Intelligent Software Engineering
Coursework Specification

Dr. Tao Chen and Dr. Rami Bahsoon

This coursework represents 70% of the total assessment of the module. As part of the coursework, you can choose one project from two **alternative** types to work on:

- **Tool Building Project:** This is an individual project that asks you to build an intelligent tool for software engineering and summarise the outcome in a short report. The actual topic and programming language are up to you, though we have provided recommended concrete examples from the labs.
- **Comparatative and Analytical Project:** This is an individual project that seeks to produce an analytical survey of various techniques for an intelligent software engineering-related problem. The actual problem is free of your choice.

Please read the details in the corresponding sections below. **Note that you can only choose one type of project**.

1 Tool Building Project

1.1 The Task

For this type of project, you will propose, specify, and develop an intelligent software engineering tool that can (hopefully) beat one of the baselines for the chosen problem (at least on one metric). This will involve programming, but the chosen programming language is entirely up to you, despite that using Python or Java is highly recommended.

Students will work on their own. Every student is required to systematically report the details and design rationales of the tool. The proposed tool will have to be in accordance with the Code of Ethics from the university; see `https://www.birmingham.ac.uk/documents/university/legal/code-of-ethics.pdf` for more detailed information. Therefore, the tool should **not** involve any of the following:

- Experiments with human participants
- Activity falling under the Human Tissues Act
- Funding by philanthropic gifts
- Military applications
- Animal testing
- Possible conflict with ethical principles partially or wholly outside the above.

1.2 The Problem

You are free to work on any problem related to intelligent software engineering, either coming from the lecture content or beyond. However, you only need to work on **ONE** problem. As part of the labs, we have provided some examples of problems with example code and baseline for you to choose from, these are:

- **Bug Report Classification**: the relevant description and code can be found under *Lab 1*. The baseline is Naive Bayes + TF-IDF.
- **Configuration Performance Learning**: the relevant description and code can be found under *Lab 2*. The baseline is Linear Regression.
- **Configuration Performance Tuning**: the relevant description and code can be found under *Lab 3*. The baseline is a Random Search.
- **AI Model Fairness Testing**: the relevant description and code can be found under *Lab 4*. The baseline is a Random Search.

In addition to those, **you are free to investigate other problems in intelligent software engineering mentioned in the lectures or even those that we have not mentioned**, but you would have to research the baselines/code during your own independent study.

1.3 Deliverable

The deliverable will mainly be a systematic report that documents your solution and proof to demonstrate its effectiveness. You do not need to submit the code, but it should be stored in a public repository with a link being inserted into the report. Specifically, the report should at least have the following sections (feel free to change their names or add sub-sections therein):

- **Introduction**: This section should provide background information about the problem you address, including some discussion as to why you have chosen this problem.
- **Related Work**: This section should provide some discussion about the existing approaches for solving the problem. You should also provide some discussion of their pros and cons. Common scientific referencing styles for Computer Science, i.e., the Chicago reference, should be used here. Details can be found here: `https://pitt.libguides.com/citationhelp/ieee`.
- **Solution**: You can articulate the proposed solution here. While we encourage you to think about your own idea, it is also fine if you want to re-implement an approach from existing work and see if it works for the systems/projects considered[1]. However, you need to discuss the designs in detail, including any rationales behind them, using your own words. You are welcome to design a GUI for it, but the most important part is the core algorithm that addresses the problem of intelligent software engineering.
- **Setup**: This section needs to include the experimental setup and procedure, such as some brief discussion of systems/projects, parameter setting, evaluation metrics, what baseline to compare, and statistical analysis used.
- **Experiments**: Here, you should quantitatively assess your solution and ideally compare it with a baseline. Some figures and/or tables are necessary. Remember to use the statistical tests introduced in the module. You would also need to consider the diverse scenarios under which the solutions are assessed, e.g., different systems/projects, metrics, and or objectives. At least, you need at least one system/project, one metric, and one objective (if applicable). As a rule of thumb, the more you consider, the more convincing your conclusion will be. In the discussion, you should talk about the observations from the results.
- **Reflection**: What are the limitations of your proposed approach? Discuss how it might be better improved here.
- **Conclusion**: Present your conclusion deriving from the observed experiment results here.
- **Artifact**: Put the link to your codebase here, containing both the source code and raw results data; you might use GitHub or Zenodo.
- **References**: Any papers/works that have been cited should be properly referred to here.

[1] You can directly call the APIs for some standard and general implementations, such as TF-IDF, Naive Bayes, etc, but for specific approaches, you must not simply copy and paste the entire code from their public repositories as this will be treated as plagiarism. If you are unsure, please ask.

1.4 Requirements

Please use Time New Roman or Arial font type, 10pt, single column, single line space to write the report. The expected length should be between 2 and 6 pages (including references), A4 size. A longer report is possible but you need to have good justifications.

1.5 Final Submission

The report should be exported in PDF and submitted to Canvas before the deadline for week 10: **Fri, 28th Mar, 16:00**. Please do not submit any other code/data as they should have been published elsewhere and included in the report via a link. In the published codebase, please include the following PDF files in the root directory (with the exact naming):

- **requirements.pdf**: Any dependencies/versions that are required to compile or run the code.
- **manual.pdf**: A manual to explain how to use the tool.
- **replication.pdf**: A clear instruction as to how we can replicate the results reported.

Failure to provide the above would affect the validity and verification of the report.

1.6 General Q&A

"What if I cannot beat the baseline? Will this fail the coursework?"

> ***Answer:*** *The short answer is no. Whether or not you can beat the baseline is only part of the marking criteria as can be seen below. We would look at the whole report, including all the sections and in particular, the rationales behind the designs and other factors such as clarity, presentation, comparison setting, statistical tests, soundness etc. So it would still be a good deliverable if you have done everything well, but the results turn out to be similar (or worse) than the baseline. Having negative results is possible in scientific research; it is more important to clearly explain the process of reaching those results. Having said that, some careless or randomly combined solutions without good justification would clearly not be encouraged.*
>
> *In addition, there could be different metrics, e.g., your solution might not optimise better but it might take less time to give a result. Being proactive and think about the positive sides on the results.*

"What if I think the baseline is exactly the best solution?"

> ***Answer:*** *This is very unlikely. If this is indeed the case, you need to give strong results as to why you think this is a good solution in the* ***Solution*** *section. In this case, there might be no comparative results but still, you need to show that the outcome is satisfactory, e.g., if the accuracy of bug report analysis is constantly over 95% for all projects then we might agree that the baseline is good enough.*

"Would it be okay to re-implement an existing approach?"

> ***Answer:*** *Yes, if you read some papers, and think that one of them could be the best solution, then you can re-implement that approach. However, you need to explain the approach in your own words, together with an explanation of the design rationales. Having said that, re-implementing an existing approach is certainly not easy as you would need to have an in-depth understanding of the approach (and its code) first.*

"What exactly should I talk about in the ***Experiments****?"*

> ***Answer:*** *Normally, you need to present the results as figures and/or tables, and then you need to provide a general summary of the results, e.g., your solution might be better on 7 systems but worse on 3, but still overall it is better. You might also discuss what aspect/system/project your solution has done better and on which ones it might have been worse, as well as over what metrics. Some hypotheses about the possible causes are also welcome here.*

"Can I get some references?"

Answer: *Yes, please refer to the lecture slides or the recommended papers on Canvas.*

1.7 Marking Criteria

Criteria	Grade
Outstanding introduction of the problem; comprehensive coverage/comparisons of the existing approaches. The solution is articulated in exceptional detail with in-depth explanations of the rationales. All the experimental setups are elaborated with excellent details. There are excellent presentations of the quantitative results, with insightful discussions, while the proposed solution beats the baseline on at least one metric. Thorough and constructive reflection and conclusion. The results are fully reproducible and verifiable.	Outstanding $\geq$70%
A good introduction of the problem; key existing approaches are mostly covered although some most noticeable ones are omitted. The solution is articulated in good detail with sufficient explanations of the rationales. All the experimental setups are elaborated with sufficient details. There are presentations of the quantitative results, with good discussions, although the proposed approach might or might not beat the baseline. Good reflection and conclusion. The results can be reproduced and verifiable although with some deviations.	Good 60%–69%
Reasonable introduction of the problem; some existing approaches are covered. The solution is articulated in some detail with some explanations of the rationales. The experimental setups have been elaborated, but key information might be missing. There are some quantitative results that might not be presented well; the discussion might be presented or might be missing; the proposed approach might or might not beat the baseline. Some reflection and conclusion, but with limited insights. The results can only be partially reproduced and verified.	Average 50%–59%
Limited introduction of the problem; some existing approaches are covered. The solution is articulated but lacks explanations of the rationales. The experimental setups are only partially explained. Quantitative results might be missing with limited discussion; the proposed approach cannot beat the baseline. Limited reflection and conclusion. The results can hardly be reproduced and verified.	Pass 40%–49%
Very weak introduction to the problem; nearly no existing approaches are discussed. The solution is articulated but lacks explanations of the rationales. The experimental setups are missing. Quantitative results are rarely presented, and nearly no discussion; the proposed approach cannot beat the baseline. No reflection and a weak conclusion. The results cannot be reproduced or no code/raw data is provided.	Fail $<$40%

2 Comparatative and Analytical Project

2.1 The Task

For this type of project, you will research, compare, and author a survey that systematically summarises the existing methods used for a chosen intelligent software engineering problem (either AI for SE or SE for AI).

Students will work on their own. Every student is required to write a survey type of easy. The topic will have to be in accordance with the Code of Ethics from the university; see `https://www.birmingham.ac.uk/documents/university/legal/code-of-ethics.pdf` for more detailed information. Therefore, the tool should **not** involve any of the following:

- Experiments with human participants
- Activity falling under the Human Tissues Act
- Funding by philanthropic gifts
- Military applications
- Animal testing
- Possible conflict with ethical principles partially or wholly outside the above.

2.2 The Problem

You are free to work on any problem related to intelligent software engineering, either coming from the lecture content or beyond. However, you only need to work on **ONE** problem and cover a set of the relevant existing work/techniques.

2.3 Deliverable

The deliverable will mainly be a systematic survey. Common scientific referencing styles for Computer Science, i.e., the Chicago reference, should be used. Details can be found here: `https://pitt.libguides.com/citationhelp/ieee`. Specifically, the survey should at least have the following sections (feel free to change their names or add sub-sections therein):

- **Introduction**: This section should provide background information about the problem you address, including some discussion as to why you have chosen this problem.
- **Methodology**: What steps have you followed to filter and search for the papers? You might want to refer to the guidance of the systematic literature review here: `https://legacyfileshare.elsevier.com/promis_misc/525444systematicreviewsguide.pdf`.
- **Categorisation of the Techniques**: This section aims to provide a complete categorisation of the techniques. The key concepts that are used to divide those techniques can vary, e.g., it could be the characteristics of the techniques or the way they formulate the problem to be addressed.
- **Techniques Details**: Here, you should articulate the working principle of the techniques. There is no need to do this for every single technique; you could group them based on the previous categorisation and discuss the general procedure as well as some design rationales.
- **Comparison of the Techniques**: This section requires you to compare the pros and cons of different techniques since no single technique is perfect. Bear in mind that justification should be given when making any claims. What we need to achieve is that, after finishing reading this section, the reader should be able to know under what circumstances which technique might be more suitable.
- **Evaluation Methods**: In this section, you can talk about how you would evaluate the techniques, what metrics will be used, as well as the statistical tests, and the reasons behind the choices.
- **Conclusion**: Present your conclusion deriving from the discussed and comparisons of the techniques.
- **References**: Any papers/works that have been cited should be properly referred to here.

2.4 Requirements

Please use Time New Roman or Arial font type, 10pt, single column, single line space to write the survey. The expected length should be at least 8 pages (including references), A4 size.

2.5 Final Submission

The report should be exported in PDF and submitted to Canvas before the deadline for week 10: **Fri, 28th Mar, 16:00**.

2.6 Marking Criteria

Criteria	Grade
Outstanding introduction of the problem; comprehensive coverage/comparisons of the existing approaches with 10+ techniques. Excellent methodology was used. Excellent categorisation. The techniques are articulated in exceptional detail with in-depth explanations of the rationales. All the techniques are compared with constructive discussion of their pros and cons. insightful discussion of the evaluation methods.	Outstanding $\geq$70%
A good introduction of the problem; key existing approaches are mostly covered with $<$10 techniques. Good methodology was used. Good categorisation. The techniques are articulated in good detail with sufficient explanations of the rationales. All the experimental setups are elaborated with sufficient details. All the techniques are compared to some extent with a good discussion of their pros and cons. Good discussion of the evaluation methods.	Good 60%–69%
Reasonable introduction of the problem; some existing approaches are covered with less than 5 techniques. Some methodology procedures were used. Limited and less sensible categorisation. The techniques are articulated in some detail with some explanations of the rationales. Some of the techniques are compared to some extent with a reasonable discussion of their pros and cons. Acceptable discussion of the evaluation methods, but key information might be missing.	Average 50%–59%
Limited introduction of the problem; some existing approaches are covered. No systematic methodology was used. Unreasonable categorisation. The solution is articulated but lacks explanations of the rationales. Some of the techniques are compared with limited discussion of their pros and cons. limited discussion of the evaluation methods.	Pass 40%–49%
Very weak introduction to the problem; nearly no existing approaches are discussed. No systematic methodology was used. The categorisation is missing or only trivially mentioned. The solution is articulated but lacks explanations of the rationales. The evaluation is missing or rather limited. The techniques are trivially compared with no or rather limited discussion of their pros and cons. No discussion nor justifications for the evaluation methods.	Fail $<$40%

2.7 General Q&A

"Do I have to follow exactly the systematic literature review methodology above?"

Answer: *You do not have to follow the complete methodology. However, we strongly encourage you to use at least some part of it. For example, the idea of setting up a clear inclusion and exclusion strategy is of great help for a formal procedure of literature review.*

"How should I categorise the techniques?"

> ***Answer:*** *There are many different ways and categories. For example, you might categorise them based on their technical foundation, e.g., some are based on machine learning, some are based on optimisation, others are based on rule-based assumptions etc. You might also categorise them using the conditions of the problem they can cover. For instance, for configuration performance learning, some techniques are for multiple workloads while some other only works on a single workload.*

"To what extent do I need to elaborate the techniques?"

> ***Answer:*** *Please keep the technique description concise and precise; do not spend a lot of text explaining the detailed working procedure. For example, if a technique is based on a tailored neural network, you would only need to briefly talk about what addition the authors have made to standard neural networks, or there might be a few words about how neural networks work. There is no need to list the equations of the internal working mechanism of the network. Most importantly, the key is to provide critical and comparative analysis, e.g., in what circumstances is one approach preferred more than the other and why?*