* **Introduction**• A statement of the problem that your project addresses including motivation, aims, objectives and challenges of the project, description of the client and the brief, how you refined and improved the original brief, and your interaction with the client to achieve this.
* **Background** 
  + A description of the background and context of the project and its relation to work already done in the area. What existing solutions are there?
  + Discussion of technical solutions or frameworks present in work/solutions already in the market/published.
  + Discussion of any related academic work related to the problem area.
  + Discussion of available tools/methodologies for implementing solution to brief and their integrability
* **Design and Implementation** 
  + A description of the methodology used to develop software as a team, organise and delegate work, track progress and ensure delivery (e.g. Agile).
  + A description of the work carried out. This should include details of technical or scientific problems tackled, solutions proposed, and the design and development of software. For example, you might provide a description of user stories in each sprint, features delivered at each sprint and how backlog was refined at each stage. We want to see evidence of clear stages of development and prototyping (including wireframes etc) with discussion of the design decisions made at each step. For example, you might include minutes from sprint planning meetings and iterative client feedback at each stage to demonstrate how your project evolved.
  + Evidence of source control. Use links to your github repo (merge/branch/commits) along with code snippets within your report to highlight important parts inline with the body of your text. A well documented repository is expected and required to be submitted.
* **Evaluation and Testing** 
  + A description and analysis of results obtained.
  + Details of Tests developed and results which demonstrate robust implementation
  + Details of User Evaluation, if appropriate, to demonstrate how objectives are met.
* **Challenges**
* **Conclusion**
* A critical evaluation of the work. This is an analysis of the extent to which the project has achieved its objectives, and whether the choices that were made were, with hindsight, the best ones.
* Suggestions on possible improvements and/or further work.
* Contribution Statement. Indicating how the team collectively believe work and effort has been split through the assessment. Please include here details of anyone who contributed but then moved to the delayed project (resit) and their contribution to this point.

1. Introduction
   1. Background of Toumetis

Toumetis is a software development company that focuses on machine learning for engineering application, and our group had been working closely with their Data Science team. There are two main parts to this project – developing a machine learning model to automate the power quality events classification process, and building an interactive web app for data visualization and events labelling.

* 1. Motivation And Aims

Toumetis’ flagship platform, Cascadence, allows their stakeholders to use it for preventive maintenance to avoid power outages and to ensure public and environmental safety. For instance, failed equipment could potentially induce power line breakage, which might lead to wildfire. In view of this, subject matter experts, like engineers, at the company would identify abnormal signals from different power grids in advance. This enables them to carry out predictive maintenance and operation performance management in order to reduce accidents related to power lines from happening.

However, their current workflow might not be efficient enough. The subject matter experts would manually classify different power quality events, they would have to go through a large amount of data one by one. Not only is this approach time-consuming, it is also more prone to inaccuracy due to human errors. In light of this issue, this project was meant to improve this situation.

Our client gave us two articles to read through, and they expected us to replicate one of the articles and then put it into practice. The said article is about a hybrid approach that combines phase space reconstruction (PSR) and convolutional neural network (CNN) to classify power quality disturbance (PQD), which was exactly what the client wanted to achieve.

* 1. Brief Refinement
     1. Original Brief

When we first received the project brief, it was rather short and high-level without much details. Our client also gave us some articles to read through. All we knew at the beginning was that we have to generate synthetic data consisting of voltage and current waveforms with disturbances using numerical methods. Then use the synthetic data to train a machine learning model for classifying different disturbances types for use on real data collected from monitors installed on power grids. We were also expected to build a visualization tool for further investigation by experts.

* + 1. Making Use of University Resources

Refining the deliverables was time-consuming yet it was one of the most important parts of the whole project as it would determine which direction our project could go. Besides Oliver, none of our group members had much knowledge on signals and disturbances. Therefore, we sought help from our courses’ teaching assistant, Craig, as he has a background in electric signals. Besides going through the basics of currents and voltages, he also helped us kickstart the project by recommending tools for generating synthetic data.

Craig suggested that we use an open-source software development toolkit called GNU Radio to explore more on synthetic data generation so we have some basic knowledge on that before our first meeting with the client.

In terms of the machine learning aspect, since none of our group members had experience in this area, we made sure to make it known to the client so they know what to expect. We then turned to our supervisor, Kerstin, for advice on where to look at for acquiring technical skills on machine learning and deep learning. As she is more experienced in the software engineering industry, even though she was not specialised in the machine learning area, she could give us recommendations on where to start. She suggested us a few articles, videos, and online courses on machine learning basics, which were of great use to us.

* + 1. Initial Meeting With Toumetis \*\*\*insert the whiteboard photo\*\*\*

Upon reading up on the articles and researching on various concepts, we arranged our first client meeting in-person to refine and improve the original brief.

We went into the meeting with an intention to walk-through the entire project process from start to finish, so we could ask questions and clarify anything we did not understand.

Edd, the Lead Data Scientist of Toumetis, walked us through the whole project and what he expected us to complete. He drew out a mind map (referring to the photo) to show us the holistic picture and for us to ask any clarifying questions related to the project.

Our client gave us direction on what they expected for synthetic signal generation, which was to follow the Institute of Electrical and Electronics Engineers (IEEE) power disturbance definitions. In terms of the CNN, they also made it clear that the accuracy should be above 80%. Since this project was aimed for real-life usage, we asked if they could provide us with some real data for model training and evaluation purposes, but due to data protection and intellectual property issues, they were not able to provide us with that. Therefore, we changed our approach – we send our trained model to the client for them to run it through with their data instead of evaluating the model on our own with real data.

Regarding the second part of the project, which is building a web application for data visualization, as there was no academic paper to follow, more clarifying questions were needed.

To start with, we had to identify the target users, which turned out to be the subject matter experts like data scientists and engineers. As they are already experts in the area, there was no need to tailor the user experience to the general public. For instance, technical jargons can be used without detailed explanation. We mainly focused on the clarifications on the application’s functionalities as the user interface could be adjusted at a later stage.

Since we adopted the agile methodology, we were able to regularly collect feedback from our client and adjust accordingly swiftly. At the end of the initial meeting, we also came to a conclusion that we should meet up once every two weeks to update our progress and collect feedback from different teams.

1. Background
   1. Project Background And Context
      1. Power Grid Signals Monitoring By Power Companies

In this day and age, there are more energy sources in power generation than ever, hence there is a need for power companies to monitor and control the signal data from the integrated power grid network. Regular maintenance of the systems helps to improve the efficiency of the power grids and the safety of the environment.

As mentioned in section 1, the current approach of signal analysis and categorization is time-consuming and is more prone to human errors, hence the aim of this project is to automate the analysis of electrical signal data generated from different power grids, allowing subject matter experts to identify different types of abnormal signals, which are also known as power quality disturbances, more efficiently and effectively.

* + 1. Existing Solutions

As briefed by our client, we first have to general time-series electrical signals to train the machine learning model. Then, the signals have to be converted into 2-D grayscale images, this process is also known as Phase Space Reconstruction. After the images have been prepared, they are then fed into the CNN for training and categorization of 10 types of power quality disturbances, and we should get a group of probabilities of their corresponding types at the end. With these probabilities, a visualization tool can be generated for subject matter experts to efficiently label and categorize different types of disturbances.

Currently, there are various academic articles on automatic power grid signals analysis. Some scholars looked into methods like decision trees, artificial neural network, probabilistic neural network, support vector machines, and so on, to automate the analysis process.

In particular, an article titled “Classifying Power Quality Disturbances Based on Phase Space Reconstruction and a Convolutional Neural Network”[1], written by Kewei Cai, Taoping Hu, Wenping Cao, and Guofeng Li, seems to be the most suitable for our project. Our client also mentioned that it would be ideal if we could replicate what this article addresses.

The article recommended using a CNN model to analyse and categorize different power events and various types of power quality disturbances, the reason behind is that it is more accurate and efficient than other methods. Comparatively, adopting the CNN approach allows automated feature extractions, it is also more accurate due to the use of convolution layers.

* 1. Existing Frameworks

According to the article mentioned in section 2.1.2., we first had to generate synthetic signals as 1-D time-series voltage data using the open-source toolkit – GNU Radio, which will then be converted into 2-D grayscale images through PSR.

As part of the project requirement, synthetic waveforms should be generated according to the IEEE power disturbance definitions, and all of the 10 disturbances listed should be used.[2]

Once we have tested that the signals generated are accurate and correct, we would then have to perform mass production of the images. Scripts should be written to automate the image generation process.

Following the recommendation given in an online machine learning course from Stanford University, the instructor, Andrew Ng, suggested the dataset ratio as follows: 60% goes to training, 20% goes to validation, and the remaining 20% goes to testing.

To evaluate the performance of the trained models, the article suggested different methods, including but not limited to precision, recall, and F1 score.

* + 1. Web Application For Data Visualization

There are comparatively more resources on building an interactive web application than automating signal analysis process. However, the app should allow the users to perform dimensionality reduction and clustering algorithms on the data. Our client has suggested us some built-in libraries to do so, like PCA and K-Means, and we would also have to do our on research on different algorithms to decide which ones fit best.

* 1. Tools For Solution Implementation

Python is the main language used for our project as it has a lot of built-in libraries that we can use. Besides, Python is also the go-to language for developing machine learning models.

* + 1. Machine Learning Model

For the first part of the project, we made use of some Python libraries for data manipulation and model training, including Matplotlib, Numpy, Pandas, Sci-kit Learn, and Tensorflow.

To train different models, we mainly used Google Colab, which is a Python development environment that runs in the browser using Google Cloud, it also provides limited computer power. Besides that, we also trained the models on Apple’s M1 MacBooks.

The client also expected us to use Click, a command line interfaces library, to take in sets of power quality events.

* + 1. Web Application

In order to make the two parts of our project integrate better, we also used Python to build the web app. As suggested by the client, we used the Bokeh model inside a Panel app, which is a Python library for creating interactive visualizations.

Another library used was Altair, which is a declarative statistical visualization library for Python. Holoviews was also incorporated for easier graph plotting.

After receiving feature from the CNN output for the probabilities of the disturbances, the app will then use dimensionality reduction to reduce the CNN feature down to 2 dimensions for a scatter plot. With suggestions from the client and our own research, we added three algorithms to the app: UMAP, PCA, and t-SNE, which are all available as built-in Python libraries. As for clustering algorithms, we included three as well: K-Means, DBSCAN, and Agglomerative Clustering.

**Design and implementation**

a. Description of the methodology used

Essentially following the Agile Methodology, our team identifies four major areas, out of the twelve principles behind the Agile Manifesto, that are implemented through this software development project: i) the early, continuous, and frequent delivery of working software, ii) the encouragement of face-to-face communication, iii) the embrace of changing requirements, and iv) the organisation of tasks among motivated, flexible team members.

*i) The early, continuous, and frequent delivery of working software*

It has been the utmost importance of this project for our team to deliver working software to the client. Throughout the project, there are two certain points that our team has emphasised on: a) the delivery of the minimum viable product, and b) the continuous updates building on the base version of the solution.

Since the official project start in June 2022, after identifying the project scope and deliverables with the client at the initial meeting, our team has produced the minimum version of the products, including both Machine Learning model and a web dashboard, within the first month. These serve as the foundational platform to introduce more features to satisfy the deliverables to the client.

For the in-person client meetings every two weeks throughout the project, our team targets to bring an update with additional features on every instance. For example, in the middle of July 2022, our team has introduced the deployable unsupervised Machine Learning algorithms onto the web dashboard to aid the user in data exploration. This work practice has ensured the continuous receipt of client feedback on the working software by walking through the updates on every client meeting. Based on this feedback, our team has been able to adjust and recalibrate the product updates clearly and efficiently.

*ii) The encouragement of face-to-face communication*

In the Agile Methodology, it is highly recommended to maintain face-to-face communication to track progress and ensure delivery. Keeping everyone on the same page is the top goal of our team. And this has been achieved by encouraging regular face-to-face communication both with the client, and within our team members.

As mentioned, the client meetings are set up on a regular basis, where our team heads to the client office to meet in-person to discuss updates, challenges, and possible changes in requirements once every two weeks between June and August 2022. This is in part thanks to the convenient location of the client office, based in Spike Island here in Bristol. With the in-person nature of the meetings, our team has been able to walk through every iteration of the product updates, introduce the User Interface of the web dashboard, discuss the accuracy levels of the Machine Learning models, and so on, in an efficient manner.

Whereas for our team, regular face-to-face meetings are also conducted once every Monday throughout the project to discuss individual progress, distribute tasks, and share learnings and understandings, with additional, flexible timeslots for in-person pair-programming as well. The meeting format has to be online on Microsoft Teams due to the geographical challenges faced with some team members back to China in later stages of the project. Nevertheless, it is encouraged within our team that members could gather in the campus together for in-person discussions, pair-programming, and so on. This ensures every one of our team members is on the same page throughout the project.

*iii) The embrace of changing requirements*

With the continuous delivery of product updates amid the regular in-person client meetings, it proves to be convenient for our team to collect feedback and discuss any changes in the deliverables and requirements on specific features of the product. A positive, two-way exchange of information is formed between the two parties. Under this exchange, our team is able to stay proactive and responsive to any changing requirements.

For example, there are meetings with an User Interface designer from the client, who proposes a series of changes to how the web dashboard is to be operated, and how the user could manipulate the different parameters and options. After collecting this feedback, our team identifies features that could be introduced, and others that are too complicated to be implemented under the project timeframe. Relaying this analysis and showcasing the newly introduced features to the client, our team has successfully responded to the changing requirements from the client in a swift manner.

*iv) The organisation of tasks among motivated, flexible team members*

With a total of five team members, each of the project tasks is essentially led by each member, where task ownership is encouraged. How this plays out throughout the project is that each member, as the task leader, would initialise on the research and information gathering for his/ her particular task, specifies the possible steps and to-dos to complete the task, and leads the other members in developing the task. With a finished task, the leader would then switch to a role where he/ she would play cover for the other members.

For example, one member of our team is responsible for the task of creating synthetic power signals according to industry standards. This task proves to be shorter than the initial expectations. With that finished, the member joins the other member leading the development of the web dashboard. This flexible arrangement encourages task ownership among our motivated team members and proves to effective for the project.

b. Description of the work carried out

The client proposes to adopt automation in analysing and labelling power signal events. Prior, the client relies on humans to study, categorise and compare the different power signal events. In details, a subject matter expert in electrical signalling investigates the waveforms of the electrical signals, analyses their 1D time-series charts, identifies whether the signals fall into specific categories of Power Quality Disturbances (PQD) according to Institute of Electrical and Electronic Engineers (IEEE) standards, and finally, compares and groups similar events. This manual process takes lengthy engineering hours. Thus, the client asks for a solution that incorporates Machine Learning to automate the process.

In initialising this project, our team expects to build a solution to deliver the said values in automating the task of analysing and labelling power signal events for the client. The target users of this solution would include the subject matter experts aforementioned, the analysts, and the data scientists from the client.

In designing the solution, our team identifies two major elements: 1) a script/ tool to handle input power events, and 2) a web dashboard to facilitate interactive data exploration. The two elements correspond to specific tasks in the project, as illustrated by the flow chart.

Specifically, the tool of the first element 1a) handles raw inputs of power signal events from the user, 1b) transforms the waveforms of the electrical signals into 2D images through Phase Space Reconstruction (PSR), and 1c) employs the trained Convolutional Neural Networks (CNN) model to predict the PQD types of the signals. The expected output is a data file in CSV format containing the prediction scores of all the waveforms of the power signal event.

The web dashboard of the second element takes the said output data file to 2a) organise all the analysed power signal events on the “Data-exploration Pane”, 2b) present the details of single elected events in the “Selected Event” tab, and 2c) identify the respective similar events in the “Similar Events” tab.

Our team works through all these major features (1a, 1b, 1c, 2a, 2b and 2c) step by step. There are different technical challenges faced, resources employed, and solutions built for each feature. This paper shall discuss each of them in details.

*1a) Handling raw user inputs of power signal events*

*What is the technical challenge faced?*

The imminent challenge faced at developing this feature is the lack of access to real-life power signal data from the client. Acting as a third-party data analytics solution provider, the client has access to real-life data generated from a number of power grids. Nevertheless, due to their business nature as a third-party service provider, they are limited by contractual obligations that none of such data could be shared with our team in this project. The lack of such data complicates the task of training the relevant Machine Learning models. Our team could not directly apply real-life data to the training datasets, hence posing a challenge to the accuracy of the models.

*What is the solution?*

In light of the challenge, the answer is to synthesis the power signal data from alternatives sources. Our team spends additional time in studying the signal standards provided by IEEE, as shown in the following exhibit, and a myriad of academic papers in calculating the required signals. The synthetic data consist of ten PQD types as a result.

Exhibit 1: Formulas to calculate the ten PQD types

Table

Description automatically generated

*Source: “Classifying Power Quality Disturbances Based on Phase Space Reconstruction and a Convolutional Neural Network” by Cai, Hu, Cao and Li in Jul 2019*

Moreover, our team takes the initiative in configuring the format of the input power signal event data files, which incorporate the event metadata, including the unique event id, and its time and location, and the electrical signals of the event. The created data files are then confirmed to the client as an acceptable alternative to the real-life data.

With these confirmed input data files, our team subsequently create the script of reading such files, extracting and processing the data, and storing the results automatically.

*How did we design the solution?*

Our team recognises the specific challenge where none of the members has any expertise or experience in electrical signalling. Thus, in designing such a solution, our team targets to seek resources from professional organisations and academic sources in such fields. As a result, our team takes IEEE standards as the main reference in generating the synthetic data.

*How did we develop and implement the solution?*

The generation of such synthetic data and their data files requires extensive mathematical calculations according to the formulas provided by IEEE. Our team, considering this requirement, elects to use Python and its libraries for mathematical calculations such as NumPy and Pandas to perform the calculations, and store the results in CSV file format for the data files. For each of the PQD types, a Python script is developed to calculate the sine waves of the electrical signals, together with the abnormal signals and noise levels. In order to automate the process of generating large volumes of the synthetic data, the input parameters of each of the formulas are then randomised within certain ranges. Throughout the project, our team has then generated more than tens of thousands of the synthetic data files.

After generating the synthetic data files, our team has elected to develop the tool to handle the CSV data files with Python and its libraries, primarily to maintain the same platform used throughout the whole project.

*1b) Transforming signals into 2D images*

*What is the technical challenge faced?*

The CNNs employed to categorise the electrical signals fundamentally perform the task of categorising 2D images. Hence, the technical challenge of this feature is to transform the synthetic data signals, which are 1D time-series data, created in feature 1a into such images.

*What is the solution?*

The solution is to apply a concept in mathematics called PSR to attach an additional dimension to the 1D data. The 1D time-series data, upon the transformation by PSR, is represented by geometric shapes, i.e. 2D images essentially, as shown in the following exhibit. Each image is in greyscale and has two axes of their normalised data values. Images of different PQD types are in different geometric shapes, hence enabling the CNNs to categorise them based on the differences in their features.

Exhibit 1: The geometric shapes of the ten PQD types

Diagram, shape

Description automatically generated

*Source: PSR images prepared by our team*

*How did we design the solution?*

Again, our team recognises that no team members are an expert in this area. To overcome such a barrier, our team seeks guidance from the client who has then suggested a number of academic papers for reference. After studying these papers, our team identifies the application of the PSR method to turn the signals into 2D images and elects to implements this as the solution.

*How did we develop and implement the solution?*

After identifying PSR, our team seeks existing resources at the Python libraries to implement the said method in the script. The method provided at the NumPy library is considered suitable after studying the documentations and illustrations of various libraries focused on mathematical calculations. Thus, the script employs this library in implementing the solution.

*1c) Employing CNNs to predict the PQD types of the signals*

*What is the technical challenge faced?*

The client proposes the application of CNNs to automatically categorise the signals into their respective PQD types, providing academic papers as reference that cover the structure of and the algorithms used in the CNNs. Thus, the technical challenge faced lies on how to develop such CNNs with the specified structure and algorithms.

*What is the solution?*

Our team relies on existing resources, primarily the TensorFlow library provided by Google, in developing the CNNs. Python scripts are developed based on the guidelines, illustrations, and documentations of the said library to read the generated 2D images, perform categorisation, and generate an output file that includes all input events by the user.

*How did we design the solution?*

Seeing a marked knowledge gap in Machine Learning, our team proactively seeks guidance from the academic supervisor, relies on related materials and textbooks, and indulges in online courses in the area, which, in particular, includes the online Machine Learning course by Professor Andrew Ng on Coursera. After getting up to speed with the fundamentals in Machine Learning, our team identifies TensorFlow as a primary platform to develop the CNNs. The extensive examples and guides on the platform provide good reference and existing resources for our team to develop scripts, as shown in the following exhibit, for the feature.

Exhibit 1: Guide on developing CNNs on TensorFlow

Graphical user interface, application

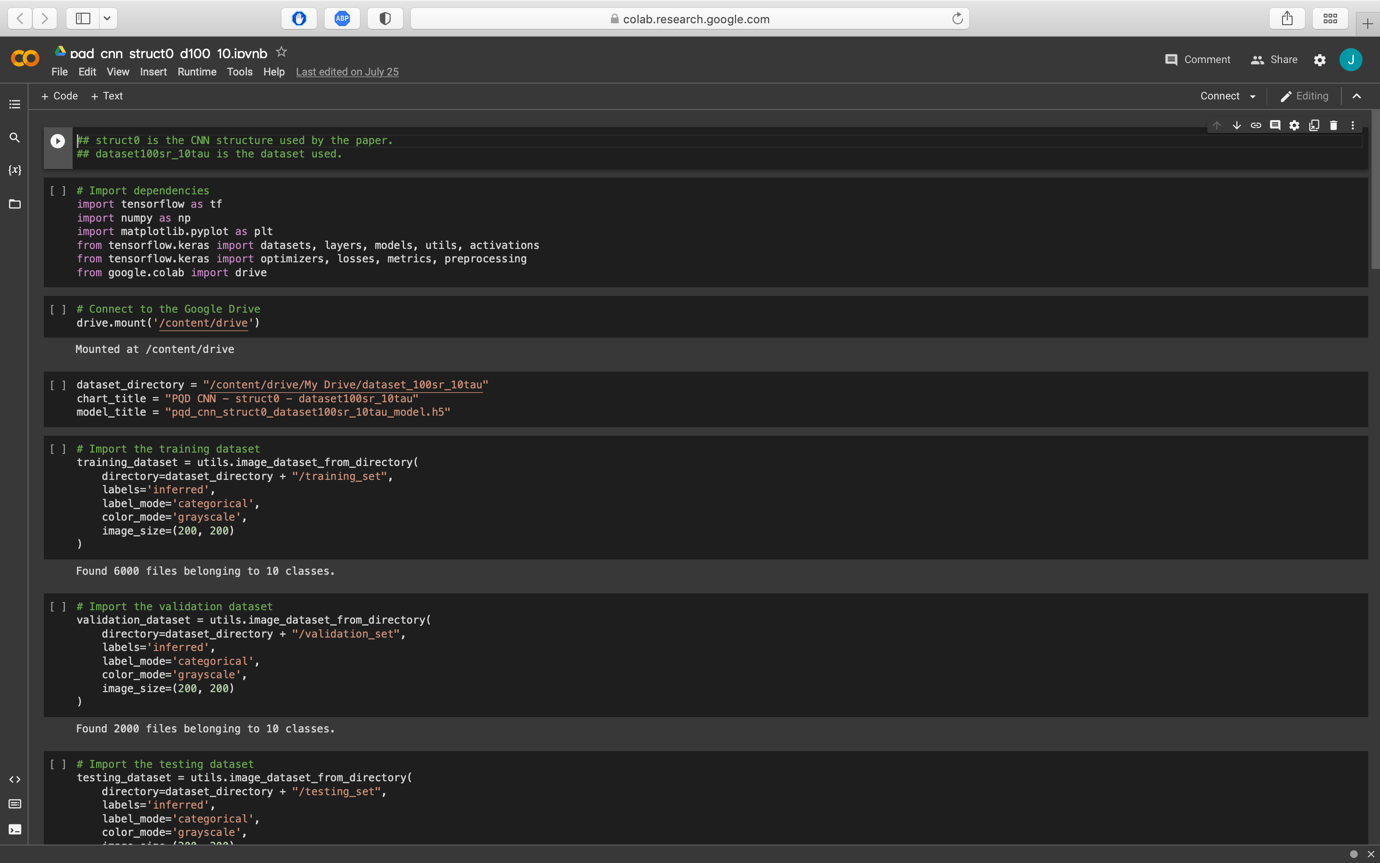
Description automatically generated

*Source: TensorFlow (https://www.tensorflow.org/tutorials/images/cnn)*

*How did we develop and implement the solution?*

With the scripts in place, our team has to manoeuvre through another obstacle, the severe lack in computing resources, especially the GPU, in developing the CNNs. Running the designed scripts on the personal laptops of our team could take more than two hours. To overcome such a challenge, our team proactively seeks resources from third parties. At the end, a platform called Google Colab with free-of-charge albeit limited GPU resources is adopted as the platform to implement the designed scripts for developing the CNNs, as show in the following exhibit.

Exhibit 1: A script for developing the CNN on Google Colab



*Source: Scripts for developing the CNNs on Google Colab prepared by our team*

*2a) Organising the signal events on the “Data-exploration Pane”*

*What is the technical challenge faced?*

The challenge faced at this feature is how to enable the user to explore the number of power signal events interactively. These events, processed by the CNNs, would be attached with the prediction probabilities of their PQD types. With six signal waveforms and ten PQD types, there would be a total of sixty prediction scores for each event. Hence, the question lies on how the user can explore, compare, and examine these events together with options to manipulate the data dynamically.

*What is the solution?*

The solution proposed would be a data dashboard presented on a local web server. This dashboard primarily incorporates a scatterplot, called the “Data-exploration Pane”, to organise the uploaded events with their prediction scores, through plotting their positions on the chart with two of the prediction scores as the x- and y-axis.

This solution is dynamic in the sense that the user can manually adjust which two prediction scores out of total of sixty would be used as the axes. For example, as the following exhibit shows, the user could choose “Vab – sags” as the x-axis, and “Ic – oscillatory transients” as the y-axis for the scatterplot. Then, the events would be plotted to their prediction scores for these two axes correspondingly. This provides a direct method to compare the events based on two specific prediction scores.

Exhibit 1: Options for adjusting the axes of the “Data-exploration Pane”

Chart

Description automatically generated

*User is enabled to adjust what PQD prediction scores are used as the axes for the “Data-exploration Pane”*

*Source: Web dashboard prepared by our team*

Another feature of the solution is the employment of Dimensionality Reduction algorithms for the “Data-exploration Pane”. These algorithms are unsupervised Machine Learning algorithms to calculate the values of a set of lower number of dimensions based on the input values. In this case, a number of Dimensionality Reduction algorithms, including PCA, UMAP, and t-SNE, are provided to the user. When deployed, the sixty prediction scores/ possible dimensions are reduced to two dimensions, enabling the user to examine the events based on the composite scores efficiently. The parameters of the algorithms are presented as adjustable user options to manipulate the data and algorithms dynamically.

Furthermore, Clustering algorithms, including K-Means, DBSCAN, and Agglomerative Clustering, are incorporated to the feature as well. As another group of unsupervised Machine Learning algorithms, these algorithms identify possible clusters of data based on their positions on the plot and their relative distance to each other. Incorporated in the “Data-exploration Pane”, these algorithms highlight the clusters of events in different colours, as shown in the exhibit. This directly provides the user insights in identifying events that share similar traits. Again, the algorithm parameters are adjustable in this dynamic dashboard.

Exhibit 1: How the Dimensionality Reduction and Clustering algorithms deployed identify clusters of similar power signal events on the “Data-exploration Pane”

Chart, scatter chart

Description automatically generated

*Five clusters of similar events are identified by the algorithms, labelled by the different colours*

*Source: Web dashboard prepared by our team*

Last but not least, the dashboard is set up on a local web server, where it has an effective user interface to dynamically adjust the different input options and parameters, and a solid mechanism to communicate between the front-end interface and the back-end data.

*How did we design the solution?*

The client proposes a web-based dashboard to upload, organise, and manipulate the processed data, allowing dynamic exploration of the power signal events. The main reason behind is that the client is developing similar dashboards of different functions on the web platform as well. Our team follows this guidance and designs such a web-based dashboard with all the said features, algorithms, and options.

*How did we develop and implement the solution?*

Our team seeks existing resources for developing the web-based dashboard, instead of creating it from scratch. Hence, our team elects to look for reference from the client who is experienced in developing such dashboards. Moreover, it would be more convenient for the client to integrate our solution when it is developed using similar technologies. As a result, the dashboard is developed with a Python library called Panel, one of the technologies that the client has adopted. This library enables the development of web dashboards on Python, with limited maintenance on the back-end needed. Our team then gets familiar with this library by studying its documentation and various illustrations.

With the web framework in place, the next task is to incorporate the said Dimensionality Reduction and Clustering algorithms into the dashboard. Similar to the feature of using the CNNs, our team relies on existing libraries for applying these algorithms. The library in use is the “scikit-learn” library, which specialises in implementing Machine Learning algorithms in Python. Since both the web framework and the algorithms are in Python, it proves to be convenient in integrating these features into the web-based dashboard.

*2b) Presenting event details in the “Selected Event” tab*

*What is the technical challenge faced? What is the solution?*

With every point on the “Data-exploration Pane” representing a single power signal event, this feature is to allow the user to view the selected event in detail. Fundamentally, the question lies on how to capture the event selection on the scatterplot, use its unique event id, reads the corresponding data file, and presents the event data to the user.

Capturing the event selection on the plot proves to be a challenge. Our team has explored a myriad of plotting libraries in Python, including Matplotlib, Plotly, and Holoviews, to check whether they support such a function. At the end, after hours of studying their documentations and illustrations, a certain plotting library called Altair is identified to be useful for such a purpose, allowing the identification of the event selected, paving the way to read and present its data to the user.

*How did we design the solution?*

Our team elects to draw the charts and create the elements on the web dashboard with the provided resources and functions from the libraries. Hence, it is of great importance that the features needed are supported by these resources and functions. In order to achieve this, our team first identifies the elements needed, and then seeks the resources that could be utilised. If yes, these elements could be swiftly introduced. Otherwise, our team seeks possible workarounds, such as presenting the elements in a different way possible. In this case, when the plotting libraries in use cannot satisfy our needs, our team searches through all deployable libraries on the web dashboard for the functions needed.

*How did we develop and implement the solution?*

The foundation of this feature is built on the “Data-exploration Pane”, where the server has to be able to identify the event selected on the scatterplot. It is thus able to do so when the plot is charted by the Altair library. With the captured event data, the server then fetches the corresponding data file with the associated event id and filename. Different data fields in the data file, including the event metadata, its axis values, the cluster value, its six waveforms in 1D time-series, and its similar events, are then presented in the “Selected Event” tab to allow a detailed examination of the single event selected in dynamic fashion.

*2c) Identifying similar events in the “Similar Events” tab*

*What is the technical challenge faced?*

For the final feature in the solution, a summary of similar events is identified and presented to the user on the “Similar Events” tab, as shown in the following exhibit. This summary takes the form of a Tabulator object, with each row of data representing a single power signal event. Similar events in the share cluster, found out by the Clustering algorithms deployed, are selected to this summary.

Exhibit 1: The “Similar Events” tab on the web dashboard

Graphical user interface, application

Description automatically generated

*1. Similar events identified by the Clustering algorithm, K-Means in this case, deployed*

*2. The Tabulator object showing the summary of similar events*

*Source: Web dashboard prepared by our team*

There are two supporting functions: displaying the details of the selected similar events in a format to the “Selected Event” tab, and labelling and exporting the data of groups of selected events. Hence, the selection of events from the Tabulator object is the key challenge here, where it proves to be essential to ensure the accuracy of extracting the data from the said object, especially when features such as sorting, filtering, and user-editing, are enabled on the object.

*What is the solution? How did we design the solution? How did we develop and implement the solution?*

Our team once again relies on the Tabulator library that is incorporated onto the Panel library to develop this summary of similar events. Studying their documentations to learn about the properties and methods of the Tabulator objects proves to be the way to go.

However, a specific complication encountered is that the Panel library supports an incomplete version of the Tabulator library, i.e. the documentation of the Tabulator library does not entirely apply in this case. As a result, our team resorts to a “trial and error” technique to check which property or method proves to be useful and relevant. The feature is successfully introduced to the solution after quite a few engineering hours.

c. Evidence of source control

Source control is rigorously performed throughout this project primarily through the use of git commits with meaningful comments. Each “git push” is further complemented by a status update, detailing what changes have been made, and how other team members could follow up on the tasks, at the Microsoft Teams chat within our team. The git commits made are mainly centred around three specific areas: 1) introduction of new features, 2) testing and bug fixes, and 3) tidy-up of the git repository.

*1) Introduction of new features*

With our team members owning and leading the development of different features of the solution, it proves to be convenient to directly use git commits to introduce new features into the repository, with little encounters of merge conflicts throughout the project. The following exhibits show evidence of how our team has introduced new features to the repository with git commits that are supplemented with on-point comments.

*(GitHub link for the introduction of the new Dimensionality Reduction algorithms onto the web dashboard: https://github.com/zzzzqi/2022-Power-Grids/commit/60b8ef7b308809654dafecf5d02130414b45bba1)*

Exhibit 1: Git commit to introduce new Dimensionality Reduction algorithms onto the web dashboard



Source: GitHub repository of the project

*2) Testing and bug fixes*

Another area which the source control applies is around testing, debugging and applying fixes. Team members play different roles of introducing the features and testing them. After a new update to the repo with the introduction of a feature, other members would take the responsibility of testing the feature, checking for possible bugs, and applying the corresponding fixes. Subsequently, the fixes are uploaded to the repository through another “git push”. This following exhibit illustrates the sequence of introducing the new feature and debugging it.

*(GitHub link for the introduction of the new feature: https://github.com/zzzzqi/2022-Power-Grids/commit/91693c262eb46d1210fb2ef756b965e0fe251f23)*

*(GitHub link for the introduction of the testing and bug fixes: https://github.com/zzzzqi/2022-Power-Grids/commit/847a39bf6e69559dca3dbd6153479f51409ffcb3)*

Exhibit 1: Git commits to introduce a new feature and the corresponding bug fixes

Graphical user interface

Description automatically generated with medium confidence

*2. Git commits to debug and apply fixes to the new features implemented*

*1. Git commits to introduce new features to web dashboard*

*Source: GitHub repository of the project*

*3) Tidy-up of the git repository*

As the project carries on, the size of the repository and the number of files and folders grow exponentially. In order to maintain the clarity and structure of the repository, it is reasonable to put forward new structures of directories to contain the different files from time to time. This following exhibit illustrates how this is performed.

*(GitHub link: https://github.com/zzzzqi/2022-Power-Grids/commit/2582c8efeb3628f5f16c8be9c03e50e6769af0d8 )*

Exhibit 1: Git commit to tidy up the repository



*Source: GitHub repository of the project*

**Evaluation and Testing**

There are two particular sections for the evaluation and testing performed in this project: 1) the part on the main operating scripts for handling inputs and the web dashboard, and 2) the part on the research of CNNs. This paper would discuss the corresponding evaluation and testing performed for both sections.

*1) Main operating scripts for handling inputs and the web dashboard*

*a. Details of user evaluation*

The purpose of the user evaluation performed on this section is to produce the operating scripts as per client requirements. These operating scripts serve as the backbone as the input handling tool for raw inputs of power signal events, and as the web dashboard for dynamic data exploration. By evaluating the scripts, collecting feedback from the client, and making improvements regularly, this process of regular user evaluation enables our team to develop the solution successfully with the client closely informed and involved of the progress throughout the project.

The project officially spans through the period from June to August 2022. In this three-month period, regular client meetings are conducted once every two weeks, where our team has made the trip to meet the client in-person on the Tuesdays, updating the correspondent Dr Edward Rowland and his Data Science team mainly. Since the first prototype in mid-June, on every occasion, a regular product update is showcased to the client, introducing all incrementally new features and bug fixes, attending to questions, enquiries, and changes in requirements from the client. The collected client feedback serves as new building blocks for the project, where our team identifies possible areas to improve and further develop the product.

For example, on 26 July 2022, our team performed a product demonstration for the client, with representatives from their Data Science team as usual in addition to their User Interface team as well. At this meeting, our team did a detailed walkthrough of the input handling tool and the web dashboard based on synthetic datasets of power signal events. Various new features at the time, including the Dimensionality Reduction and Clustering algorithms, and the options to adjust the axes of the “Data-exploration Pane”, were introduced. At the meeting, the client opined that a series of adjustments could be helpful, including that i) options for adjusting the algorithm parameters could be added, ii) a separate “Similar Events” page for increased clarity, and iii) related to the user interface, particular warnings for the user on possible slow performance at certain features. In response to the feedback collected at this face-to-face product demonstration at the client office, our team swiftly engineered a number of updates that feature on the finalised product, as the following exhibits show.

Exhibit 1: Options for adjusting algorithm parameters on the web dashboard

Chart, scatter chart

Description automatically generated

*1. Options for adjusting parameters for Dimensionality Reduction and Clustering algos*

*Source: Web dashboard prepared by our team*

Exhibit 1: The separate “Similar Events” page for showing the events and the user warning

Graphical user interface, text, application

Description automatically generated

*2. The “Similar Events” page*

*3. User warning for slow performance*

*Source: Web dashboard prepared by our team*

Another example is the penultimate client meeting in August 2022. With the operating scripts nearly finalised with most of the deliverables to the client completed, at the meeting, the client requested a detailed documentation to be provided to guide their engineers through. Responding to this request, as shown in the following exhibits, our team proactively identifies the Sphinx library for producing the said documentation, prepares the installation file for all the libraries used in the project, and delivers a complete package to the client at the final meeting.

Exhibit 1: The project documentation prepared with the Sphinx library

Graphical user interface, text, application, email

Description automatically generated

*Source: Project documentation prepared by our team*

Exhibit 1: Project dependency installation file for Python Package Index (PyPI)

Graphical user interface, application, Word

Description automatically generated

*Source: Project dependency installation file prepared by our team*

The face-to-face meetings have been a proven channel for our team to perform user evaluation, collect their feedback, and improve the product throughout this project. Due to the specialised nature of the product, alternative methods for evaluating human-computer interaction, such as mass surveys and questionnaires have little relevance.

*b. Details of tests performed and their results*

The tests performed on this section mainly centre around the behaviour of the operating scripts. Integration and functional tests are performed on both scripts to check whether the actual behaviour matches the corresponding expectations. Testing is vastly performed before every client meeting where, as discussed, is for showcasing every updated feature to the client.

For example, for the input handling tool, integration tests are frequently performed to check if i) the input power signal events, in their CSV file format, could have their data extracted to produce the 2D images for each of their waveforms through PSR, and ii) the trained CNN model could be used to predict the PQD types of the waveform images. Different command line options, datasets and CNNs are used to test if the expected actions would be performed. Thus, by integrating the use of files of multiple formats, our team is able to test if the input handling tool performs as expected.

Another example is how our team uses functional tests for the web dashboard. In particular, functions on the dashboard such as displaying the selected similar events, and grouping and labelling a number of events, are tested rigorously, in order to make sure the functions perform correctly. For the function of displaying the selected similar events, as shown in the following exhibit, different events are selected with different data files uploaded to see if the chosen events are correctly displayed with their event metadata and waveforms. Whereas for the function for grouping and labelling the events, different selections are made and their export data files are tested well. These functional tests assure the correct behaviour of the web dashboard.

Exhibit 1: How the web dashboard displays the selected similar events

Graphical user interface, text, application

Description automatically generated

*1. Select a similar event*

*2. Click this button to call the function*

Graphical user interface

Description automatically generated

*3. Show the originally selected event for reference*

Graphical user interface, application

Description automatically generated

*4. Show the selected similar event for comparison*

*Source: Web dashboard prepared by our team based on synthetic datasets*

*2) The research of CNNs*

*a. Details of user evaluation*

The user evaluation of the research of CNNs proves to be straightforward, with the main focus being the task of communicating with the client the differences in the accuracy levels of the CNNs with different structures, algorithms, and training datasets. At the client meetings, the accuracy levels, measured by their classification accuracy levels, marco-F1 scores, and confusion matrices, are discussed with the client, in order to know which CNN could be the most suitable to the real-life datasets. Granted that our team is no expert in Machin Learning, and has no access to the real-life datasets on the client side, this user evaluation is fairly academic and based on CNNs trained with synthetic datasets. For the most accurate CNN trained by our team, which is the model with 256 units of sample rate, 20 units of the time lag factor, and with 2 additional layers, its multiple accuracy measurement metrics are shown in the following exhibits for reference.

Exhibit 1: Accuracy measurement metrics of the most accurate CNN trained







*Source: Team analysis*

*b. Details of tests performed and their results*

As discussed, our team has conducted a rigorous, academic research to assess and document the differences in the accuracy levels of the CNNs with different structures, algorithms, and training datasets.

Specifically, a total of a hundred and eight models are trained. There are six different model structures: i) the fundamental structure with two convolution layers, two pooling layers and the use of the average pooling algorithm, ii) an enlarged convolution size for the layers, iii) an increased convolution and pooling layer, iv) two increased convolution and pooling layers, v) the use of the maximum pooling algorithm in replacement, and vi) an reversed order of the layers against the fundamental structure. There are six different data sample rates for the datasets from 100 to 500 units. And there are three different time lag factors used for the PSR method converting the waveforms into 2D images. Hence, there a total of a hundred and eight models for our team to assess their accuracy levels.

There are three metrics used to assess their respective accuracy levels: i) the classification accuracy level calculated by the number of correct categorisations over the total attempts, ii) the macro-F1 scores covering the different true and false positives and negatives, calculated by the formula shown in the following exhibit, and iii) their confusion matrices which show a detailed categorisation result for every PQD type. A summary of their results is shown in the following exhibit.

Exhibit 1: Formula for Macro-F1 Score



Exhibit 1: Macro-F1 Score of each of the trained CNNs 

*Source: Team analysis*

Since our team is again no expert in Machine Learning, and possesses fairly limited computing power in the student laptops, a simple conclusion that can be drawn from this test and research is that the training dataset for the CNN plays an important role in correctly categorising the images. It would seem essential for the CNN to be trained by a dataset that matches the sample rate and time lag factor used in the target dataset. This finding has been subsequently relayed to the client for their Data Science team.

# Challenges

Although we completed the project successfully, it was not without its challenges. In fact, since this is a team project, we have faced challenges such as managing the team in addition to the project itself. These challenges were on the one hand a hindrance to the project, but on the other hand they allowed us to understand the project better and facilitate communication between us. In this section, we will talk about what kinds of challenges we have faced and how we solve them.

## Manage team

At the beginning of our project, how to manage our team is the first challenge we faced. Specifically, how we should meet and how often. How to distribute tasks and how to keep everyone in touch. We also need to communicate with our supervisor and clients about progress and make adjustments to the project at any time based on the idea of agile development.

In order to keep in touch with each other, we had group meeting every week. Also, there was meeting with supervisor Bi-weekly, so we are able to talk to and get advice from our supervisor. After each meeting, we uploaded a brief meeting summary in the team’s channel for every member to review. Besides, we keep weekly email exchange with our supervisor so that our supervisor could give us proper advice.

## Manage client and project

Managing client and project are other two challenges. We need to work with our clients to define the objectives and outcomes of the project. And we should be able to respond the follow-up needs from meetings.

Firstly, we had Bi-weekly in person meetings with our client. At each meeting, we show the latest version of our product and get the feedback and requirement from them. For example, according to their requirement, we divided the ‘selected event’ and ‘similar events’ into two panes.

In order to show the obvious progress to our client, we need to manage our project. Therefore, we used GitHub to control the version of the product, and communicate with each other about the tasks we need to do in the next week at the weekly group meeting. Besides, we shared the latest update in the team’s channel so that everyone could check the update and give feedback.

## Search and learn lots of new things in a short time

At the process of project, we are required to search proper libraries which can be used in python. The question is that we can not learn lots of new knowledge at the same time. Therefore, we divided the libraries we need to learn according to the sections we are responsible for. When learning a new library, we share the information and learning routes we find with the team in parallel, making it efficient for everyone to have a clear understanding of the parts for which different team members are responsible. For example, Riley is in charge of scikitlearn and umap, Qi is responsible for panel and sphinx.

## Handle the loss of manpower due to university’s re-sit arrangements

Although we wanted our team to work together to complete the project, two members – Qi and Xihao – had to leave the project due to the university’s re-sit arrangements. It basically means a 40 percent loss of the available manpower. When we heard the bad news, we knew we needed to take up their part and continue the project. Thanks to the constant communication and active sharing of our parts as the project progressed, we were able to quickly understand the idea and code logic of their parts. For example, Jasper help Qi with bugs fixes and feature additions to the web side. We write the document of the project according to the information of sphinx which searched by Qi.

In a word, to handle these challenges, communication is the most important point we need to keep and what we have learnt.

# Conclusion

In this project, we created a tool which can convert the 1D power grid signal to 2D images and classify the type of images. Besides, we designed a web dashboard which can show the detail of the chosen signal and the relationship between the chosen signal with other signals. There are the achievements of this project.

1. The first part we convert the raw signal to images and pass the images to the cnn model and then export the CSV file which contains the result of the classification and meta data for part two:
   1. We wrote a series of scripts to generate different Power Quality Disturbances (PQD) according to Institute of Electrical and Electronic Engineers (IEEE) standards. The transformed synthetic signals are highly compatible with the relevant images in the reference.
   2. The project has the ability that convert the 1D voltage and current signal to 2D images automatically. Here we chose to use the idea of Phase Space Reconstruction to convert these signals. After converting, the type of disturbances can be clearly figure out by eyes.
   3. After training 108 models with different parameter values, we successfully trained a Convolutional Neural Network model which can classify the type of disturbances, which is able to achieve 97.7% accuracy.
   4. The project ended up putting the above results together with a command line tool. This command tool can handle the input and export a CSV file which includes the classification data of all signals.
2. The second part we designed a web dashboard:
   1. To show the signals interactively, we designed a data-exploration pane that the users can present the signal according to different algorithms and types of perturbations.
   2. For a single selected signal, there is another pane which can show the detail of the signal such as six waveforms, predicted value and the table of similar signals.
   3. After choosing similar signals, the users can compare the waveforms and the detail of signals. Besides, these data can be exported and saved as a CSV file.

Although it is fully meeting the needs of the client, there is still potential for further improvement in this project. At the back-end part, we can add more options for the users to choose, such as the delay time of phase space reconstruction, the path of the CSV file etc. At the front-end part, we can use JavaScript to refactor the web dashboard for higher design freedom and better User Interaction.

1. The front-end:
   1. Synthetic data generation: Jasper Li
   2. Data conversion via Phase Space Reconstruction: Oliver Lin, Jasper Li
   3. Convolutional Neural Network training: Xihao Wang, Jasper Li
   4. Command Tool: Oliver Lin, Riley Lee
2. The back-end:
   1. The structure of web dashboard: Jasper Li, Qi Zhao, Riley Lee
   2. The Clustering algorithms: Riley Lee
3. Documentation of project: Qi Zhao, Jasper Li, Oliver Lin
4. Group report division:
   1. Riley Lee: Introduction, Background
   2. Jasper Li: Design and Implementation, Evaluation and Testing
   3. Oliver Lin: Challenges, Conclusion