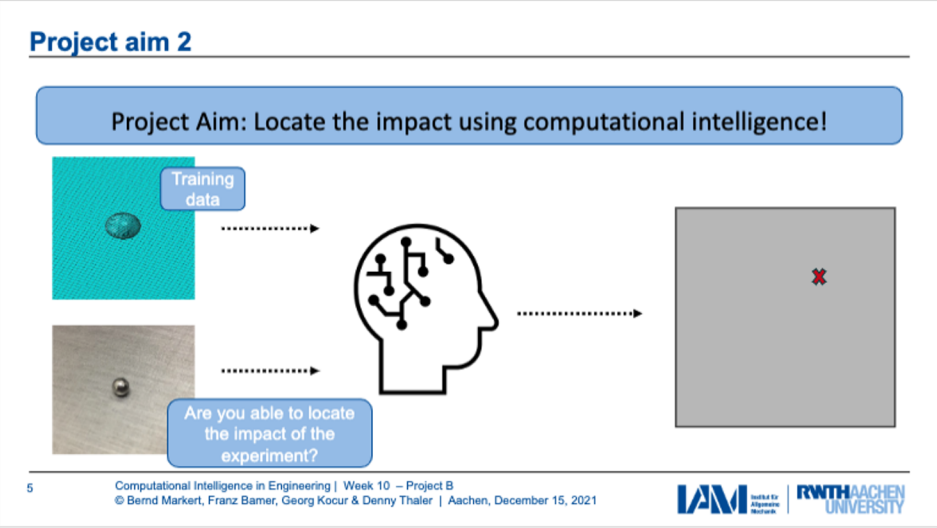
Project B

Group 5

Institut für Allgemeine Mechanik
RWTH AACHEN UNIVERSITY

Motivation and Aim



Data

- Numerical Data
  - EPOT data “EPOT_X_Y.mat”
  - Validation_augmented_data
- Experimental Data
  - Experimental_data “impact_xy_###.txt” (group5)
  - Experimental_validation “impact_xy_X_Y_run_###.txt”

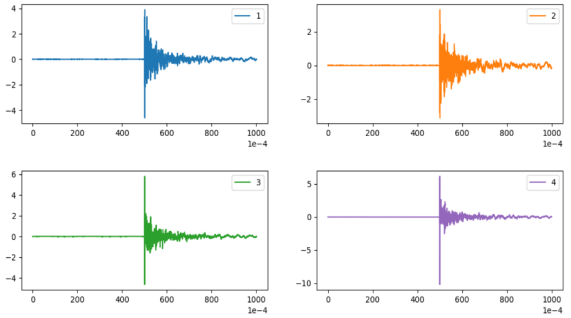
Team Introduction

Responsibilities

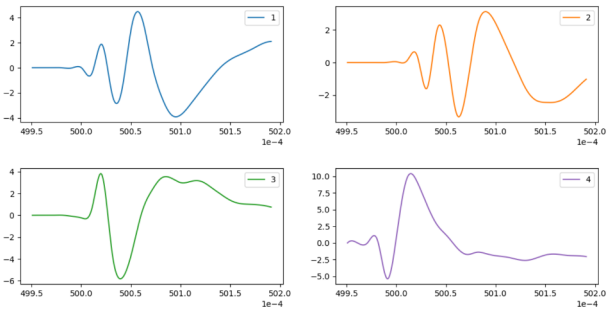
- Alejandro Delgadillo, Data augmentation
- Jinyang Yu, Data preprocessing
- Nayoung Ahn, Data preprocessing
- Shuteng Wang, Neural network model
- Xiaoyang Liu, Neural network model

Data Preprocessing - Summary

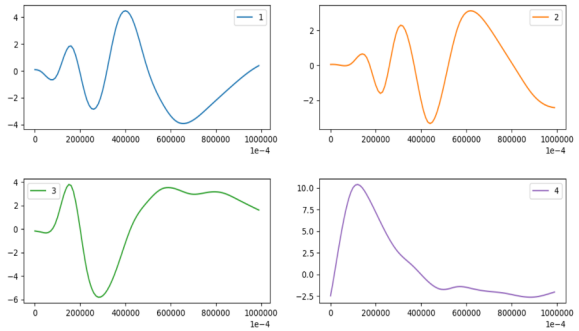
Read data



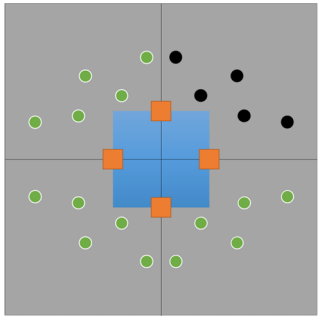
Cut exp. data



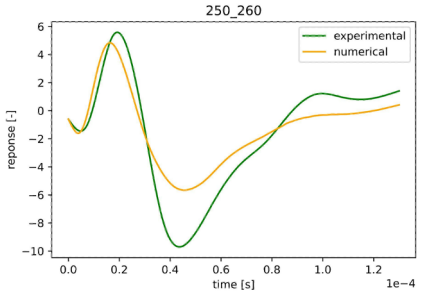
Resample exp. data



Augmentation of num. data



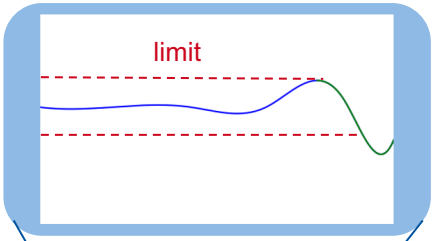
Align num. and exp. data



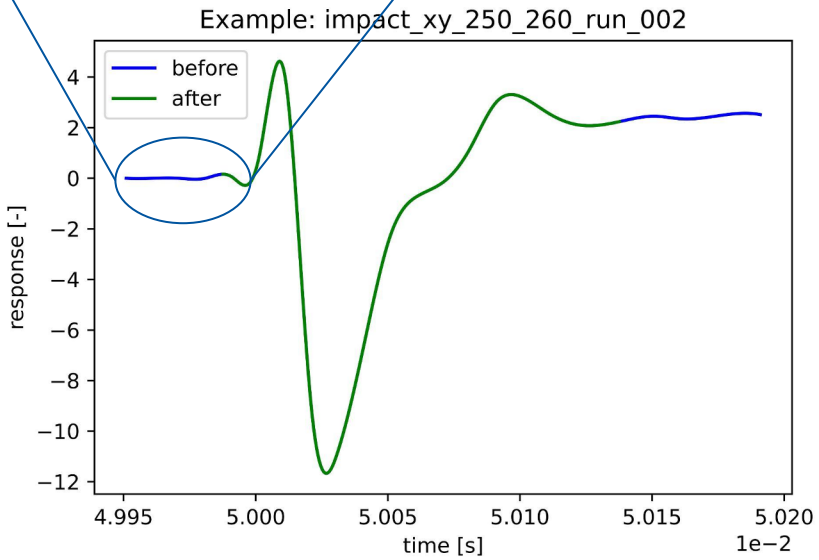
Generate dataset

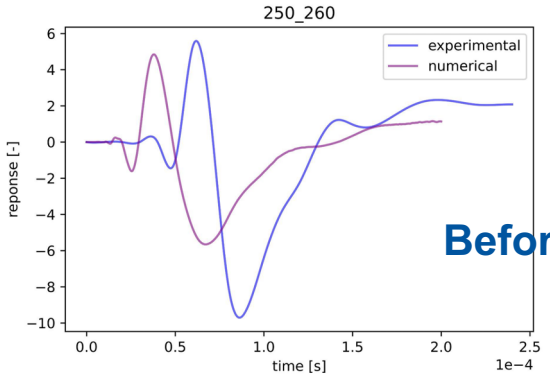
Numerical Data	[244, 100,4]
Augmented Data	[976,100,4]
Experimental Data	[24,100,4]
Prediction Data	[18,100,4]

Data Preprocessing: Alignment of Numerical and Experimental Data

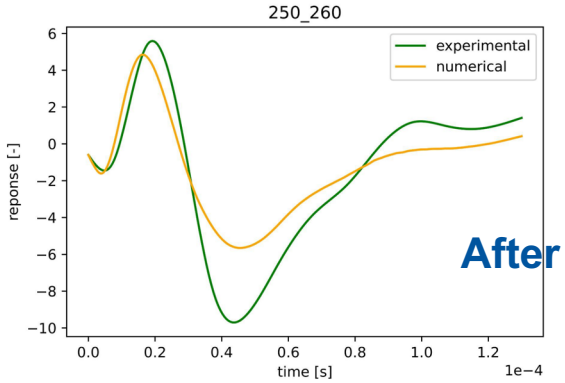


Remove the first part of |y|<limit to obtain the main shape of the impact wave



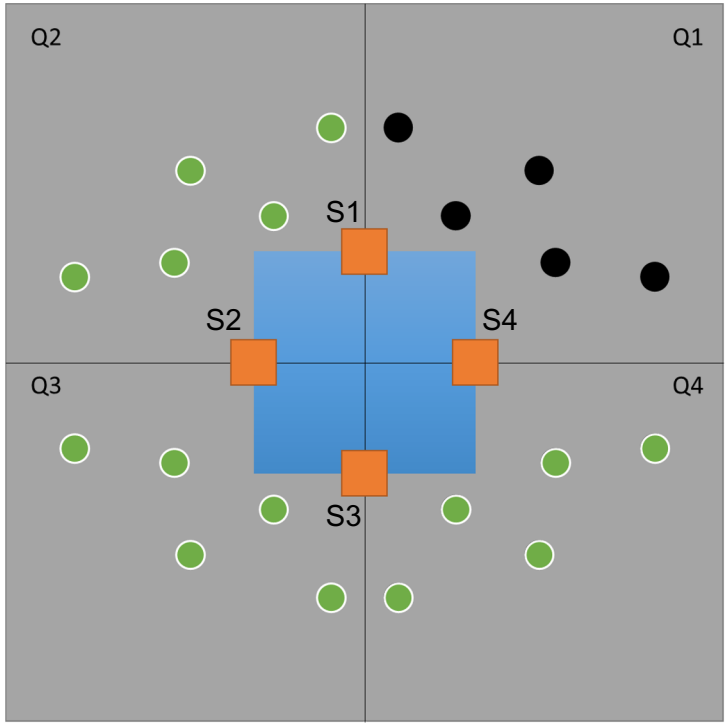


Before alignment



After alignment

Data Preprocessing: Augmentation of Numerical Data



Generated data for collisions taking place in quadrants 2, 3 and 4 by rearranging the input sensor input channels

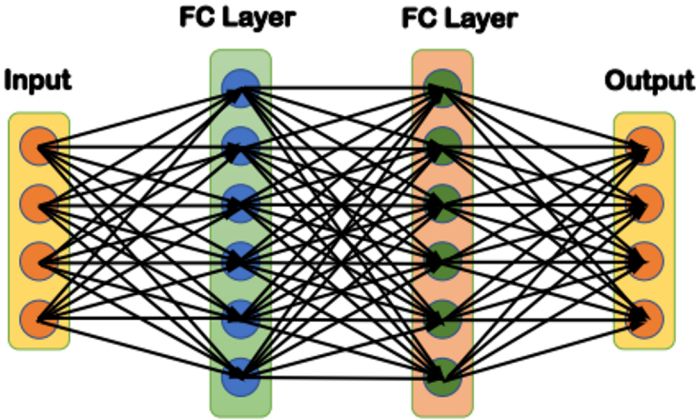
Sensor signal arrangement for data augmentation

	Ch.1	Ch.2	Ch.3	Ch.4
Q1	S1	S2	S3	S4
Q2	S1	S4	S3	S2
Q3	S3	S4	S1	S2
Q4	S3	S2	S1	S4

Neural Network - Summary

1. Load preprocessed data ".pickle"
2. Split numerical data into training, validation, test set (8:1:1)
3. Normalize input and/or target
4. Determine and implement calculation of error metrics
5. Implement baseline NN architecture (FCN/CNN/RNN(LSTM/GRU))
    a. Tune the architecture
    b. Tune hyperparameters
6. Repeat above steps with augmented numerical data

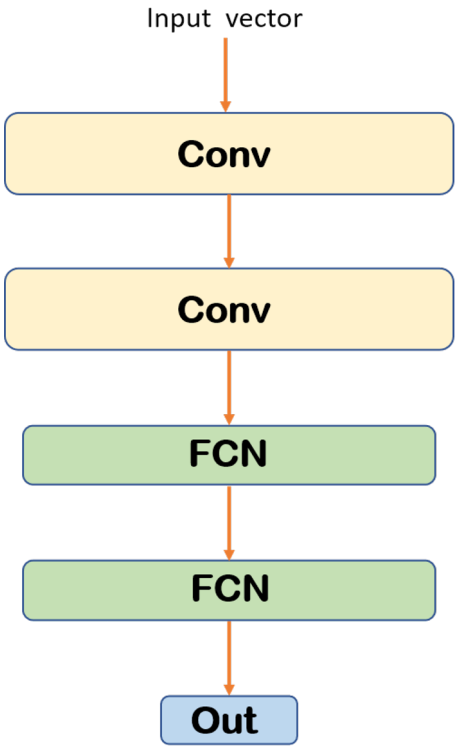
Neural Network: FCN Architecture



Tab1. FCN architecture and parameters

Layers	Parameters
Input	Input shape = (100,4)
FC layer	128 units, ReLU activation, L2 regularizer
FC layer	64 units, ReLU activation, L2 regularizer
Output	2 units, tanh activation, MSE loss

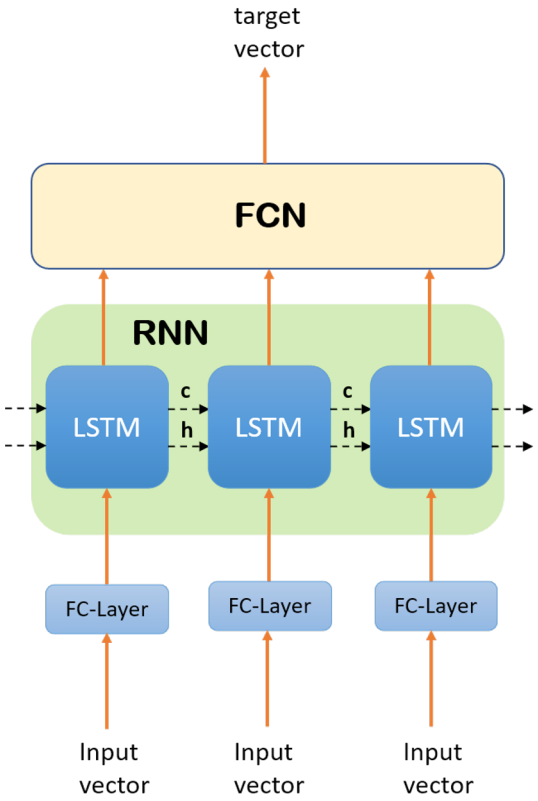
Neural Network: CNN Architecture



Tab2. CNN architecture and parameters

Layers	Parameters
Input	Input shape = (100,4)
Conv layer	Conv kernel = (50,2), depth = 8, ReLU activation
Max Pooling	Pooling kernel = (3,1)
Conv layer	Conv kernel = (10,2), depth = 8, ReLU activation
FC layer	Units = 32, ReLU activation, L2 regularizer
FC layer	Units = 8, ReLU activation, L2 regularizer
Output	Units = 2, tanh activation, MSE loss

Neural Network: RNN Architecture



Tab3. RNN architecture and parameters

Layers	Parameters
Input	Input shape = (20,4)
FC layer	8 units, ReLU activation, L2 regularizer
LSTM (GRU)	64 units, ReLU activation
FC layer	32 units, ReLU activation, L2 regularizer
FC layer	16 units, ReLU activation, L2 regularizer
Output	2 units, tanh activation, MSLE loss

First Run: Train, Validate and Test on Numerical Data (without augmentation)

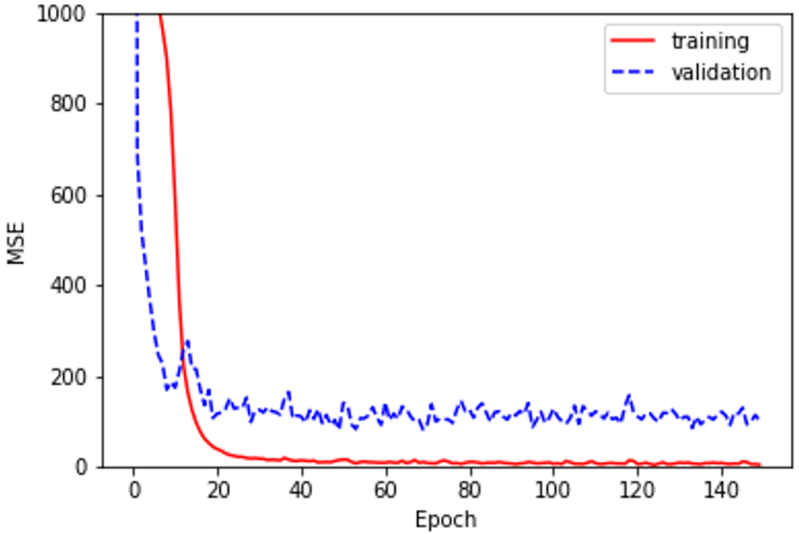
Tab4. MAE for different models on training, validation and test set

Model	Training	Validation	Test
FCN	3.12	16.09	23.57
CNN	1.52	9.75	19.16
RNN (LSTM)	4.08	11.25	17.37
RNN (GRU)	5.39	12.36	16.42

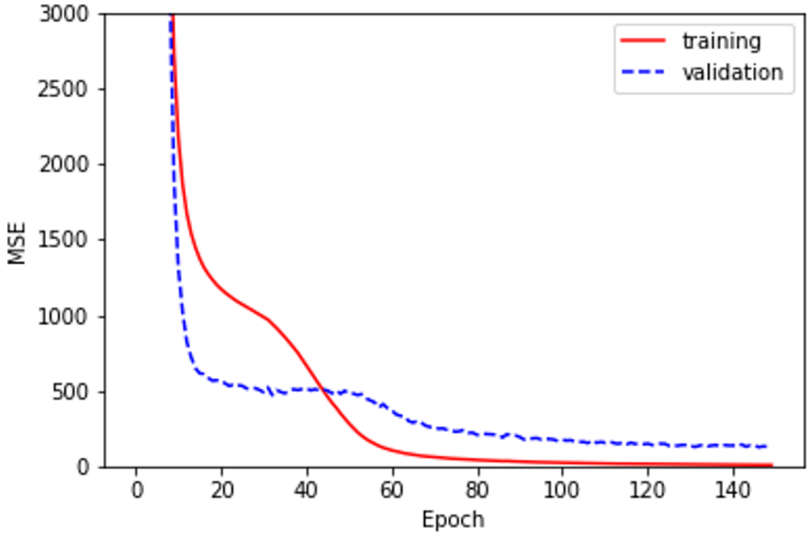
Implementation details

- Data normalization
- Training set used for training
  Validation set used for parameters tuning
  Test set used for performance test on numerical data.
- MSE loss functions for CNN and FCN
  MSLE loss functions for RNN
  MAE loss functions for model evaluation
- Adam optimizer used as default choice
- Data shuffling used for errors reduction

Neural Network: Losses of Training and Validation for Different Batch Sizes



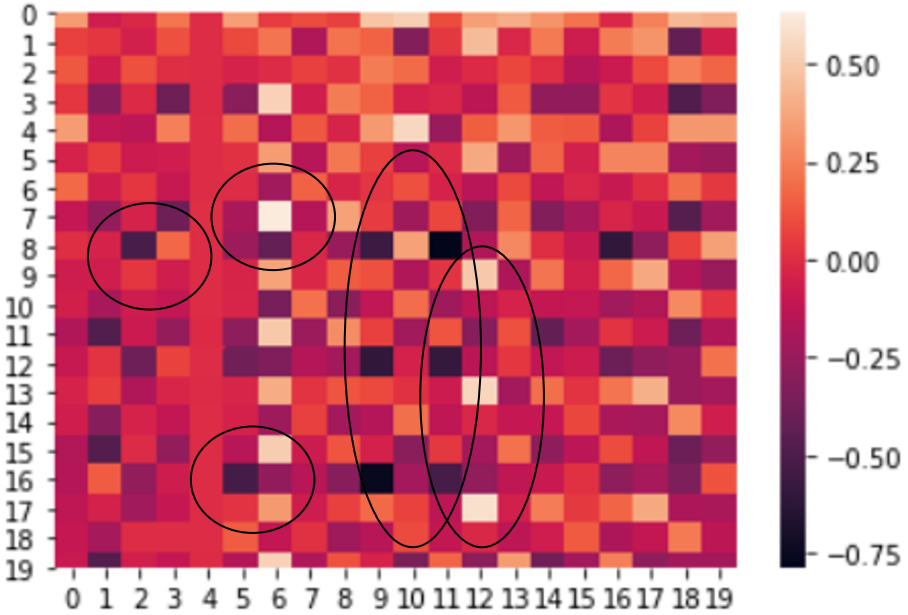
Training loss and validation loss all in mean squared error for batch size = 5



Training loss and validation loss all in mean squared error for batch size = 40

For small batch sizes, training and validation loss converge more quickly but in an unstable way

Neural Network: Automatic Feature Engineering



Heatmap of a part of the 1st layer's weights

- The first few layers would be responsible for the feature extraction
- The white and black spots show the positions where the network thought as useful features

```
fig = sns.heatmap((model.layers[1].get_weights()[0])[0:20,0:20])
plt.show(fig)
plt.savefig("heat_map.png")
```

Second Run: Train on Augmented Num. Data and Test on Experimental Data

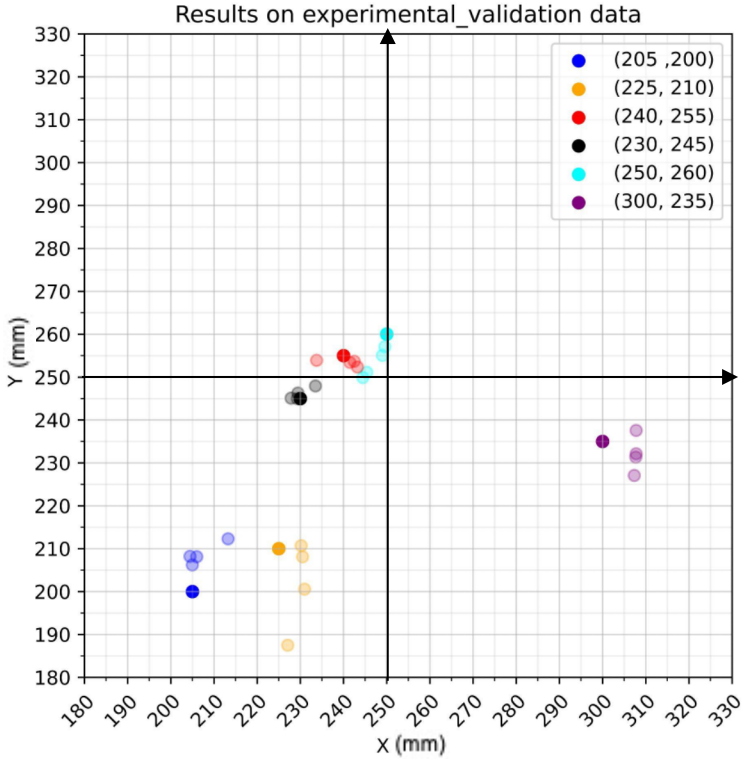
Tab5. MAE for different models on experimental data

Model	MAE
FCN	9.67
CNN	17.85
RNN (LSTM)	4.78
RNN (GRU)	4.48

Data Augmentation

- use the whole numerical dataset after augmentation as training set
- This greatly improved the results on experimental dataset.

Visualization of GRU Predictions on Experimental Validation Data



Observation

- Center is (250, 250)
- Impacts closer to the sensors are predicted better than those far away.

Final Prediction

Final Prediction to the Experimental Data given to Group 5

```
[[245.59564 250.28908]
 [261.75955 220.44699]
 [233.2645  234.77596]
 [230.66547 248.74158]
 [249.03458 256.65952]
 [247.60269 246.97462]
 [186.9534  224.94916]
 [224.73698 212.21803]
 [255.09892 298.44788]
 [247.69572 244.09216]
 [255.3009  228.51135]
 [230.11415 308.94138]
 [261.03564 245.68115]
 [230.63318 231.75743]
 [247.78885 254.24402]
 [259.27243 309.27954]
 [243.00551 253.67584]
 [283.48468 284.2997 ]]
```

Conclusion

Data Preprocessing

- Filtering was not necessary for this project
- Setting limit value of 0.2 yields best performance (grid search)
- Augmented data greatly improves prediction on experimental validation data

Neural Network

- Best prediction was achieved by the GRU model
- Impacts closer to the sensors are predicted better than those far away

Discussion and Possible Extensions

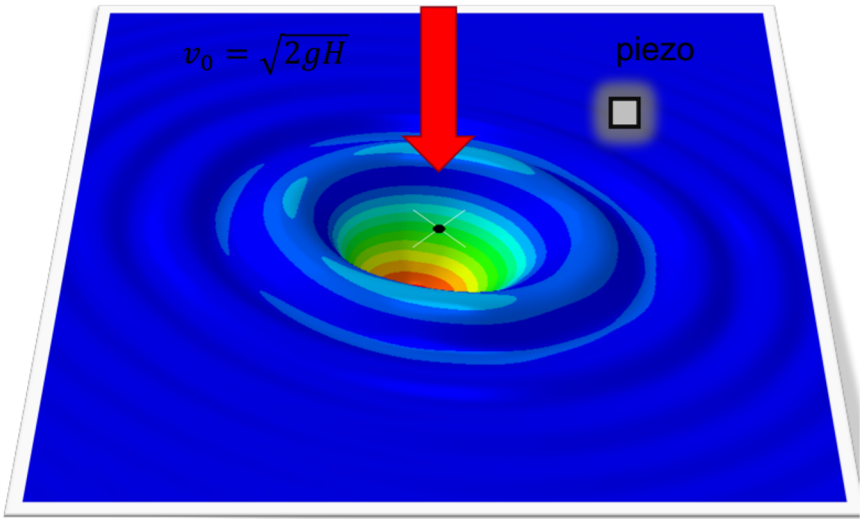
Data Preprocessing

- Align both numerical and experimental data along time and frequency
- Analysis of the data obtained from repeated experiments may be used to further augment the training data by incorporating the uncertainty observed in the experimental data to the numerical data

Neural Network

- RNN can extract information along time series, which is suitable for this task
- GRU compared to LSTM, has a simpler structure and fewer parameters to reduce overfitting and get better results on validation set
- Implement different combination of architectures e.g. CNN+LSTM

Discussion: Possible Experimental Extension



- In addition to the falling position, we could also predict the height of the falling ball. Since the height determines the terminal velocity just before the ball reaches the plate, it would influence the time that the sensors react to the wave and possibly the wave's shape etc.
- The temperature, moisture and plate materials should also be able to be predicted since they would impact the wave propagation behaviour.

Questions?

Thank you for your attention!