Project Machine Learning
— Milestone 1 —

Ricardo Fleck, Hassan Bassiouny, Augustin Krause

November 24, 2023

1 Introduction

In this report, we detail our implementation of 'Attention-based Multiple Instance Learning' (MIL), following the framework presented by (Ilse et al., 2018). MIL addresses the machine learning challenge of assigning a class label to a bag of instances. Specifically, we focus on the binary case, wherein our model predicts a label $Y \in \{0, 1\}$ for a bag of instances $X = \{x_1, ..., x_K\}$. Each individual instance x_k is paired with a corresponding label y_k for $k \in \{1, ..., K\}$, and the length K of the bags may vary. We call an instance positive if its associated label $y_k = 1$ and negative if $y_k = 0$. The label of the bag depends on the labels assigned to individual instances within it. A bag assumes the label 1 if at least one of its instances carries a 1 label. It is then called positive. Conversely, we call it negative, if and only if all instances within it are negative. Crucially, the labels of the instances are unknown during training. From these assumptions, we can write the bag label as the maximum over its instance labels:

$$Y = \max_k y_k. \tag{1}$$

This formulation presents two primary challenges: (i) gradient-based methods encounter issues with vanishing gradients, and (ii) it is limited to single-instance-based approaches. Both challenges can be mitigated by optimizing our model's training over the log-likelihood function instead of the maximum. This adjustment allows our model to learn the probability $\theta(X) \in [0, 1]$ of $Y = 1$ given a bag of instances X, following a *Bernoulli* distribution with parameter $\theta(X)$.

Structural Overview As can be seen from Figure 1, our code is organized into two core directories: `data` and `model`. Within `data`, we further subdivide into `datasets`, housing all loaded datasets, and `datasets_loader`, containing a loader for classical MIL datasets like `MUSK1`, `MUSK2`, `TIGER`, `FOX`, `ELEPHANT`, and one for `MNIST-bags`. These loaders are complemented by utility functions in the `data_utils` folder. Some of these will be detailed in Section 2.1.1. In the `model` directory, we have stored both the model and the training script.

2 Data Set Overview

In the work by Ilse et al. (2018), various datasets are referenced and categorized into three distinct groups:

- the `MNIST-bags` dataset.
- Classical MIL datasets such as `MUSK1`, `MUSK2`, `TIGER`, `FOX`, and `ELEPHANT`.
- `Histopathology` datasets.

In this milestone, we have opted to explore both the `MNIST-bags` and Classical MIL datasets.

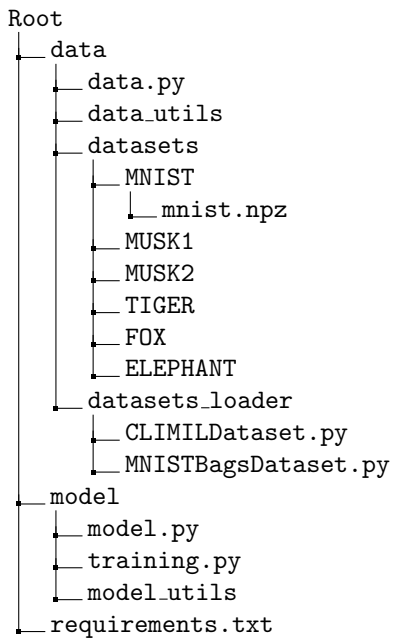


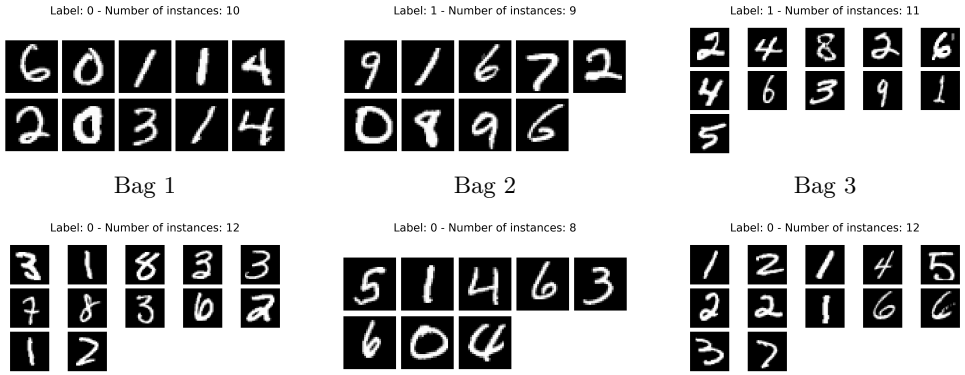
Figure 1: Our project structure.

2.1 Data Composition and Quality Assessment

2.1.1 MNIST-bags

Creation of Bags The "MNIST" dataset is a widely known benchmark in image recognition and related tasks. It is a collection of 28x28 grayscale images of handwritten digits from 0 to 9. In total, its training set comprises 60,000 examples, with an evenly distributed 6,000 examples per digit. Similarly, the test set consists of 10,000 examples, with 1,000 examples allocated per digit.
In their paper, Ilse et al. (2018) used MNIST to create an `MNIST-bags` dataset, suitable for MIL. We have followed their creation scheme meticulously and our implementations can be found in the `data_utils` folder (see Figure 1).
To construct bags from these instances, a bag is formed by sampling a random bag length from a Gaussian distribution. Each bag is assigned a label based on the presence of a pre-selected target number. Ilse et al. (2018) chose 9 as their target number, due to its susceptibility to being mistaken with '7' or '4'. If a bag contains one or more images depicting the chosen target number, it is labeled as positive; otherwise, it receives a negative label. This is done until a set number of training and test bags have been created.
From this creation scheme it follows that the bags within the training set could overlap with other bags from the training set. The same holds true for the test set. We do not deem this a significant problem since the instances are sampled from large selections of different data points, and therefore overlap should be rare. We implemented it this way to not diverge from the `MNIST-bags` implementation of Ilse et al. (2018).
We showcase visual examples of MNIST-based bags that feature the digit 9 as their target number in Figure 2.

Quality of Data Maintaining data integrity involves balancing the representation of positive and negative bags within the dataset. An equal count of positive and negative bags is ensured by the creation scheme in the following way:
To achieve parity, after the sampling of a negative bag, a positive bag is forcibly selected. Conversely, after a positive bag is chosen, a negative bag follows suit.
Moreover, the MNIST dataset inherently lacks duplicates, missing values, or any form of irrelevant or corrupt data.



Bag 1 Bag 2 Bag 3

Bag 4 Bag 5 Bag 6

Figure 2: Examples of `MNIST-bags`

2.1.2 Classical MIL datasets

There are several established benchmark datasets to evaluate algorithms and models in MIL. Among the classical MIL datasets commonly employed for benchmarking are `MUSK1`, `MUSK2`, `TIGER`, `FOX`, and `ELEPHANT`. These datasets represent diverse real-world scenarios and encapsulate challenges inherent in MIL problems. The data in these classical MIL datasets are structured in the form of vector embeddings, where features are already extracted.
For instance, the `MUSK1` and `MUSK2` datasets involve molecules, where the task is to distinguish between musks and non-musks based on molecular embeddings (Dietterich et al. (1997)).
Similarly, the `TIGER`, `FOX`, and `ELEPHANT` datasets encompass image classification scenarios, where the data is presented as pre-processed feature vectors instead of individual images (Andrews et al. (2002)).

Shape and Quality of the Data The Classical MIL datasets are already organized into bags, with each bag having its corresponding label. Relevant metadata of the classical MIL datasets can be found in Table 1.

Name	Features (Non-Zero)	Positive Bags	Negative Bags	Positive Instances	Negative Instances
`ELEPHANT`	230 (143)	100	100	762	629
`FOX`	230 (143)	100	100	647	673
`TIGER`	230 (143)	100	100	544	676
`MUSK1`	166	47	45	207	269
`MUSK2`	166	39	63	1017	5581

Table 1: Metadata about Classical MIL datasets (*Source: http://www.cs.columbia.edu/ andrews/mil/datasets.html*)

The classical MIL datasets are devoid of duplicates, missing values, or any form of erroneous or irrelevant data. We therefore consider them to be of high quality standards.
Additionally, most of the datasets contain an equitable distribution of positive and negative bags. The dataset where this balance is least guaranteed is `MUSK2`. In this dataset, negative bags are roughly 1.5 times more frequent than positive ones. This could lead to problems further down the line, since the identification of a positive bag is a more ambiguous task than the identification of a negative bag. (Andrews et al. (2002)).
Moreover, most of the classical MIL datasets are relatively small compared to what a neural networks usually needs in order to generalize well (Liu et al. (2017)). This means that neural networks could inherently be a sub-optimal choice for solving MIL problems on the data.

2.2 Feature Extraction Method

2.2.1 MNIST

The feature extraction process for MNIST datasets in the model architecture used by Ilse et al. (2018) consists of two convolutional layers equipped with multiple filters. These are theoretically able to extract specific and discriminating features from the input images. Subsequently, a fully connected layer is employed to map these extracted features into a meaningful embedding. The layers are activated by the Rectified Linear Unit (ReLU). This established architecture (called the LeNet5 architecture) has demonstrated high performance in extracting relevant features from MNIST(-based) datasets (see LeCun et al. (1998)).
How we specifically implemented this architecture is outlined in more detail in section 3.1.

2.2.2 Classical MIL datasets

The classical MIL datasets arrive in the form of pre-processed embedding vectors that already consist of extracted features. In Ilse et al. (2018), they advocate for an architecture with the capability to further extract features from these embeddings. This architecture leverages three fully connected layers, activated by the ReLU function, and is described in more detail in Section 3.1. It is based on the work of Wang et al. (2018). In this work, they have empirically validated that this architecture is able to extract nuanced and meaningful features from the existing embeddings.

2.3 Normalization

For the Classical MIL datasets, provided in pre-processed vector embeddings, no additional normalization is required. They are already in a normalized form.
In the case of `MNIST-bags`, the individual pixel values initially range from 0 to 255, potentially impeding training efficiency. A normalization step mitigates any potential problems arising from that. We normalize by dividing each pixel value in the dataset by 255, to cause the range of the pixel values to lie between 0 and 1. This normalization process facilitates smoother and more effective training of the model, aiding convergence and preventing saturation of network activations by ensuring a suitable input range for the neural network.

2.4 Machine Learning Potential and Queries

The approach of MIL allows models trained within this setting to work with "weakly-labeled" data. This has the potential to allow for faster labeling processes in, for instance, the medical domain. Here, exact labeling of where a specific region of interest lies within an image is time-consuming for doctors who usually have more pressing responsibilities. To instead just assign a label for the whole image can be done much faster (Andrews et al. (2002)).
Moreover, the use of attention mechanisms in MIL enables the identification of those instances within a bag that trigger positive classification (called 'key instances'). It could also shed light on whether multiple instances collectively contribute to the outcome. Such analyses can potentially uncover previously undiscovered latent patterns or anomalies within the data.
Figure 3 is taken from Ilse et al. (2018) and displays the attention values assigned to each element within an MNIST bag by a trained model. This shows how it could be possible to identify key instances on the basis of attention values in a well-trained network.

Human Queries The trained attention-based MIL model is capable of answering queries on entire bags, and of supporting queries trying to identify significant areas within those bags. Here are some examples of such queries:

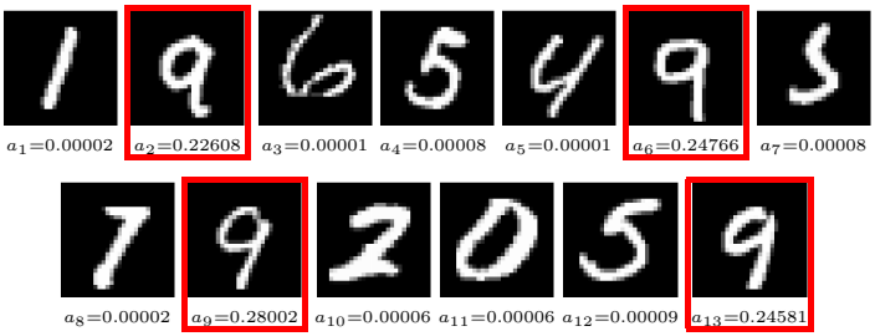


Figure 3: Example of attention weights for a positive bag.

- Can we identify the photos within a set of histopathological photos of a specific organ that showcase an organ with high risk of being cancerous? Can we pinpoint where in these photos we have indicators of potential malignancy?
- Can we segment this extensive image containing numerous animals into smaller images and identify which of these sub-images contain a tiger?
- Within the MNIST dataset comprising digit images, can the model distinguish unique features indicative of particular digits? Which images exhibit prominent characteristics representing distinct digits?
- Analyzing sequences of embedded features within Classical MIL datasets, can the model identify instances representing specific patterns or anomalies? Could it highlight the instances where these patterns are most prevalent?

3 Baseline Method and Evaluation

3.1 Simple Baselines

Modeling Approaches In MIL, the predicted probability of a bag X having label 1 can be modeled by $\hat{\theta}(X) = g(\sigma(f(X)))$, with suitable functions f, σ, and g. Depending on the choice of f, this is called either an "instance-based" approach, or an "embeddings-based" approach (Ilse et al. (2018)).
In the former, f outputs a 'score' per instance in X, representing the estimated probability of the instance being 'positive'. These scores are then aggregated by the pooling-operation given by σ. This will yield a final score for the whole bag, which does not need any further processing. Hence, g is the identity here.
In the latter, the instances in X are first transformed to an embedding-space. σ then pools these extracted embeddings together, to form a bag score that does not depend on the number of instances in the bag. In this approach, g is a classifier working on the bag-level that outputs the probability $\hat{\theta}(X)$.
f and g in both approaches can feasibly be learned by neural networks. In most previous attempts, σ was not adaptable and commonly was either the max-function, or the mean (Ilse et al. (2018)).

Concrete Baseline Implementations A simple, baseline approach would be (i) to model f in either approach using a neural network architecture that is proven to work well on image classification and related tasks, (ii) to fix σ to be one of the two common pooling operations mentioned above, and, in the embeddings-based scenario, (iii) to learn g with a fully connected layer.
To achieve the (i)st part, we choose the 'LeNet5' architecture for the `MNIST-bags` dataset (LeCun et al. (1998)) and the architecture of Wang et al. (2018) for the classical MIL datasets. These two structures are also what Ilse et al. (2018) used to evaluate their 'attention-based'-pooling. This will make it easier to compare the

results from their approach to the baseline results.
The concrete implementations of this for the classical MIL datasets and the `MNIST-bags` dataset are outlined in Table 2 and Table 3. Note that the overall used architecture in the embeddings-based approach differs from the instance-based approach in the ordering of the last two layers. Within these tables "fc-x" stands for a fully-connected layer with x output neurons, "mil-max" stands for a max-pooling layer, and mil-mean stands for a mean-pooling layer.

Layer	**Embeddings-based**	**Instance-based**
1	fc-256 + ReLU	fc-256 + ReLU
2	dropout	dropout
3	fc-128 + ReLU	fc-128 + ReLU
4	dropout	dropout
5	fc-64 + ReLU	fc-64 + ReLU
6	dropout	dropout
7	mil-max/mil-mean	fc-1 + sigm
8	fc-1 + sigm	mil-max/mil-mean

Table 2: The neural network architectures used to predict bag probabilities for the classical MIL datasets.

Layer	**Embeddings-based**	**Instance-based**
1	conv(5,1,0)-20 + ReLU	conv(5,1,0)-20 + ReLU
2	maxpool(2,2)	maxpool(2,2)
3	conv(5,1,0)-50 + ReLU	conv(5,1,0)-50 + ReLU
4	maxpool(2,2)	maxpool(2,2)
5	fc-500 + ReLU	fc-500 + ReLU
6	mil-max/mil-mean	fc-1 + sigm
7	fc-1 + sigm	mil-max/mil-mean

Table 3: The neural network architectures used to predict bag probabilities for the `MNIST-bags` dataset.

3.2 Results from the Baselines

Evaluation Metrics To evaluate how well specific sets of hyper-parameters and trained parameters for the above explained functions f and g work, we can use the standard accuracy measure, i.e. the fraction of bags from the test set the model correctly predicts.
The classical MIL datasets are small, which means a fixed train-test split should not yield a reliable estimate of the accuracy the model can achieve. Hence, Ilse et al. (2018) use 10-fold-cross-validation to make the most of the available data in this case. In this milestone, and for the sake of providing simple baseline results, we have opted for a fixed test-train split for all the evaluated datasets. In the case of `MNIST-bags` this should still yield a reliable estimate of the achievable accuracy, since the dataset is comparatively large. In the case of the other datasets, we used 80% of the available data for training, and the remaining 20% for testing.
Other metrics that can be used to evaluate the quality of a prediction include: 'precision', 'recall', 'F-Score', and 'AUC' (Powers (2020)).

- **Precision** is calculated as the fraction $\frac{\text{true\_positives}}{\text{all\_positives}}$, where 'true_positives' is the number of positive bags the model **correctly** labeled as positive (i.e., as having a label of 1), and 'all_positives' is the number of **all bags** the model predicted as positive.

- **Recall** is the fraction $\frac{\text{true\_positives}}{\text{real\_positives}}$, where 'real_positives' is the number of positive bags in the test set.
- The **F-Score** is calculated as the harmonic mean of precision and recall.
- **AUC** represents the area under the 'ROC'-curve. The ROC-curve is obtained by plotting the 'false positive rate' (FPR) against the recall of the model. The different FPR and recall values are obtained by sorting the final scores of all the bags and splitting this sorted array in different places. Values lower than the split will be classified as negative and values higher than the split as positive. For each of these splits the FPR and recall are determined and plotted against one another.

Baseline Results For the purpose of providing some baseline results that can be improved upon, we are leaving out a comparison of different hyper-parameters, and opt for the ones found to be best performing by Ilse et al. (2018). We will also not use any special initialization scheme for the parameters of the networks. All the optimization parameters are given in Table 4.

Dataset	Training Size	Test Size	Optimizer	Learning Rate	Weight Decay	Epochs
`MNIST-bags`	1000	1000	Adam	0.0005	0.0001	200
`MUSK1`	73	19	SGD	0.0005	0.005	100
`MUSK2`	81	21	SGD	0.0005	0.03	100
`TIGER`	160	40	SGD	0.0001	0.01	100
`FOX`	160	40	SGD	0.0005	0.005	100
`ELEPHANT`	160	40	SGD	0.0001	0.005	100

Table 4: The parameters used during training to obtain the baseline results for the models described in Table 2 and Table 3. These values are taken from the paper of Ilse et al. (2018). All of the models using SGD had a momentum of 0.9. The Adam Optimizer used $(\beta_1, \beta_2) = (0.9, 0.99)$.

(Dataset, Approach, Pooling-Type)	Accuracy	Precision	Recall	F-Score	AUC
(`MNIST-bags`, embeddings-based, max)	0.967	0.961	0.974	0.967	0.967
(`MNIST-bags`, instance-based, max)	0.969	0.972	0.966	0.969	0.969
(`MNIST-bags`, embeddings-based, mean)	0.964	0.955	0.974	0.964	0.964
(`FOX`, embeddings-based, max)	0.600	0.609	0.667	0.636	0.597
(`FOX`, instance-based, max)	0.650	0.667	0.667	0.667	0.649
(`FOX`, instance-based, mean)	0.550	0.560	0.667	0.609	0.544

Table 5: A selection of results from our trained baseline models.

We present a representative selection of the results from the baseline experiments in Table 5. These results were obtained by training the models on one GPU on the "Hydra"-Cluster of the "IDA"-group at TU Berlin.
We can observe that we were able to achieve comparatively high scores in terms of the tested metrics on the `MNIST-bags` dataset. The results on the classical MIL datasets are low in comparison.
This could be because the training conditions were much more favorable for the models training on the MNIST-based dataset, since they had far more training data available. These models also were allowed to train for a higher number of epochs. Since the size of the test set was also larger in this case, the estimates of the evaluation metrics should additionally be more reliable.
Another insight from the baselines is that the max-pooling seems to work better than the mean-pooling. This is true for both the instance-based approach, as well as the embeddings-based approach. At least for the instance-based approach, we deem that this makes sense. The label of a bag is determined by the maximum label of all the instances it contains. Therefore it makes sense for the model to predict the

label of the bag on the basis of its predictions for the instances in the same manner. In general it is noteworthy that instance-level classifiers seem to work better for this task. It will be an interesting question to answer, whether or not this also holds true when using the suggested attention-based pooling approach from the work of Ilse et al. (2018).

3.3 Overfitting

The classical MIL datasets are all small in comparison to dataset sizes typically necessary for training well-generalizing neural networks (Liu et al. (2017)). Here, overfitting of the models is very likely if no form of regularization is employed. To prevent this, dropout layers are used at various stages of the computational graph of the network.
Data augmentation is not easily possible for these datasets, since they consist of abstract, pre-computed features. For these it is not entirely clear, how to augment them, without possibly significantly changing the distributions of positive and negative instances within the bags.
The `MNIST-bags` dataset is large enough to make overfitting less of an issue than for the classical MIL datasets. Therefore, no form of regularization is used.

3.4 Predicting the Different Categories

Since a positive bag of size K can contain 1 to K positive instances, the task of identifying what constitutes a positive bag is far more ambiguous than identifying what constitutes a negative bag (Andrews et al. (2002)).
To learn the distinction between positive and negative bags more clearly, it is helpful to have a roughly equal number of positive and negative bags (Cardie and Howe (1997)). From Figure 1 we can see that this is the case for almost all of the classical MIL datasets. The only exception is formed by the `MUSK2` dataset. This could lead to the classification task being more difficult to solve on this dataset; a question we will be investigating in subsequent milestones.
From the creation scheme of the `MNIST-bags` dataset (outlined in Section 2.1.1) a roughly even distribution of positive and negative bags follows. This should lead to the models trained on the MNIST-based dataset to be able to predict both classes equally well.

4 Discussion

4.1 Progress

In this first Milestone we have achieved the implementation of data loaders and visualization tools, as well as a model prototype with a training script. We implemented a data loader for the classical MIL datasets and one for `MNIST-bags`.
Considering that the classical MIL datasets contain pre-computed features, we did not find it useful to implement a visualization tool for them. For the MNIST-based dataset our tool accepts a limited number of bags given as a list, which is saved to a file `outfile` when specified.
As before, we have to differentiate between the classical MIL datasets and the `MNIST-bags` dataset in the implementation of our model. The classical MIL model requires the dataset, approach (embedding- or instance-based), and pooling type to be specified. The chosen approach determines the architecture of the model as shown in Table (2). The model for `MNIST-bags` requires the approach and pooling type to be specified. Its architecture is shown in Table 3. Lastly we have also implemented a training script, which provides the results displayed in Table 5.

4.2 Challenges

The primary challenge with the classical MIL datasets lies in the small amount of samples. These datasets, composed of vector embeddings with pre-extracted features, pose a unique challenge for data augmentation strategies. Conversely, the `MNIST-bags` dataset presents an abundance of training data. However, the absence of pre-computed features introduces complexity, creating a more difficult task for our model. The histopathology datasets fall in between, grappling with a shortage of data samples, partly solvable through data augmentation.

4.3 Explainability

A distinguishing feature of the attention-based model is its interpretability. The attention mechanism enables the identification of key instances, as exemplified in Figure 3. Ideally, the model should extend this capability to highlight approximate regions of interest (ROIs) in histopathological datasets, as interpretability is indispensable for medical diagnoses.
The attention mechanism achieves this by assigning high attention weights to instances likely to have a positive label. While distinct from the instance-based classifier that assigns scores to individual instances, it can be seen as a proxy to it.

4.4 Goals

Considering data augmentation for histopathology datasets, we anticipate training a model capable of delivering high performance while providing explanatory insights through the attention mechanism. This could prove helpful for offering a reliable secondary diagnosis, potentially identifying conditions that may elude human observation.
The implementation of the model on histopathology datasets remains a pending goal of significant interest, given its potential impact. Thus, evaluating the effectiveness of the attention mechanism in providing explanations for the model's results on histopathological datasets is an essential next step.
We would also like to test different hyper-parameters, as done in the original paper by Ilse et al. (2018). So far, we have also not trained and tested the models relying on the attention mechanism. This will be a crucial part of the coming milestones.

References

S. Andrews, I. Tsochantaridis, and T. Hofmann. Support vector machines for multiple-instance learning. *Advances in neural information processing systems*, 15, 2002.

C. Cardie and N. Howe. Improving minority class prediction using case-specific feature weights. In *Proceedings of the Fourteenth International Conference on Machine Learning*, 1997.

T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. *Artificial intelligence*, 89(1-2):31–71, 1997.

M. Ilse, J. Tomczak, and M. Welling. Attention-based deep multiple instance learning. In *International conference on machine learning*, pages 2127–2136. PMLR, 2018. URL `https://arxiv.org/abs/1802.04712`.

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

B. Liu, Y. Wei, Y. Zhang, and Q. Yang. Deep neural networks for high dimension, low sample size data. In *IJCAI*, pages 2287–2293, 2017.

D. M. Powers. Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. *arXiv preprint arXiv:2010.16061*, 2020.

X. Wang, Y. Yan, P. Tang, X. Bai, and W. Liu. Revisiting multiple instance neural networks. *Pattern Recognition*, 74:15–24, 2018.