What Can We Learn from Small Data

Tamas Nyiri and Attila Kiss

Abstract—**Over the past decade, deep learning has profoundly transformed the landscape of science and technology, from refining advertising algorithms to pioneering self-driving vehicles. While advancements in computational capabilities have fueled this evolution, the consistent availability of high quality training data is less of a given. In this work, the authors aim to provide a bird's eye view on topics pertaining to small data scenarios, that is scenarios in which a less than desirable quality and quantity of data is given for supervised learning. We provide an overview for a set of challenges, proposed solution and at the end tie it together by practical guidelines on which techniques are useful in specific real-world scenarios.**

Index Terms—**deep learning, small data, small sample learning, few shot learning.**

I. Introduction

A. Background

Supervised Learning is a type of Machine Learning (ML) which itself is a sub-field of Artificial Intelligence (AI). In supervised learning, the algorithm learns from labeled data to make predictions or decisions about new, unseen data. In this type of learning, the algorithm is trained on a data set that includes both input data and corresponding output labels. The algorithm then uses this training data to learn a mapping function that can predict the output labels for new, unseen input data. [1]

Early works in AI focused on rule-based systems and expert systems, where human experts would define rules and logic for the system to follow. However, these systems were limited by the complexity and variability of real-world data.

In the 1980s, the field of Machine Learning emerged, which focused on algorithms that could automatically learn patterns from data. Early Machine Learning algorithms were primarily based on statistical models, such as linear regression and logistic regression.

In the 1990s, the development of Artificial Neural Networks brought new advances in Supervised Learning. These networks were inspired by the structure of the human brain and were capable of learning complex patterns from data through a process called back-propagation. [2]

In the early 2000s, larger neural network models, such as Convolutional Neural Networks (CNNs) [3] and Recurrent Neural Networks (RNNs) [4], were developed, leading to breakthroughs in areas such as image recognition and Natural Language Processing (NLP).

In recent years, the development of even larger Neural Network models, such as Deep Neural Networks (DNNs), has led to even greater advances in Supervised Learning. These models can learn from vast amounts of training data, and their performance has been shown to improve with increasing amounts of data.

Overall, the history of AI, ML, and Neural Networks has been characterized by a gradual progression towards larger models and more training data, which has enabled breakthroughs in Supervised Learning and other areas of Machine Learning.

This has worked well for the most part on well defined problems where large, good quality data sets are available. We focus on the situations where this assumption does not necessarily hold.

B. Related Work

The topic of learning with limited amounts of data is not a recent one. There does seem to be however many takes on what constitutes "small data" and many techniques developed to be able to achieve competitive results on less than desirable data sets.

There have been several surveys written on this topic, looking at the problem from different directions. One example, "Generalizing from a Few Examples: A Survey on Few-shot Learning" by Wang et al [5] gave a unique taxonomy of Few Shot Learning (FSL) methods, dividing them into three main categories: ones that incorporate prior knowledge into the data, model or the algorithm of the learning system. Another one "Small Sample Learning in Big Data Era" by Shu et al [6] divided Small Sample Learning (SSL) techniques into two main branches: Concept Learning which emphasizes learning new concepts from few related observations, and Experience Learning which focuses on learning with insufficient samples, co-existing with the Large Sample Learning (LSL) manner of conventional machine learning.

Even though there has also been attempts on more theoretical explanations with promising results [7][8], there does still seem to be a large gap to traverse until we see these results used in more practical settings.

Tamas Nyiri and Attila Kiss, Department of Information Systems, ELTE Eötvös Loránd University, Budapest, Hungary. Attila Kiss was also with J. Selye University, Komárno, Slovakia (E-mail: nytuaai@inf.elte.hu, kiss@inf.elte.hu)

DOI: 10.36244/ICJ.2023.5.5

C. Objectives

In subsequent sections, we do not plan to provide an exhaustive overview of the subject, given the vast scope of the topic. Instead, our objective is to highlight key challenges associated with data scarcity and the strategies formulated to tackle them. Furthermore, we delve into select real-world scenarios one could encounter during their data scientific journey, and offer practical solutions to them.

II. Small Data Sources

A. Limited annotations

The lack of annotations, refers to a common challenge faced in Supervised Learning, where the required data lacks the necessary annotations (a.k.a. labels) to train a model. An annotation or label is a piece of information that is associated with each data point that is required for the ML algorithm to learn from it. Without these annotations, the algorithm is unable to differentiate between the correct and incorrect outputs.

An example of this annotation in Computer Vision (CV) can be the names of the objects identified by their bounding boxes used for object detection or in NLP, a sentiment associated with each example sentence, that can be used for Sentiment Analysis.

The lack of annotations can occur for several reasons, including the cost, difficulty, or time-consuming nature of annotation. In many cases, it may simply be impossible to obtain annotations for certain types of data, such as historical archives or rare events. The lack of annotations can also occur in situations where data is unstructured or noisy, making it difficult to label accurately.

B. Limited diversity

Limited diversity of data points refers to a situation in which the data set used to train a machine learning model contains a small number of examples that are not representative of the entire population. This can lead to bias in the model, resulting in poor performance on new, unseen data.

An example of this phenomenon can be seen in facial recognition systems. If the data set used to train the model contains a mostly images of people with lighter skin tones or darker hair color, the model may perform poorly when presented with images of individuals with darker skin tones or lighter hair color. This is because the model has not been trained on a diverse set of data, resulting in a biased model.

The limited diversity of data points can occur for several reasons, including the difficulty in obtaining diverse data or the availability of biased data sources.

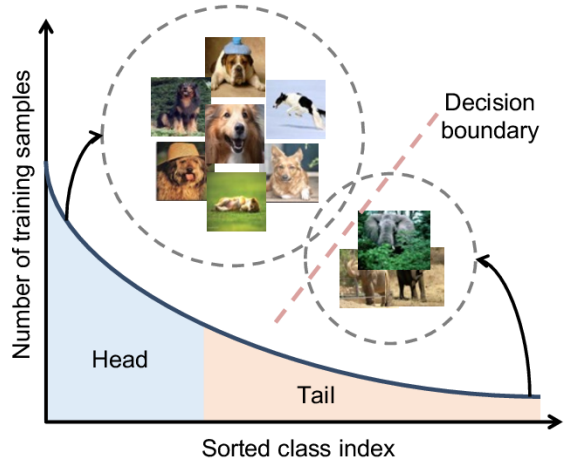

Fig. 1. Example of a long tailed dataset [9]

C. Long tail distribution

Long-tail distribution refers to a situation in which a small number of categories occur frequently, while the vast majority of categories occur infrequently.

An example of this phenomenon can be seen in the recommendation systems that suggest products to users. In many cases, a small number of popular product categories account for the majority of the purchases, while the vast majority of product categories are purchased infrequently. If the recommendation system is trained only on the popular products, it may perform poorly when making recommendations for less popular products.

The long-tail distribution can occur for several reasons, including the inherent nature of the data and the data collection process. In some cases, it may be difficult or expensive to collect data on the less popular data points, resulting in a bias towards the more popular data points.

To understand the origins of this phenomena, it is important to understand that long tail distributions are abundant in nature and thus will naturally show up in randomly sampled data.

One example of this would be the Pareto distribution, which describes the relative wealth distribution in sociology, or Zipf's law which states that in a given corpus of natural language, the frequency of any word is approximately inversely proportional to its rank in the frequency table. [10]

D. Concept drift

Concept drift refers to a situation in which the statistical properties of the target variable in a Machine Learning problem change over time, resulting in a decrease in the performance of the trained model. This can be

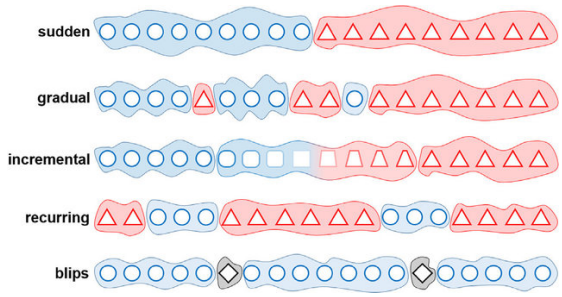

Fig. 2. Five types of Concept Drift according to [11]

a challenge in ML, as the model may become outdated and unable to accurately predict new, unseen data.

An example of this phenomenon can be seen in a spam email classifier. The distribution of spam emails may change over time, with new types of spam emails appearing that are not similar to those seen before. If the model has been trained only on the earlier types of spam emails, it may perform poorly when presented with the new types of spam emails, resulting in increased proportion of false negatives or false positives.

Concept drift can occur for several reasons, including changes in the behavior of users, changes in the environment, and changes in the data generation process. In many cases, the drift is gradual, making it difficult to detect and correct.

III. Small Data Solutions

A. Smart Sampling

The very first step in most practical machine learning is to ensure the data collected is the best quality possible, that is our sample is closest possible to our population. There are several statistical techniques developed over the years. Here we will only show a few that are most useful in a limited data environment.

1) Under-sampling and Over-sampling: When one has to deal with imbalanced dataset, a common approach is to either over-sample the minority class(es) or under-sample the majority class(es) until the desired distribution is reached.

This can be done by randomly removing samples (under-sampling) or adding multiple copies of the same sample (over-sampling) at random.

They both have their disadvantages. Under-sampling can lead to a loss of information since we leave out potentially relevant information from our training dataset. On the other-hand, over-sampling can reinforce existing biases in the over-sampled instances. For these reasons, it's usually better to use a more sophisticated method where possible.

2) Importance Sampling: Importance sampling is particularly useful for catching rare events in long-tail distributions. This involves creating a new distribution where rare events become not-so-rare, sampling from this new distribution, then re-weighing the samples to adjust for the bias introduced.

Let's say we have a target distribution $p(x)$ and an importance distribution $q(x)$. In order to arrive at an approximation of the expectation of a function $f(x)$ under the target distribution:

$$\mathbb{E}_p[f(x)] = \int f(x)p(x)\,dx \tag{1}$$

we can sample from the importance distribution (where x_i are samples drawn from $q(x)$) and then re-weight the samples:

$$\mathbb{E}_p[f(x)] \approx \frac{1}{N}\sum_{i=1}^{N}\frac{p(x_i)}{q(x_i)}f(x_i) \tag{2}$$

3) Active Learning: Active learning is an iterative process where the model selects the most informative samples to be labeled, thus reducing the amount of labeling resources to be used.

An early example for an active learning algorithm introduced by Cohn, Atlas and Ladner [12] is Query by Committee (QBC). The central idea behind QBC is to maintain a committee of models (set of hypotheses) over the data, and to obtain labels for instances about which the committee members disagree the most. This can reduce the amount of labeling by focusing only on the most informative (highest entropy) examples, but at the same time can introduce a large computational overhead. For the detailed algorithm, see Algorithm 1

Algorithm 1 Query By Committee (QBC)

Input: Dataset $\mathcal{D}$, committee of models C
Train each model $c_i \in C$ on $\mathcal{D}$
while stopping criterion not met **do**
  for each unlabeled x_u **do**
    Calculate disagreement:
    $D(x_u) = \sum_{c_i,c_j\in C, i\neq j} \mathbb{I}(c_i(x_u) \neq c_j(x_u))$
  end for
  Query label for instance $x^* = \arg\max_{x_u} D(x_u)$
  Add labeled instance (x^*, y^*) to $\mathcal{D}$
  Re-train each model $c_i \in C$ on $\mathcal{D}$
end while
Output: Labeled dataset $\mathcal{D}$

It has to be mentioned that this is just an early example, since its introduction several other Active Learning techniques have developed, such as Uncertainty Sampling [13] Expected Model Change [14], Expected Error

Reduction [15], Variance Reduction [16], Bayesian Active Learning by Disagreement [17], Diversity Sampling [18], Hierarchical Sampling [19], and Online Active Learning [20], to name a few.

B. Expert knowledge

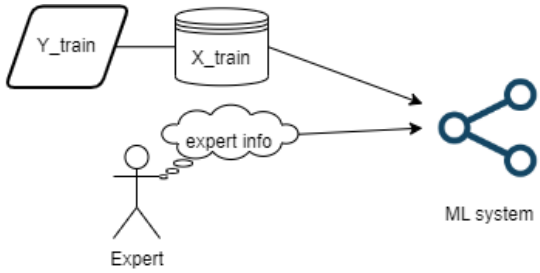

Fig. 3. Schematic drawing of "Expert Knowledge"

1) Taxonomy: One way in which expert knowledge can be incorporated into Deep Learning is through the use of taxonomies. A taxonomy is a hierarchical or otherwise structured organization of data that categorizes items or concepts based on their similarities, differences, and relationships.

In many machine learning tasks, a binary decision can be expanded into a multi-class decision using a taxonomy. This taxonomy divides the initial binary classes into more specific sub-categories, organized hierarchically in a tree-like fashion. Rather than training the model on the initial binary labels, one can use the more detailed labels corresponding to the leaves of this hierarchical structure. This approach allows the model to establish more intricate decision boundaries, capturing subtleties that might be overlooked in a simple binary classification.

Once the model is trained and deployed, predictions made at the leaf-level can be aggregated back to the original binary classification, if necessary.

2) Hand-crafted features: Another way in which expert knowledge can be incorporated into deep learning is through the use of hand-crafted features. Hand-crafted features are manually designed features that can be used as inputs to a neural network. These features are often designed based on domain-specific knowledge or prior research, and can be used to capture important characteristics of the data that may not be captured by the network's automatic feature learning.

These techniques used to be the back-bone of many AI algorithms before Deep Learning came into the picture, but have quickly fallen out of favor due to Deep Neural Networks' ability to learn similar but more complicated features. Examples of such techniques in Computer Vision include Histogram of Oriented Gradients (HOG) [21] and Local Binary Patterns (LBP) [22].

C. Data Augmentation

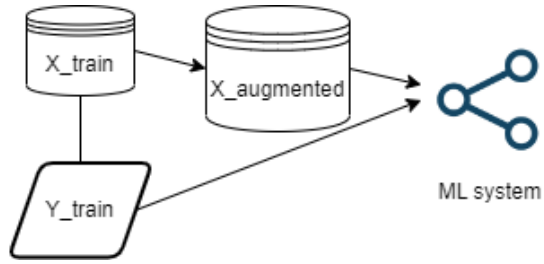

Fig. 4. Schematic drawing of "Data Augmentation"

1) Heuristic-based Methods: Heuristic-driven data augmentation techniques apply specific rules or heuristics to original data, generating new data samples. Designed to imitate natural data variations, these methods produce samples closely resembling, but not identical to, the original ones.

For image data, examples include geometric and color-space adjustments like random cropping, rotation, shifting, and variations in color through flips and jitter.

The same in the case of text-based input can involve: synonym replacement, back-translation, random deletion/insertion, random swap, etc..

It's important to note that these transformations need to be invariant with respect to the labels associated with the input data.

2) Data Generation: Data generation is a data augmentation method in Deep Learning that involves generating new synthetic data from scratch instead of transforming or manipulating existing data samples. This is typically done using generative models, which are deep learning models designed to learn the underlying patterns and structure of the data and generate new samples that are similar to the original data.

One of the most common generative models used for data generation is the generative adversarial network (GAN). GANs consist of two deep neural networks: a generator network and a discriminator network. The generator network takes a random input vector and generates a synthetic data sample, while the discriminator network tries to distinguish between the synthetic data and the real data.

During training, the generator and discriminator networks are trained together in a zero-sum game, where the generator tries to generate synthetic data that fools the discriminator, and the discriminator tries to correctly distinguish between the synthetic and real data. Over time, the generator becomes better at generating realistic data samples, and the discriminator becomes better at distinguishing between the synthetic and real data.

Given a noise variable z drawn from a prior distribution $p(z)$, generator G tries to produce something similar to a sample, $G(z)$.

Given a real sample x, drawn from the observed distribution $p_{data}(x)$ and the fake sample $G(z)$, the discriminator tries to differentiate between the two and outputs a probability associated with its confidence that the generated sample is from the observed distribution.

This way, we get a two-player minimax game, with the value function $V(D,G)$:

$$V(D,G) = \mathbb{E}_{x\sim p_{\text{data}}(x)}[\log D(x)] \quad (3)$$

$$+ \mathbb{E}_{z\sim p(z)}[\log(1 - D(G(z)))] \quad (4)$$

The discriminator tries to maximize $V(D,G)$ with respect to D, while the generator tries to minimize the same.

This results in the following optimization problem:

$$\min_G \max_D V(D,G) \quad (5)$$

Where the generator and discriminator are trained alternatively step-by-step.

D. Semi-supervised Learning

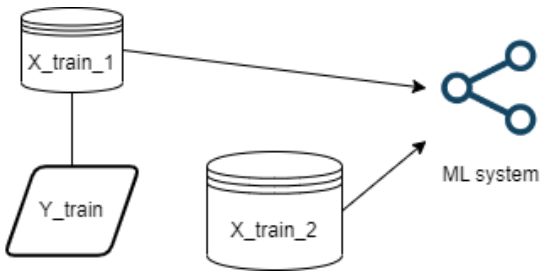

Fig. 5. Schematic drawing of "Semi-supervised Learning"

Semi-supervised learning is a type of machine learning that involves training a model on both labeled and unlabeled data. The core idea is that even though the unlabeled data doesn't provide direct supervision, it still contains valuable information about the underlying data distribution that can assist the learning process. This approach is especially useful when labeled data is limited, expensive, or time-consuming to obtain, but unlabeled data is abundant.

An example of Semi-Supervised Learning is Consistency Regularization, where the model is trained to be robust against different data augmentations by ensuring consistent predictions for different augmented views of the same input, even if the input is unlabeled. [23]

Another is MixMatch, which leverages the MixUp process: a data augmentation technique that creates virtual training examples by linearly interpolating between pairs of examples and their associated labels. MixMatch takes a pair of data points (one from the labeled set and one from the unlabeled set with its guessed label) and applies the MixUp process on them. [24]

Pseudo-labeling is another straightforward way to utilize unlabeled data. The idea is to train a model on all the labeled data and then predict on the rest (unlabeled) data points. If the prediction certainty reaches a certain confidence, we assign the data with the predicted label and use it to retrain the model. [25]

E. Self-supervised Learning

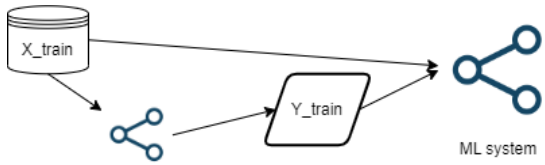

Fig. 6. Schematic drawing of "Self-supervised Learning"

In self-supervised learning, the algorithm learns to generate its own labels or representations from the input data itself, without any explicit supervision. This is usually achieved by defining a "pretext task" or "auxiliary task" that helps the model learn useful features or representations from the data. The idea is that these learned features will be useful for downstream tasks, like classification or regression.

Common examples of self-supervised learning in a text-based context include language model pre-training (e.g., BERT [26], GPT [27]), where the model learns to predict the next word in a sentence, based on huge amounts of unlabeled text data, where the training consists of hiding certain parts of the input and letting the model try to guess the right answer. In this was, the labels are the masked parts of the unlabeled input data, which are hidden from the models during training.

In the realm of computer vision, a famous self-supervised learning method is Contrastive Learning [28]. Here, the model tries to learn an embedding space where similar images are closer to each other, and dissimilar images are farther away. This is done by utilizing concepts such as positive pairs and negative pairs, where positive pairs can be different augmentations of the same data point.

Given:

- $D_w(x_i, x_j)$ as the Euclidean distance between two data point embeddings x_i and x_j, where w represents the parameters of the neural network.
- y as a binary label indicating whether the pair is a positive pair ($y = 1$) or a negative pair ($y = 0$).
- m as a predefined margin to ensure that negative pairs have distances greater than this margin.

The contrastive loss L for a single pair is:

$$L_{ij} = \frac{1}{2} y D_w(x_i, x_j)^2 + \frac{1}{2}(1-y)\max(0, m - D_w(x_i, x_j))^2$$

F. Transfer Learning

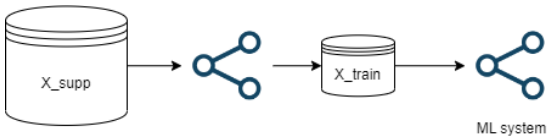

Fig. 7. Schematic drawing of "Transfer Learning"

Transfer learning is a Machine Learning technique that involves using knowledge gained from solving one problem to improve the performance of a model on a different, but related, problem. Transfer learning is based on the intuition that the knowledge and representations learned by a model on one task can be transferred to a different task that shares similar features or structure. The review by Pan et al [29] identified three main subcategories: Inductive, Transductive and Unsupervised Transfer Learning. Below is a quick summary of each:

1) Inductive Transfer Learning: This technique involves transferring knowledge across domains or tasks, where labeled data is available in both the source and the target domain.

2) Transductive Transfer Learning: Here we focus on domain adaptation, where the task remains unchanged, but the data distribution differs, and no labeled data is available in the target domain.

3) Unsupervised Transfer Learning: In this scenario, we attempt to transfer knowledge from the source domain/task to improve the learning of an entirely different task in the target domain, where no labeled data is available.

G. Meta learning

In meta-learning, the goal is to train a model to learn how to learn from a small number of examples, and then use this learned knowledge to rapidly adapt to new, unseen tasks.

There are different approaches to meta-learning, but a common one is to use a "meta-learner" that learns to update the parameters of a "base-learner" model based on a small amount of data from a new task. This process of updating the base-learner parameters based on new tasks is sometimes referred to as "meta-training.". We can identify several categories of Meta Learning, such as Model based, Metric based and Optimization based meta-learning. [30]

1) Model based meta-learning: Here we are training a meta-learner on a set of training tasks, each with limited number of labels. Whenever a new task is presented, the meta-learner adjusts its internal parameters based on the training examples and desired labels for the new task.

One example of a model-based meta-learning algorithm is Memory-Augmented Neural Networks. The core idea is to augment the model architecture (neural network) with an external memory mechanism. This introduces an extra memory component to the training process. Instead of only updating the weights of the network as traditional neural networks do, they can also update the content of their memories to perform better on new tasks. [31]

2) Metric based meta-learning: In metric based meta-learning, we have a distance metric in the space of tasks that can be used to quickly identify similar tasks and generalize to new tasks. The meta-learner can be given new tasks and a few related examples and is trained to be able to identify the similarity between these new tasks and the old ones in its space of tasks.

An example of this is Prototypical Networks. For each class, it computes a prototype (mean representation) from the embedding associated with the examples in that class. For a new data point, its class is determined by its proximity to these prototypes. [32]

3) Optimization based meta-learning: Finally, optimization based meta-learning approaches meta-learning as a bi-level optimization problem. At the inner-level, a base-learner makes task-specifc updates using some optimization strategy (such as gradient descent). At the outer-level, the performance across tasks is optimized.

Here, we can look at Model-Agnostic Meta Learning where the aim is to find a set of model parameters that are not optimal for any single task, but can be quickly adapted to any of the tasks within the desired set of tasks. [31]

IV. Small Data Scenarios

Now let's examine some scenarios a practitioner in the field might encounter in the real-world. For each we will list the most likely problems that can arise and recommended solutions from our list of techniques examined.

A. Diagnosis of Rare Diseases from Medical Images

- Small Data Sources:
  - Limited annotations: Obtaining labels might involve invasive/expensive procedure.
  - Limited diversity: Examples might come from a few specialized hospitals/geographical regions only.

  - Long tail distribution: Many common diseases and a few rare ones.
- Small Data Solutions:
  - Expert knowledge: Incorporate knowledge from medical professionals. [33]
  - Taxonomy: Subdivide diseases based on origin. [33]
  - Data Augmentation: Generate more images through invariant transformations. [33]
  - Transfer Learning: Use models pre-trained on larger dataset. [33]

B. Predicting Customer Churn in a New Market

- Small Data Sources:
  - Limited annotations: Company is still new, so small existing dataset.
  - Concept drift: Customer behavior might change over time, especially in new markets.
- Small Data Solutions:
  - Active Learning: Keep updating the model by querying the most uncertain predictions. [34]
  - Expert knowledge: Incorporate business intelligence and market insights. [35]
  - Semi-supervised Learning: Incorporate information about customer interactions. [36]

C. Sentiment Analysis for a Less Common Language

- Small Data Sources:
  - Limited annotations: Fewer examples in rare languages.
  - Limited diversity: Most examples might come from a limited set of sources (people who like to leave reviews).
  - Concept drift: Words and phrases change their meanings over time, sense of humor might evolve.
- Small Data Solutions:
  - Smart Sampling: Choose diverse examples across different all possible languages.
  - Data Generation: Use translation tools to augment data. [37]
  - Self-supervised Learning: Predict which words are the best sentiment predictors. [38]
  - Transfer Learning: Transfer user biases to textual features. [39]

D. Self-driving in a New Environment

- Small Data Sources:
  - Limited diversity: Training data might not include all types of environments.
  - Concept drift: Environment can change over time (e.g., changes in climate/lighting conditions).
  - Long tail distribution: Certain events in driving happen rarely (e.g., crashes).
- Small Data Solutions:
  - Data Augmentation: Simulate different lighting and object placements. [40]
  - Importance Sampling: Weight experiences that are less frequent but important (like crashes) more heavily.
  - Meta-learning: Use knowledge from common objects to help detect rare ones. [41]

V. Discussion

The era of big data has led to a vast landscape of deep learning techniques, leaving the average practitioner uncertain about which direction to take for unfamiliar challenges. Furthermore, much of the theoretical groundwork is done on unrealistically large and good quality data sources that doesn't take into account the natural shift in the specific domain studied.

Through this paper, our aim is to guide practitioners by offering a concise summary of frequently faced challenges and potential solutions. This is complemented by a curated set of real-world examples. It is our sincere hope that readers find value in our efforts.

References

[1] E. Alpaydin, *Introduction to machine learning*. MIT press, 2020.
[2] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *nature*, vol. 323, no. 6088, pp. 533–536, 1986.
[3] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
[4] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
[5] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni, "Generalizing from a few examples: A survey on few-shot learning," *ACM computing surveys (csur)*, vol. 53, no. 3, pp. 1–34, 2020.
[6] J. Shu, Z. Xu, and D. Meng, "Small sample learning in big data era," *arXiv preprint arXiv:1808.04572*, 2018.
[7] L. Szymanski, B. McCane, and C. Atkinson, "Conceptual capacity and effective complexity of neural networks," *arXiv preprint arXiv:2103.07614*, 2021.
[8] A. M. Saxe, Y. Bansal, J. Dapello, M. Advani, A. Kolchinsky, B. D. Tracey, and D. D. Cox, "On the information bottleneck theory of deep learning," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2019, no. 12, p. 124020, 2019.
[9] Y. Zhang, B. Kang, B. Hooi, S. Yan, and J. Feng, "Deep long-tailed learning: A survey," *arXiv preprint arXiv:2110.04596*, 2021.
[10] M. E. Newman, "Power laws, pareto distributions and zipf's law," *Contemporary physics*, vol. 46, no. 5, pp. 323–351, 2005.
[11] B. Krawczyk and A. Cano, "Online ensemble learning with abstaining classifiers for drifting and noisy data streams," *Applied Soft Computing*, vol. 68, pp. 677–692, 2018.
[12] D. Cohn, L. Atlas, and R. Ladner, "Improving generalization with active learning," *Machine learning*, vol. 15, pp. 201–221, 1994.
[13] D. D. Lewis, "A sequential algorithm for training text classifiers: Corrigendum and additional data," in *Acm Sigir Forum*, vol. 29, no. 2. ACM New York, NY, USA, 1995, pp. 13–19.

[14] D. A. Cohn, Z. Ghahramani, and M. I. Jordan, "Active learning with statistical models," *Journal of artificial intelligence research*, vol. 4, pp. 129–145, 1996.

[15] N. Roy and A. McCallum, "Toward optimal active learning through monte carlo estimation of error reduction," *ICML, Williamstown*, vol. 2, pp. 441–448, 2001.

[16] A. Kapoor, K. Grauman, R. Urtasun, and T. Darrell, "Active learning with gaussian processes for object categorization," in *2007 IEEE 11th international conference on computer vision*. IEEE, 2007, pp. 1–8.

[17] N. Houlsby, F. Huszár, Z. Ghahramani, and M. Lengyel, "Bayesian active learning for classification and preference learning," *arXiv preprint arXiv:1112.5745*, 2011.

[18] K. Brinker, "Incorporating diversity in active learning with support vector machines," in *Proceedings of the 20th international conference on machine learning (ICML-03)*, 2003, pp. 59–66.

[19] S. Vijayanarasimhan and K. Grauman, "What's it going to cost you?: Predicting effort vs. informativeness for multi-label image annotations," in *2009 IEEE conference on computer vision and pattern recognition*. IEEE, 2009, pp. 2262–2269.

[20] N. Cesa-Bianchi, C. Gentile, A. Tironi, and L. Zaniboni, "Incremental algorithms for hierarchical classification," *Advances in neural information processing systems*, vol. 17, 2004.

[21] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, vol. 1. IEEE, 2005, pp. 886–893.

[22] T. Ahonen, A. Hadid, and M. Pietikäinen, "Face recognition with local binary patterns," in *Computer Vision-ECCV 2004: 8th European Conference on Computer Vision, Prague, Czech Republic, May 11-14, 2004. Proceedings, Part I 8*. Springer, 2004, pp. 469–481.

[23] S. Laine and T. Aila, "Temporal ensembling for semi-supervised learning," *arXiv preprint arXiv:1610.02242*, 2016.

[24] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. A. Raffel, "Mixmatch: A holistic approach to semi-supervised learning," *Advances in neural information processing systems*, vol. 32, 2019.

[25] D.-H. Lee et al., "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," in *Workshop on challenges in representation learning, ICML*, vol. 3, no. 2. Atlanta, 2013, p. 896.

[26] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[27] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever *et al.*, "Improving language understanding by generative pre-training," 2018.

[28] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 2. IEEE, 2006, pp. 1735–1742.

[29] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.

[30] M. Huisman, J. N. Van Rijn, and A. Plaat, "A survey of deep meta-learning," *Artificial Intelligence Review*, vol. 54, no. 6, pp. 4483–4541, 2021.

[31] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *International conference on machine learning*. *PMLR*, 2017, pp. 1126–1135.

[32] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," *Advances in neural information processing systems*, vol. 30, 2017.

[33] A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, and S. Thrun, "Dermatologist-level classification of skin cancer with deep neural networks," *nature*, vol. 542, no. 7639, pp. 115–118, 2017.

[34] W. Verbeke, D. Martens, C. Mues, and B. Baesens, "Building comprehensible customer churn prediction models with advanced rule induction techniques," *Expert systems with applications*, vol. 38, no. 3, pp. 2354–2364, 2011.

[35] E. Lima, C. Mues, and B. Baesens, "Domain knowledge integration in data mining using decision tables: case studies in churn prediction," *Journal of the Operational Research Society*, vol. 60, no. 8, pp. 1096–1106, 2009.

[36] X. Liu, M. Xie, X. Wen, R. Chen, Y. Ge, N. Duffield, and N. Wang, "A semi-supervised and inductive embedding model for churn prediction of large-scale mobile games," in *2018 IEEE international conference on data mining (ICDM)*. *IEEE*, 2018, pp. 277–286.

[37] A. Balahur and M. Turchi, "Multilingual sentiment analysis using machine translation?" in *Proceedings of the 3rd workshop in computational approaches to subjectivity and sentiment analysis*, 2012, pp. 52–60.

[38] J. Tang, Z. Lu, J. Su, Y. Ge, L. Song, L. Sun, and J. Luo, "Progressive self-supervised attention learning for aspect-level sentiment analysis," *arXiv preprint arXiv:1906.01213*, 2019.

[39] P. H. Calais Guerra, A. Veloso, W. Meira Jr, and V. Almeida, "From bias to opinion: a transfer-learning approach to real-time sentiment analysis," in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2011, pp. 150–158.

[40] Y. Chen, F. Rong, S. Duggal, S. Wang, X. Yan, S. Manivasagam, S. Xue, E. Yumer, and R. Urtasun, "Geosim: Realistic video simulation via geometry-aware composition for self-driving," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 7230–7240.

[41] Y.-X. Wang, D. Ramanan, and M. Hebert, "Meta-learning to detect rare objects," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9925–9934.

Tamas Nyiri finished him Masters degree in Computer Science in 2019 and started his PhD in 2022, both at Eötvös Loránd University. His research is mainly focused on deep learning scenarios involving data quality and interpretability issues.

Attila Kiss defended his PhD in the field of database theory in 1991. His research is focused on information systems, data mining, and artificial intelligence. He has more than 190 scientific publications. Seven students received their PhD degrees under his supervision. Since 2010, he has been the head of Department of Information Systems at Eötvös Loránd University, Hungary. He is also teaching at J. Selye University, Slovakia.