To generate counterfactuals, we need to establish a conceptual framework, which consists of three main components. The three components are shown in Figure 1.

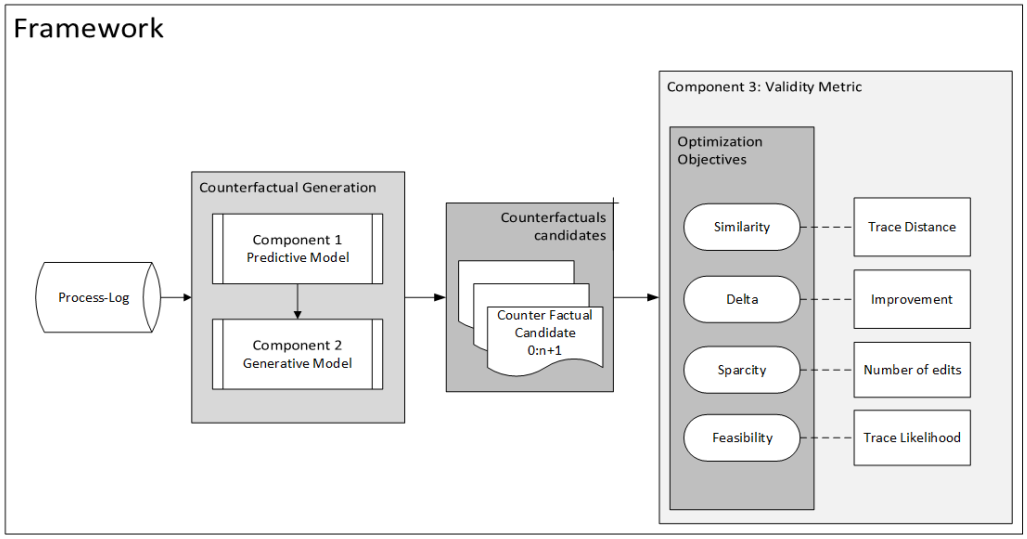

Figure 1: The methodological framework of this thesis. The input is the process log. The log will be used to train a predictive model (Component 1) and the generative model (Component 2). This process produces a set of candidates which are subject to evaluation via the validity metric (Component 3).

The first component is a predictive model. As we attempt to explain model decisions with counterfactuals, the model needs to be pretrained. We can use any model that can predict the probability of a sequence. This condition holds for models trained for process outcome classification and next-activity prediction. The model used in this thesis is a simple LSTM model using the process log as an input. The model is trained to predict the next action given a sequence.

The second component is a generative model. The generative model produces counterfactuals given a factual sequence. In our approach, each generative model should be able to generate a set of counterfactual candidates given one factual sequence. Specifically, we compare an evolutionary approach against 3 different generative baseline approaches. The baselines do not iteratively optimise towards viability criterions. All approaches allow us to use a factual sequence as a starting point for the generative production of counterfactuals. Furthermore, they also generate multiple variations of the final solution.

The generated candidates are subject to the third major component's scrutiny. To select the most *viable* counterfactual candidate, we evaluate their viability score using a custom metric. The metric incorporates all main

criterions for viable counterfactuals mentioned in **??**. We measure the *similarity* between two sequences using a multivariate sequence distance metric. The *delta* between the likelihood of the factual and the counterfactual. For this purpose, we require the predictive model, as it computes a predictions score that reflects the likelihood. We measure *sparsity* by counting the number of changes in the features and computing the edit distance. Lastly, we need to determine the *feasibility* of a counterfactual. This requires splitting the feasibility into two parts. First, the likelihood of the sequence of each event and second, the likelihood of the features given the event that occurred.