**Methodology**

### 1. Data Preprocessing

* A MinMaxScaler is defined to normalize the wind speed values between 0 and 1, ensuring that inputs remain within a consistent range suitable for the trained model.
* The scaler is pre-fitted with assumed wind speed values ranging from 0 to 100 km/h.

### 2. Model Loading

* A pre-trained Variational Autoencoder (VAE) with a Bidirectional Long Short-Term Memory (BiLSTM) network is loaded using tensorflow.keras.models.load\_model.
* Custom objects, including a KL Divergence loss layer and a sampling function, are registered and utilized during the model loading process.
* The model is compiled using the Adam optimizer with Mean Squared Error (MSE) loss function.

### 3. Wind Speed Prediction Process

* The user enters the current day’s wind speed through a graphical user interface (GUI).
* The input value is validated to ensure it is a numerical entry.
* The wind speed value is scaled using the pre-defined MinMaxScaler.
* A sequence of wind speed values is created, assuming the model expects an input shape of (None, 10, 4).
* The model predicts wind speed for the next 5 days based on the provided input.
* The predicted values are inverse-transformed to obtain the actual wind speed values in km/h.

### 4. Graphical User Interface (GUI) Development

* The GUI is implemented using customtkinter for an interactive user experience.
* Users can input the wind speed, trigger predictions, and view the predicted wind speed values for the next five days.
* A CTkEntry field allows users to enter wind speed values, and a CTkButton triggers the prediction function.
* The results are displayed using a CTkLabel.

### 5. Error Handling

* The system handles incorrect inputs (non-numeric values) by displaying an error message.
* Any exceptions encountered during prediction are caught and displayed appropriately in the GUI.

### 6. Execution and Deployment

* The GUI is initialized with a dark theme for better readability.
* The application is designed to run as a standalone script, integrating model inference seamlessly within the user interface.