**Methodology**

**1. Data Collection**

The source of the main data in this study is the Trans-African Hydro-Meteorological Observatory-TAHMO, operating a network of five stations across Kampala. This dataset provided high-resolution records of relative humidity, precipitation, maximum temperature, minimum temperature, and mean temperature, which we used for the flood prediction. These are the meteorological and hydrological variables that precede flood events and have been deemed essential for an accurate urban flood forecast by Musoke et al. (2022) and the International Rescue Committee (2018).

**2. Data Preprocessing**

The preprocessing is also another important aspect of this project. We merged five stations data into one which can represent the whole Kampala. Missing data values were imputed, and outliers were removed to ensure dataset quality. The data was labeled by 7 flood events for the model to learn. Standardization of variables via normalization ensures uniform scales, enhancing the model's performance.

**3. Model Selection**

For flood prediction in Kampala, we utilized Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) models. RNN’s main distribution lies in its ability to model and process sequential data, unlike traditional neural networks that handle inputs independently. Its recurrent nature and hidden state give it a significant edge in tasks requiring temporal understanding, though it comes with challenges in training and efficiency for long sequences. LSTMs are superior to RNNs because they can capture long-term dependencies and mitigate the vanishing gradient problem. This makes LSTM more suitable for handling hourly data, which involves extensive temporal dependencies. For this flood prediction study, LSTMs are optimal due to their robust gating mechanisms, which enable them to learn from detailed, sequential data. LSTM is particularly effective for managing non-linear relationships and avoiding vanishing gradient issues, making it ideal for rainfall prediction tasks (Haidar & Verma, 2018; Babar et al., 2022).

**4. Model Validation**

The dataset was divided into subsets for testing and training, each serving a distinct purpose in model development (Hastie et al., 2009). The training set is used to adjust the model's internal parameters by minimizing the error during learning. The validation set guides the tuning of hyperparameters, such as learning rates, dropout rates, and hidden layer configurations, helping to prevent overfitting and improve generalization. The testing set, at the end, verifies that the model generalizes well when exposed to previously unseen data and works reliably in real-world situations. This formalized approach balances the fitting of data with predictive accuracy.

**5. Model Evaluation**

The performance was evaluated on several key metrics to ensure accuracy and reliability such as the Confusion Matrix. The F1 Score balances precision (correct flood predictions) and recall (detecting most flood events), ensuring the model provides dependable flood event predictions (Chollet, 2017).

**6. Python Libraries**

This project used several Python libraries. Pandas is used for data manipulation and analysis, providing efficient tools for handling structured data. Sklearn (Scikit-learn) is used for machine learning, offering algorithms for classification, regression, clustering, and evaluation metrics. Torch (PyTorch) is also used, which is a deep learning framework that enables building and training neural networks with flexibility. Seaborn and Matplotlib are visualization libraries, with Seaborn specialized in statistical graphics and Matplotlib providing comprehensive plotting capabilities for data analysis.