Algorithm 1: Data Acquisition and Preprocessing

Input: Raw data sources $S = \{s_1, s_2, \ldots, s_n\}$ (e.g., IoT sensors, fleet logs, manual entries), emission factors $E = \{e_1, e_2, \ldots, e_m\}$.
Output: Validated and preprocessed data S', securely stored for further analysis.

Algorithm 1 Data Acquisition and Preprocessing

1: Identify stakeholders $\mathcal{ST}$ and system objectives $\mathcal{O}$.
2: Define emission categories:

- **Scope 1:** Direct emissions from controlled sources.
- **Scope 2:** Indirect emissions from purchased energy.
- **Scope 3:** Value chain emissions (upstream and downstream).

3: Establish automated data collection via APIs for sources S_{auto}:

$$\text{API.connect}(S_{\text{auto}})$$

4: Enable manual data entry for S_{manual} with built-in validation checks.
5: Validate data S to ensure accuracy:

$$S' = \{s \mid \text{validate}(s) = \text{true}\}$$

6: Normalize S' to a unified schema:

$$\mathcal{N}(S') \rightarrow \text{Preprocessed Data}$$

7: Encrypt and securely store S' in a database:

$$\mathcal{E}(S') \rightarrow \text{Secure Database}$$

Algorithm 2: Emissions Analysis and Insights

Input: Validated data S', emission factors E, compliance thresholds C, historical data H.
Output: Emission metrics D, predictive insights P, reports R, and compliance alerts A.

Algorithm 2 Emissions Analysis and Insights

1: Compute emissions for each data point $s \in S'$:

$$d_i = \sum_{j=1}^{m} f(s_i, e_j), \quad \forall s_i \in S'$$

2: Timestamp emissions data $t(d_i)$.

3: Analyze data $D = \{d_1, d_2, \ldots, d_k\}$ for:

- Trends $\mathcal{T}$: Identify patterns in historical data.
- Anomalies $\mathcal{A}$: Detect deviations or unexpected spikes.

4: Generate predictive models P using:

$$P = \text{Forecast}(H, D)$$

5: Perform scenario modeling $\mathcal{M}$ for emission reduction strategies:

$$\mathcal{M} = \text{Simulate}(D, \text{Reduction Strategies})$$

6: Automate reporting R:

$$R = \text{GenerateReport}(D, \text{Formats: PDF, CSV})$$

7: Monitor progress against thresholds C:

$$A = \{d_i \mid d_i > C_i, \forall d_i \in D\}$$