MLPC 2025 Task 4: The Challenge

Tara Jadidi, Florian Schmid and Paul Primus

May 2025

Context

Kepler Intelligent Audio Labs (KIAL), a soon-to-be-founded innovative AI startup, aims to collect a general-purpose dataset with strong temporal annotations to train sound event detection systems for their customer (see Figure 1 for a schematic illustration of sound event detection).

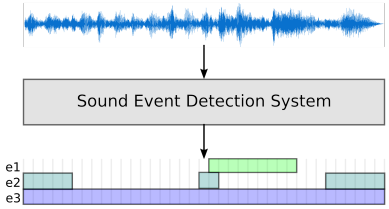

Figure 1: **Sound Event Detection (SED)** systems take an audio recording as input and predict acoustic events of interest $(e1, e2, e3)$ *including* their temporal onsets and offsets.

To make this data set reusable for many potential future customers, KIAL decides to annotate sounds with free text annotations instead of relying on a fixed set of categories (see Figure 2). For a detailed description of the project, look at the project description on our Moodle page.

Figure 2: *Left:* Typically, SED data sets are only annotated for a fixed set of labels $(e1, e2, e3)$. *Right:* Instead of relying on a fixed set of labels, KIAL decided to use arbitrary free text descriptions. Text annotations can be mapped to labels of interest $(e1, e2, e3)$ by matching synonyms, using word embedding models, or leveraging LLMs.

KIAL has divided the project into these five stages:

1. **Data Collection:** Collect a large dataset of audio recordings, containing a large variety of potential target sounds and events. (✓)
2. **Data Annotation:** Once the data is gathered, human annotators carefully annotate sounds in the recordings with textual descriptions and temporal onsets and offsets. (✓)
3. **Data Analysis:** The dataset undergoes a thorough examination through exploratory data analysis, allowing for a deeper understanding of its characteristics and potential challenges before further processing. (✓)
4. **Model Training & Selection:** KIAL's R&D team needs to demonstrate the usefulness of the new approach to the managers by *training machine learning models* for one or multiple fictitious customers (each customer is represented by a fixed set of labels). (✓)

5. **Challenge:** A customer requests a sound event detection system for a fixed set of labels. Provide predictions on a hidden test set to prove to the customer that the developed system has a high detection performance and win an (also fictitious) contract. (← we are here)

KIAL has already collected the dataset by scraping publicly available audio recordings on the web (step 1) and annotated the recordings with textual annotations (step 2). Additionally, they conducted a thorough analysis of the dataset (step 3) and laid the groundwork for training the sound event detectors (step 4).

Task Outline: Challenge (30 points + 10 bonus points)

Deadline: Thursday, June 18th, 23:59
Team Restructering Deadline: Friday, May 30th, 23:59
Submission: Submit your team's report, slides and the predictions via Moodle. Only **one** submission per team is allowed. The final submission via Moodle counts.
Group Work: Coordinate early with your team to distribute tasks, set a schedule, and define deadlines. *Check in regularly* and plan buffer time to assist teammates if needed. Inform us promptly if a team member is unresponsive.

This task is mandatory. All team members are required to contribute. Content and formal requirements are outlined below.

1 Overview

A (fictitious) customer is interested in monitoring environmental noise in an urban environment to study noise pollution patterns. To this end, the customer defined ten sound event categories of interest, which will be the prediction targets for this phase of the project. The primary objective will be to use the classifiers developed in the previous project phase to *detect the temporal occurrence* of the noise events.

To find the system with the highest detection performance, the customer provides a set of audio recordings without annotations (i.e., a secret test set; available on Moodle). Developed system are expected to predict the absence or presence of the 10 event classes for 1.2-second segments (without overlap between the segments). The resulting predictions for all 1.2-second segments of all test files must then be submitted in a single CSV file, which has the following columns:

```
filename,onset,Speech,Dog Bark,Rooster Crow,Shout,Lawn Mower,Chainsaw,Jackhammer,
Power Drill,Horn Honk,Siren
```

- `filename`: name of the audio file (e.g., `1922.mp3`)
- `onset`: the start time (in seconds) of corresponding 1.2-second segment; first onset of a file is 0, second onset is 1.2, etc.; Note that since the audio files are of varying length, the last segment is probably shorter than 1.2 seconds.
- `speech, ..., siren`: binary indicators for each of the ten acoustic classes, where `1` indicates presence and `0` indicates absence; the order of the class columns does not matter
- Each row provides the file name, the onset of the 1.2-second segment and 10 predictions.

The customer is interested in establishing a lower bound of the noise pollution in certain areas; for their investigation, false positives are less critical than false negatives, and mechanical sounds are more interesting than human or animal sounds. The performance will therefore be assessed using a custom cost-based evaluation metric that aligns with the client's goals. Specifically, for each sound category, the customer defined costs for correct and incorrect classifications, which are outlined in Table 1. The total cost on the test set is computed separately for each class, normalized by the data set size to obtain the cost per 1 minute, and summed to get a single score for ranking. To secure the (fictitious) contract, your system must demonstrate lower cost compared to your competitors' systems.

Table 1: Cost Matrix for Evaluation

Class	TP	FP	TN	FN
Speech	0	1	0	5
Dog Bark	0	1	0	5
Rooster Crow	0	1	0	5
Shout	0	2	0	10
Lawn Mower	0	3	0	15
Chainsaw	0	3	0	15
Jackhammer	0	3	0	15
Power Drill	0	3	0	15
Horn Honk	0	3	0	15
Siren	0	3	0	15

1.1 Evaluation Script

Use our Python evaluation script `compute_cost.py` (available on Moodle) to *check the format of your upload file* and *to compute the cost of your model's predictions on a custom test split.*

Usage: To asses the cost for a custom test set, first create the ground truth CSV file and the predictions CSV file by calling `get_ground_truth_df` and `get_segment_prediction_df` in your implementation., e.g.,

```
from compute_cost import get_ground_truth_df, get_segment_prediction_df
...
get_ground_truth_df(
    list_of_files_in_custom_test_split,     # ['123.wav', ...
    path_to_the_development_set             # 'path/to/MLPC2025_dataset'
).to_csv("ground_truth.csv", index=False)

get_segment_prediction_df(
    dictionary_containing_model_outputs     # {'123.wav: {'Siren': [0,1,1,1 ....
    class_names                             # ['Speech', 'Dog Bark', ...
).to_csv("predictions.csv", index=False)
```

This essentially converts the labels and the predictions of selected files in the development set into labels for 1.2-second segments and stores them in a single CSV called `ground_truth.csv`.

You can the use the following command to compute the total cost:

```
python compute_cost.py \
--dataset_path=path/to/MLPC2025_dataset \
--ground_truth_csv=path/to/ground_truth.csv \
--predictions_csv=path/to/predictions.csv
```

- `dataset_path`: Path to the root directory of the dataset; this folder must contain 'audio_features' to check the expected number of 1.2-second segments.
- `ground_truth_csv`: Path to the CSV file containing the ground truth labels. Format is described above.
- `predictions_csv`: Path to the CSV file containing the binary predictions. Format is described above.

To check the format of your predictions CSV file, run:

```
python compute_cost.py \
--dataset_path=path/to/MLPC2025_dataset \
--predictions_csv=path/to/predictions.csv
```

2 Report (max. 27 points)

For the first part of this assignment, you will again have to write a report based on the template provided on Moodle. Your report needs to cover *all of the following points*:

1. Describe your *evaluation setup*.
   (a) How did you split the development dataset (provided in phase three of the project) into a training and evaluation set while avoiding information leakage?
   (b) Establish and describe a naive baseline system (i.e., a baseline without a classifier). How much cost can we expect from it?
2. *Build a simple SED system*, that predicts the presence and absence of the ten sound categories in 1.2-second audio segments. **Hint**: You may re-use a classifier from the previous phase of the project. Your systems from the previous project phase were most likely trained to predict the presence or absence of sound events for 120ms frames; in this case, you will have to aggregate your predictions to obtain a single prediction for 1.2-second segments. For this, you can modify the `get_segment_prediction_df` function provided in `compute_cost.py`.
   (a) How did you threshold and combine predictions to derive a label for the 1.2-second segments? Describe any heuristics or post-processing strategies used.
   (b) What strategies did you apply to minimize the task-specific cost function?
   (c) Does your classifier-based SED system achieve lower costs compared to the naive baseline system you established in step 1? If not, describe possible issues.
3. Investigate at least *three diverse strategies to improve your starting point* (e.g., via hyperparameter tuning, ensembling, data augmentation, cost-specific tuning, post-processing, etc.)
   (a) For each of these strategies: Describe the main ideas you had and hypotheses you tried, as well as their outcome after experimentally verifying or falsifying the hypotheses (via validation on (parts of) the development set). Do not hesitate to report negative outcomes.
4. *Do you think your final system could be deployed in a real-world application?* In your opinion, which aspects of the project or your system would need to be adapted to fulfill possible real-world requirements?
5. *Bonus (up to 10 points)*: So far, we've worked with audio embeddings, that were created from a pre-trained and frozen sound event detection model. To receive the bonus points, train one of our pre-trained sound event detection models on the MLPC dataset to predict the ten sound categories of interest in an end-to-end manner. You can find an implementation, a detailed documentation and pre-trained checkpoints here: https://github.com/fschmid56/PretrainedSED. Use one of the checkpoint that was pre-trained on AudioSet-strong as a starting point. You can use any of the model architectures. Compare the performance of the resulting system to the best model from the previous section.

The report **must not exceed a page limit of 6 pages**, of which **in total at most 4 pages should be text**. In addition to addressing these questions above, **add a statement of the contributions of all team members** as indicated in the template. If you address the bonus questions, you can use up to 7 pages with 5 pages max for the text.

In addition to the report, submit a CSV-file containing your predictions. The expected format is described in Section 1. Not submitting the predictions or submitting them in an incorrect format will deduct up to 10 points.

3 Slide Set (3 points)

As always, create a *short complementary slide set* that could be used to present your results clearly and concisely to your fellow course participants. For this project phase, your slide set should contain a **description of the general architecture of your system** and the **most interesting hypotheses you investigated and their outcomes**. The **upper limit for the number of slides you should prepare is 6** (excluding an additional title slide that should contain your group name and the member names).

4 Grading

Each of the six topics given in the task outline above will be evaluated according to the following criteria:

- Thoroughness & Completeness: Have you thought about the problem, and answered every question?
- Clarity: Are the ideas, features, algorithms, and results described clearly? Based on your descriptions, could the reader reconstruct your experiments?
- Presentation: Did you select an appropriate way of communicating your results, e.g., did you use meaningful plots where helpful?
- Correctness: Is the proposed procedure/experiment sound, correct?
- Punctuality: The reports must be submitted in time. Any delay will result in reduced grades. Specifically, submitting on the day after the deadline will deduct 1/3 of the points, submitting on the second day after the deadline will deduct 2/3 of the points; submissions on the third day after the deadline or later will be rejected.

You will be awarded all points for the slide set if it addresses the assigned topic, is within the slide limit, and is submitted before the deadline. You will receive all points for the predictions if the file is correctly formatted and submitted before the deadline.

5 Summary

- Completing Task 4 is a requirement to pass this course.
- Look at the given questions and answer all of them appropriately in a written report. Make sure to use the report template provided to you via Moodle. Adhere to the given page limit (max. 6 pages where at most 4 pages can be text) and include a statement about the contributions of all team members.
- Create a set of slides describing your system and interesting findings. Make sure to adhere to the slide limit for this step as well (max. 6 + 1 title slide).
- Create predictions for the test set as described in Section 1; check your upload file with the evaluation script.
- Upload the written report as well as your predictions and the slides to Moodle before the deadline.
- You will get a maximum number of 27 points for your written report and the predictions, and 3 points for the slide set.