Abstract:

Chapter 1. Introduction:

- Context and motivation
- Problem statement
- Research questions
- Objectives
- Contributions
- Structure of the thesis

Chapter 2. Related work:

At least two pages + table

Chapter 3. Background:

- V-Model-based testing phases (MIL, SIL, PIL, HIL and VIL)
- Hardware in the loop simulation (testing process, architecture, dSPCAE tools)
- Faults in automotive software systems, i.e., sensor signals (types, causes, effects)
- Fault detection and classification of simultaneous faults in time series data (current approaches and limitations, i.e., model-based, signals-based, knowledge-based and data-driven approach)
- Describe Deep learning/Machine Learning techniques for fault detection and classification, more details about the **used** technique in your thesis, e.g., DAE.

Chapter 4. Methodology:

Architecture of the proposed methodology including the entire phases <u>**in detail**</u>

Chapter 5. Implementation and setup:

- Case study description (ASM vehicle dynamic model with digital test drive) briefly
- Data collection including Setup of the model executions on HIL and faults injection briefly
- Data description **in detail**
- machine learning-based fault classification development, step by step according to flow chart (pre-processing phase, training phase, validation phase, testing phase) **in detail**

Chapter 6. Result and discussion:

As a result of the implementation, please consider the following points as a requirement for the fulfilment of your work:

Phase 1: unknown fault detection:

4 variants of DAE should be trained using **only healthy data** under different noise levels, i.e., GRU-DAE, CNN-DAE, CLSTM-DAE and ANN-DAE.

1. **Training process:** model performance during the training process under different hyperparameters (LR, epoch, layers, latent size..) should be documented. In particular, the reconstruction error of the model should be calculated under different training trials until the best model with optimal hyperparameters is found.
2. **Testing process1**: Fault detection performance of GRU-DAE (reconstruction error, recall, precision and F1 score) should be calculated and compared with other variants, e.g., CNN-DAE, CLSTM-DAE and ANN-DAE under different thresholds.
3. **Testing process2**: Fault detection performance of GRU-DAE (reconstruction error, recall, precision and F1 score) should be calculated and compared with other variants, e.g., CNN-DAE, CLSTM-DAE and ANN-DAE under different level of noise.
4. **Testing process3**: Fault detection performance of GRU-DAE (reconstruction error, recall, precision and F1 score) using testing data from different driving scenarios should be calculated.

Phase 2: fault classification:

1. **Training process:** model performance during the training process against different hyperparameters (LR, epoch, layers) should be documents. Specifically, the training and validation accuracy of the model should be calculated under different training trials until the best model with optimal hyperparameters is found.
2. **Testing process1**: Fault classification performance of trained ensemble model (recall, precision and F1 score) should be calculated and compared with other traditional classifier, e.g., CNN, SVM and LSTM under single and concurrent faults.

3. **Testing process2**: Fault classification performance of trained ensemble model (recall, precision and F1 score) should be calculated and compared with other traditional classifier, e.g., CNN, SVM and LSTM under different level of fault combinations (2, 3,4).
4. **Testing process3**: Fault classification performance of ensemble model (recall, precision and F1 score) should be analysed under different data size (number of input variables, number of training samples)

Note: the training and testing time of each model should be documented.

The results can be presented using tables, confusion matrix, AUC-ROC curve, feature visualisation and signal diagrams.

Chapter 7. Conclusion and future work