model. This offers numerous benefits and opens a new avenue for researchers to explore previously unseen classes in their field of study. The primary advantages of Meta-sense include, but are not limited to, high accuracy in unseen recognition and detection, cost and time efficiency, and no erroneous data inclusion. However, Meta-sense may have one disadvantage that could negatively impact its performance if researchers are unable to generate a suitable semantic space for latent space representation. Figure 4.9 shows that meta-sense has dealt with CoD and machine learning problems to make a promising result for recognizing classes that haven't been seen before.

4.9 Conclusion

Meta-learning addressed the emerging problems in machine learning by utilizing an additional semantic space of labels, where semantic information is transformed into a new dimension to accommodate unobserved test data in the new feature space.

In this chapter, we investigated the zero-shot learning solution to enable our novel paradigm, i.e. meta-sensing, in the human sensing domain. This is the first study to our knowledge that employs MTL to enable real-time learning for device-free human sensing. It enables a paradigm shift from current sensing solutions to models of learning that are revolving and device-free. In the next chapter, we'll try to use Meta-sense on datasets of human movement and activity to recognize motion that can't be seen. This is useful for healthcare applications.

CHAPTER 5

APPLIED META-SENSE FOR HUMAN MOTION RECOGNITION IN THE APPLICATION OF HEALTH CARE [1]

---

[1] This chapter is a reprint and extended version of the following papers, with the permission of publishers:
1) Farid Ghareh Mohammadi, Ahmed Imteaj, M. Hadi Amini and Hamid R. Arabnia, Human Motion Recognition Using Zero-Shot learning, Advances in Artificial Intelligence and Applied Cognitive Computing, pp. 171-181,Las Vegas, July 27-30, 2021.
2) Farid Ghareh Mohammadi*, Farzan Shenavarmasouleh*, and Hamid R. Arabnia, Internet of Things (IoT) in Healthcare: Applications, Challenges, and Solutions, IEEE Xplore The 2021 International Conference on Computational Science and Computational Intelligence, CSCI'21: December 15-17, 2021, Las Vegas, USA

5.1 Abstract

Recognizing unseen objects and categories has become more popular and challenging in advanced machine learning as a result of significant advancements in sensor technology. Motion recognition addresses some of the emerging obstacles in computer vision applications, such as the analysis of actions in surveillance footage with insufficient training data. Human action and behavior recognition also require motion recognition. Real-world motion and activity recognition techniques must account for all motions and activities in the majority of domains, especially the target domain. However, motion and activity classes are not displayed during training time. To close this gap, a complex and expensive procedure requiring extensive knowledge and technical specialists is necessary. In general, we seek a method that can extend the rich model's recognition of previously unseen actions and movements. In this chapter, we propose using meta-sense as a novel action and motion recognition solution that employs zero-shot learning to recognize and detect unseen categories, enabling specialists to monitor patients even during activities not visible during training. By overcoming the limitations of machine learning in the field of human action recognition, we identify previously unseen classes. We utilize a dataset from the machine learning repository at UCI to evaluate the efficacy of the proposed solution. We can apply zero-shot learning to human motion and action recognition using this UCI repository. According to what we found, the proposed method is better than the best algorithms that are available right now.

5.2 Introduction

Motivation: The digitalization of the healthcare industry has been made possible by the development of smart devices. For instance, smartphones and Internet of Things (IoT) devices allow us to live a better and more comfortable life. As a result, people all over the world utilize a variety of devices, such as wearable sensors and smartphones, on a regular basis. Smartphones contain a variety of sensors that capture vital body data and detect activities such as steps. In addition, these smart devices have been used to collect a vast amount of sensor data on the vitals of patients [152]. These devices are used for real-time patient monitoring. Numerous healthcare systems monitor patients' diabetes using smartphones and wearable sensors [10], [152], [189]. Due to the lack of motions and activities, these systems are limited in their ability to collect data in real time. As a result, knowledge extraction and analysis from healthcare data has become an ongoing challenge for healthcare monitoring systems [152]. Unlabeled data generated by these smart devices is one of the obstacles, making it difficult to analyze patient data. So, an intelligent method must be able to deal with data that hasn't been labeled using advanced machine learning techniques like zero-shot learning [149].

Activity recognition is considered a difficult task because the number of states in which to recognize activity is increasing, making it difficult to compile a dataset that includes all sample motion and activity states [22]. Using zero-shot learning (ZSL), which creates a mapping from available features to semantic meaning and enables the recognition of categories despite the absence of test data, this predicament can be resolved. For classification and prediction, existing ZSL-based works frequently consider a projection function that takes feature and semantic space into account. But older models can't predict when the test data contains data that hasn't been seen before. Because of this, the project domain shift problem creates problems. In [99], the concept of an encoder-decoder paradigm for ZSL is presented using semantic space and encoder-decoder learning. Their proposed model can be used as an under-complete auto-encoder, i.e., it is capable of learning data structure and data visualization and regenerating the input signal. They considered an auxiliary constraint for the decoder, which stipulates that the decoder must be able to regenerate the original output of a visual feature, thereby accelerating the performance of their algorithm.

The author in [223] proposed a zero-shot emotion recognition from speech samples in which the samples are considered to be extracted from never-before-seen emotion states. They used attribute learning to make a connection between each emotion attribute and a label. They then used label learning to predict the unknown emotional state. Besides, a video action recognition technique using a self-training and data aggregation approach based on ZSL is propounded in [224] for improving the performance of mapping between the video action features and semantic space. To devise this, they project their class labels to the semantic embedding space, learn to map visual attributes to the semantic space, and aggregate their training and auxiliary data to attain zero-shot learning. Some social learning study [80] demonstrated human behavior learning during the purchase of a product by learning from the knowledge of other consumers. It shows how different people's willingness to buy a product is related to what other people have said about the product's quality.

Zero-shot learning is essential for recognizing unseen, unlabeled objects and significantly overcoming the limitations of machine learning. This algorithm is used in this chapter to recognize unseen human motions by learning from observed human motions. ZSL provides a projection function to identify classes that are not visible. Learning a visual-semantic function and representing objects semantically is the crux of ZSL. In order to accomplish this, researchers have proposed complex projection functions, which increase the risk of overfitting and consequently produce poor performance on unseen classes. However, simple linear functions have low classification accuracy for the observed classes and perform poorly for the unseen classes.

Contributions: This chapter's primary contribution consists of two essential steps. One developing semantic auto-encoder (SAE) and extending a work in [99]. Kodirov et al. applied SAE to image classification in order to identify classes that were not visible. We apply SAE to the dataset of human motion to achieve high performance in human motion recognition. In addition, in the second step, we tune the SAE's parameters to improve its performance. We tune $lambda$ in addition to other hyper parameters such as $HITK$ and $DIST$. After examining and experimenting with various parameters, we discovered that only HITK has a positive and direct effect on the proposed method's accuracy.

5.3 Related work

5.3.1 Machine learning algorithms

In order to thoroughly cover the entire sample space, machine learning algorithms use the inductive learning process to learn from training data and identify a pattern that corresponds to a model. Inductive learning strictly acquires knowledge from labeled observed classes. Failure occurs if the number of labeled seen classes in training datasets is not captured. As a result, we have a model with poor performance for testing on test datasets. There are two distinct types of learning: supervised and unsupervised. In the subsequent subsections, we will briefly discuss these two categories.

Supervised Learning

Supervised learning occurs when the input dataset contains a target value for each instance, requiring the learning model to map instances to their respective target values. This mapping function refers to a model or pattern that is designed to make predictions on test datasets. A human activity recognition model has been developed utilizing mobile embedded sensors and the fixed points arithmetic of the support vector machine (SVM) developed in [16][18]. All of these studies utilized mobile sensor data to predict human activity and motion, but they may encounter difficulties if a test activity is unknown. Due to the diversity of human activity and the heterogeneity of activity patterns, the model is likely to receive unspecified test data. In such a scenario, a conventional supervised learning model would incorrectly predict the category, thereby decreasing the system's performance accuracy. All of these works utilized mobile sensor data to predict human activity and motion, but they may encounter difficulties if a test activity is unknown. Due to the diversity of human activity and the heterogeneity of activity patterns, the model is likely to receive unspecified test data. In such a scenario, a conventional supervised learning model predicts the incorrect category, degrading the performance accuracy of the system.

Unsupervised Learning

Vaha *et al* [201] used hip-worn accelerometer to recognition of lying, sitting, standing motion. Ghareh Mohammadi and Amini [146] proposed meta-sense, which utilizes device-free human sensing to learn to sense. In [98], a regularized sparse coding-based unsupervised zero-shot learning model is made. This model learns from labeled source data and unlabeled test data to create a projection function. The authors in [73] He focused on the disadvantages of traditional ZSL, in which the association between attributes must be manually defined. In this study [73], the authors came up with a way for the unsupervised learning environment to automatically make connections between the attributes of the dataset and unknown classes.

5.3.2 Human action and human-object interaction recognition

Action modeling was the first step in understanding human activities. Numerous studies have modeled and interpreted activities utilizing semantics. J. Gibson's introduction of the affordances concept initiated the HOI modeling procedure [65], followed by some work in the field of object and verb functionality understanding [193]. For HOI understanding, several techniques have been utilized to model semantic relationships [236]. Delaitre et al. [45] presented a method for modeling the spatial interactions between individuals and things using interactional elements. Poselet [124] and phraselet [47] proposed learning distributed representations of individuals and things for HOI recognition. The majority of these approaches require costly labeled data (position, body parts, and object segmentation, for instance), making it difficult to collect data for any type of activity and apply it to cases with a small number of classes. In reality, they fail in situations involving multiple classes.

5.3.3 Zero-shot learning

Zero-shot learning is an appealing method [28], [62], [72] in a variety of disciplines. The majority of cutting-edge zero-shot learning algorithms [88], [156], [232]have two stages and a focus on quality. The first step predicts properties, while the second infers class labels. Compositional learning has been studied for Visual Question Answering (VQA) [15], where the VQA problem is broken down into a series of modular subproblems.

For zero-shot action recognition [91], which employs the two-stream faster R-CNN with one fc layer operating on the concatenated features of both streams, it is suggested to train object-action detectors in movies simultaneously. This method is not limited to recognizing human actions; it also recognizes "cat eating" and "dog jumping." The characteristics have also been utilized in an autonomous learning framework for recognizing objects and actions in order to comprehend human activities [117]. The strong link between objects and actions is exploited for zero-shot action recognition [87].

5.4 Machine learning challenges in IoT healthcare

This section primarily enumerates and discusses the most significant limitations and obstacles of machine learning in IoT healthcare. Figure 5.2 illustrates the close relationship between IoT, machine learning (ML), and individual healthcare (PH). Numerous studies [3] describe the IoT and PH applications of machine learning. Figure demonstrates that IoT generates data that feeds machine learning (ML) algorithms, whose outputs then provide solutions for PH, such as disease diagnosis, patient behavior analysis, and advice for assistive care.

Due to technological advancement, ML and IoT-based assistive PH services have already had and will continue to have a significant impact on people's lives. However, assistive PH will need to address difficult issues such as usability and cost [184]. Additionally, privacy and authentication problems in IoT devices can attract hackers' attention and cause issues as they will be hacked if not correctly secured [3], [184].

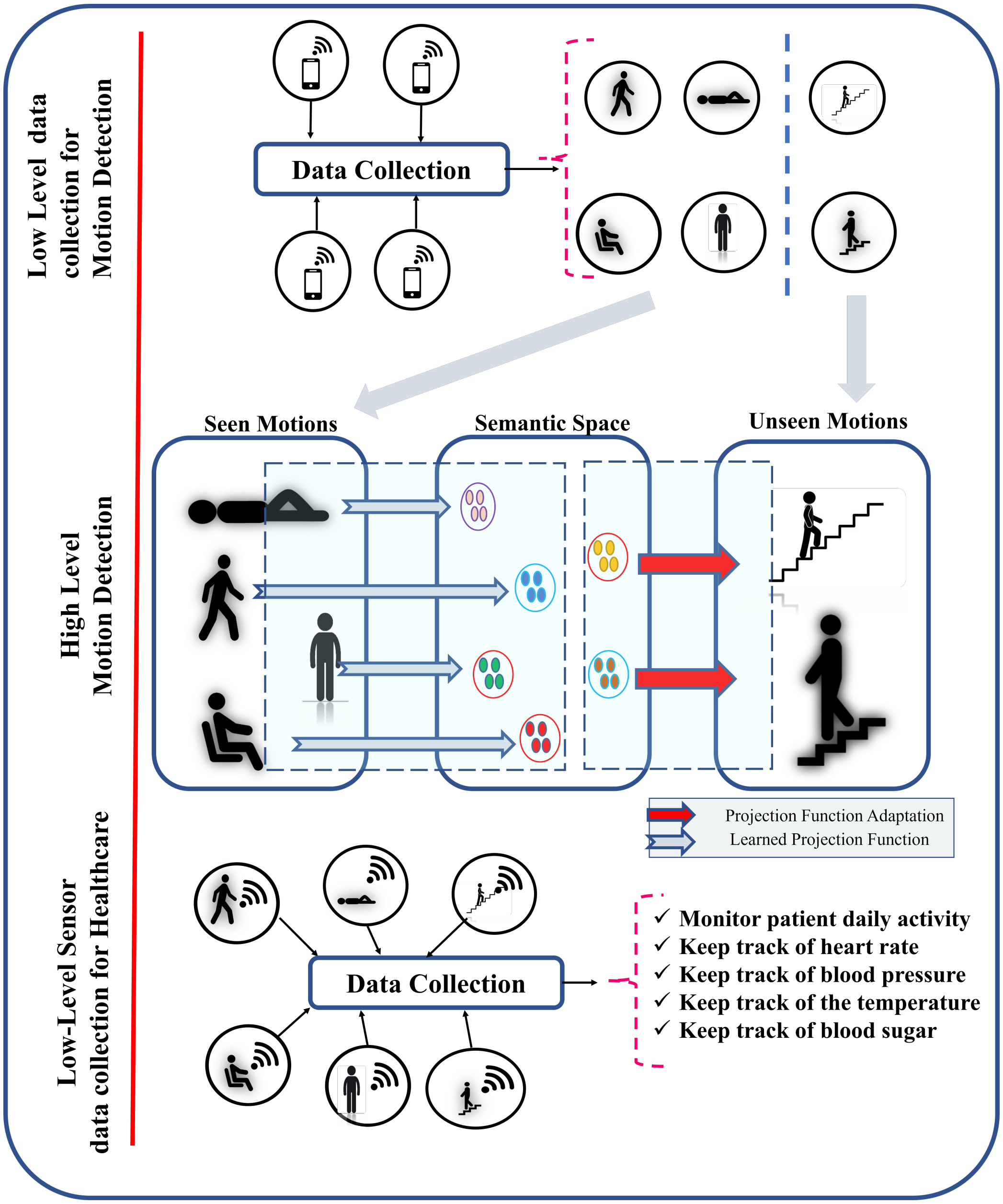


Figure 5.1: Projection function for human motion recognition

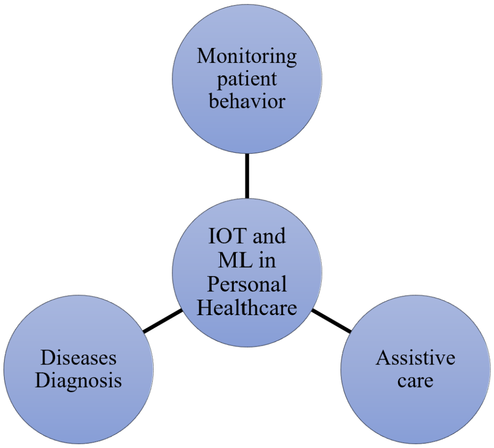


Figure 5.2: A general schema of IoT and Machine Learning applications in Personal Healthcare (PH)

The additional discovery demonstrates that using an ML-based PH service enables us to use a predictive analysis approach that aids discharged hospital patients who may need to be readmitted. The objective of predictive analysis is to develop a risk classification model in which high-risk patients are controlled with additional effort and assistive care, such as providing additional monitoring IoT devices/sensors and constant (real-time) monitoring and analysis. These models are heavily generated based on historical data and previous experience. The dynamic PH system that would aid in re-admission avoidance must also utilize dynamic patient data to predict future possibilities and initiate action to mitigate potential complications [3].

Figure 5.3 depicts an obsolete model that leads to erroneous decisions and subpar performance, as well as an abstract view of the primary challenge and a promising solution. Additionally, the diagram illustrates the associated solution: Meta-sense. In this section, we elaborate on the possible occurrences of this difficulty. The following is a discussion of the corresponding solution.

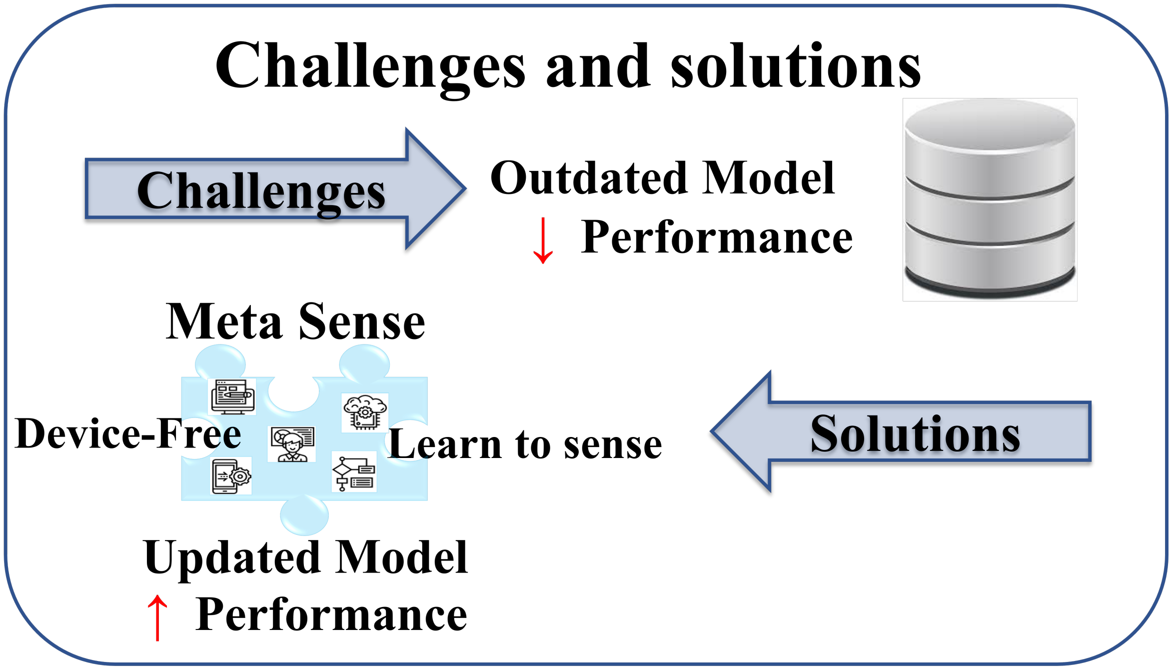


Figure 5.3: A general ML challenges in IoT healthcare and associated solutions

5.5 Proposed method: applied Meta-Sense

Traditional machine learning only learns from observed classes and predicts the same observed classes with new data samples; it no longer generates promising results for unseen classes. In this chapter, we propose an advanced algorithm (Meta-sense) to extend the Zero-Shot learning application for human motion recognition. We extend the semantic auto-encoder (SAE) method, which is advanced zero-shot learning, to learn from four motions and activities, such as sitting, laying, standing, and walking, and then to distinguish between walking downstairs and walking upstairs. The main benefit of the proposed method is that it uses a linear projection function, which gives better results than other methods.

For the classification of unseen categories, it is essential that some of the seen categories have semantic relationships with the unseen categories. For human motion recognition, we have identified two unseen motions and four seen motions. Clearly, walking up- and down-stairs classes from unseen classes are related to walking classes in seen classes. Upstairs classes are walking classes that want to go upstairs, so there will only be one-bit differences between them in semantic space. This means that there is only one column between walking and walking up-stairs.

5.5.1 Preliminary

To understand the idea of zero-shot learning, let's suppose we have a distribution of seen classes, ($\mathscr{D}$)=($\mathscr{X}$, $\mathscr{Y}$) where $\mathscr{Y}$ goes for the sets of seen classes. We need to find a projection function, which takes a training dataset $\mathscr{D}$, then maps them to $\mathscr{S}$ in an optimized way such that we could use that one to map test data to unseen classes. The primary distinction between zero-shot learning and machine learning resides in the input data and the training phases. All of the classes in the test dataset are represented in the training dataset, from which machine learning learns. Moreover, machine learning only learns the data and identifies patterns that may exist in the data based on numbers. However, zero-shot learning also requires a semantic space in which the features accept binary values of either 0 or 1, where 1 indicates that a particular class possesses the feature and 0 indicates that it does not. In addition, zero-shot learning behaves like a human cognitive system, i.e., it comprehends data that enables humans to recognize unseen objects. Figure 5.1 shows how the projection function maps motions that can't be seen to semantic space and how the data collection process can separate data based on how people move.

5.5.2 Semantic Auto-encoder Adaptation on human motion recognition

In the machine learning and computer vision literature, zero-shot learning has demonstrated promising performance results [137]. Using semantic space between the motion and other rooted words, this type of learning has enabled experts to predict and identify newly observed types of human motion (events). We use a popular natural language processing method called "word2vec" [128] to model semantic space. This method takes each motion label as input and turns it into a high-dimensional vector space.

Figure 5.4 illustrates the three primary sections of our proposed method. One prepares a dataset to be utilized in zero-shot learning. To apply our proposed method, we must first convert the dataset into a standard dataset using pre-processing techniques. The raw dataset contained only training and testing datasets. Each dataset contains six types of motion and actions, implying that machine learning algorithms examine test datasets to learn and predict all six types. In this chapter, however, we only divide our dataset into seen and unseen classes, with the seen class consisting of four action types: standing, walking, laying down, and sitting. Descending and ascending stairs constitute the unseen class.

The second section describes the search for the optimal projection matrix or function, which is informed by observed actions and motions. The strength of our method is a linear projection

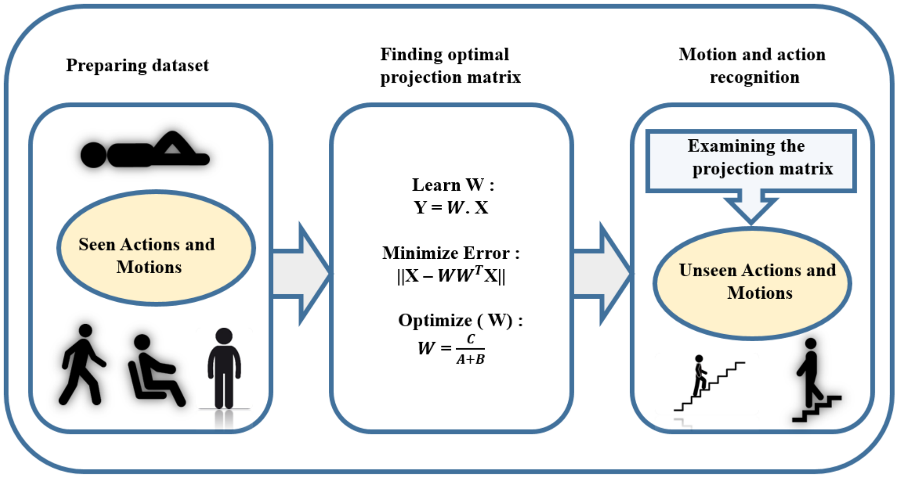


Figure 5.4: Proposed framework for human motion recognition

function that yields the lowest error rate when mapping to a semantic space. In addition, we reduce the time complexity of the learning model by employing a linear function as opposed to a complex model, which could lead to overfitting. Consequently, this section prevents us from settling for the local minimum and running into the overfitting issue. For zero-shot learning, we use a semantic auto-encoder (SAE) to generate a projection function. In [99] used Sylvester equation to compute an optimal projection function, $\mathscr{W}$, using three parameters A = SS^T, B= λXX^T and C= $(1 + \lambda)SX^T$.

The third section is a crucial phase in which the projected function is applied to unanticipated motions and actions in order to improve the outcome. We use the same projected function to embed semantic space for unseen movements and actions. The goal of an auto-nature encoder is to use a projection function $\mathscr{W}$ to turn a training dataset into a semantic space.

5.5.3 Zero-Shot learning tuning

The proposed method consists of one general parameter, $lambda$, and several embedded parameters, including HITK, DIST, etc. We first examine the generated dataset and proposed method with the embedded parameters' default values and then with tuned parameters. When the value of HITK is set to 2, the proposed methods achieve a 100 percent performance rate for unseen motions and actions. We proposed the image recognition performance of tuned-SAE and obtained promising results. This section also confirms the effectiveness of tuned-SAE.

5.6 Experimental result

We create a zero-shot learning technique utilizing a semantic auto-encoder. In addition, we tune the hyper-parameters to achieve optimal performance. This parameter change shows that our algorithm, which used meta-sense to detect human movement, has changed in the right way.

5.6.1 Time efficiency

In this chapter, we utilize smartphones and sensor-generated signal data to address challenging applications such as healthcare. We propose a novel method that requires minimal operating resources and provides a solution for energy efficiency, which is one of the limitations of using smart devices. This algorithm is the most efficient for use in smart devices because it executes in linear time. We wish to increase recognition precision while decreasing computational expenses.

5.6.2 Dataset

In this chapter, we use a dataset [2] including 561 features and 6 labels. This dataset is the result of experiments conducted with 30 individuals ranging in age from 19 to 48 years old. Each participant performed six distinct motions and activities, including standing, sitting, lying, walking, walking up and down stairs, and walking downstairs, while wearing a smartphone (Samsung Galaxy S II). Researchers captured one 3-axial linear acceleration and one 3-axial angular velocity using the smartphone's accelerometer and gyroscope at a static rate of 50Hz. The dataset is generated manually by recording activities, and then labeling them. To obtain the dataset, 70% and 30% of the population were selected to generate the training and test data, respectively.

To apply ZSL to this dataset, we must first convert it to the classes considered seen and unseen. In zero-shot learning, a training dataset consists of classes that have been observed, such as standing, sitting, laying, and walking, whereas a testing dataset consists of classes that have not been observed, such as walking upstairs and downstairs. Zero-shot learning generates an optimal projection matrix or function for recognizing previously unseen classes based on previously observed classes. The training and testing instances in the zero-shot learning process do not have to have the same duration. Additionally, there are no classes shared by "seen" and "unseen". (i.e. Seen classes $\cap$ unseen classes= $\emptyset$)

---

[2] UCI dataset for human action recognition : https://www.kaggle.com/uciml/human-activity-recognition-with-smartphones : access date: Oct. 30, 2019

5.6.3 Supervised learning results

In [16] [18] [170] [17], the researchers applied their method to this dataset in order to detect human motion and action. The results are shown in table 5.1, which shows that the supervised learning algorithms they proposed can work with this dataset.

Table 5.1: Comparing related works using different motion/activity recognition techniques and different input sources, note that SC: Supervised Classification. Notably, the 100% accuracy is based on a small number of simulated experimental samples, and it may not be able to reach 100% accuracy in real-world situations.

Criteria / **Methods**	Classifier	Activity / Motion	Input source	Accuracy (%)
Semantic Auto-encoder- [99]	ZSL	Walking, Laying, Siting, Standing, W. Upstairs, W. Downstairs	Smartphone	47
SC - [16]	Multi class (MC) SVM	Walking, Laying, Siting, Standing, W. Upstairs, W. Downstairs	Smartphone	89
SC - [18]	$MC - HF$ SVM	Walking, Laying, Siting, Standing, W. Upstairs, W. Downstairs	Smartphone	89
SC - [16]	$MC - HF$ SVM	Walking, Laying, Siting, Standing, W. Upstairs, W. Downstairs	Smartphone	89
SC - [17]	Binary SVM	Walking, Laying, Siting, Standing, W. Upstairs, W. Downstairs	Smartphone	96
Mapping of motion recognition for image classificaiton [194]	Alexnet,CaffeRef, K-NN, SVM	Communicating, staying, sleeping, reading, writing, work at computer, eating, studying, drinking	Image	90.75
Deep CNN for motion recognition [172]	SVM, CNN	Walking, Laying, Siting, Standing, W. Upstairs, W. Downstairs	Smartphone	95.75
Optimal ZSL for motion recognition	Meta-sense	Walking, Laying, Siting, Standing, W. Upstairs, W. Downstairs	Smartphone	100

5.6.4 Unsupervised learning

The primary contribution of this chapter is the introduction of a new unsupervised learning algorithm (applied Meta-sense) for classifying unseen objects. We propose and adapt zero-shot learning to identify human motion and actions. The results of the proposed method are displayed in table 5.1; our method outperforms related work and other cutting-edge algorithms.

5.7 Alternative solution for outdated models

A promising algorithmic procedure for meta-sense has already been discussed and demonstrated. In this section, we also suggest a way for scientists to deal with the problem effectively by using an online learning algorithm.

There is a fundamental algorithm in machine learning known as "online learning" or "adaptive learning," which is highly recommended and used to update the best classifier for future data after training is complete. In contrast to batch learning algorithms, which produce the best classifier by learning only from the training dataset and are never updated afterward, reinforcement learning algorithms produce the best classifier by learning from both the training and test datasets. In other words, online or adaptive learning takes into account tasks arriving in a stream (during and after training) as opposed to an offline, finite dataset. Instead of remembering old tasks, the tasks are linked to how well they can adapt to the current task in the stream based on their learning rate [75].

Furthermore, online learning is a new machine learning technique in which it is physically and logically impossible to train over training data such as patient data over therapy sessions, necessitating the generation of data in the future. It is also used when the algorithm needs to train the whole dataset because it doesn't have enough memory [59].

5.7.1 General process of online machine learning

This section aims to provide a concise introduction to an online machine learning algorithm. Figure 5.5 depicts the algorithm for online machine learning in its general form. In general, the online machine learning schema works with three major packages: 1) a wearable or healthcare IoT device input package; 2) a five-step online machine learning package; and 3) the results and projections projected on AI devices. The primary steps of online machine learning are identical to those of traditional machine learning, with the exception of retraining the learning model with a learning rate that indicates the significance of new input data. The greater the significance of the input data, the faster the learning rate. If a faster learning rate is selected, the model will be more susceptible to outliers and may become stuck in a local minimum or maximum. Conversely, the slower the learning rate, the less susceptible the system is to outliers. Because the learning rate is slower, it takes a long time to update the model with new data.

Figure 5.5 depicts the five steps of the online machine learning package that are crucial to the application of online learning in IoT healthcare. Reading input data (training data), training a model, and applying the model are the first three steps in traditional machine learning. Step 4 of the online learning algorithm involves accepting a new sequence of data, while step 5 involves learning from the newly added data and updating the model with a customized value for the learning rate.

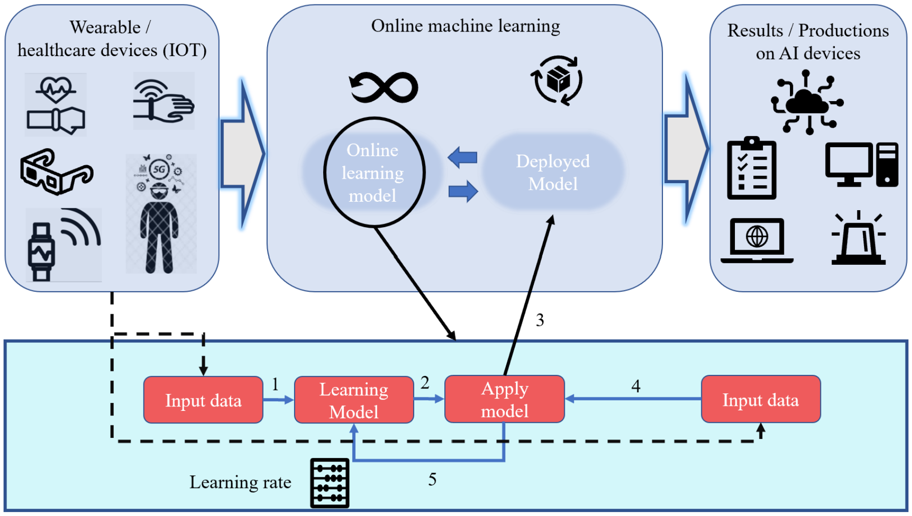


Figure 5.5: A general schema of Online Machine Learning

Online Learning applications in IoT healthcare

With the development of technologies such as artificial intelligence, specifically machine learning and Internet of Things (IoT) devices, AI applications and IoT object usage have proliferated across many fields, most notably and urgently in medicine and healthcare. Never-ending learning, also known as adaptive learning [131] or online machine learning [59], is a new fundamental concept in machine learning in which models continuously learn from data and evolve at a predetermined learning rate while retaining previously acquired knowledge. This non-static process of supervised machine learning algorithms permits the model to iteratively learn from data and update its behavior as the learning rate increases. The narrower the range of behavior, the slower the learning rate. In other words, we do not give additional weight to the upcoming input data. To have the model updated quickly based on the new data, we may need to initialize the learning rate with a relatively large value that is not affected by outliers. The use of recommender systems by companies such as Netflix and Amazon is an example of this online learning. However, such systems present their own set of challenges, such as fairness in the machine learning process [4].

Big data streaming: Widespread use of big data streaming computation as a key component of real-time healthcare analytics. In the healthcare and medical domain, online machine learning applications such as real-time monitoring and tracking systems play a crucial role. Mobile applications and sensors, as well as wearable medical devices/sensors, are examples of common sources that have consistently produced a large volume of data, namely streaming data [75]. Using traditional machine learning algorithms to make real-time decisions based on streaming data in the event of an emergency appears to be a challenging endeavor. Therefore, a solution for real-time processing of large amounts of data is required to ensure that the results are effective and easily understood. Using the Twitter mining tool, for instance, a real-time solution is proposed for monitoring flu and cancer patients [112]. Researchers in [7] investigated another big data model for real-time medical analysis. The model evaluated a stream of healthcare data using Spark streaming and Apache Kafka. The [43] paper proposed a solution for real-time health status prediction using Apache Spark, a powerful tool for analyzing large amounts of data. Due to the limitations of traditional machine learning and the need to manage distributed computations, Spark was successfully implemented for streaming data.

Online meta-learning is the second important extension of online machine learning. Finn *et. al.* [55] introduced an online meta-learning strategy based on regretful meta-learners. The method utilizes MAML-style meta-learning for adaptive deep learning and online meta-training throughout the task sequence. In this approach, there are meta-training and meta-testing phases in which we learn from input data during both the training and testing phases. Even while testing, a training phase is beneficial to the meta-testing phase. This capability makes MAML appear to be more powerful than competing deep learning methods. In addition, MAML is one of the best approaches for few-shot learning (FSL) and one-shot learning (OSL), which enables classifiers to learn from a small number of examples of each category in order to accurately predict class labels that have not yet been observed [145].

5.7.2 Further challenges of online learning

Although online machine learning appears ideal for IoT healthcare applications, its accurate application remains a challenge. One of the hardest problems to solve is catastrophic forgetting, which happens when a model learns something new that makes it forget what it has already learned [111]. This impediment causes the classifier's performance to degrade significantly while the new input data is being integrated, or it generates a new model rather than retaining prior knowledge [126], [127]. The majority of online learning applications in non-medical fields are less affected by this restriction [164]. In the health care industry, online learning models address a variety of challenges where multiple complex tasks are required.

A simple solution to the catastrophic interference problem is to retrain the model whenever new data becomes available; however, this is computationally expensive and prevents real-time inferences[111].

5.8 Other challenging issue of machine learning in healthcare with associated solution

5.8.1 Challenge: data confidentiality & correctness

The development of medical Internet of Things (IoT) devices and the widespread popularity of ubiquitous wearable devices enables us to have continuous personal healthcare monitoring anywhere, including at home, at work, and in hospitals, for a variety of applications such as assisted living and childcare. These devices collect electronic health measurements in real time from a variety of objects and sensors and transmit them to an application server, where they are pre-analyzed and pre-processed before being stored on a data server [90].

In this processing and analysis, machine learning algorithms are used to provide services such as motion tracking, displaying the number of steps taken, calories burned, sleep monitoring, traveled distance, and vital signs measurements including heart rate, electrocardiogram (ECG), skin temperature, and electroencephalogram (EEG) [71]. Consequently, analyzing data generated by IoT devices and sharing it with a server via a network connection raises concerns regarding unpredictability, trust, and privacy. Both server-stored data and communication networks are susceptible to compromise. Consequently, data security and privacy are crucial considerations. Using encryption algorithms and keeping the data safe are simple ways to increase data security. However, if a hacker discovers the key to the decryption algorithm and decrypts the message, the confidential information will be widely available. Also, it's possible that some data will be lost during the encryption process, and if the decryption algorithms can't get back all the original data, the whole process of encryption and decryption will be useless.

ML Challenges in Monitoring Patient Activities

A number of research studies use data generated by motion sensors to define physical activity in real-world settings in a variety of case studies for patient activity monitoring. Research shows that motion sensors are one of the most accurate ways to track how active cancer patients are over time while they are getting treatment [68].

Due to the nature of IoT devices, the collection of data from medical IoT sensors and devices for healthcare applications is extremely delicate. Recent advances in web technologies and wireless communication/transformation enhance remote real-time monitoring and data collection [90]. But the complicated workflow involved in collecting medical data raises security and privacy concerns throughout the entire data collection life cycle, including data collection, transmission, processing, and storage [176].

The challenging aspect of activity recognition using mobile devices as opposed to wearable devices is identifying the data that preserves user privacy while remaining relevant and important for machine learning tasks [90]. To address this complex issue, we must pose and respond to two questions: First, determine if the collected information is protected by multiple layers of security or an encryption protocol that prevents unauthorized access. How can we determine if the protected data is being maintained with the same accuracy as when it was captured? Finding a balance between how data is processed and how private it is is important for safe data transfer, protecting data on mobile devices, and building trust with end users.

Security, specifically the privacy of patient data during the machine learning analysis process, is one of the greatest obstacles to utilizing IoT for healthcare monitoring. A robust user authentication framework ensures that only authorized account holders have access to data and services. Therefore, the issue is user access to data, which leaves sensitive information vulnerable to hackers. Currently, sharing data from IoT devices will be challenging [90].

5.9 Alternative promises in IoT healthcare

As indicated by the Internet of Things usage statistics, access to large volumes of data collected by IoT sensors and devices has the potential to be applied to a variety of medical and healthcare domains. Nonetheless, the increasing demand for high levels of healthcare data security and privacy causes IoT devices to be known as data-isolated islands [229]. Due to the lack of updated data model issues, data privacy, and security, which are the primary challenges with traditional machine learning algorithms, we also explore the potential of advanced machine learning in IoT-healthcare applications in this section. In the subsequent sections, we elaborate on the promises made. First, we will examine the online learning algorithm, which provides a solution to the problem of outdated data by allowing the classifier to learn from incoming new data and update the model in real-time on each learning iteration. Second, we show a way to learn from distributed end-user data without sending it to an application server. This is called "federated learning."

5.9.1 Solution: federated learning

The presence of heterogeneous IoT devices and associated different end-users' (patients') information on them makes them highly susceptible to hacking during information transfer to an application server or when data is restored on the server, raising data privacy and security concerns. In this section, we will examine a solution that will allow us to solve these problems to an acceptable degree. This solution must be able to integrate a model and keep user data while training on devices that are not connected to each other.

In a traditional machine learning-based approach, data collected by IoT devices is uploaded to an application/data server before trained models are utilized by machine learning algorithms. However, data owners (devices) have proliferated, and the importance of data privacy is growing [84], [116], particularly in the medicine and healthcare field. To address the privacy requirement regarding individually identifiable information, we propose using federated learning, which was first proposed by Google [102], which is a novel approach to resolving this data conundrum, which is defined as the difficult problem of training a high-quality shared global model on a centralized server with decentralized data distributed across a large number of devices or end-users [221].

We propose using federated learning (FL), a machine learning algorithm designed to train a centralized rich model using training data distributed across a large number of users and devices with unreliable and slow network connections [116]. In this instance, we employ the federated learning algorithm, in which IoT devices independently update an input model (received from a central server) based on locally gathered data on demand and transfer this updated model to the central server, where the user-side updates are aggregated to generate/update a new global or generalized model. Mobile phones are the most common device in this setting, where the speed with which data can be sent and shared locally is very important.

General process of federated learning

Federated learning is a method for generating models that differs from other machine learning algorithms, which require all training data to be centralized. In federated learning, a model is trained iteratively by aggregating collective models gathered from multiple sources (devices: information extracted from cell phone users). Users' privacy and local (training) datasets are kept confidential, but they may participate in a shared federated learning process. The combined model is then sent to all users by an aggregator, which also adds the updated model that has learned from the local training data to the devices. [116].

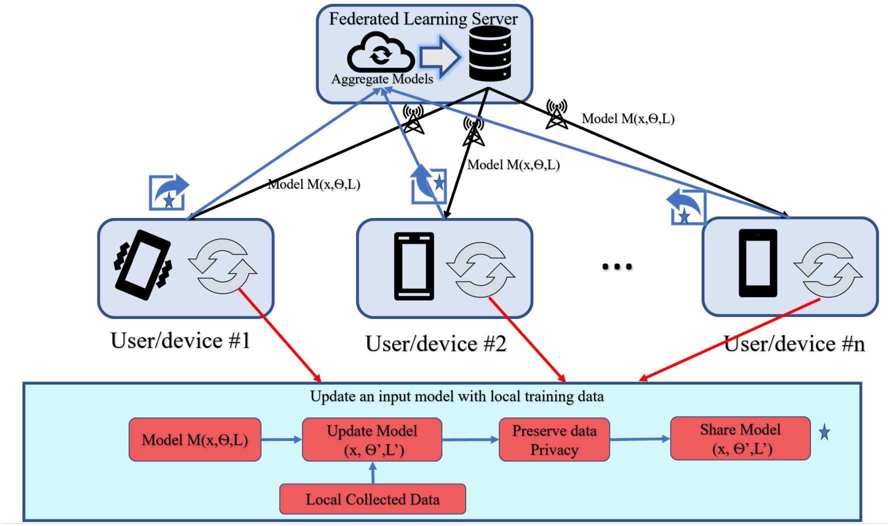


Figure 5.6: A general schema of Federated Learning

We present a very abstract overview of the federated learning process in figure 5.6. This figure shows that a federated learning server communicates with all end-users / devices and gets them to check in the server and then start broadcasting the current model. The end-users / devices receive the model M with input data (x), parameters θ and loss function . Next, the device starts to update the model by learning from the local training dataset which is collected. The updated model M with updated values of parameters θ' and loss function ', per each device will send back the updated model. The server then aggregates the models received from devices, stores them on the server, and repeats the entire process. This process enables specialists to accurately monitor their patients with updated information in near real-time. Moreover, this FL prohibits the dissemination of sensitive information. But federated learning only uses mobile phones as smart devices, which is a problem because of how things work.

Federated learning applications in IoT healthcare

aFederated learning enhances collaborative model training to the extent that raw data sharing is no longer required [81]. We need a federated learning system that can prevent inference over both training messages and the final trained model, as well as ensure that the resulting model has acceptable predictive accuracy [199]. In [212] PerFit, a FL-based solution for learning a machine learning model, was proposed by researchers. PerFit utilizes a cloud-based architecture to provide IoT devices with computing power. The significance of computational capacity is examined in depth in [83] . A well-structured architecture enables IoT devices to offload their computing tasks due to requirements for efficiency and low latency. Due to the fact that FL communicates via multiple devices, servers, and the cloud, models can be shared locally without compromising sensitive data. The learning process of the PerFit framework consists primarily of three steps: 1) the phase of task unloading, 2) the phase of learning, and 3) the phase of personalization. Researchers assess the effectiveness of PerFit in [212] using a data set called Mobile-Act, which focuses on human activity recognition and includes ten different activities such as walking, jumping, and jogging.

FedHealth, a new FL-based framework based on transfer learning, was proposed in [35] for IoT healthcare applications. FedHealth has utilized transfer learning to reduce distribution divergence in a variety of fields. FedHealth also employs a specific encryption algorithm to enhance the security of current model update communications between the aggregation server and end users, and vice versa. FedAvg has also been utilized as a federated optimization algorithm in the proposed FedHealth. Although all end-users have access to the global model, which is kept up to date on the FL centralized server, the personalized model in each user does not guarantee that it will perform well locally [94]. To solve this problem, the researchers in [35] trained a personalized model for each user using transfer learning to improve the performance of the federated learning model for end users.

Researchers proposed [50], a FL framework for decentralized healthcare applications based on deep learning that protects data privacy and enhances data security in a decentralized architecture. In addition, they proposed a plan that employs an automated training data acquisition method. They also tested the algorithm on skin diseases and used transfer learning to solve the problem of making deep learning models without healthcare data.

5.9.2 Further challenges of federated learning

Even though FL has high levels of data security and privacy, some posted attacks found that sharing only local data during training and updating the model M does not provide enough data privacy [199]. Researchers in [199] came up with a hybrid strategy called Privacy-Preserving Federated Learning to make a model with good predictive power while keeping people from figuring out anything from the messages sent locally during training or the final model.

Furthermore, FL on IoT devices provides the following challenges: [8]:

Heterogeneity of devices: In terms of technologies and hardware, IoT devices used for medical and healthcare issues differ from those used for general purposes in several ways, including CPU version, memory storage, network connection bandwidth, storage capacity, and, finally, power. Numerous factors must be configured as fault tolerance, which may increase the cost of implementing FL. Also, some devices may stop learning for a variety of reasons, like when they can't connect to the network or when they run out of power.

Statistical heterogeneity and heterogeneity of models : User distributions of activity and settings play a crucial role in data collection for all forms of machine learning, particularly FL. Due to their vastly different physical characteristics and behaviors, devices in the FL generate diverse data samples under various circumstances and settings, which may hinder the development of a comprehensive model. Because of this, the server application must combine several different models before sending them out again.

5.10 Discussion and conclusion

The Internet of Things (IoT) implementation of machine learning algorithms for network monitoring and user activity management is gaining strength. On the other hand, traditional machine learning algorithms may fail when applied to decentralized IoT device data. Because the nature of the algorithms is to collect all of the training data at once and generate a robust model capable of predicting unknown class labels during the testing phase, it is impractical to collect the training data sequentially. Due to the dispersed nature of the data, conventional machine learning algorithms cannot produce promising outcomes. In this chapter, we discuss the Internet of Things, its pipeline, and its application in healthcare, as well as the challenges that ML algorithms may encounter.

Lastly, we look into the process of collecting data and the problems that come with it, the effect of Big Data on IoT health, general IoT challenges, especially machine learning challenges in IoT healthcare, and new ways to deal with them.

Recognition of human motion is a challenging problem in human sensing. Human motion detection and recognition is one of the most challenging problems in the field of human interaction computing, and its study has grown exponentially year after year. This chapter proposes the use of Meta-sense, a promising zero-shot learning method for recognizing unseen human motions. By employing semantic space and zero-shot learning, we can overcome the inability of traditional machine learning to recognize unseen human motion. The results demonstrate that zero-shot learning with a semantic auto-encoder outperforms other state-of-the-art approaches. Based on the idea of zero-shot learning, we plan to propose in the next chapter a classification method based on meta-learning for putting observed classes into 3D point cloud models.

CHAPTER 6

META-SEL: 3D POINT CLOUDS MODELS CLASSIFICATION USING META-SEMANTIC LEARNING [1]

---

[1] This chapter is a reprint and extended version of the following accepted paper, with the permission of publishers. Farid Ghareh Mohammadi, Cheng Chen, Farzan Shenavarmasouleh, M. Hadi Amini , Beshoy Markos and Hamid R. Arabnia, Meta-SeL: 3D-model ShapeNet Core Classification using Meta-Semantic, The 6th International Conference on Applied Cognitive Computing (ACC'22), CSCE'22, 2022

6.1 Abstract

Understanding 3D point cloud models for learning purposes has become a critical challenge for real-world identification applications such as autonomous driving systems. A wide range of deep learning-based solutions for point cloud segmentation, object detection, and classification have been proposed. However, these methods frequently necessitate a large number of model parameters and are computationally expensive. We investigate a semantic dimension of given 3D data points and propose a practical method known as Meta-Semantic Learning (Meta-SeL). Meta-SeL is an integrated framework that uses two input 3D local points (input 3D models and part-segmentation labels) to provide a precise projection model for a variety of 3D recognition tasks. The results show that Meta-SeL outperforms other complex state-of-the-art work in terms of performance. Furthermore, because it is random shuffle invariant, Meta-SeL is resistant to translation and jittering noise.

6.2 Introduction

◇Motivation

Analyzing 3D point cloud models has become an essential component of many AI systems such as additive manufacturing [123], autonomous driving [95], and augmented reality [9]. Classifying 3D point clouds accurately in real-time can be a challenging task due to the complexity of objects that exist in a real-world environment [200]. Among various 3D models in computer vision, point cloud representations have become increasingly popular as more data is collected by devices such as LiDAR. The goals of modeling 3D objects can be divided into three categories: part segmentation, semantic segmentation, and shape classification. Literature shows promising results utilizing sophisticated PointNet-based deep learning techniques for understanding object classifications from point clouds [120], [200]. These methods, however, often require a considerable number of model parameters and are computationally expensive. In this study, we present a framework for simplifying the classification process by executing only one epoch while achieving competitive results.

point cloud models with 1024 points are shown in figures 6.2 and 6.3. The input dense points, point segmented labels, and labels for each complete point, which is an object, are divided into three pieces in this diagram. There are 10 different things in our collection. such as *airplane, chair, table,* *earphone, pistol, car, lamp*, and *guitar*. These objects are chosen because they have three-part segmentation labels.

◇**Gap:** One major challenge in classifying 3D models is to improve algorithm efficiency and execution in real-time. Several deep neural networks were developed to address this issue, including PointNet, DGCNN, and SimpleView. With a unified architecture, PointNet utilizes permutation invariance of points and processes each independently with a symmetric function that aggregates features. DGCNN introduces the EdgeConv block, which exploits both local and global shape properties for each point as topological information. In SimpleView, 3D point clouds are converted into 2D depth images using a projection-based method as a type of dimension reduction technique. In general, deep neural models require a long training time and rely on many parameters to reach higher accuracy. As yet, these techniques do not convert feature space into semantic embedding space for the problem of 3D model classification.

◇**State of the art solutions:** Recently 3D point cloud datasets and associated solutions for segmentations and classifications are presented and the need to conduct research on 3D Models is proliferated [79], [165], [220]. Here, we summarise the most popular ones. The most common benchmarks for comparing methods for point cloud classification are ShapeNet Core [227] and Modelnet40 [214]. There are current solutions tested on ModelNet40, such as Point-Net [165], DGCNN [210] and RSCNN [120]. In this research, we apply two of these current solutions, DGCNN and Point-Net, with which we compare the results.

◇**Contribution:** In this chapter, we propose an efficient solution for classifying 3D point cloud models that outperforms other state-of-the-art methods. It is also efficient in terms of resource usage. Furthermore, we develop Meta-SeL, a simple framework for performing linear computations on CPUs rather than GPUs. We originally, as shown in Figure 6.1, proposed the idea of Meta-sense in [144], to perform a zero-shot learning (ZSL) task [145], and, further, we presented another work on computer vision using tuned-ZSL [148], and another on signal processing for human motion detection using ZSL [149].

We conduct this research based on previous work, inspired by one part of the semantic auto-encoder solution presented initially by Kodirov et al. [99] in which we generate a projection function (W) from vector space to a semantic space. The key contributions of our research in this collaborative work are as follows:

◇ Design a novel framework, namely Meta-SeL, suitable for 3D point cloud models.

◇ Posit that training and testing phases only need one *epoch* to yield a promising result. This means that Meta-SeL has a high convergence rate, so it is extremely time and cost-efficient.

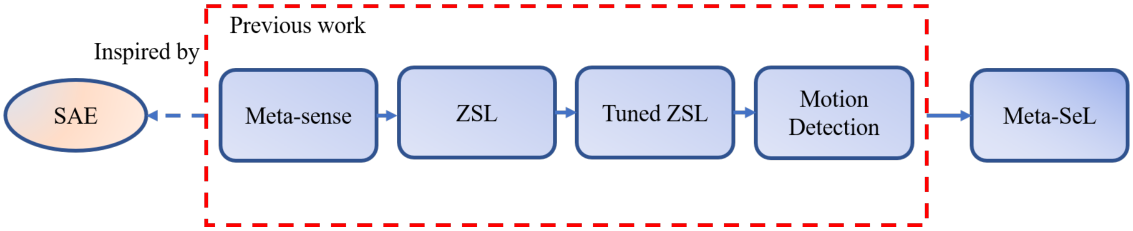


Figure 6.1: The progress of the work from Meta-sense to Meta-Semantic Learning (Meta-SeL)

◇ Demonstrate how Meta-Sel has a large positive impact on the point cloud, particularly shapeNet Core dataset, classification performance.

◇ Demonstrate that Meta-SeL is resistant to jittering noise and translation.

◇ Illustrate that Meta-SeL generates semantic information for each entry on the fly.

6.3 Related work

Researchers investigated 3D point clouds models to understand and analyze them using handy-crafted features [20], [178]. These features are extracted and categorized into two parts: local features and global features. Researchers feed these features into traditional machine learning algorithms to obtain results. However, these features are more biased and may not work for all 3D model shapes. To address this challenge, scientists use deep learning with PointNet [165] in 2017 and further PointNet++ [166]. So, deep learning has been widely used in 3D model classification with the base of PointNet such as DensePoint [119]. We discuss the 3D dataset and the difference between deep learning algorithms such as PointNet and DGCNN.

6.3.1 3D point cloud data:

In this Chapter, we use 3D point cloud models to discover a new application of Meta-learning. Figure 6.2 and 6.3 illustrate data format. The figures show that our input data appeared in 3D including the x,y, and z-axis. Originally, we applied Meta-learning to images and signal processing data that are 2D data. Having the third axis provide more challenge to learn, yet gives us more information about some object.

3D point cloud model classification highly depends on a priori knowledge about associated objects. Thus, it is recommended to have a rich part-segmented and labeled dataset. There are a few

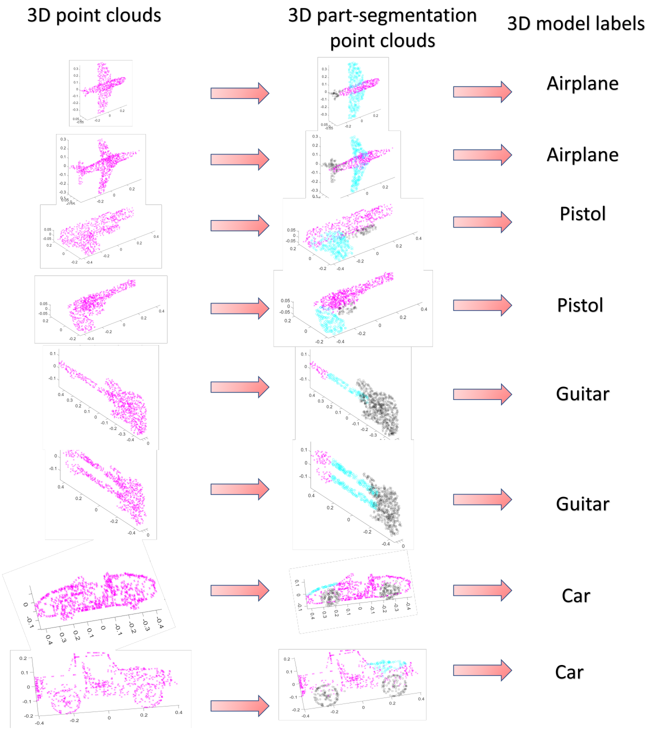


Figure 6.2: The process of segmentation and 3D model classification

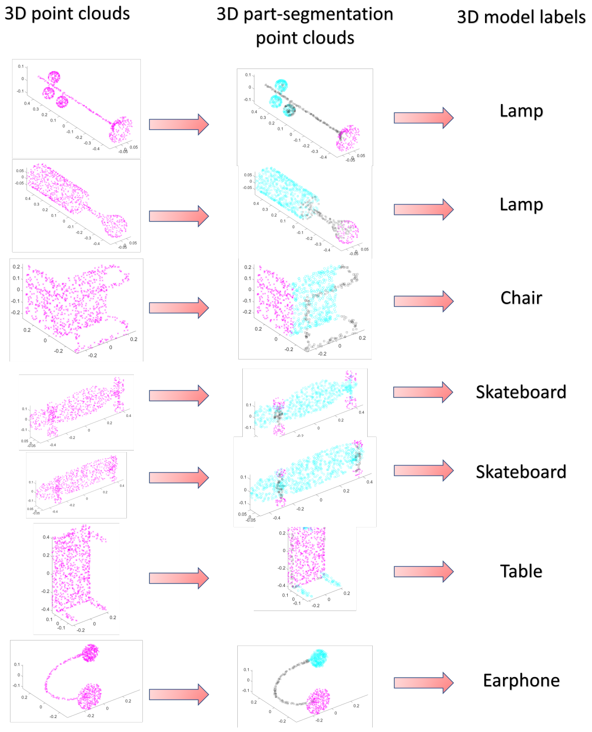


Figure 6.3: The process of segmentation and 3D model classification

publicly available datasets and are expected to have many in the future due to advanced technologies. In this study, ShapeNetCore data set is selected. This is a subset of ShapeNet dataset that has rich 3D point cloud models, which collected by the Intel RealSense ZR300 (Intel Corporation, Santa Clara, CA, USA) device.

The subset of the ShapeNetCore dataset is created to propagate manually verified labels from a large dataset to a small dataset [227], which includes 17,775 models from 16 categories. Each model is annotated in two to six parts. The dataset is initially divided into 15990 and 1785 models, respectively, for training and testing. In each model, we have 1024 points and generate label for points.

6.3.2 Approaches:

PointNet [165] was the first deep neural network algorithm to run on raw point clouds at the network's input. PointNet uses a multi-layer perceptron (mlp) [173] to extract features from point sets, as well as max-pooling layers to eliminate the permutation problem in point clouds. PointNet has revolutionized the process of classifying and segmenting 3D point cloud datasets [165]. There are three appealing properties of this approach: unordered input, local and global feature extraction, and invariance under transformations. Through a local network, i.e. T-net, spatial information has been transformed into local and global features. In contrast to PointNet, DGCNN [210] incorporates structural information by computing the pairwise distance using k-nearest neighbors(KNN) of points.

6.3.3 Point-based methods:

Handling 3D point cloud datasets using deep neural networks(DNN) has evolved recently [66], [101], [118]. One of the first DNN solutions presented by Point-Net [213] is that features are updated for each point using MLP layers and aggregated using global max pooling. PointNet, on the other hand, does not do local comparisons, which is why PointNet++ [167] exists. PointNet++ divides subsets of points into local areas, which are the first point to be analyzed. More modern methods are proposed for more explicit modeling of spatial relationships between points [120], [213].

6.3.4 Projection-based methods

Researchers in [175] proposed an approach to learn a model to predict viewing angles and categorise photos in a way that is end-to-end distinguishable. As a backbone, they use the ResNet50 model, which has been pre-trained on ImageNet. Further, researchers in [180] proposed a unique multi-height rendering and feature merging approach, as well as a larger backbone network that has been pre-trained on ImageNet. Addtionally, another work on projection-based methods [5] in which to train an aggregation of PointNet++ and CNN,they manually defined important views for each object category, generated binary edge maps, and directly defined important views for each object category.

There is a substantial list of work on the application of projection-based approaches to various point-cloud analytic challenges, such as segmentation [42], [198], rendering [11] and reconstruction [163]. Researcher in [27] use point cloud density to construct scene models, which are then fed into a mesh renderer to generate a variety of image perspectives at various scales. Furthermore, a research work in [110] renders a scene point cloud from 120 images for several modalities such as color, depth, and surface normal. The data from several modalities is then combined to produce point-wise predictions. We advise readers to read [67]'s new survey study for a full examination of several projection algorithms on various point-cloud processing assignments.

In addition to these research areas, for 3D model analysis, many research studies use pictures produced from object meshes [70], [192], [228].

6.3.5 Meta-learning:

Meta-learning has three main promises to classify objects and images. One of which is zero shot-learning [145], in which we learn from seen classes to predict unseen classes. Meta-sense [144], image recognition [148], and human motion recognition [149] are a few examples to present the process of generating a projection function from a vector space to a semantic space. These research studies are inspired by semantic auto-encoder (SAE) [99].

To work with 3D data, early deep learning approaches for functioning on 3D point clouds used multi-view [196] representations. The tendency in this area has recently switched to directly employing raw point clouds without any preprocessing [209]. These methods do not have the same scaling limitations as the volumetric representation, and unlike view-based methods, they do not make any a priori assumptions about which 2D planes the point cloud should be projected on or how many.

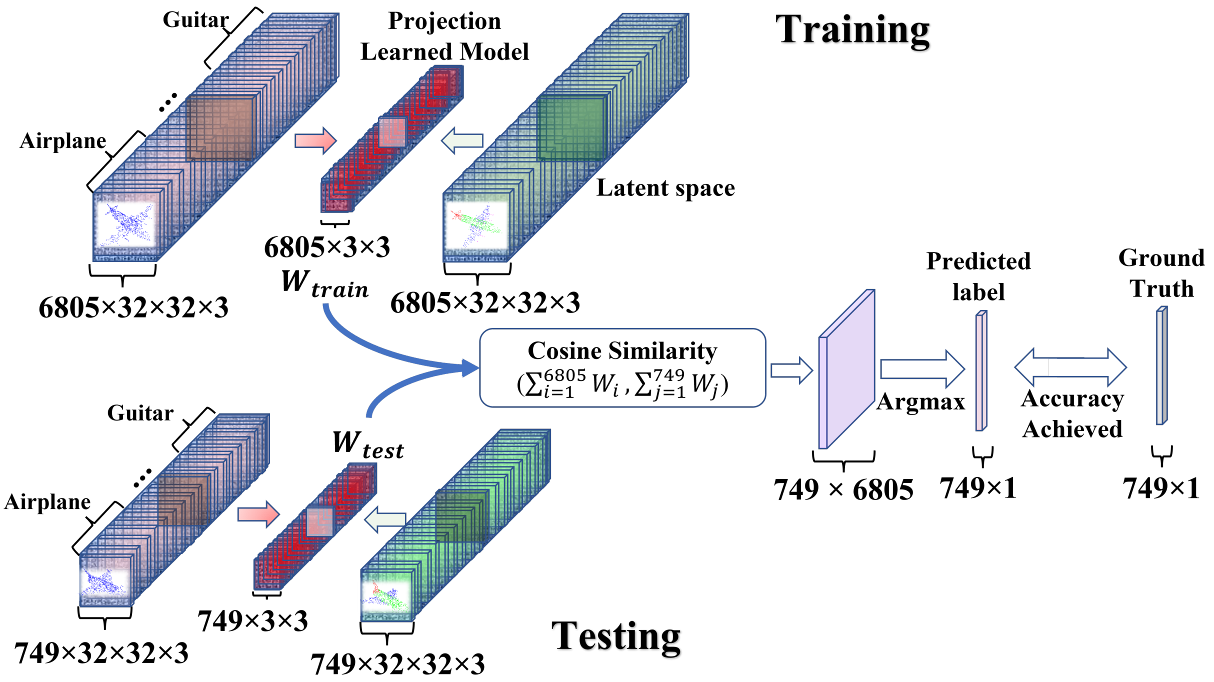


Figure 6.4: Meta-SeL framework for 3D point cloud models classification

Recent deep learning architectures are fairly capable at recognizing instances of 3D point cloud objects belonging to previously encountered classes. Simultaneously, current 3D depth camera technology enables for the generation/segmentation of a large number of 3D point cloud objects from any image, even if no previous training data exists. The classification of items from new, unexplored classes poses a difficulty for a 3D point cloud identification system. Similar to the 2D image version of the problem, this issue can be overcome by using a zero-shot learning (ZSL) strategy for 3D data. ZSL compares semantic information (attribute/word vector) of seen and unseen classes to classify unseen objects. Researchers in [38] stated that the work is the first research work to classify unseen 3D point could models, they adapted various contemporary 3D point cloud identification methods to the ZSL context with some architecture adjustments.

6.4 Proposed method

Since the number of 3D point cloud models classification algorithms has increased, an efficient method to perform this task must be developed. To that end, we present an optimized framework leveraging part-segmentation label information to classify 3D points cloud models. Figure 6.4 presents a comprehensive framework of Meta Semantic Learning (also known as Meta-SeL). We start with point cloud and elaborate on Meta-SeL. Meta-SeL is an optimized version from zero shot learning to few shot learning that works for 3D model classification.

6.4.1 Point cloud:

To leverage prior information on both classification and segmentation labels, we select the models with three segmentation labels. As shown in Table 6.3, the total number of models has reduced to 7,555 with 10 categories. Note that each filtered model contains three parts of segmentation labels.

To train the network, PointNet and PointNet++ use a fixed set of 1024 points per object. The fixed points strategy is what we call it. During the training process, RSCNN [120] and PointCNN [114] sample points at random during each phase, effectively exposing the model to more than 1024 points per object. However, in order to match all research studies, we use 1024 points for each each object in the 3D model.

6.4.2 Pre-processing:

Before starting Meta-SeL process, we have to make sure all input 3D could point models include three-part segmentation labels. Each model has a dimension of [32 * 32 * 3] and associated part-segmentation labels which stand for 1,2, or 3. We take advantage of the three-dimensionality of models and present a framework to achieve promising results for model classification. Thus, we maintain models that have a number of part segmentation labels equal to three. Next, we convert the input models (models and segmentation labels) into a new dimension 1024*3 models. Finally, we sort segmentation labels in ascending order and update the main models accordingly to have a consistent order of part-segmentation labels for all input 3D point cloud models. Note that we assume that every single point cloud in the 3D point clouds models is independent.

6.4.3 Latent space (semantic space):

Latent space represent a compressed version of input data in a way that helps decoders to generate an output and predict a label to a given sample instance. In prior research studies, scientists used a static semantic space similar to [99] or leveraged auto-encoders to generate the semantic space using deep neural network. But they made a static general semantic space for all input data.

Unlike other semantically based algorithms, Meta-SeL performs a classification task with dynamic semantic space for each 3D model separately. Using consistent ordering of the associated part-segmentation labels, Meta-SeL generates a semantic space for each new 3D model instantaneously. Each column in the semantic space matrix(S) represents the semantic space for cloud points. We use this dimension [32*32*3] consistent with the input 3D model dimensions and size.

6.4.4 Projection learned model:

Projection Learned Model(PLM) plays a significant role in Meta-SeL during the process of understanding input 3D point cloud models.

We generate PLM Ws in both phases: training W_{train} and testing W_{test} separately in real-time while executing each step. We are inspired by the idea of the semantic auto-encoder [99], [148], [149] in which researchers applied the framework of an auto-encoder to generate a projection function (W) to map input vectors to a latent space. To generate the W, we use only the encoder part of the semantic auto-encoder to generate a project function for each model. The encoder uses the Sylvester equation to find W for each input 3D model. Figure 6.5 presents a conceptual view of the projection learned model generation where the Sylvester equation takes input data(X,S andλ) to generate projection learned models (Ws) for both train and test. Thus, the output of the function is Ws with associated labels. Then, we calculate an average weight (W') of each category to present distinct distributions of categories. It is obvious that Meta-SeL generates complete separate Ws that enables us to predict test 3D models accurately and precisely. We have three dimensions that are shown in figure 6.5. First, On the x-axis we have the 3D model categories. Second, we have associated average weights (W's), which are flattened into 1*9 on z-axis. Finally, on Y-axis we have scalar values for the W's.

After the training phase is successfully completed, we execute the testing phase and evaluate the Meta-SeL results using a cosine similarity metric. We use this metric to seek the closest model to the test model. The cosine similarity metric seeks the angles between two vectors rather than their

Calculating a W per each 3D point cloud model

$$\underbrace{\begin{bmatrix} 0.4 & -0.3 & 0.1 \\ 0.1 & \cdots & -0.2 \\ \vdots & & \vdots \\ \vdots & .. & \vdots \\ 0.7 & & -0.1 \\ -0.2 & \cdots & 0.5 \end{bmatrix}}_{1024\ \times 3}^{\text{X}} \times \underbrace{\begin{bmatrix} 0.3 & 0.01 & -0.1 \\ 0.12 & 0.2 & 0.01 \\ -0.1 & 0.01 & 0.04 \end{bmatrix}}_{3\ \times 3}^{W} = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ \vdots & 2 & \vdots \\ \vdots & 2 & \vdots \\ 0 & 0 & 3 \\ 0 & 0 & 3 \end{bmatrix}}_{1024\ \times 3}^{\text{S}}$$

X: Input 3D Point Cloud Models
S: Semantic Space
W: Projection Learned Model
$W=\{w_{i,j} \in \mathbb{R} | i, j = [0:2],\ -1 < w_{i,j} < 1\}$

$$\Rightarrow \quad \text{X} \cdot W = \text{S}$$

$$\Downarrow$$

$$SS^T \cdot W + W \cdot (\lambda \times XX^T) = (1 + \lambda)\, SX^T$$

Average Weight(W') of total Ws:

$W'[11, 3, 3] = [Average(W_{(Airplan)}[471{,}3{,}3]), \ldots, Average(W_{(Table)}[1349{,}3{,}3])]$

$$W'[11, 3, 3] = W'[11, 9] \Rightarrow$$

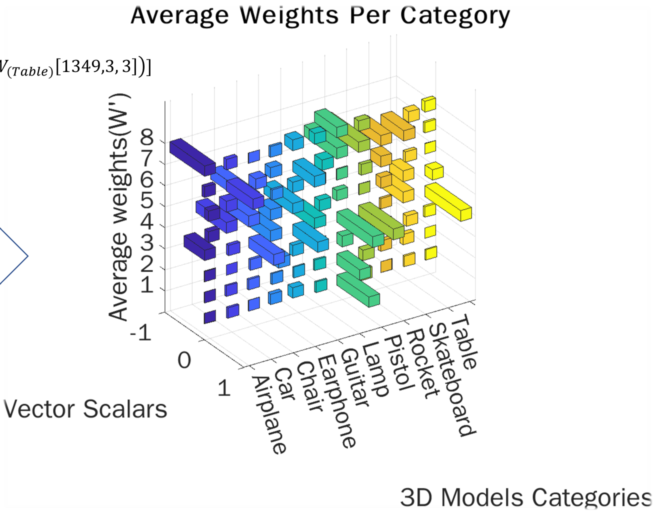


Figure 6.5: Averages Weights (Ws)

Figure 6.6: Presents the process to generate Ws, together with average weight W' per each category.

weights. Thus, if two models are far apart from each other, this metric can still determine if they are similar or not using equation 6.1.

$$\cos(\mathrm{W}_{\text{train}}, \mathrm{W}_{\text{test}}) = \frac{\mathrm{W}_{\text{train}}\mathrm{W}_{\text{test}}}{||\mathrm{W}_{\text{train}}||\,||\mathrm{W}_{\text{test}}||} = \frac{\sum_{i=1}^{n} \mathrm{W}_{\text{train}i} \sum_{i=1}^{n} \mathrm{W}_{\text{test}i}}{\sqrt{\sum_{i=1}^{n} (\mathrm{W}_{\text{train}i})^2}\sqrt{\sum_{i=1}^{n} (\mathrm{W}_{\text{test}i})^2}} \tag{6.1}$$

6.4.5 Accuracy computation:

We use an arg-max function to predict the test 3D model labels using a calculated cosine-similarity matrix and compare them with the ground truth to compute the accuracy of Meta-SeL. Meta-SeL computes the final result with only one *epoch* and does not require further iterations, as its accuracy does not change. Figure 6.7 presents Meta-SeL evaluation performance with DGCNN and PointNet on different types of customized and augmented datasets. We observe that the accuracy of Meta-SeL in the beginning reaches the highest rate and remains there.

Table 6.2: Results comparison with other state of the art

Methods	Accuracy(%)
DGCNN:	99.83
PointNet:	88.22
Meta-SeL:	
Base	93.19
Random Shuffle (RS)	93.19
Sorted (Ascending)	93.19
Sorted (Descending)	93.19
$Normalization(N)$	**95.59**
$Normalization$+ Random angle Rotation (x-axis)	90.25
$Normalization$+ Random angle Rotation (y-axis)	93.59
$Normalization$+Random angle Rotation (z-axis)	83.97
$Normalization$+Random angle Random Rotation (x, y, z -axis)	88.38
$Normalization + Translation$	**95.99**
$Normalization + RR + Translation$	77.03
$Normalization + Jittering$	**95.95**
$Normalization + Jittering + Translation$	**95.46**

6.5 Experimental results

We evaluate Meta-SeL performance on a subset of the ShapeNet Core dataset. Proving that Meta-SeL yields promising results. Meta-SeL results are competitive with respect to other state-of-the-art

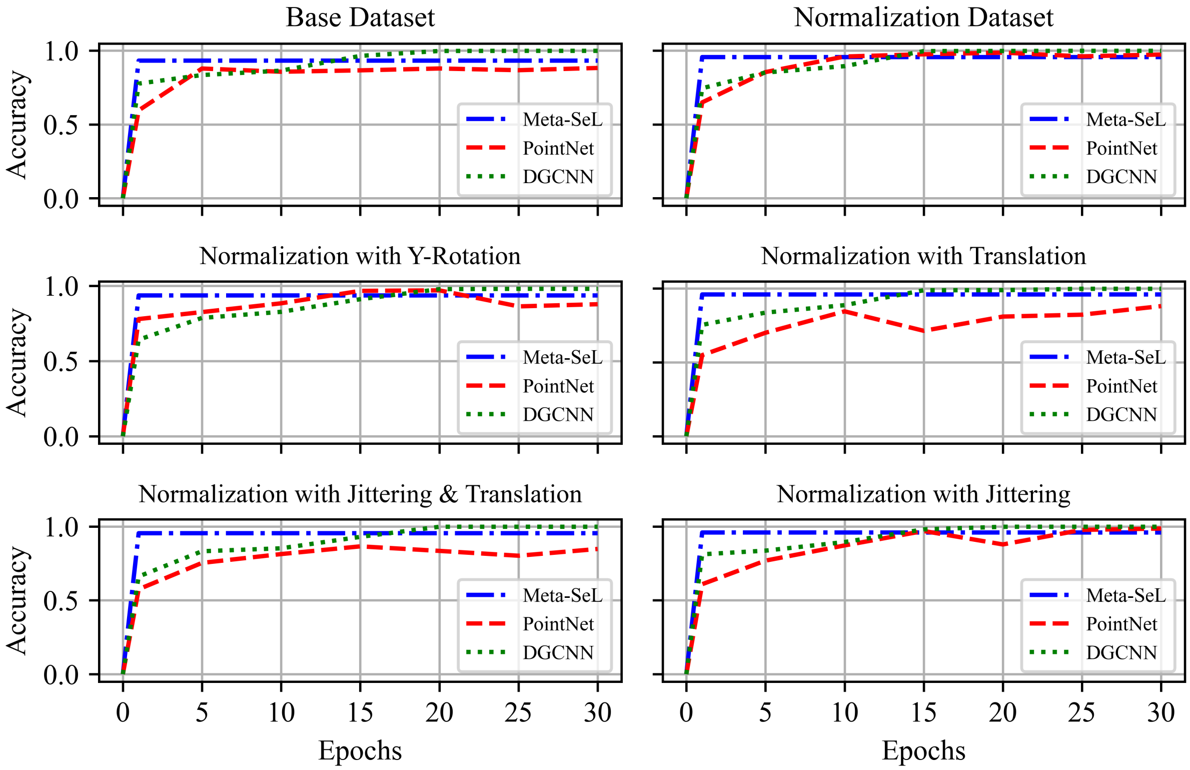


Figure 6.7: Accuracy Comparison

Table 6.3: Presents all class distribution during training and testing phase, together with *recall* and *precision*

Measurements **Categories**	Training	Testing	Recall(%)	Precision(%)
Airplane:	471	51	94.11	96
Car:	257	31	93.33	96.55
Chair:	2508	281	99.64	98.93
Earphone:	31	2	50	25
Guitar:	706	79	98.73	98.73
Lamp:	1086	115	91.30	92.10
Pistol:	244	28	100	90.32
Rocket:	51	6	100	66.66
Skateboard:	102	11	36.66	66.66
Table:	1349	145	96.55	97.90
Total / Weighted Average:	6806	749	95.99	96.12

Table 6.4: Meta-SeL's Confusion matrix for normalized dataset

Categories	Airplane	Car	Chair	Earphone	Guitar	Lamp	Pistol	Rocket	Skateboard	Table
Airplane:	48	0	0	0	0	1	0	0	0	2
Car:	1	28	1	0	1	0	0	0	0	0
Chair:	0	0	280	0	0	0	0	0	0	1
Earphone:	0	1	0	1	0	0	0	0	0	0
Guitar:	0	0	1	0	78	0	0	0	0	0
Lamp:	6	0	0	2	0	105	0	0	2	0
Pistol:	0	0	0	0	0	0	28	0	0	0
Rocket:	0	0	0	1	0	1	0	4	0	0
Skateboard:	0	0	0	0	0	7	0	0	4	0
Table:	1	0	1	0	0	0	3	0	0	140

methods. We apply different filters to the dataset and elaborate on them. The experimental results show that Meta-SeL becomes resistant to normalization, random angle random rotations, translation, and added jittering noise.

6.5.1 Evaluation metrics

We evaluate the performance of Meta-SeL using three important measurements. Table 6.5 illustrates these three measurements: Accuracy, recall and precision. In table 6.3 Meta-SeL depicts that *Pistol* and *Rocket* classes have 100% recall and *Earphone* has the minimum recall among other classes as it only has two instances that would be ignored in long round. A

Table 6.5: Evaluation criteria

Assessment Criteria	Formula
Accuracy (ACC)	$\frac{TP+TN}{TP+FP+TN+FN}$
Specificity	$\frac{TN}{TN+FP}$
Recall or sensitivity	$\frac{TP}{TP+FN}$

6.6 Data augmentation

In this section, we discuss different setup experiments that evaluate the performance of the Meta-SeL. To that end, we various data augmentation strategies such as normalization, random rotation, jittering, and translation.

6.6.1 Random shuffle-invariant, Sorted-invariant:

We evaluate the performance of Meta-SeL on different orders of labels. We obtain the same result with different orders of labels including random shuffle, ascending sort, and descending sort. It proves that Meta-SeL is highly invariant to the order of input data. Additionally, it does not have bias [185] towards classes with more instances over the ones with just a few instances.

6.6.2 Normalization:

We normalize the base dataset into a unit sphere using the following steps [165]. First, we calculate the center by computing the mean of point clouds for each model. Then, we find the point which has the maximum distance from this center. Finally, we divide all points over this calculated maximum distance. By normalizing each model, the data is located between -1 and +1.

6.6.3 Rotation for 3D shapes:

We further evaluate Meta-SeL on a different angle with different rotations including the x-axis, y-axis, and z-axis. The result (accuracy= 88.38%) shows that we still have a better result than DGCNN. We obtain an accuracy of approximately five percent less than the base accuracy. It is noteworthy that we check which axis rotation has affected adversely to the accuracy.

We test Meta-SeL with random angle rotation on x-axis. We randomly choose angles to rotate each model in train and test datasets. The result for *N+RR(x-axis)* in Table 6.2 shows that the accuracy has decreased by approximately three percent in comparison with base accuracy. We further test Meta-SeL with random angle rotation on the y-axis. The result of this rotation states that the y-axis has the highest accuracy in comparison with other axis rotations. Additionally, we rotate the datasets using different angle rotations on the z-axis. The result shows that 3D models have the worst accuracy for z-axis rotation mode.

6.6.4 Translation and rotation for 3D shapes:

Further, we test Meta-SeL to check if it predicts as high as possible if translation happened to each 3D model or not. We realize that a combination of normalization and translation has the highest accuracy. However, we test Meta-SeL on a combination of random angle random rotation and a translation. The result looks surprisingly expected due to the z-axis rotation and adding translation decreases its accuracy.

6.6.5 Jittering for 3D shapes:

We add jittering to data to evaluate the performance of the Meta-SeL. Jittering is the process of adding random jittering noise to data to prevent overplotting in 3D models. Overplotting occurs when some 3D point clouds in the models overlap into a single point. Jittering can alleviate the problem and help Meta-SeL to generate an efficient projection learned model. Although jittering enables us to obtain a better result, adding translation due to jittering noise data decreases the accuracy. The final accuracy for this combination of jittering and translation is superior to the base result.

6.7 Result analysis :

In this study, we have available data of 3D point cloud and associated 3D part-segmentation labels for each pixel. Figures 6.2 and 6.3 provide us with deep understanding of 3D model labels and part segmentation labels. It is obvious that all part segmentation labels for each model follow the same trend. It means that the relation among part-segmentation labels remains identical. Here, we discuss the reasons why one sample is falsely predicted for some other classes. Each adjacent graph presented in this section provides a combination of three labels for each node on the right

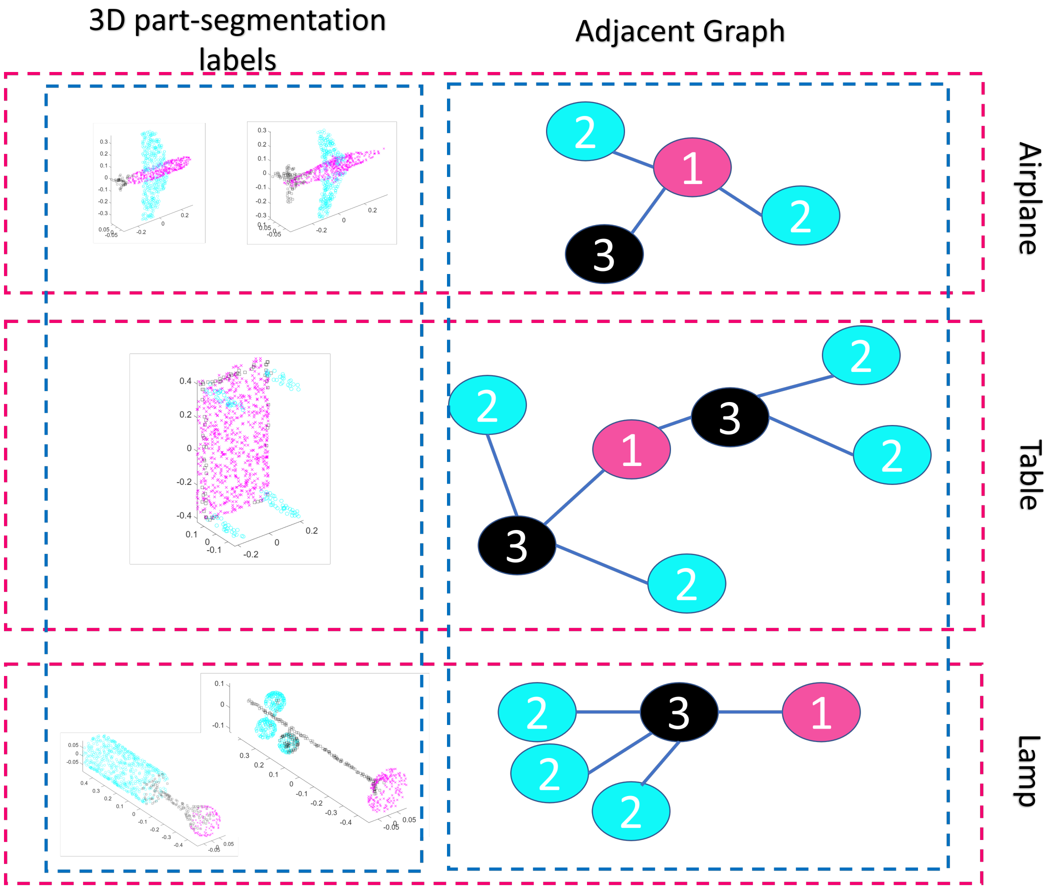


Figure 6.8: The similarity of *airplane* with *Lamp* and *Table* samples

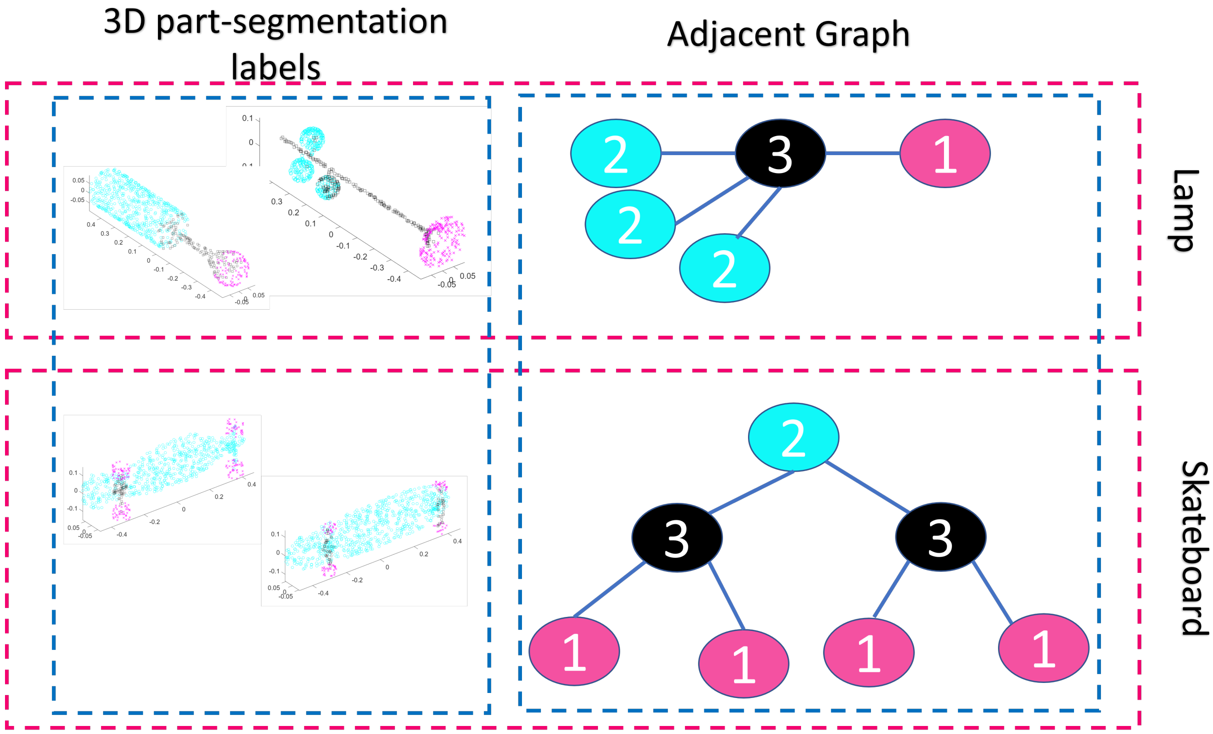


Figure 6.9: The similarity of *Lamp* with *Lamp* and *Skateboard* samples

side of figures like figure 6.8. Light blue, black and pink are three labels that match the 3D part-segmentation labels that appeared on the left side of the figures.

6.7.1 Confusion matrix:

We compute a confusion matrix as shown in table 6.4 for Meta-SeL evaluation on the Normalization dataset to tabulate how the work obtains competitive results. We show which category has the highest and lowest result. Table 6.4,6.3 depicts that *Chair* and *Pistol* have the highest predicted statistic and recall. We notice that since *Chair* and *Table* might look similar to each other, one *Chair's* wrongly predicted sample goes to *Table* and one of the wrongly predicted *table* instances goes to *Chair*. Furthermore, *Airplane* wrongly labeled samples go to *Table* and one of the wrongly predicted *Table* instances goes to *Airplane*. Additionally, two of the wrongly predicted *Lamp* instances go to *Skateboard* and all wrongly predicted labels for *Skateboard* go to *Lamp*. We observe that there is a trend among predicted labels. Whether Meta-SeL predicted correctly or wrongly, it predicts precisely.

6.7.2 *Airplane*

Let's start with *airplane* class, figure 6.8 shows that *Airplane* has four part-segmentation lables where we have two labels of *2* and one label of *1* and *3*. Label *2* goes for the airplane wings. label *1* is associated with the body of airplane. Finally, label *3* also is associated with the tail of airplane. These three-part-segmentation labels and the relationship remain identical for all 3D models of airplane which this enable us to generate the semantic space out of this. Further in figure 6.8, we have class *Table* which also has two class of *3* and *2*. Additionally, we have a class of *Lamp* which has three part-segmentation labels and sometimes it has more than one label *2*. Figure 6.8 illustrates that *airplane* has similar adjacency part-segmentation label graph with *Table* and *Lamp* classes. So, it is now clear that why three samples of *Airplane* are falsely predicted as *Table* and *Lamp*. Additionally, six samples of *Lamp* also falsely presented as *Airplane*.

6.7.3 *Lamp*

Class of *Lamp* has different segmentation formats, but in general it has three part-segmentation labels. one of the formats is shown in figure 6.9 in which we have three labels of *2*s and one label for *1* and *3*. We also have a class of *Skateboard* in which we have a adjacent graph like a binary tree with the height of two. Unlike the class of *Lamp*, the class of *Skateboard* has an identical adjacency graph for all samples of this class. Figure 6.9 presents that *Lamp* part-segmentation labels are close to *Skateboard* ones and that's why two samples of *Lamp* are falsely predicted as *Skateboard*. Vice versa, seven samples of *Skateboard* also falsely classified as *Lamp*.

6.7.4 *Pistol*, *Chair* and *Guitar*

These three classes have provided the best prediction accuracy among others because they have the simplest part-segmentation labels. Each of them has three part-segmentation labels and the whole structure of them also look the same and the simplest one. The class of *Pistol* has the most straightforward structure that provides the best recall as well as *rocket*.

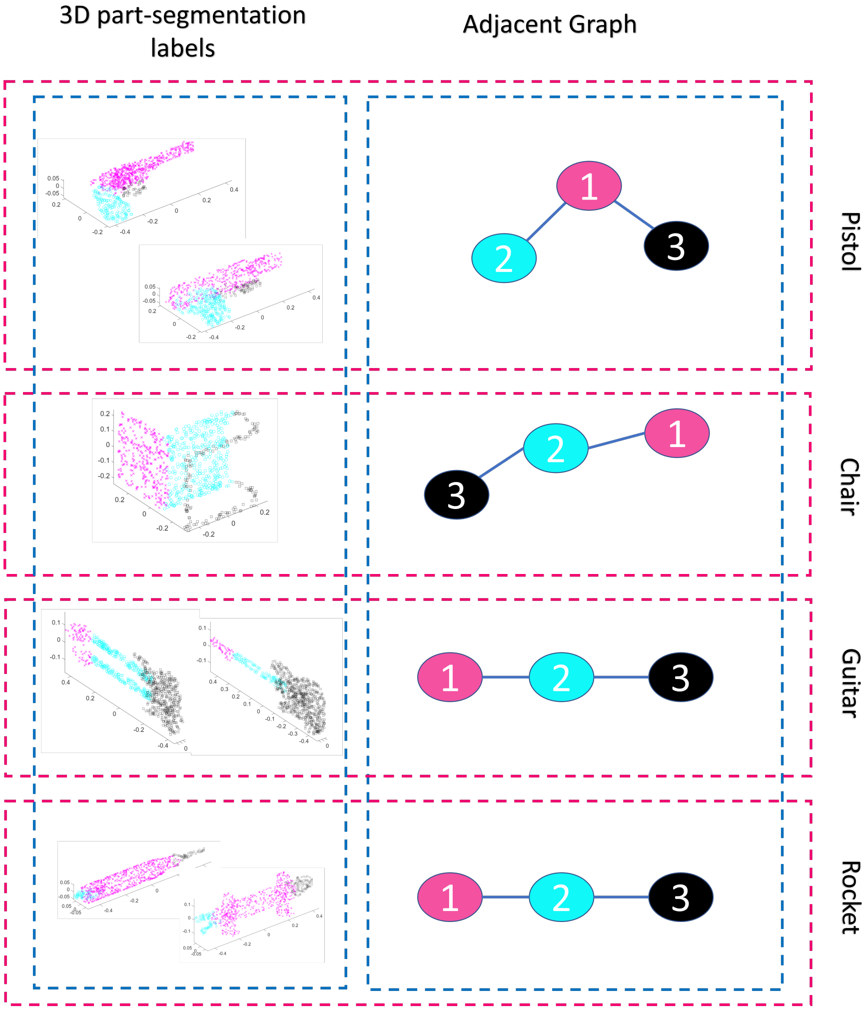


Figure 6.10: The similarity of *Pistol* with *Chair*, *Guita* and *Rocket* samples

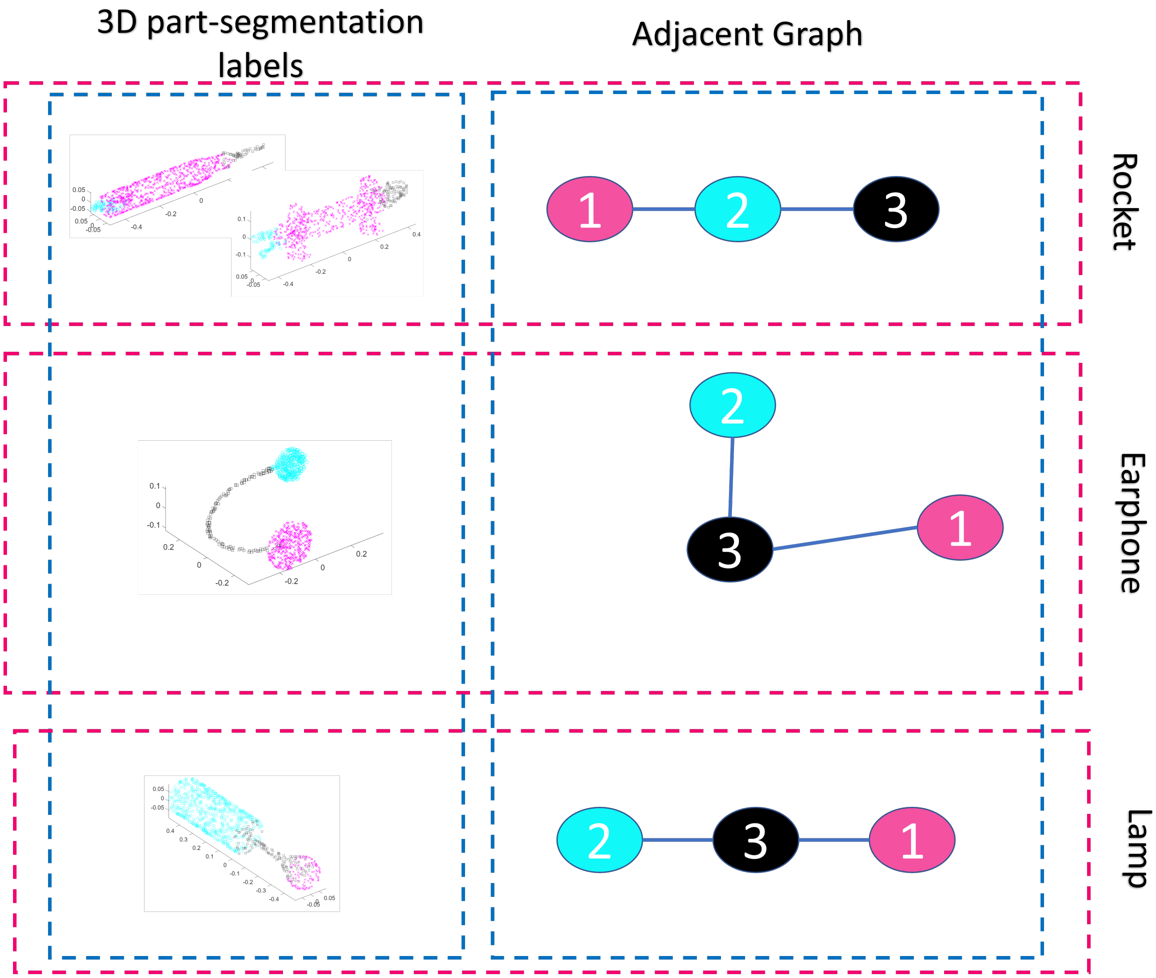


Figure 6.11: The similarity of *Rocket* with *Earphone* and *Lamp* samples

6.7.5 *Rocket*

The class of *Rocket* as well as the class of *Earphone* and *Lamp* has three part-segmentation labels. The structure and these three classes look simple, thus one of *Rocket*' s sample, two of *Lamp*'s samples are falsely predicted as *Earphone*. *Guitar*'s some sample adjacency graph matches exactly some *rocket*s samples. In addition to this, *Chair* and *Pistol* also offer simple three node graph, but in *Chair*'s graph we have label *2* as a root or center node and in *Pistol*'s graph we have label *1* as a root and center node. It is noteworthy that *Chair*, *Guitar* and *Rocket* has label *2* as their center and root node.

6.8 Conclusion

In this chapter, we presented Meta-SeL for 3D point cloud classification on the ShapeNet core dataset. We explore each model to generate semantic space during training time. We calculated the projection function from vector space which is 3D point clouds models to the semantic space for both the training and testing process. Further, we employed a cosine similarity function to find the top match projection function among training and testing models. Then, we used an arg-max function to predict labels per each 3D model in the testing process. Additionally, we modified the input dataset with different orders, augmented data, and rotations to check the performance of Meta-SeL. The results indicate that Meta-SeL obtains competitive accuracy with an efficient time and cost.

CHAPTER 7

CONCLUSION AND FUTURE DIRECTION

In this dissertation, we explored machine learning algorithms and addressed their limitations when machine learning fails to produce a promising result. Meta-learning (MTL), in particular zero-shot learning, is one of the alternate solutions we have proposed. In this dissertation, we focused on zero-shot learning and proposed several algorithms to overcome the limitation of a lack of training samples. Four solutions are illustrated in figure 7.1 : meta-learning for computer vision, meta-Sense for signal processing, applied meta-sense for healthcare, and Meta-SeL for classifying 3D point cloud models. These solutions have applications in computer vision, health care, and signal processing. This dissertation uses 3D point clouds, 2D images, and signal data. This chapter provides an overview of these solutions, applications, and data.

7.1 Solutions

In **Chapter 2**, we investigated Meta-Learning (MTL) to address the problem of insufficient samples. MTL provides three crucial solutions for new data and complex issues. Zero-shot learning (ZSL) is one of the effective meta-learning solutions when there are no samples from previously unseen classes and therefore insufficient data for training. In addition, in **Chapter 3**, we investigated image recognition for unseen categories using the ZSL. Using the zero-shot learning (ZSL) solution, a semantic auto-encoder was implemented (SAE). The effect of hyper-parameter tuning on semantic auto-encoder performance is investigated further.

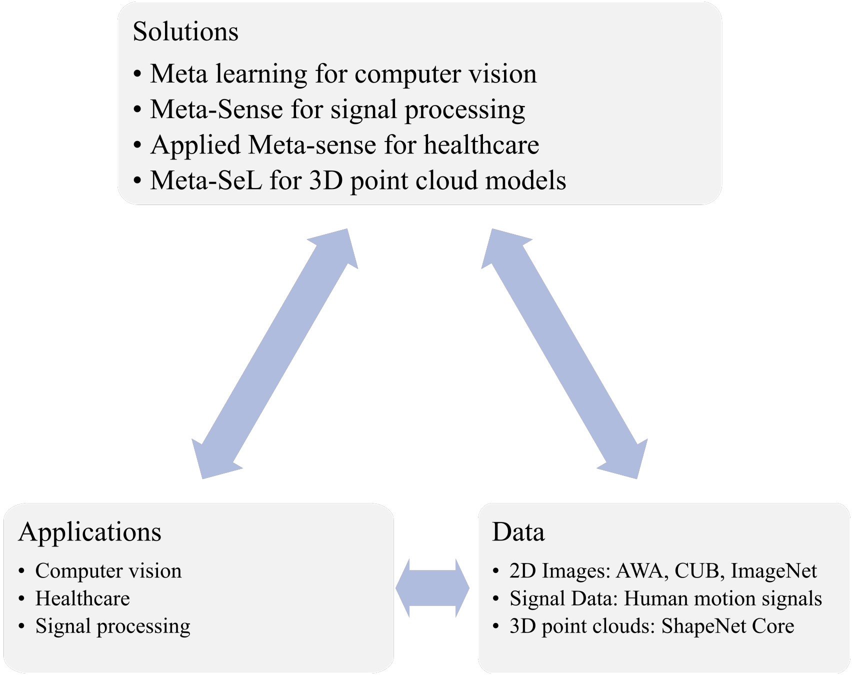


Figure 7.1: Solutions, dataset, and applications of this dissertation at a glance

In **Chapter 4**, we investigated how meta-learning solutions can be utilized to enable device-free sensing. As an alternative to adding more sensors, we proposed Meta-sensing, which is essentially learning to sense. Thus, in **Chapter 5**, we overcome the limitation of machine learning by identifying new classes in the field of human action recognition. In the absence of sufficient training data, we used zero-shot learning with Meta sense to recognize unseen motion.

In **chapter 6**, we proposed Meta-Semantic Learning (Meta-SeL), a classification method inspired by the zero-shot learning concept and applicable to all zero-shot learning algorithms. Meta-SeL is an integrated framework that provides a time- and cost-efficient solution for classifying 3D point cloud models using two input 3D local points.

Overall, we began with ZSL and concluded with the efficient classification algorithm Meta-semantic learning (Meta-SeL). We applied ZSL to unseen animal images using static semantic space to recognize animal labels. In addition, we introduce Meta-sense and apply it to the recognition of human motion. Finally, we presented Meta-SeL, an efficient algorithm for classifying 3D point cloud models.

7.2 Data

This dissertation included three types of data. The first dataset is comprised of three distinct 2D image datasets: AWA, CUB, and ImageNet. In zero-shot learning, they are utilized to identify unseen animals. The use of smartphone signal data for human motion detection in healthcare applications is another illustration. The final dataset is the ShapeNetCore dataset, which includes 3D point cloud models for 3D model classification using meta semantic learning.

7.3 Applications

There are three primary applications for this dissertation. The first significant application discussed in chapters 2 and 3 is computer vision. In addition, we investigated healthcare by detecting unseen human motion and activity with meta-sense. The objective is to monitor the activity of patients and record any health issues that may arise as a result of their motions and actions. Signal processing is the final application, which collects data from sensors embedded in smartphones and wearable devices.

7.4 Future directions

I'd like to provide open-ended instructions for prospective students who are eager to continue this work. I strongly suggest that students investigate the following areas:

◇ Analyzing various types of input data for Meta-SeL, including 2D images. They should expand their investigation and evaluate Meta-SeL on two-label segmented 2D images.

◇ Investigating a few-shot learning framework and generating meta-training and meta-testing datasets for the Meta-SeL evaluation.

◇ Using aggregation or another optimal projection function to combine the weighted models from each category into a single weighted model.

BIBLIOGRAPHY

[1] O. Abu Arqub, M. Al-Smadi, S. Momani, and T. Hayat, "Numerical solutions of fuzzy differential equations using reproducing kernel hilbert space method," *Soft Computing*, vol. 20, no. 8, pp. 3283–3302, 2016.

[2] A. Adeli and A. Broumandnia, "Image steganalysis using improved particle swarm optimization based feature selection," *Applied Intelligence*, pp. 1–14, 2018.

[3] F. Ahamed and F. Farid, "Applying internet of things and machine-learning for personalized healthcare: Issues and challenges," in *2018 International Conference on Machine Learning and Data Engineering (iCMLDE)*, IEEE, 2018, pp. 19–21.

[4] M. A. Ahmad, A. Patel, C. Eckert, V. Kumar, and A. Teredesai, "Fairness in machine learning for healthcare," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 3529–3530.

[5] S. M. Ahmed, P. Liang, and C. M. Chew, "Epn: Edge-aware pointnet for object recognition from multi-view 2.5 d point clouds," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2019, pp. 3445–3450.

[6] Z. Akata, S. Reed, D. Walter, H. Lee, and B. Schiele, "Evaluation of output embeddings for fine-grained image classification," 2015, pp. 2927–2936.

[7] U. Akhtar, A. M. Khattak, and S. Lee, "Challenges in managing real-time data in health information system (his)," in *International Conference on Smart Homes and Health Telematics*, Springer, 2016, pp. 305–313.

[8] M. Aledhari, R. Razzak, R. M. Parizi, and F. Saeed, "Federated learning: A survey on enabling technologies, protocols, and applications," *IEEE Access*, vol. 8, pp. 140 699–140 725, 2020.

[9] E. Alexiou, E. Upenik, and T. Ebrahimi, “Towards subjective quality assessment of point cloud imaging in augmented reality,” in *2017 IEEE 19th International Workshop on Multimedia Signal Processing (MMSP)*, IEEE, 2017, pp. 1–6.

[10] F. Ali, S. El-Sappagh, S. R. Islam, *et al.*, “An intelligent healthcare monitoring framework using wearable sensors and social networking data,” *Future Generation Computer Systems*, vol. 114, pp. 23–43, 2021.

[11] K.-A. Aliev, A. Sevastopolsky, M. Kolos, D. Ulyanov, and V. Lempitsky, “Neural point-based graphics,” in *European Conference on Computer Vision*, Springer, 2020, pp. 696–712.

[12] M. H. Amini, J. Mohammadi, and S. Kar, “Distributed holistic framework for smart city infrastructures: Tale of interdependent electrified transportation network and power grid,” *IEEE Access*, vol. 7, pp. 157 535–157 554, 2019.

[13] S. Amirian, Z. Wang, T. R. Taha, and H. R. Arabnia, “Dissection of deep learning with applications in image recognition,” in *Proceedings of International Conference on Computational Science and Computational Intelligence (CSCI 2018: December 2018, USA); "Artificial Intelligence" Research Track (CSCI-ISAI)*, 2018, pp. 1132–1138.

[14] S. Amirian, K. Rasheed, T. R. Taha, and H. R. Arabnia, “Image captioning with generative adversarial network,” in *2019 International Conference on Computational Science and Computational Intelligence (CSCI)*, IEEE, 2019, pp. 272–275.

[15] J. Andreas, M. Rohrbach, T. Darrell, and D. Klein, “Neural module networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 39–48.

[16] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, “Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine,” in *International workshop on ambient assisted living*, Springer, 2012, pp. 216–223.

[17] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, “A public domain dataset for human activity recognition using smartphones.,” in *Esann*, 2013.

[18] ——, “Energy efficient smartphone-based activity recognition using fixed-point arithmetic.,” *J. UCS*, vol. 19, no. 9, pp. 1295–1314, 2013.

[19] A. Antoniou, H. Edwards, and A. Storkey, “How to train your maml,” *arXiv preprint arXiv:1810.09502*, 2018.

[20] M. Aubry, U. Schlickewei, and D. Cremers, "The wave kernel signature: A quantum mechanical approach to shape analysis," in *2011 IEEE international conference on computer vision workshops (ICCV workshops)*, IEEE, 2011, pp. 1626–1633.

[21] J. M. Banda, M. Seneviratne, T. Hernandez-Boussard, and N. H. Shah, "Advances in electronic phenotyping: From rule-based definitions to machine learning models," *Annual review of biomedical data science*, vol. 1, pp. 53–68, 2018.

[22] F. Baradel, C. Wolf, J. Mille, and G. W. Taylor, "Glimpse clouds: Human activity recognition from unstructured feature points," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 469–478.

[23] Y. Bengio, S. Bengio, and J. Cloutier, *Learning a synaptic learning rule*. Université de Montréal, Département d'informatique et de recherche . . ., 1990.

[24] Y. Bengio, Y. LeCun, *et al.*, "Scaling learning algorithms towards ai," *Large-scale kernel machines*, vol. 34, no. 5, pp. 1–41, 2007.

[25] Y. Bengio, Y. Lecun, and G. Hinton, "Deep learning for ai," *Communications of the ACM*, vol. 64, no. 7, pp. 58–65, 2021.

[26] M. Boroumand, M. Chen, and J. Fridrich, "Deep residual network for steganalysis of digital images," *IEEE Transactions on Information Forensics and Security*, 2019.

[27] A. Boulch, B. Le Saux, and N. Audebert, "Unstructured point cloud semantic labeling using deep segmentation networks.," *3DOR@ Eurographics*, vol. 3, 2017.

[28] M. Bucher, S. Herbin, and F. Jurie, "Improving semantic embedding consistency by metric learning for zero-shot classiffication," in *European Conference on Computer Vision*, Springer, 2016, pp. 730–746.

[29] N. Cesa-Bianchi and G. Lugosi, *Prediction, learning, and games*. Cambridge university press, 2006.

[30] S. S. Chaeikar and A. Ahmadi, "Ensemble sw image steganalysis: A low dimension method for lsbr detection," *Signal Processing: Image Communication*, vol. 70, pp. 233–245, 2019.

[31] S. Changpinyo, W.-L. Chao, B. Gong, and F. Sha, "Synthesized classifiers for zero-shot learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 5327–5336.

[32] C. Chen and Y. Q. Shi, "Jpeg image steganalysis utilizing both intrablock and interblock correlations," in *2008 IEEE International Symposium on Circuits and Systems*, IEEE, 2008, pp. 3029–3032.

[33] L. Chen, H. Zhang, J. Xiao, W. Liu, and S.-F. Chang, "Zero-shot visual recognition using semantics-preserving adversarial embedding networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 1043–1052.

[34] M. Chen, U. Challita, W. Saad, C. Yin, and M. Debbah, "Artificial neural networks-based machine learning for wireless networks: A tutorial," *IEEE Communications Surveys & Tutorials*, 2019.

[35] Y. Chen, X. Qin, J. Wang, C. Yu, and W. Gao, "Fedhealth: A federated transfer learning framework for wearable healthcare," *IEEE Intelligent Systems*, vol. 35, no. 4, pp. 83–93, 2020.

[36] Y. Chen, A. L. Friesen, F. Behbahani, *et al.*, "Modular meta-learning with shrinkage," 2019.

[37] A. Cheraghian, S. Rahman, D. Campbell, and L. Petersson, "Transductive zero-shot learning for 3d point cloud classification," *arXiv preprint arXiv:1912.07161*, 2019.

[38] A. Cheraghian, S. Rahman, and L. Petersson, "Zero-shot learning of 3d point cloud objects," in *2019 16th International Conference on Machine Vision Applications (MVA)*, IEEE, 2019, pp. 1–6.

[39] R. R. Chhikara, P. Sharma, and L. Singh, "A hybrid feature selection approach based on improved pso and filter approaches for image steganalysis," *International Journal of Machine Learning and Cybernetics*, vol. 7, no. 6, pp. 1195–1206, 2016.

[40] ——, "An improved dynamic discrete firefly algorithm for blind image steganalysis," *International Journal of Machine Learning and Cybernetics*, vol. 9, no. 5, pp. 821–835, 2018.

[41] J. Choi, J. Lee, J. Park, and J. Nam, "Zero-shot learning and knowledge transfer in music classification and tagging," *arXiv preprint arXiv:1906.08615*, 2019.

[42] A. Dai and M. Nießner, "3dmv: Joint 3d-multi-view prediction for 3d semantic scene segmentation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 452–468.

[43] A. Ed-daoudy and K. Maalmi, “Application of machine learning model on streaming health data event in real-time to predict health status using spark,” in *2018 International Symposium on Advanced Electrical and Communication Technologies (ISAECT)*, IEEE, 2018, pp. 1–4.

[44] F. De Terán and F. Dopico, “Consistency and efficient solution of the sylvester equation for*-congruence,” *The Electronic Journal of Linear Algebra*, vol. 22, pp. 849–863, 2011.

[45] V. Delaitre, J. Sivic, and I. Laptev, “Learning person-object interactions for action recognition in still images,” *Advances in neural information processing systems*, vol. 24, 2011.

[46] W. Deng, X. Zeng, and Y. Hong, “Distributed computation for solving the sylvester equation based on optimization,” *IEEE Control Systems Letters*, vol. 4, no. 2, pp. 414–419, 2019.

[47] C. Desai and D. Ramanan, “Detecting actions, poses, and objects with relational phraselets,” in *European conference on computer vision*, Springer, 2012, pp. 158–172.

[48] Z. Ding, H. Zhao, and Y. Fu, “Zero-shot learning,” in *Learning Representation for Multi-View Data Analysis: Models and Applications*. Springer International Publishing, 2019, pp. 127–144, ISBN: 978-3-030-00734-8. DOI: `10.1007/978-3-030-00734-8_6`.

[49] H. Dong, T. Li, R. Ding, and J. Sun, “A novel hybrid genetic algorithm with granular information for feature selection and optimization,” *Applied Soft Computing*, vol. 65, pp. 33–46, 2018.

[50] H. Elayan, M. Aloqaily, and M. Guizani, “Deep federated learning for iot-based decentralized healthcare systems,” in *2021 International Wireless Communications and Mobile Computing (IWCMC)*, IEEE, 2021, pp. 105–109.

[51] A. Farahani, S. Voghoei, K. Rasheed, and H. R. Arabnia, “A brief review of domain adaptation,” *arXiv preprint arXiv:2010.03978*, 2020.

[52] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth, “Describing objects by their attributes,” in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 2009, pp. 1778–1785.

[53] R. Felix, I. Reid, G. Carneiro, *et al.*, “Multi-modal cycle-consistent generalized zero-shot learning,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 21–37.

[54] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, JMLR. org, 2017, pp. 1126–1135.

[55] C. Finn, A. Rajeswaran, S. Kakade, and S. Levine, "Online meta-learning," *arXiv preprint arXiv:1902.08438*, 2019.

[56] C. Finn, K. Xu, and S. Levine, "Probabilistic model-agnostic meta-learning," in *Advances in Neural Information Processing Systems*, 2018, pp. 9516–9527.

[57] C. Finn, T. Yu, T. Zhang, P. Abbeel, and S. Levine, "One-shot visual imitation learning via meta-learning," in *1st Conference on Robot Learning (CoRL)*, 2017.

[58] T. M. Fischer, "On the algorithm by al-mohy and higham for computing the action of the matrix exponential: A posteriori roundoff error estimation," *Linear Algebra and its Applications*, vol. 531, pp. 141–168, 2017.

[59] Ó. Fontenla-Romero, B. Guijarro-Berdiñas, D. Martinez-Rego, B. Pérez-Sánchez, and D. Peteiro-Barral, "Online machine learning," in *Efficiency and Scalability Methods for Computational Intellect*, IGI Global, 2013, pp. 27–54.

[60] J. Fridrich and J. Kodovsky, "Rich models for steganalysis of digital images," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 3, pp. 868–882, 2012.

[61] Y. Fu and L. Sigal, "Semi-supervised vocabulary-informed learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 5337–5346.

[62] Z. Fu, T. Xiang, E. Kodirov, and S. Gong, "Zero-shot object recognition by semantic manifold distance," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 2635–2644.

[63] M. R. Garey, R. L. Graham, D. S. Johnson, and D. E. Knuth, "Complexity results for bandwidth minimization," *SIAM Journal on Applied Mathematics*, vol. 34, no. 3, pp. 477–495, 1978.

[64] A. L. Georgiana Dinu and M. Baroni, "Improving zero-shot learning by mitigating the hubness problem," in *Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015)*, workshop track, 2015.

[65] J. J. Gibson, *The ecological approach to visual perception: classic edition*. Psychology Press, 2014.

[66] F. Groh, P. Wieschollek, and H. Lensch, "Flex-convolution (million-scale point-cloud learning beyond grid-worlds)," *arXiv preprint arXiv:1803.07289*, 2018.

[67] Y. Guo, H. Wang, Q. Hu, H. Liu, L. Liu, and M. Bennamoun, "Deep learning for 3d point clouds: A survey," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 12, pp. 4338–4364, 2020.

[68] A. Gupta, T. Stewart, N. Bhulani, *et al.*, "Feasibility of wearable physical activity monitors in patients with cancer," *JCO clinical cancer informatics*, vol. 2, pp. 1–10, 2018.

[69] J. B. Guttikonda and S. R., "A new steganalysis approach with an efficient feature selection and classification algorithms for identifying the stego images," *Multimedia Tools and Applications*, vol. 78, no. 15, pp. 21 113–21 131, Aug. 2019. DOI: `10.1007/s11042-019-7168-5`.

[70] T. Hackel, N. Savinov, L. Ladicky, J. D. Wegner, K. Schindler, and M. Pollefeys, "Semantic3d. net: A new large-scale point cloud classification benchmark," *arXiv preprint arXiv:1704.03847*, 2017.

[71] M. Haghi, K. Thurow, and R. Stoll, "Wearable devices in medical internet of things: Scientific research and commercially available devices," *Healthcare informatics research*, vol. 23, no. 1, pp. 4–15, 2017.

[72] Z. Al-Halah and R. Stiefelhagen, "How to transfer? zero-shot object recognition via hierarchical transfer of semantic attributes," in *2015 IEEE Winter Conference on Applications of Computer Vision*, IEEE, 2015, pp. 837–843.

[73] Z. Al-Halah, M. Tapaswi, and R. Stiefelhagen, "Recovering the missing link: Predicting class-attribute associations for unsupervised zero-shot learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 5975–5984.

[74] J. Hannan, "Approximation to bayes risk in repeated play," *Contributions to the Theory of Games*, vol. 3, pp. 97–139, 1957.

[75] F. Hassan, M. E. Shaheen, and R. Sahal, "Real-time healthcare monitoring system using online machine learning and spark streaming," *(IJACSA) International Journal of Advanced Computer Science and Applications*, vol. 11, no. 9, 2020.

[76] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[77] V. Holub and J. Fridrich, “Random projections of residuals for digital image steganalysis,” *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 12, pp. 1996–2006, 2013.

[78] V. Holub and J. Fridrich, “Phase-aware projection model for steganalysis of jpeg images,” in *Media Watermarking, Security, and Forensics 2015*, International Society for Optics and Photonics, vol. 9409, 2015, 94090T.

[79] C. Huang, Z. Cao, Y. Wang, J. Wang, and M. Long, “Metasets: Meta-learning on point sets for generalizable representations,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 8863–8872.

[80] B. Ifrach, C. Maglaras, M. Scarsini, and A. Zseleva, “Bayesian social learning from consumer reviews,” *Operations Research*, 2019.

[81] A. Imteaj, R. Alabagi, and M. H. Amini, “Exploiting federated learning technique to recognize human activities in resource-constrained environment,” in *International Conference on Intelligent Human Computer Interaction*, Springer, 2021, pp. 659–672.

[82] A. Imteaj, M. H. Amini, and J. Mohammadi, “Leveraging decentralized artificial intelligence to enhance resilience of energy networks,” *arXiv preprint arXiv:1911.07690*, 2019.

[83] A. Imteaj, U. Thakker, S. Wang, J. Li, and M. H. Amini, “Federated learning for resource-constrained iot devices: Panoramas and state-of-the-art,” *arXiv preprint arXiv:2002.10610*, 2020.

[84] ——, “A survey on federated learning for resource-constrained iot devices,” *IEEE Internet of Things Journal*, vol. 9, no. 1, pp. 1–24, 2021.

[85] A. Iranmehr, H. Masnadi-Shirazi, and N. Vasconcelos, “Cost-sensitive support vector machines,” *Neurocomputing*, vol. 343, pp. 50–64, 2019.

[86] A. Isazadeh, H. Izadkhah, and A. Mokarram, “A learning based evolutionary approach for minimization of matrix bandwidth problem,” *Appl. Math*, vol. 6, no. 1, pp. 51–57, 2012.

[87] M. Jain, J. C. Van Gemert, T. Mensink, and C. G. Snoek, “Objects2action: Classifying and localizing actions without any video example,” in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 4588–4596.

[88] D. Jayaraman and K. Grauman, “Zero-shot recognition with unreliable attributes,” *Advances in neural information processing systems*, vol. 27, 2014.

[89] Z. Ji, Y. Fu, J. Guo, Y. Pang, Z. M. Zhang, *et al.*, “Stacked semantics-guided attention model for fine-grained zero-shot learning,” *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[90] T. Jourdan, A. Boutet, A. Bahi, and C. Frindel, “Privacy-preserving iot framework for activity recognition in personal healthcare monitoring,” *ACM Transactions on Computing for Healthcare*, vol. 2, no. 1, pp. 1–22, 2020.

[91] V. Kalogeiton, P. Weinzaepfel, V. Ferrari, and C. Schmid, “Joint learning of object and action detectors,” in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 4163–4172.

[92] J. Kaur and S. Singh, “Feature selection using mutual information and adaptive particle swarm optimization for image steganalysis,” in *2018 7th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO)*, IEEE, 2018, pp. 538–544.

[93] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in *Proceedings of IEEE international conference on neural networks*, IEEE Press, vol. 4, 1995, pp. 1942–1948.

[94] L. U. Khan, W. Saad, Z. Han, E. Hossain, and C. S. Hong, “Federated learning for internet of things: Recent advances, taxonomy, and open challenges,” *IEEE Communications Surveys & Tutorials*, 2021.

[95] K. Kidono, T. Miyasaka, A. Watanabe, T. Naito, and J. Miura, “Pedestrian recognition using high-definition lidar,” in *2011 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2011, pp. 405–410.

[96] J. Kim, T. Kim, S. Kim, and C. D. Yoo, “Edge-labeling graph neural network for few-shot learning,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11–20.

[97] G. Koch, R. Zemel, and R. Salakhutdinov, “Siamese neural networks for one-shot image recognition,” in *ICML deep learning workshop*, vol. 2, 2015.

[98] E. Kodirov, T. Xiang, Z. Fu, and S. Gong, “Unsupervised domain adaptation for zero-shot learning,” in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2452–2460.

[99] E. Kodirov, T. Xiang, and S. Gong, “Semantic autoencoder for zero-shot learning,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3174–3183.

[100] J. Kodovsk and J. Fridrich, “Calibration revisited,” in *Proceedings of the 11th ACM workshop on Multimedia and security*, ACM, 2009, pp. 63–74.

[101] A. Komarichev, Z. Zhong, and J. Hua, “A-cnn: Annularly convolutional neural networks on point clouds,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7421–7430.

[102] J. Konečn, H. B. McMahan, D. Ramage, and P. Richtárik, “Federated optimization: Distributed machine learning for on-device intelligence,” *arXiv preprint arXiv:1610.02527*, 2016.

[103] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[104] A. Kumar, A. Levine, T. Goldstein, and S. Feizi, “Curse of dimensionality on randomized smoothing for certifiable robustness,” *arXiv preprint arXiv:2002.03239*, 2020.

[105] M. Kumari and R. Chhikara, “Blind image steganalysis using neural networks and wrapper feature selection,” in *2017 International Conference on Computing, Communication and Automation (ICCCA)*, IEEE, 2017, pp. 1065–1069.

[106] B. M. Lake, R. Salakhutdinov, and J. B. Tenenbaum, “Human-level concept learning through probabilistic program induction,” *Science*, vol. 350, no. 6266, pp. 1332–1338, 2015.

[107] C. H. Lampert, H. Nickisch, and S. Harmeling, “Attribute-based classification for zero-shot visual object categorization,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 3, pp. 453–465, 2013.

[108] H. Larochelle, D. Erhan, and Y. Bengio, “Zero-data learning of new tasks.,” in *AAAI*, vol. 1, 2008, p. 3.

[109] M. T. Law, Y. Yu, M. Cord, and E. P. Xing, “Closed-form training of mahalanobis distance for supervised clustering,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3909–3917.

[110] F. J. Lawin, M. Danelljan, P. Tosteberg, G. Bhat, F. S. Khan, and M. Felsberg, “Deep projective 3d semantic segmentation,” in *International Conference on Computer Analysis of Images and Patterns*, Springer, 2017, pp. 95–107.

[111] C. S. Lee and A. Y. Lee, “Clinical applications of continual learning machine learning,” *The Lancet Digital Health*, vol. 2, no. 6, e279–e281, 2020.

[112] K. Lee, A. Agrawal, and A. Choudhary, “Real-time disease surveillance using twitter data: Demonstration on flu and cancer,” in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2013, pp. 1474–1477.

[113] A. Li, Z. Lu, L. Wang, T. Xiang, and J.-R. Wen, “Zero-shot scene classification for high spatial resolution remote sensing images,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 7, pp. 4157–4167, 2017.

[114] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen, “Pointcnn: Convolution on x-transformed points,” *Advances in neural information processing systems*, vol. 31, 2018.

[115] A. Lim, J. Lin, B. Rodrigues, and F. Xiao, “Ant colony optimization with hill climbing for the bandwidth minimization problem,” *Applied Soft Computing*, vol. 6, no. 2, pp. 180–188, 2006.

[116] W. Y. B. Lim, N. C. Luong, D. T. Hoang, *et al.*, “Federated learning in mobile edge networks: A comprehensive survey,” *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 2031–2063, 2020.

[117] J. Liu, B. Kuipers, and S. Savarese, “Recognizing human actions by attributes,” in *CVPR 2011*, IEEE, 2011, pp. 3337–3344.

[118] X. Liu, Z. Han, Y.-S. Liu, and M. Zwicker, “Point2sequence: Learning the shape representation of 3d point clouds with an attention-based sequence to sequence network,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 8778–8785.

[119] Y. Liu, B. Fan, G. Meng, J. Lu, S. Xiang, and C. Pan, “Densepoint: Learning densely contextual representation for efficient point cloud processing,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 5239–5248.

[120] Y. Liu, B. Fan, S. Xiang, and C. Pan, “Relation-shape convolutional neural network for point cloud analysis,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8895–8904.

[121] X. Lu, W. Wang, C. Ma, J. Shen, L. Shao, and F. Porikli, "See more, know more: Unsupervised video object segmentation with co-attention siamese networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3623–3632.

[122] Z. Lu, J. Guan, A. Li, T. Xiang, A. Zhao, and J.-R. Wen, "Zero and few shot learning with semantic feature synthesis and competitive learning," *arXiv preprint arXiv:1810.08332*, 2018.

[123] J. Lyu, J. Akhavan Taheri Boroujeni, and S. Manoochehri, "In-situ laser-based process monitoring and in-plane surface anomaly identification for additive manufacturing using point cloud and machine learning," in *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, American Society of Mechanical Engineers, vol. 85376, 2021, V002T02A030.

[124] S. Maji, L. Bourdev, and J. Malik, "Action recognition from a distributed representation of pose and appearance," in *CVPR 2011*, IEEE, 2011, pp. 3177–3184.

[125] D. Mandal, S. Narayan, S. K. Dwivedi, *et al.*, "Out-of-distribution detection for generalized zero-shot action recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9985–9993.

[126] J. L. McClelland, B. L. McNaughton, and R. C. O'Reilly, "Why there are complementary learning systems in the hippocampus and neocortex: Insights from the successes and failures of connectionist models of learning and memory.," *Psychological review*, vol. 102, no. 3, p. 419, 1995.

[127] M. McCloskey and N. J. Cohen, "Catastrophic interference in connectionist networks: The sequential learning problem," in *Psychology of learning and motivation*, vol. 24, Elsevier, 1989, pp. 109–165.

[128] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, *Advances in neural information processing systems*, vol. 26, 2013.

[129] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in engineering software*, vol. 69, pp. 46–61, 2014.

[130] P. Mitchell, M. Spence, J. Aldridge, A. Kotilainen, and M. Diesing, "Sedimentation rates in the baltic sea: A machine learning approach," *Continental Shelf Research*, vol. 214, p. 104 325, 2021.

[131] T. Mitchell, W. Cohen, E. Hruschka, *et al.*, “Never-ending learning,” *Communications of the ACM*, vol. 61, no. 5, pp. 103–115, 2018.

[132] T. M. Mitchell, *Machine Learning*. New York: McGraw-Hill, 1997, ISBN: 978-0-07-042807-2.

[133] F. G. Mohammadi and M. S. Abadeh, “An improved image steganalysis using a novel feature selection algorithm based on artificial bee colony,” 2013, pp. 593–598.

[134] ——, “Image steganalysis using a bee colony based feature selection algorithm,” *Engineering Applications of Artificial Intelligence*, vol. 31, pp. 35–43, 2014.

[135] ——, “Image steganalysis using a bee colony based feature selection algorithm,” *Engineering Applications of Artificial Intelligence*, vol. 31, pp. 35–43, 2014.

[136] F. G. Mohammadi and M. H. Amini, “Promises of meta-learning for device-free human sensing: Learn to sense.,” *1st ACM Workshop on Device-Free Human Sensing (A ACM Buildsys 2019Workshop), NY, USA*, 2019.

[137] F. G. Mohammadi, M. H. Amini, and H. R. Arabnia, “An introduction to advanced machine learning: Meta learning algorithms, applications and promises,” 2019.

[138] ——, “Applications of nature-inspired algorithms for dimension Reduction: Enabling efficient data analytics,” in *Optimization, Learning, and Control for Interdependent Complex Networks*, Springer, 2019.

[139] ——, “Evolutionary computation, optimization and learning algorithms for data science,” Optimization, Learning, and Control for Interdependent Complex Networks, Springer, 2019.

[140] F. G. Mohammadi and H. Sajedi, “Region based image steganalysis using artificial bee colony,” *Journal of Visual Communication and Image Representation*, vol. 44, pp. 214–226, 2017.

[141] F. G. Mohammadi and M. S. Abadeh, “A survey of data mining techniques for steganalysis,” in *Recent Advances in Steganography*, H. Sajedi, Ed., Rijeka: IntechOpen, 2012, ch. 1. DOI: `10.5772/53989`. [Online]. Available: `https://doi.org/10.5772/53989`.

[142] ——, “A new metaheuristic feature subset selection approach for image steganalysis,” *Journal of Intelligent & Fuzzy Systems*, vol. 27, no. 3, pp. 1445–1455, 2014.

[143] ——, “A new metaheuristic feature subset selection approach for image steganalysis,” *Journal of Intelligent & Fuzzy Systems*, vol. 27, no. 3, pp. 1445–1455, 2014.

[144] F. G. Mohammadi and M. H. Amini, "Promises of meta-learning for device-free human sensing: Learn to sense," in *Proceedings of the 1st ACM International Workshop on Device-Free Human Sensing*, 2019, pp. 44–47.

[145] F. G. Mohammadi, M. H. Amini, and H. R. Arabnia, "An introduction to advanced machine learning: Meta-learning algorithms," *Optimization, Learning, and Control for Interdependent Complex Networks*, vol. 1123, p. 129, 2020.

[146] F. G. Mohammadi and M. H. Amini, "Promises of meta-learning for device-free human sensing: Learn to sense," in *Proceedings of the 1st ACM International Workshop on Device-Free Human Sensing*, ser. DFHS'19, New York, NY, USA: ACM, 2019, pp. 44–47, ISBN: 978-1-4503-7007-3. DOI: `10.1145/3360773.3360884`. [Online]. Available: `http://doi.acm.org/10.1145/3360773.3360884`.

[147] F. G. Mohammadi, M. H. Amini, and H. R. Arabnia, "On parameter tuning in meta-learning for computer vision," in *2019 International Conference on Computational Science and Computational Intelligence (CSCI)*, IEEE, 2019.

[148] F. G. Mohammadi, H. R. Arabnia, and M. H. Amini, "On parameter tuning in meta-learning for computer vision," in *2019 International Conference on Computational Science and Computational Intelligence (CSCI)*, IEEE, 2019, pp. 300–305.

[149] F. G. Mohammadi, A. Imteaj, M. H. Amini, and H. R. Arabnia, "Human motion recognition using zero-shot learning," in *Advances in Artificial Intelligence and Applied Cognitive Computing*, Springer, 2021, pp. 171–181.

[150] F. G. Mohammadi, F. Shenavarmasouleh, M. H. Amini, and H. R. Arabnia, "Malware detection using artificial bee colony algorithm," in *Adjunct Proceedings of the 2020 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2020 ACM International Symposium on Wearable Computers*, 2020, pp. 568–572.

[151] ——, "Evolutionary algorithms and efficient data analytics for image processing," in *2021 15th International Conference on Ubiquitous Information Management and Communication (IMCOM)*, IEEE, 2021, pp. 1–8.

[152] F. G. Mohammadi, F. Shenavarmasouleh, and H. R. Arabnia, "Applications of machine learning in healthcare and internet of things (iot): A comprehensive review," *arXiv preprint arXiv:2202.02868*, 2022.

[153] F. G. Mohammadi, F. Shenavarmasouleh, K. Rasheed, T. Taha, M. H. Amini, and H. R. Arabnia, “The application of evolutionary and nature inspired algorithms in data science and data analytics,” *arXiv preprint arXiv:2202.03859*, 2022.

[154] A. Nagabandi, C. Finn, and S. Levine, “Deep online learning via meta-learning: Continual adaptation for model-based rl,” *arXiv preprint arXiv:1812.07671*, 2018.

[155] M. Nasiri, B. Minaei, and Z. Sharifi, “Adjusting data sparsity problem using linear algebra and machine learning algorithm,” *Applied Soft Computing*, vol. 61, pp. 1153–1159, 2017.

[156] M. Norouzi, T. Mikolov, S. Bengio, *et al.*, “Zero-shot learning by convex combination of semantic embeddings,” *arXiv preprint arXiv:1312.5650*, 2013.

[157] D. Pathak, P. Mahmoudieh, G. Luo, *et al.*, “Zero-shot visual imitation,” in *International Conference on Learning Representations (ICLR)*, 2018.

[158] ——, “Zero-shot visual imitation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 2050–2053.

[159] Y. Pathak, K. Arya, and S. Tiwari, “Feature selection for image steganalysis using levy flight-based grey wolf optimization,” *Multimedia Tools and Applications*, vol. 78, no. 2, pp. 1473–1494, 2019.

[160] G. Patterson, C. Xu, H. Su, and J. Hays, “The sun attribute database: Beyond categories for deeper scene understanding,” *International Journal of Computer Vision*, vol. 108, no. 1-2, pp. 59–81, 2014.

[161] T. Pevny and J. Fridrich, “Merging markov and dct features for multi-class jpeg steganalysis,” in *Security, Steganography, and Watermarking of Multimedia Contents IX*, International Society for Optics and Photonics, vol. 6505, 2007, p. 650 503.

[162] T. Pevny, P. Bas, and J. Fridrich, “Steganalysis by subtractive pixel adjacency matrix,” *IEEE Transactions on information Forensics and Security*, vol. 5, no. 2, pp. 215–224, 2010.

[163] F. Pittaluga, S. J. Koppal, S. B. Kang, and S. N. Sinha, “Revealing scenes by inverting structure from motion reconstructions,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 145–154.

[164] I. Portugal, P. Alencar, and D. Cowan, “The use of machine learning algorithms in recommender systems: A systematic review,” *Expert Systems with Applications*, vol. 97, pp. 205–227, 2018.

[165] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, “Pointnet: Deep learning on point sets for 3d classification and segmentation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 652–660.

[166] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, “Pointnet++: Deep hierarchical feature learning on point sets in a metric space,” *Advances in neural information processing systems*, vol. 30, 2017.

[167] ——, “Pointnet++: Deep hierarchical feature learning on point sets in a metric space,” *Advances in neural information processing systems*, vol. 30, 2017.

[168] S. Ravi and H. Larochelle, “Optimization as a model for few-shot learning,” in *International Conference on Learning Representations (ICLR)*, 2017, pp. 281–288.

[169] S. Reed, Z. Akata, H. Lee, and B. Schiele, “Learning deep representations of fine-grained visual descriptions,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 49–58.

[170] J. L. Reyes-Ortiz, A. Ghio, X. Parra, D. Anguita, J. Cabestany, and A. Catala, “Human activity and motion disorder recognition: Towards smarter interactive cognitive environments.,” in *ESANN*, Citeseer, 2013.

[171] B. Romera-Paredes and P. Torr, “An embarrassingly simple approach to zero-shot learning,” in *International Conference on Machine Learning*, 2015, pp. 2152–2161.

[172] C. A. Ronao and S.-B. Cho, “Human activity recognition with smartphone sensors using deep learning neural networks,” *Expert systems with applications*, vol. 59, pp. 235–244, 2016.

[173] F. Rosenblatt, “Principles of neurodynamics. perceptrons and the theory of brain mechanisms,” Cornell Aeronautical Lab Inc Buffalo NY, Tech. Rep., 1961.

[174] V. Rostami and A. S. Khiavi, “Particle swarm optimization based feature selection with novel fitness function for image steganalysis,” in *2016 Artificial Intelligence and Robotics (IRANOPEN)*, IEEE, 2016, pp. 109–114.

[175] R. Roveri, L. Rahmann, C. Oztireli, and M. Gross, “A network architecture for point cloud classification via automatic depth images generation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4176–4184.

[176] M. Rushanan, A. D. Rubin, D. F. Kune, and C. M. Swanson, "Sok: Security and privacy in implantable medical devices and body area networks," in *2014 IEEE symposium on security and privacy*, IEEE, 2014, pp. 524–539.

[177] O. Russakovsky, J. Deng, H. Su, *et al.*, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015. DOI: `10.1007/s11263-015-0816-y`.

[178] R. B. Rusu, N. Blodow, Z. C. Marton, and M. Beetz, "Aligning point cloud views using persistent feature histograms," in *2008 IEEE/RSJ international conference on intelligent robots and systems*, IEEE, 2008, pp. 3384–3391.

[179] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. Lillicrap, "Meta-learning with memory-augmented neural networks," in *International conference on machine learning*, 2016, pp. 1842–1850.

[180] K. Sarkar, B. Hampiholi, K. Varanasi, and D. Stricker, "Learning 3d shapes as multi-layered height-maps using 2d convolutional networks," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 71–86.

[181] J. Schmidhuber, J. Zhao, and M. Wiering, "Simple principles of metalearning," *Technical report IDSIA*, vol. 69, pp. 1–23, 1996.

[182] J. Schmidhuber, "Evolutionary principles in self-referential learning," *Diploma thesis*, 1987.

[183] J. Schmidhuber, J. Zhao, and M. Wiering, "Shifting inductive bias with success-story algorithm, adaptive levin search, and incremental self-improvement," *Machine Learning*, vol. 28, no. 1, pp. 105–130, 1997.

[184] S. Shahrestani, "Assistive iot: Deployment scenarios and challenges," in *Internet of Things and Smart Environments*, Springer, 2017, pp. 75–95.

[185] F. Shenavarmasouleh and H. R. Arabnia, "Causes of misleading statistics and research results irreproducibility: A concise review," in *2019 International Conference on Computational Science and Computational Intelligence (CSCI)*, IEEE, 2019, pp. 465–470.

[186] ——, "Drdr: Automatic masking of exudates and microaneurysms caused by diabetic retinopathy using mask r-cnn and transfer learning," *arXiv preprint arXiv:2007.02026*, 2020.

[187] S. Shi, I. Malhi, K. Tran, A. Y. Ng, and P. Rajpurkar, "Unseen disease detection for deep learning interpretation of chest x-rays," in *Medical Imaging with Deep Learning*, PMLR, 2021, pp. 699–712.

[188] Y. Shigeto, I. Suzuki, K. Hara, M. Shimbo, and Y. Matsumoto, "Ridge regression, hubness, and zero-shot learning," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, Springer, 2015, pp. 135–151.

[189] S. A. Siddiqui, Y. Zhang, J. Lloret, H. Song, and Z. Obradovic, "Pain-free blood glucose monitoring using wearable sensors: Recent advancements and future prospects," *IEEE reviews in biomedical engineering*, vol. 11, pp. 21–35, 2018.

[190] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," in *Advances in Neural Information Processing Systems*, 2017, pp. 4077–4087.

[191] R. Socher, M. Ganjoo, C. D. Manning, and A. Ng, *Zero-shot learning through cross-modal transfer*, Advances in neural information processing systems, 2013, pp. 935–943.

[192] S. Song and J. Xiao, "Deep sliding shapes for amodal 3d object detection in rgb-d images," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 808–816.

[193] L. Stark and K. Bowyer, "Achieving generalized object recognition through reasoning about association of function to structure," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 10, pp. 1097–1104, 1991.

[194] A. Štulienė and A. Paulauskaite-Taraseviciene, "Research on human activity recognition based on image classification methods," *Comput. Sci*, 2017.

[195] S. Stupar, M. Bičo Ćar, E. Kurtović, and G. Vico, "The importance of machine learning in intelligent systems," in *International Conference "New Technologies, Development and Applications"*, Springer, 2021, pp. 638–646.

[196] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, "Multi-view convolutional neural networks for 3d shape recognition," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 945–953.

[197] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. Torr, and T. M. Hospedales, "Learning to compare: Relation network for few-shot learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1199–1208.

[198] M. Tatarchenko, J. Park, V. Koltun, and Q.-Y. Zhou, "Tangent convolutions for dense prediction in 3d," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3887–3896.

[199] S. Truex, N. Baracaldo, A. Anwar, *et al.*, "A hybrid approach to privacy-preserving federated learning," in *Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security*, 2019, pp. 1–11.

[200] M. A. Uy, Q.-H. Pham, B.-S. Hua, T. Nguyen, and S.-K. Yeung, "Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 1588–1597.

[201] H. Vähä-Ypyä, P. Husu, J. Suni, T. Vasankari, and H. Sievänen, "Reliable recognition of lying, sitting, and standing with a hip-worn accelerometer," *Scandinavian journal of medicine & science in sports*, vol. 28, no. 3, pp. 1092–1102, 2018.

[202] S. Veena, S. Arivazhagan, and W. S. L. Jebarani, "Improved detection of steganographic algorithms in spatial lsb stego images using hybrid grasp-bgwo optimisation," in *Nature Inspired Optimization Techniques for Image Processing Applications*, Springer, 2019, pp. 89–112.

[203] R. Vilalta and Y. Drissi, "A perspective view and survey of meta-learning," *Artificial intelligence review*, vol. 18, no. 2, pp. 77–95, 2002.

[204] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra, *et al.*, "Matching networks for one shot learning," in *Advances in neural information processing systems*, 2016, pp. 3630–3638.

[205] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The caltech-ucsd birds-200-2011 dataset," 2011.

[206] Q. Wang and K. Chen, "Zero-shot visual recognition via bidirectional latent embedding," *International Journal of Computer Vision*, vol. 124, no. 3, pp. 356–383, 2017.

[207] Q. Wang, Y. Ming, Z. Jin, *et al.*, "Atmseer: Increasing transparency and controllability in automated machine learning," in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, ACM, 2019, p. 681.

[208] Y.-X. Wang, R. Girshick, M. Hebert, and B. Hariharan, "Low-shot learning from imaginary data," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2018.

[209] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph cnn for learning on point clouds," *Acm Transactions On Graphics (tog)*, vol. 38, no. 5, pp. 1–12, 2019.

[210] ——, "Dynamic graph cnn for learning on point clouds," *ACM Trans. Graph.*, vol. 38, no. 5, 2019, ISSN: 0730-0301. DOI: `10.1145/3326362`.

[211] Z. Wang, F. Li, T. R. Taha, and H. R. Arabnia, "Improved automating seismic facies analysis using deep dilated attention autoencoders," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 0–0.

[212] Q. Wu, K. He, and X. Chen, "Personalized federated learning for intelligent iot applications: A cloud-edge based framework," *IEEE Open Journal of the Computer Society*, vol. 1, pp. 35–44, 2020.

[213] W. Wu, Z. Qi, and L. Fuxin, "Pointconv: Deep convolutional networks on 3d point clouds," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9621–9630.

[214] Z. Wu, S. Song, A. Khosla, *et al.*, "3d shapenets: A deep representation for volumetric shapes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1912–1920.

[215] Y. Xian, T. Lorenz, B. Schiele, and Z. Akata, "Feature generating networks for zero-shot learning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 5542–5551.

[216] Y. Xian, B. Schiele, and Z. Akata, "Zero-shot learning-the good, the bad and the ugly," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4582–4591.

[217] Y. Xian, S. Sharma, B. Schiele, and Z. Akata, "F-vaegan-d2: A feature generating framework for any-shot learning," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2019.

[218] G.-S. Xie, L. Liu, F. Zhu, *et al.*, "Region graph embedding network for zero-shot learning," in *European conference on computer vision*, Springer, 2020, pp. 562–580.

[219] G.-S. Xie, X.-Y. Zhang, W. Yang, M. Xu, S. Yan, and C.-L. Liu, "Lg-cnn: From local parts to global discrimination for fine-grained recognition," *Pattern Recognition*, vol. 71, pp. 118–131, 2017.

[220] J. Xie, Y. Xu, Z. Zheng, S.-C. Zhu, and Y. N. Wu, “Generative pointnet: Deep energy-based learning on unordered point sets for 3d generation, reconstruction and classification,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 14 976–14 985.

[221] J. Xu, B. S. Glicksberg, C. Su, P. Walker, J. Bian, and F. Wang, “Federated learning for healthcare informatics,” *Journal of Healthcare Informatics Research*, vol. 5, no. 1, pp. 1–19, 2021.

[222] J. Xu, R. Zhao, F. Zhu, H. Wang, and W. Ouyang, “Attention-aware compositional network for person re-identification,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 2119–2128.

[223] X. Xu, J. Deng, N. Cummins, Z. Zhang, L. Zhao, and B. Schuller, “Autonomous emotion learning in speech: A view of zero-shot speech emotion recognition,” *Proc. Interspeech 2019*, pp. 949–953, 2019.

[224] X. Xu, T. Hospedales, and S. Gong, “Semantic embedding space for zero-shot action recognition,” in *2015 IEEE International Conference on Image Processing (ICIP)*, IEEE, 2015, pp. 63–67.

[225] X.-S. Yang, *Nature-Inspired Metaheuristic Algorithms, chap 10,2nd ed.* Luniver Press, 2008, ISBN: 1905986106, 9781905986101.

[226] S. Y. Yerima and S. Sezer, “Droidfusion: A novel multilevel classifier fusion approach for android malware detection,” *IEEE transactions on cybernetics*, vol. 49, no. 2, pp. 453–466, 2018.

[227] L. Yi, V. G. Kim, D. Ceylan, *et al.*, “A scalable active framework for region annotation in 3d shape collections,” *ACM Transactions on Graphics (ToG)*, vol. 35, no. 6, pp. 1–12, 2016.

[228] T. Yu, J. Meng, and J. Yuan, “Multi-view harmonized bilinear network for 3d object recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 186–194.

[229] B. Yuan, S. Ge, and W. Xing, “A federated learning framework for healthcare iot devices,” *arXiv preprint arXiv:2005.05083*, 2020.

[230] C. Zhang, P. Patras, and H. Haddadi, “Deep learning in mobile and wireless networking: A survey,” *IEEE Communications Surveys & Tutorials*, 2019.

[231] Z. Zhang and V. Saligrama, “Zero-shot learning via semantic similarity embedding,” in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 4166–4174.

[232] ——, “Zero-shot learning via joint latent similarity embedding,” in *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 6034–6042.

[233] A. Zhao, M. Ding, J. Guan, Z. Lu, T. Xiang, and J.-R. Wen, “Domain-invariant projection learning for zero-shot recognition,” in *Advances in Neural Information Processing Systems*, 2018, pp. 1019–1030.

[234] Y. Zhao, R. Wu, and H. Dong, “Unpaired image-to-image translation using adversarial consistency loss,” in *European Conference on Computer Vision*, Springer, 2020, pp. 800–815.

[235] W. Zhu, W. Yeh, J. Chen, D. Chen, A. Li, and Y. Lin, “Evolutionary convolutional neural networks using abc,” in *Proceedings of the 2019 11th International Conference on Machine Learning and Computing*, 2019, pp. 156–162.

[236] M. Ziaeefard and R. Bergevin, “Semantic human activity recognition: A literature review,” *Pattern Recognition*, vol. 48, no. 8, pp. 2329–2345, 2015.

[237] Y. Zou and J. Feng, “Hierarchical meta learning,” *arXiv preprint arXiv:1904.09081*, 2019.