Having discussed the previous work on counterfactual sequence generation, a couple of challenges emerge. First, we need to generate on a set of criteria and therefore, require complex loss and evaluation metrics, that may or may not be differiantiable. Second, they cannot to be logically impossible, given the data set. Hence, we have to restrict the space to counterfactuals of viable solutions, while being flexible enough to not just copy existing data instances. Third, using domain knowledge of the process significantly reduces the practicality of any solution. Therefore, we have to develop an approach, which requires only the given log as input while not relying on process specific domain knowledge. This begs the question, whether there is a method to generate sequential counterfactuals that are viable, without relying on process specific domain knowlege. In terms of specific research questions we try to answer:

RQ: Which existing counterfactual approaches can be applied to generate sequences?

- RQ1: Which evaluation metric, reflects the viability of counterfactuals?
- RQ2: To which extend do viable counterfactuals align with domain experts?

We approach these questions, by proposing a schematic framework which allows the exploration of several independent components. shows the conceptual framework of the base approach visually.

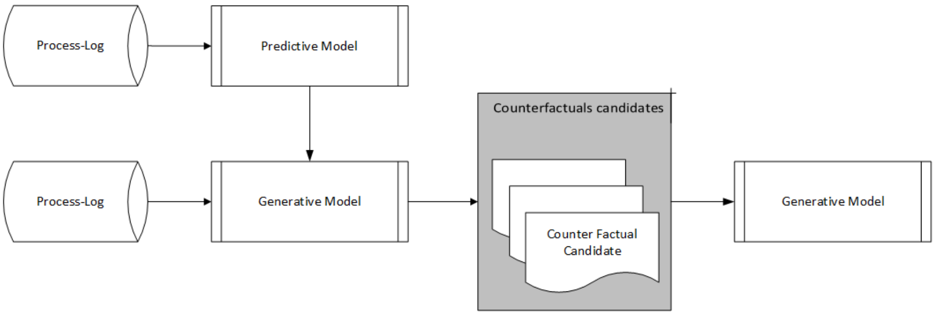

Figure 1: This figure shows a simplified schematic representation of the framework which is explored in this thesis.

The framework contains three parts. First, we need a pretrained predictive component which we aspire to explain. The component should be capable of *accurately* predicting the outcome of a process at any step. The accuracy-condition is favorable but not necessary. If the component is accurately modelling the real world, we can draw real-world conclusions from the explanations generated. If the component is inaccurate, the counterfactuals only explain the prediction decisions and not the real world. The second part requires a generative component. The generative component needs to generate viable sequential counterfactuals which are logically *plausible*. A plausible counterfactual is one whose outcome can be predicted by the predictive component. If the predictive component cannot predict the counterfactual sequence, we can assume that the generative model is *unfaithful* to the predictive component, we want to explain. The third component is the evaluation metric upon which we decide the viability of the counterfactual candidates.

This part is not fully thought out. For the evaluation, we have to test the following hypotheses:

RQ1-H1: If we use a viability function to determine valid counterfactuals, we consistently retrieve more viable counterfactuals, than randomly choosing a counterfacual.

RQ2-H1: The counterfactual generation consistently identifies the most viable counterfactual in the dataset faster than a random search.

RQ2-H2: The generated counterfactual consistently outperforms the most viable counterfactuals among examples in the dataset.

The first hypothesis RQ1-H1 appears to be trivial, if it does not have a probablistic component. Therefore, we need to use a model which probablistically generates the most viable counterfactuals. Hence, we use an evolutionary algorithm, which is intrinsically probablistic.