Department of Mathematics and Computer Science
Process Analytics

The Generation of interpretable counterfactual examples by finding minimal edit sequences using event data in complex processes

Master Thesis

Olusanmi A. Hundogan

Supervisors:
Xixi Lu
Yupei Du
March 14, 2022

Abstract

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Contents

List of terms

BI Business Intelligence. 24, *see:* Business Intelligence

BPM Business Process Management. 8, 24, *see:* Business Process Management

Business Intelligence XXX. 24

Business Process Management XXX. 8, 24

Causal Inference A discipline which seeks to incoporate causal relations to model phenomena in the real world.. 15, 24

Comma Seperated Values A structured data format to store information. Every line relates to a data point and every feature is seperated by a seperator. The seperator is commonly a comma but other characters like tabs or semicolons are valid as well.. 8, 24

Concept Drift The phenomenon of changing processes over time. The changes may occur gradually or sudden and be irregular or recurrent.. 24

Continuous Process Improvement XXX. 8, 24

Corporate Performance Management XXX. 24

CPI Continuous Process Improvement. 8, 24, *see:* Continuous Process Improvement

CPM Corporate Performance Management. 24, *see:* Corporate Performance Management

CSV Comma Seperated Values. 8, 24, *see:* Comma Seperated Values

Data Mining XXX. 8, 24

Deep Learning A sub-discipline of machine learning which focuses on neural networks as primary tool. The discipline emphasises the research and development of neural network architectures. Data preprocessing and feature engineering play a secondary role.. 13, 20, 24

DKF Deep Kalman Filter. 13, 24

DMM Deep Markov Model. 13, 24

ELBO Evidence Lower-Bound. 14, 24

event log A collection of event data, that's produced by the process. They are the main input of every process mining venture.. 8, 19, 24

eXtensible Event Stream An XML-based data format to store event logs. The format was developed and adopted by the IEEE Task Force on Process Mining.. 8, 24

GAN Generative Adversarial Model. 13, 24, *see:* Generative Adversarial Model

Generative Adversarial Model A model in which two neural network train simultaneously. The generative model tries to generate instance examples, while the discriminative models tries to distinguish real examples from generated ones. Both, the discriminator and the generator can be used in isolation afterwards.. 13, 24

HMM Hidden Markov Model. 13, 20, 24

Information System XXX. 8, 24

IS Information System. 8, 24, *see:* Information System

Markov Decision Process A probablistic process which assumes that an agent can influence the outcome of the process by choosing decisions given a state and expecting a return for its performance.. 15, 24

Markov Process A probablistic process whose outcomes depend on the process' state.. 15, 24

MDP Markov Decision Process. 15, 24, *see:* Markov Decision Process

ML Machine Learning. 3, 24

MP Markov Process. 15, 24, *see:* Markov Process

Natural Language Processing A discipline that is mainly concerned with the analysis and modelling of natural language.. 9, 24

NLP Natural Language Processing. 9, 12, 13, 24, *see:* Natural Language Processing

PGM Probalistic Graphical Model. 19, 24

PM Process Mining. 15, 24, *see:* Process Mining

process event Also called activities. A discrete step in the process.. 7, 24

process instance Also called case. A collection of activities that belong to a common entity that is produced by the process.. 7–9, 12, 19, 24

Process Mining A subdiscipline of Data Mining, which uses process logs to analyse and utilise data which was produced by processes.. 15, 24

Rashommon Effect Rashomon is a classic Japanese movie in which multiple witnesses tell a different equally valid story about the murder of a samurai. Although, each story acts as a valid explanation, they contradict each other. The same effect may apply to equally valid counterfactuals.. 12, 24

Reinforcement Learning An are within Machine Learning, which seeks to allow an intelligent agent to choose the right actions by maximizing the cumulative rewards of a task setting.. 15, 24

RL Reinforcement Learning. 15, 24, *see:* Reinforcement Learning

SCM Structural Causal Model. 15, 24, *see:* Structural Causal Model

Structural Causal Model Also called *Structural Equation Model (SEM)*. An SCM is a set of variables and equations. The equations' the relationship of outputs from other equations and the variables. The relationship between functions and variables inform the causal relationship. SCMs can be represented as directed graphs in which nodes describe variables and equations and the vertices dependencies. Furthermore, a SCM can produce a dataset containing all the values thhat where generated.. 15, 24

Total Quality Management XXX. 24

TQM Total Quality Management. 24, *see:* Total Quality Management

VAE Variational Autoencoder. 24

XAI eXplanable AI. 3, 10, 11, 15, 24

XES eXtensible Event Stream. 8, 24, *see:* eXtensible Event Stream

Chapter 1

Introduction

1.1 Motivation

Many processes, often medical, economical, or administrative in nature, are governed by sequential events and their contextual environment. Many of these events and their order of appearance play a crucial part in the determination of every possible outcome. With the rise of AI and the increased abundance of data in recent years several techqniques emerged that help to predict the outcomes of complex processes in the real world.

Research in the Process Mining discipline has shown that is possible to predict the outcome of a particular process fairly well CITE . For instance, in the medical domain, models have been shown to predict the outcome or trajectory of a patient's condition CITE . In the private sector, process models can be used to detect faults or outliers. Deep Learning, in particular, has shown promising results within domains that have been considered difficult for decades CITE Moravec Paradox . However, while many prediction models can easily predict certain outcomes, it remains a difficult challenge to understand their reasoning. This difficulty arises from models, like neural networks, that are so-called blackbox models. Meaning, that their inference is imcomprehensible, due to the vast amount of parameters involved. This lack of comprehension is undesirable for many fields like IT or finance. Not knowing why a loan was given, makes it impossible to rule out possible biases. Knowing what will lead to a system failure, will help us knowing how to avoid it. In critical domains like medicine, the reasoning behind decisions become crucial. For instance, if we know that a treatment process of a patient reduces the chances for survival, we want to know which treatment step is the critical factor we ought to avoid. To summarise, knowing the outcome of a process often leads us to questions on how to change it. Formally, we

want to change the outcome of a process, by making it maximally likely, with as little interventions as possible CITE . Figure 1.1 is a visual representation of the desired goal.

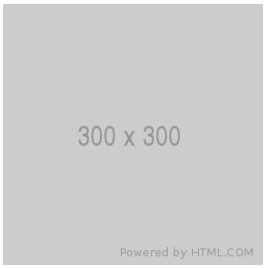

Figure 1.1: This figure illustrates a model, that predicts a certain trajectory of the process. However, we want to change the process steps in such a way, that it changes the outcome.

One-way to better understand the Machine Learning (ML) models lies within the eXplanable AI (XAI) discipline. XAI dedicates its research to the research and development of blackbox models. Most of the discipline's techniques produce explanations that guide our understanding. Explanations can come in various forms, such as [examples] CITE , but some are more comprehensible for humans than others.

A prominent and human-friendly approach are counterfactuals CITE . Counterfactuals within the AI framework help us to answer hypothetical "what-if" questions. Basically, if we know *what* would happen *if* we changed the process, we could change it for the better. In this thesis, we will raise the question, how we can use counterfactuals to change the trajectory of a process models' prediction towards a desired outcome. Knowing the answers will not only increase the understanding of blackbox models, but also help us avoid or enforce certain outcomes.

1.2 Problem Space

In this paper, we will approach the problem of generating counterfactuals for processes. The literature has provided a multitude of techniques to generate counterfactuals for AI models, that are derived from static data[1]. However, little research has focussed on counterfactuals for dynamic data[2]. A major

[1]With static data, we refer to data that does not change over a time dimension.

[2]With dynamic data, we refer to data that has time as a major component, which is also inherently sequential

reason, emerges from a multitude of challenges, when dealing with counterfactuals and sequences.

Three of these challenges are particularly important. First, counterfactuals within AI attempt to explain outcomes which never occured. A *what-if* questions often refer to hypothetical scenarios. Therefore, there is no evidential data, from which we can infer predictions. Subsequently, this lack of evidence further complicates the evaluation of generated counterfactuals. In other words, you cannot validate the correctness of a theoretical outcome that has never occured. Second, sequential data is highly variable in length, but processes steps have complicated factors, too CITE . The sequential nature of the data impedes the tractability of many problems due to the combinatorial explosion of possible sequences. Furthermore, the data generated is seldomly one-dimensional or discrete. Henceforth, each dimension's contribution can vary in dependance of its context, the time and magnitude. Third, process data often requires knowledge of the causal structures that produced the data in the first place. However, these structures are often hidden and it is a NP-hard problem to elicit them CITE Check process discovery literature . Just these challenges, makes the field in which we can contribute a vast endeavor.

1.3 Outline

Due to the challenges imposed by process data, we have to restrict the solution space by imposing limitations and assumptions. By knowing the restrictions, we can train a predictive model and compare counterfactual generation methods. We evaluate the generated counterfactuals based on their *validity*. In short, *valid* counterfactuals are those that help us understand a predictive model better. Both, restrictions and validity are further discussed in chapter 2. For the counterfactual generation process, we limit our focus on exploring an evolutionary computing approaches for their ability to optimise non-differential criterions and deep generative models as their samples are directly informed by the data distribution. The reasons for these approaches, become clear in section 2.4.

Chapter 2

Background

2.1 Process Mining

This thesis will focus on processes and the modelling of process generated data. Hence, it is important to establish a common understanding for this field.

2.1.1 A definition for Business Processes

Before elaborating on Process Mining, we have to establish the meaning of the term *process*. The term is widely-used and therefore has a rich semantic volume. A process generally refers to something that advances and changes over time[6]. Despite, legal or biological processes being valid understandings, too, we focus on *business processes*.

An example is a loan application process in which an applicant may request a loan. The case would then be assessed and reviewed by multiple examiners and end in a final decision. The loan might end up in an approval or denial. The *business* part is misleading as these processes are not confined to commercial settings alone. For instance, a medical business process may cover a patients admission to a hospital, followed by a series of diagnostics and treatments and ending with the recovery or death of a patient. Another example from a Human Computer Interaction (HCI) perspective would be an order process for an online retail service like Amazon. The buyer might start the process by adding articles to the shopping cart and proceeding with specifying their bank account details. This order process would end with the submission or receival of the order.

All of these examples have a number of common characteristics. They have a clear starting point which is followed by numerous intermediary steps and end in one of the possible sets of outcomes. For this work we will mainly

follow the understanding outlined in W. van der Aalst et al.[25]. Each step, including start and end points, is an process event which was caused by an *activity*[1]. Each process event may contain additional information in the form of event attributes. If a collection of events *sequentially* relate to a single run through a process, we call them *process instance* or *trace*. These instances do not have to be completed. Meaning, the trace might end prematurely. In line with the aforementioned examples, these process instances could be understood as a single loan application, a medical case or a buy order. We can also attach process instance related information to each instance. Examples would be the applicants location, a patients age or the buyers budget. In its entirety, a business process can be summarised as a *graph*, a *flowchart* or another kind of visual representation. Figure 2.1's graphical representation is an example of such a *process map*[25].

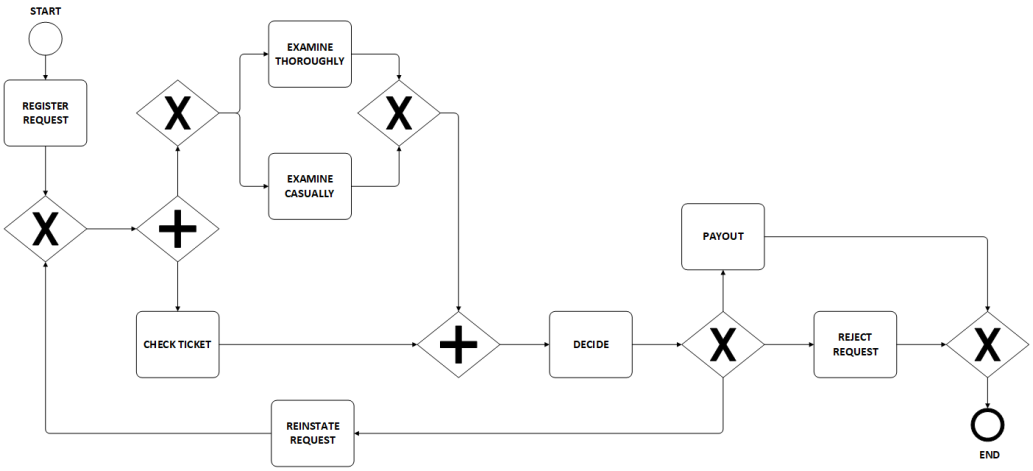

Figure 2.1: This graph shows an example of a BPMN process map.

In conclusion, in this thesis a *business process* refers to

> *A finite series of discrete events with one or more starting points, intermediary steps and end points. Each intermediate step has at least one precedent and at least one antecedent step.*

However, we have to address a number of issues with this definition.

First, it excludes infinite processes like **[EXAMPLE NEEDED]** or continuous processes such as **[EXAMPLE NEEDED]**. There may be valid arguments to include processes with these characteristics, but they are not relevant for this thesis.

[1] In this paper we will use the terms *event* and *activity* interchangebably as their practical difference is quite subtle.

Second, in each example, we deliberately used words that accentuate modality such as *may*, *can* or *would*. It is important to understand that each process anchors its definition within an application context. Hence, what defines a business process is indisputably subjective. For instance, while an online marketplace like Amazon might be interested in the process from the customers first click to the successful shipment, an Amazon vendor might only be interested in the delivery process of a product.

Third, the example provided in Figure 2.1 may not relate to the *real* underlying data generating process. As process *models* are inherently simplified, they may or may not be accurate. The *true* process is often unknown. Therefore, we will distinguish between the *true process* and a *modelled process*. The *true process* is a hypothetical concept whose *true* structure remains unknown. In, contrast, a process *model* simplifies and approximates the characteristics of the *true process*.

2.1.2 What is Process Mining?

Having established a definition for a process, we next discuss *Process Mining*. This young discipline has many connections to other fields that focus on the modeling and analysis of processes such as Continuous Process Improvement (CPI) or Business Process Management (BPM)[25]. However, its data-centric approaches originate in Data Mining. The authors W. van der Aalst et al. describe this field as a discipline "to discover, monitor and improve real processes (i.e., not assumed processes) by extracting knowledge from event logs readily available in today's (information) systems"[25]. The discipline revolves around the analysis of event logs. A event log is a collection of process instances, which are retrieved from various sources like an Information System (IS) or database. Logs are often stored in data formats such as Comma Seperated Values (CSV) or eXtensible Event Stream (XES) **CITE** .

2.1.3 The Challenges of Process Mining

As mentioned in chapter 1, process data modelling and analysis is a challenging task. W. van der Aalst et al. mentions a number of issues that arise from processes[25].

The first issue arises from the quality of the data set. Process logs are seldomly collected with the primary goal of mining information and hence, often appear to be of subpar quality for information mining purposes. The information is often incomplete, due to a lack of context information, the ommision of logged process steps, or wrong levels of granularity[25].

This issue is exacerbated by the second major issue with process data. Mainly, its complexity. Not only does a process logs' complexity arise from the variety of data sources and differing levels of complexity, but also from the datas' characteristics. The data can often be viewed as multivariate sequence with discrete and continuous features and variable length. This characteristic alone creates problems explored in section 2.2. However, the data is also just a *sample* of the process. Hence, it may not reflect the real process in its entirety. In fact, mining techniques need to incorporate the *open world assumption* as the original process may generate unseen process instances[25].

A third issue which contributes to the datasets' incompleteness and complexity is a phenomenon called *concept drift*. This phenomenon relates to the possibility of changes in the *true* process. The change may occur suddenly or gradually and can appear in isolation or periodically. An expression of such a drift may be a sudden inclusion of a new process step or domain changes of certain features. These changes are not uncommon and their likelihood increases with the temporal coverage and level of granularity of the dataset CITE . In other words, the more *time* the dataset covers and the higher its detail, the more likely a change might have occured over the time.

All three issues relate to the *representativeness* of the data with regards to the unknown *true* process that generated the data. However, they also represent open challenges that require research on their own. For our purpose, we have to assume that the data is representative and its underlying process is static. These assumptions are widely applied in the body of process mining literature CITE .

2.2 Multivariate Time-Series Modelling

The temporal and multivariate nature of process instance often turns Process Mining into a Multivariate Time-Series Modelling problem. Therefore, it is necessary to establish an understanding for this type of data structure.

The data which is mined in Process Mining is typically a multivariate time-series. It is important to establish the characteristics of time-series.

2.2.1 What are Time Series Models?

A time series can be understood as a series of observable values and depend on previous values. The causal dependence turns time-series into a special case of sequence models. Sequences do not *have to* depend on previous values. They might depend on previous and future values or not be interdependent

at all. An example of a sequence model would be a language model. Results in Natural Language Processing (NLP), that the words in a sentences for many languages do not seem to only depend on prior words but also on future words CITE . Hence, we can assume that a human has formulated his sentence in the brain before expressing it in a sequence of words CITE . In contrast to sequences, time series cannot depend on future values. The general understanding of *time* is linear and forward directed CITE . The notion of time relates to our understanding of *cause and effect*. Hence, we can decompose any time series in a precedent (causal) and an antecedent (effect) part (CHECK [14]). A time series model attempts to capture the relationship between precedent and antecedent.

2.2.2 The Challenges of Time Series Modelling

The analysis of unrestricted sequential opens up a myriad of challenges. First, sequential data introduces a combinatorial set of possible realisations (often called *productions*). For instance, a set of two objects $\{A, B\}$ *produces* 7 theoretical combinations ($\{\emptyset\}$, $\{A\}$, $\{B\}$, $\{A, B\}$, $\{B, A\}$, $\{A, A\}$, $\{B, B\}$). Just by adding C and then D to the object set increases the number of combinations to 40 and 341 respectively. Second, sequential data may contain cyclical patterns which increase the number of possible productions to infinity CITE . Both, the combinatorial increase and cycles, yield a set of a countable infinite number of possible productions. However, as processes may also contain additional information a third obstacle arises. Including additional information extends the set to an uncountable number of possible productions. With these obstacles in mind, it often becomes intractable to compute an exact model.

Hence, we have to include restrictive assumptions to reduce the solution space to a tractable number. A common way to counter this combinatorial explosion is the inclusion of the *Granger Causality* assumption CITE see Anastasiou's refererences for G . This idea postulates the predictive capability of a sequence given its preceding sequence. In other words, if we know that C must be followed by D, then 341 possible combinations reduce to 156. All of these possible 156 combinations are now temporally-related and hence, we speak of a *time-series*.

However, the prediction of sequences recontextualises the issue two new questions: First, if we know the precedence of a time-series, what is the antecedent? And second, if we can predict the antecedent accurately, what caused it? We often use data-driven AI-methods like Hidden-Markov-Models or Deep Learning to solve the first question. However, the second question is more subtle. At first glance, it is easy to believe that both questions are quite similar, because we could assume that the precedent causes the

antecedent. Meaning, that we can use the data available to elicit sequential correlative patterns. In reality, the latter question is much more difficult as data often does not include any information about the **inter-relationships**. To illustrate this difficulty, we could say that the presence of C causes D. But if D also appears valid in a sequence 'AABD', it cannot be caused by the presence of C alone.

Answering this question requires additional tools within the XAI framework. One such method is the focus of this thesis and is further explored in section 2.3.

2.3 Counterfactuals

Counterfactuals are an important explanatory tool to understand a models' cause for decisions. Generating counter factuals is main focus of this thesis. Hence, we will establish the most important chateristics of counterfactuals in this section.

2.3.1 What are Counterfactuals?

Counterfactuals have various definitions. However, their semantic meaning refers to "a conditional whose antecedent is false"[4]. A simpler definition from Starr states, counterfactual modality concerns itself with *what is not, but could or would have been.* Both definitions are related to linguistics and philosophy. Within AI and the mathematical framework various formal definitions can be found within causal inference[8]. Here, citeauthor describes a counterfactual as Causal inference definition. What binds all of these definitions is the notion of causality within "what if" scenarios.

However, for this paper, we will use the understanding established within the XAI context. Within XAI, counterfactuals act as a prediction which "describes the smallest change to the feature values that changes the prediction to a predefined output"[17]. Note that XAI mainly concerns the explanation of models, which are always subject to inductive biases of the model itself and therefore inherently subjective. The idea behind counterfactuals as explanations[2] is that we understand the output of a model, if we know what change caused would cause a different outcome. For instance, lets denote a sequence 1 as $ABCDE\boldsymbol{FG}$, then a counterfactual $ABCDE\boldsymbol{XZ}$ would tell us that **F** (probably) caused **G** in sequence 1. As counterfactuals only address

[2]There are other explanatory techniques in XAI like *feature importances* but counterfactuals are considered the most human-understandable

explanations of single model instances and not the model as a whole, they are called *local* explanation.

Valid counterfactuals satisfy four criteria. First, a counterfactual should be minimally different from the true instance. If the counterfactual to sequence 1 was *A**A**CDE**XZ*** we would already have difficulties to discern whether B or F or both caused G at the end of sequence 1. Second, a counterfactual should produce a predefined outcome as closely as possible. This characteristic is ingrained in Molnars definition. If the counterfactual *ABCDE**XZ*** ends with Z but this sequence is highly unrealistic, then cannot be certain of our conclusion for sequence 1. Third, we typically desire multiple diverse counterfactuals. One counterfactual might not be enough to understand the causal relationships in a sequence. In the example above we might have a clue that F causes G but what if G is not only caused by F? If we are able to find counterfactuals ***V**BCDEF**H*** and *ABCDE**XZ*** but all other configurations lead to G, then we know positions 1 and 6 cause G. As last criterion, each counterfactual should be possible. A sequence *ABCDE**1G*** would not be possible if numericals are not allowed. All four criteria allow us to assess the validity of each generated counterfactual and thus, help us to define an evaluation metric.

2.3.2 The Challenges of Counterfactual Sequence Generation

The current literature surounding counterfactuals expose a number of challenges when dealing with counterfactuals.

The most important disadvantage of counterfactuals is the Rashommon Effect[17, ch.9.3]. If all of the counterfactuals are valid, but contradict eachother, we have to decide which of the *truths* are worth considering.

This decision reveals the next challenge of evaluation CITE . Although, the criteria can support us with the decision, it remains a question *how* to evaluate counterfactuals. Every automated measure comes with implicit assumptions and often do not guarantee a realistic explanation. We still need domain experts to assess their validity.

The generation of counterfactual sequences contribute to both former challenges, due to the combinatorial expansion of the solution space. This problem is common for counterfactual sentence generation and has been adressed within the NLP CITE . However, as process mining data not only consist of discrete objects like *words*, but also event and case features, the problem remains a daunting task. So far, little work has gone into the generation of multivariate counterfactual sequences like process instances CITE .

2.4 Related Literature

Many researchers have worked on counterfactuals and process mining. Here, we will combine the important concepts and discuss the various contributions to this thesis.

2.4.1 Generating Counterfactuals

The topic of counterfactual generation as explanation method was introduced by Wachter et al. in 2017 CITE . The authors defined a loss function which incorporates the criteria to generate a counterfactual which maximizes the likelihood for a predefined outcome and minimizes the distance to the original instance. However, the solution of Wachter et al. did not account for the minimalisation of feature changes and does not penalize unrealistic features. Furthermore, their solution cannot incorporate categorical variables.

A newer approach by Dandl et al. incoporates all four main criteria for counterfactuals (see section 2.3) by applying a genetic algorithm with a fitness function that incorporates all of the main criteria CITE . This approach strongly differs from gradient-based methods, as it does not require a differentiable function which requires optimization. However, their solution only works with structured data.

2.4.2 Generating Counterfactual Sequential

When it comes to sequential data most researchers work on ways to generate counterfactuals for natural language. This often entails generating univariate discrete counterfactuals with the use of Deep Learning techniques. Martens and Provost and later Krause et al. are early examples of counterfactual NLP research. Their approach strongly focuses on the manipulation of sentences to achieve the desired outcome. However, as Robeer et al. puts it, their counterfactuals do not comply with *realisticness*.

Instead, Robeer et al. showed that it is possible to generate realistic counterfactuals with a Generative Adversarial Model (GAN). They use the model to implicitly capture a latent state space and sample counterfactuals from it. Apart from implicitly modelling the latent space with GANs, it is possible to sample data from an explicit latent space. Examples of these approaches often use an encoder-decoder pattern in which the encoder encodes a data instance into a latent vector, which will be peturbed and then decoded into a a similar instance[16][27]. By modelling the latent space, we can simply sample from a distribution conditioned on the original instance. Bond-Taylor et al.

provides an overview of the strengths and weaknesses of common generative models.

Eventhough, a single latent vector model can theoretically produce multivariate sequences, it may still be too restrictive to capture the combinatorial space of multivariate sequences. Hence, most of the models within NLP were not used to produce a sequence of vectors, but a sequence of discrete symbols. For process instances, we can assume a causal relation between state vectors in a sequential latent space. We call models that capture a sequential latent state-space which has causal relations *dynamic*[14]. Early models of this type of dynamic latent state-space models are the well-known *Kalman-Filter* for continous states and Hidden Markov Model (HMM) for discrete states. In recent literature, many techniques use Deep Learning to model complex state-spaces. The first models of this type were developed by Krishnan et al. Their Deep Kalman Filter (DKF) and subsequent Deep Markov Model (DMM) approximate the dynamic latent state-space by modeling the latent space given the data sequence and all previous latent vectors in the sequence. There are many variations of Krishnan et al.'s model, but most use Evidence Lower-Bound (ELBO) of the posterior for the current Z_t given all previous $\{Z_{t-1}, \ldots, Z_1\}$ and X_t.

2.4.3 Generating Counterfactual Time-Series

Within the *multivariate time-series* literature two recent approaches yield ideas worth discussing.

First, Delaney et al. introduces a case-based reasoning to generate counterfactuals. Their method uses existing counterfactual instances, or *prototypes*, in the dataset. Therefore, it ensures, that the proposed counterfactuals are *realistic*. However, case-based approaches strongly depend on the *representativeness* of the prototypes CITE . In other words, if the model displays behaviour, which is not capture within the set of prototypical instances, most case-based techniques will fail to provide valid counterfactuals. The likelihood of such a break-down increases due to the combinatorial explosion of possible behaviours if the *true* process model has cycles or continuous event attributes. Cycles may cause infinite possible sequences and continous attributes can take values on a domain within infinite negative and positive bounds. These issues have not been explored in the paper of Delaney et al., as it mainly deals with time series classification. However, despite these shortcomings, case-based approaches may act as a valuable baseline against other sophisticated approaches.

The second paper within the multivariate time series field by Ates et al. also uses a case-based approach. However, it contrasts from other approaches,

as it does not specify a particular model but proposes a general framework instead. Hence, within this framework, individual components could be substituted by better performing components. Describing a framework, rather than specifying a particular model, allows to adapt the framework, due to the heterogeneous process dataset landscape. In this paper, we will also introduce a framework that allows for flexibility depending on the dataset. **The framework will be evaluated in two steps. The first step aims to compare various model types against eachother based on the countefactual validiy. The second step scrutinizes the best framework configurations from step one, by presenting its results to a domain expert.**

2.4.4 Generating Counterfactuals for Business Processes

So far, none of the models have been applied to process data.

Within Process Mining (PM), Causal Inference has long been used to analyse and model business processes. Mainly, due to the causal relationships underlying each process. However, early work has often attempted to incorporate domain-knowledge about the causality of processes in order to improve the process model itself[2, 9, 22, 28]. Among these, Narendra et al. approach is one of the first to include counterfactual reasoning for process optimization. Oberst and Sontag use counterfactuals to generate alternative solutions to treatments, which lead to a desired outcome. Again, the authors do not attempt to provide an explanation of the models outcome and therefore, disregard multiple **[criterions for viable counterfactuals]** in XAI. [20] published the most recent paper on the counterfactual generation of explanations.**[Explain their approach.]** **The authors, use a known Structural Causal Model (SCM), to guide the generations of counterfactuals. However, this approach requires a process model which is as close as possible to the *true* process model. For our approach, we assume that no knowledge about the dependencies are known.**

Within the XAI context, Tsirtsis et al. develop the first explanation method for process data. However, their work closely resembles the work of Oberst and Sontag and treat the task as Markov Decision Process (MDP). This extension of a regular Markov Process (MP) assumes that an actor influences the outcome of a process given the state. This formalisation allows the use of Reinforcement Learning (RL) methods like Q-learning or SARSA **CITE .** However, this often requires additional assumptions such as a given reward function and an action-space. For counterfactual sequence generation, there is no obvious choice for the reward function or the action-space. Nonetheless, both Tsirtsis et al. and Oberst and Sontag contribute

an important idea to incrementally generate the counterfactual. Hsieh et al. build on this concept by proposing a system that generates counterfactuals milestone-wise. Their work is the closest to our approach. The authors recognised that some processes have critical events, which govern the overall outcome. Hence, by simply avoiding the undesired outcome from milestone to milestone, it is possible to limit the search space and compute valid counterfactuals with counterfactual generation methods, such as the DiCE alogorithm. However, their approach presupposes that the critical event points are known and the outcome is binary. Especially, the first condition makes their approach heavily dependent on the data which is used.

2.5 Research Question

As we seek to make data-driven process models interpretable, we have to understand the exact purpose of this thesis. Hence, we will establish the challenges that are open and how this thesis attempts to solve them.

Having discussed the previous work on generating counterfactual sequences, 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 to viable solutions, while being flexible enough to not just be copies of already 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 process-agnostic method to generate sequential counterfactuals that are viable. In terms of specific research questions we try to answer:

RQ: Is there a process-agnostic method to generate viable counterfactuals?

- RQ1: Is there an evaluation metric, which reflects the viability of counterfactuals?
- RQ2: Is it possible to generate viable counterfactuals, even if the counterfactual has not been present in the data set?
- RQ3: Can we generate viable counterfactuals which are logically plausible?

We approach these questions, by proposing a schematic framework which allows the exploration of several independent components. The framework

contains three parts. First, we need a predictive component which needs to be explained. The component should be capable of accurately predicting the outcome of a process at any step. This 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 that the predictive component can predict. 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. Figure 2.2 displays this approach visually.

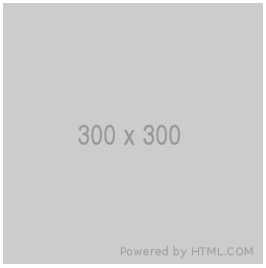

Figure 2.2: This figure shows a schematic representation of the framework which is explored in this thesis.

2.6 Formal Definitions

Before diving into the remainder of this thesis, we have to establish preliminary definitions, we use in this work. With this definitions, we share a common formal understanding of mathematical descriptions of every concept used within this thesis.

2.6.1 Process Logs, Cases and Instance Sequences

We start by formalising the log and its elements. Let $\mathcal{E}$ be the universe of event identifiers and $E \subseteq \mathcal{E}$ a set of events. Let C be a set of case identifiers and $\pi_\sigma : E \mapsto C$ a surjective function that links every element in E to a case $c \in C$ in which c signifies a case. For a set of events $E \subseteq \mathcal{E}$, we use a

shorthand $\sigma \in C$ being a particular sequence $\sigma^c = \langle e_1, e_2, \ldots, e_t \rangle$ as a case of length t.

Furthermore, let $\mathcal{T}$ be the time domain and $\pi_t : E \mapsto \mathcal{T}$ a surjective linking function which strictly orders a set of events.

We assume that every e contains a set of data attributes. Hence, let $A = a_1, \ldots, a_i$ be a set of attribute names (e.g., timestamp, resource, action, etc.) and $\mathcal{F}$ be the universe of each attribute's representations $F \in \mathcal{F}$. For each element in A, we have a surjective function $\pi_{a_i} : A \mapsto F_i$ which links each event attribute a_i to its value $f_i \in F_i$. Each attribute value is represented with a vector space of varying dimensions $f_i \in R^d$.

We represent each event as a concatenated vector of its attribute values as $e_t^c = [f_1^c; f_2^c; \ldots; f_i^c]$, in which c specifies a particular case, a specific event within the case at time t and i its event attributes.

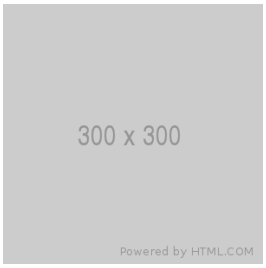

Figure 2.3: This figure show the representation of a particular event e and a case c.

2.6.2 State-Space Models

Generally speaking, every time-series can be represented as a state-space model[11]. Within this framework the system consists of *input states* for *subsequent states* and *subsequent outputs*. A mathematical form of such a system is shown in Equation 2.1.

$$\begin{aligned} \mathbf{z}_{t+1} &= \mathbf{h}(t, \mathbf{z}_t, \mathbf{u}_t) \\ \mathbf{x}_t &= \mathbf{g}(t, \mathbf{z}_t, \mathbf{u}_t) \\ \dot{\mathbf{x}}(t) &:= \frac{d}{dt}\mathbf{x}(t) \end{aligned} \tag{2.1}$$

[Here, u represents the input, x the state, t the time. The function g maps t, $z(t)$ and $u(t)$ to the next state $z(t+1)$. x acts

as an output computed by function f which takes the same input as h. The variables z, u, t and y are vectors with discrete or continuous features. The distinction of $z(t+1)$ and $x(t)$ decouples *hidden*[3] states, from *observable* system outputs.] Figure 2.4 shows a graphical representation of these equations.

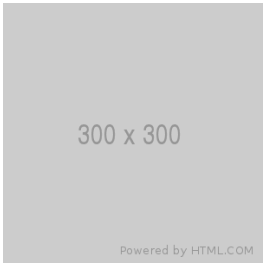

Figure 2.4: This figure shows a simplified graphical representation of a state-space model. Each arrow represents the flow of information.

The body of literature for state-space models is too vast to discuss it in detail[4]. However, for process mining we can use this representation to discuss the necessary assumptions with regards to process mining. In line with the process-definition in section 2.1, we can understand the event log as a collection of observable outputs of a state-space model. The state of the process is hidden as the *true* process which generated the data cannot be observed as well. The time t is a step within the process. Hence, we will treat t as a discrete skalar value to denote discrete sequential time steps. The input u represents all context information of the process. Here, u subsumes observable information such as the starting point and process instance-related features. The functions h and g determine the transition of a process' state to another state and its output over time. **Note, that this formulation disregards any effects of future timesteps on the current timestep.** As we establish in section 2.1, we can assume that a process is a discrete sequence, whose transitions are time-variant. In this framework, the goal becomes to identify the parameters of the functions h and g. Knowing the functions it is simple to infer valid counterfactuals. However, the function parameters are often unknown and therefore, require probablistic approaches.

We can formulate Equation 2.1 probablistically as shown in Equation 2.2.

[3] A state does not have to be hidden. Especially, if we know the process and the transition rules. However, those are often inaccessible if we only use log data. Instead, many techniques try to approximate the hidden state given the data instead.

[4] For an introduction to state-space models see: XXX

$$\mathbb{E}[p\left(z_{t+1} \mid t, z_{1:T}, u_{1:T}, x_{1:T}, \theta_h\right)] = \int z_{t+1} \cdot p\left(z_{t+1} \mid t, z_{1:T}, u_{1:T}, x_{1:T}, \theta_h\right) \tag{2.2}$$

$$\mathbb{E}[p\left(x_t \mid t, z_{1:T}, u_{1:T}, \theta_g\right)] = \int x_t \cdot p\left(x_t \mid t, z_{1:T}, u_{1:T}, \theta_g\right)$$

Note, that h and g are substitued with probability density functions parametrized with θ_h and θ_g. T signifies the full sequence including future timesteps. Both expectations are intractable as they require integrating over n-dimensional vectors. To solve the intractability, we characterize the system as a *Hidden Markov Process* and Probalistic Graphical Model (PGM). This framework allows us to leverage simplifying assumptions such as the independece from future values and *d-seperation*. The stochastic process is shown in Figure 2.5.

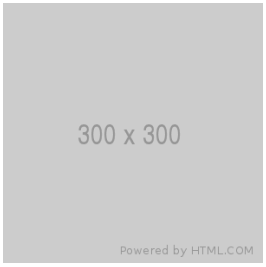

Figure 2.5: Figure shows a graphical representation of the stochastic process.

These characteristics change the probabilities in Equation 2.2 to Equation 2.3:

$$p\left(z_{t+1} \mid z_{1:t}, u_{1:t}, \theta_h\right) = \prod_{1}^{t} p\left(z_t \mid z_{1:t}, u_t, \theta_h\right) \tag{2.3}$$

$$p\left(x_t \mid z_{1:t}, \theta_g\right) = \prod_{1}^{t} p\left(x_{t-1} \mid z_{1:t}, \theta_f\right) \tag{2.4}$$

For $p\left(z_{t+1} \mid t, z_{1:T}, u_{1:T}, x_{1:T}, \theta_h\right)$, we ignore future timesteps, as T changes into t. *d-seperation* allows us to ignore all x of previous timesteps. The graphical form also decomposes the probability into a product of probabilities that each depend on all previous states and its current inputs. Previous x_t are ignored due to *d-seperation*. $p\left(x_t \mid t, z_{1:T}, u_{1:T}, \theta_g\right)$ only depends on its current state, which is in line with HMMs. Note, that we deliberately not

assume a *strong Markov Property*, as the Deep Learning-Framework allows us to take all previous states into account. The *strong Markov Property* would assume that only the previous state suffices. At last, we assume that we do not model automatic or any other process whose state changes without a change in the input or previous states. Hence, we remove the dependency on the independet t variable. Only the previous states $z_{1:T}$ and the input information u_t remain time-dependent.

In this probablistic setting, the generation of counterfactuals, amounts to drawing samples from the likelihood of Equation 2.3. We then use the samples to reconstruct the most-likely a counterfactual $x^*_{1:t}$. Hence, our goal is to maximize both likelihoods.

[A number of AI techniques where developed to model this representation bla bla bla (HMM, Kalman, etc – Has further formalisation).]

Chapter 3

Methods

3.1 Validity Measure

3.2 Trace Distance Measure

3.3 Models

3.3.1 Framework Architecture

3.3.2 Predictive Model: LSTM

3.3.3 Predictive Model: Transformer

3.3.4 Generative Model: Case-Based Approach

3.3.5 Generative Model: Evolutionary Approach

3.3.6 Generative Model: Deep Generative Approach

3.4 Datasets

3.4.1 Datasets

3.4.2 Preprocessing

Chapter 4

Results

4.1 Evaluation Procedure

4.1.1

Chapter 5

Discussion

Chapter 6

Conclusion

Bibliography

Ates, E., Aksar, B., Leung, V. J., & Coskun, A. K. (2021, May 19). Counterfactual Explanations for Multivariate Time Series. In *2021 International Conference on Applied Artificial Intelligence (ICAPAI)* (pp. 1–8). 2021 International Conference on Applied Artificial Intelligence (ICAPAI). doi:10.1109/ICAPAI49758.2021.9462056

Baker, J., Song, J., & Jones, D. R. (2017). Closing the Loop: An Empirical Investigation of Causality in IT Business Value. *undefined*. Retrieved March 1, 2022, from https://www.semanticscholar.org/paper/Closing-the-Loop%3A-An-Empirical-Investigation-of-in-Baker-Song/df210060211bdc598f2d3382c68c615319287f71

Bond-Taylor, S., Leach, A., Long, Y., & Willcocks, C. G. (2021, April 14). Deep Generative Modelling: A Comparative Review of VAEs, GANs, Normalizing Flows, Energy-Based and Autoregressive Models. arXiv: 2103.04922 [cs, stat]. Retrieved October 1, 2021, from http://arxiv.org/abs/2103.04922

Counterfactual. (n.d.). doi:10.1093/oi/authority.20110803095642948

Dandl, S., Molnar, C., Binder, M., & Bischl, B. (2020). Multi-Objective Counterfactual Explanations. In T. Bäck, M. Preuss, A. Deutz, H. Wang, C. Doerr, M. Emmerich, & H. Trautmann (Eds.), *Parallel Problem Solving from Nature – PPSN XVI* (pp. 448–469). doi:10.1007/978-3-030-58112-1_31

Definition of PROCESS. (n.d.). Retrieved February 17, 2022, from https://www.merriam-webster.com/dictionary/process

Delaney, E., Greene, D., & Keane, M. T. (2021). Instance-Based Counterfactual Explanations for Time Series Classification. In A. A. Sánchez-Ruiz & M. W. Floyd (Eds.), *Case-Based Reasoning Research and Development* (pp. 32–47). doi:10.1007/978-3-030-86957-1_3

Hitchcock, C. (2020). Causal Models. In E. N. Zalta (Ed.), *The Stanford Encyclopedia of Philosophy* (Summer 2020). Metaphysics Research Lab, Stanford University. Retrieved February 10, 2022, from https://plato.stanford.edu/archives/sum2020/entries/causal-models/

Hompes, B. F. A., Maaradji, A., La Rosa, M., Dumas, M., Buijs, J. C. A. M., & van der Aalst, W. M. P. (2017). Discovering Causal Factors Explaining Business Process Performance Variation. In E. Dubois & K. Pohl (Eds.), *Advanced Information Systems Engineering* (pp. 177–192). doi:10.1007/978-3-319-59536-8_12

Hsieh, C., Moreira, C., & Ouyang, C. (2021, October 31). DiCE4EL: Interpreting Process Predictions using a Milestone-Aware Counterfactual Approach. In *2021 3rd International Conference on Process Mining (ICPM)* (pp. 88–95). 2021 3rd International Conference on Process Mining (ICPM). doi:10.1109/ICPM53251.2021.9576881

Kalman, R. E. (1960). A New Approach to Linear Filtering and Prediction Problems. *Transactions of the ASME–Journal of Basic Engineering*, *82*, 35–45.

Krause, J., Perer, A., & Ng, K. (2016, May 7). Interacting with Predictions: Visual Inspection of Black-box Machine Learning Models. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (pp. 5686–5697). doi:10.1145/2858036.2858529

Krishnan, R., Shalit, U., & Sontag, D. (2017). Structured Inference Networks for Nonlinear State Space Models. *Proceedings of the AAAI Conference on Artificial Intelligence*, *31*(1). Retrieved February 22, 2022, from https://ojs.aaai.org/index.php/AAAI/article/view/10779

Leglaive, S., Alameda-Pineda, X., Girin, L., & Horaud, R. (2020, February 10). A Recurrent Variational Autoencoder for Speech Enhancement. arXiv: 1910.10942 [cs, eess]. Retrieved February 7, 2022, from http://arxiv.org/abs/1910.10942

Martens, D., & Provost, F. (2014). Explaining data-driven document classifications. *MIS Quarterly*, *38*(1), 73–100. doi:10.25300/MISQ/2014/38.1.04

Melnyk, I., Santos, C. N. dos, Wadhawan, K., Padhi, I., & Kumar, A. (2017, December 4). Improved Neural Text Attribute Transfer with Non-parallel Data. arXiv: 1711.09395 [cs]. Retrieved February 28, 2022, from http://arxiv.org/abs/1711.09395

Molnar, C. (2019). *Interpretable machine learning. A Guide for Making Black Box Models Explainable*. Retrieved from https://christophm.github.io/interpretable-ml-book/

Narendra, T., Agarwal, P., Gupta, M., & Dechu, S. (2019). Counterfactual Reasoning for Process Optimization Using Structural Causal Models. In T. Hildebrandt, B. F. van Dongen, M. Röglinger, & J. Mendling (Eds.), *Business Process Management Forum* (pp. 91–106). doi:10.1007/978-3-030-26643-1_6

Oberst, M., & Sontag, D. (2019, June 6). Counterfactual Off-Policy Evaluation with Gumbel-Max Structural Causal Models. arXiv: 1905.05824 [cs, stat]. Retrieved September 22, 2021, from http://arxiv.org/abs/1905.05824

Qafari, M. S., & van der Aalst, W. M. P. (2021). Case Level Counterfactual Reasoning in Process Mining. In S. Nurcan & A. Korthaus (Eds.), *Intelligent Information Systems* (pp. 55–63). doi:10.1007/978-3-030-79108-7_7

Robeer, M., Bex, F., & Feelders, A. (2021, November). Generating Realistic Natural Language Counterfactuals. In *Findings of the Association for Computational Linguistics: EMNLP 2021* (pp. 3611–3625). EMNLP-Findings 2021. doi:10.18653/v1/2021.findings-emnlp.306

Shook, C. L., Ketchen Jr., D. J., Hult, G. T. M., & Kacmar, K. M. (2004). An assessment of the use of structural equation modeling in strategic management research. *Strategic Management Journal*, *25*(4), 397–404. doi:10.1002/smj.385

Starr, W. (2021). Counterfactuals. In E. N. Zalta (Ed.), *The Stanford Encyclopedia of Philosophy* (Summer 2021). Metaphysics Research Lab, Stanford University. Retrieved February 9, 2022, from https://plato.stanford.edu/archives/sum2021/entries/counterfactuals/

Tsirtsis, S., De, A., & Gomez-Rodriguez, M. (2021, July 6). Counterfactual Explanations in Sequential Decision Making Under Uncertainty. arXiv: 2107.02776 [cs, stat]. Retrieved September 9, 2021, from http://arxiv.org/abs/2107.02776

van der Aalst, W., Adriansyah, A., de Medeiros, A. K. A., Arcieri, F., Baier, T., Blickle, T., ... Wynn, M. (2012). Process Mining Manifesto. In F. Daniel, K. Barkaoui, & S. Dustdar (Eds.), *Business Process Management Workshops* (pp. 169–194). doi:10.1007/978-3-642-28108-2_19

Wachter, S., Mittelstadt, B., & Russell, C. (2017). Counterfactual Explanations Without Opening the Black Box: Automated Decisions and the GDPR. *ArXiv*. doi:10.2139/ssrn.3063289

Wang, K., Hua, H., & Wan, X. (2019, December 12). Controllable Unsupervised Text Attribute Transfer via Editing Entangled Latent Representation. arXiv: 1905.12926 [cs]. Retrieved November 9, 2021, from http://arxiv.org/abs/1905.12926

Wang, Z., Zhang, J., Xu, H., Chen, X., Zhang, Y., Zhao, W. X., & Wen, J.-R. (2021, July 11). Counterfactual Data-Augmented Sequential Recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 347–356). doi:10.1145/3404835.3462855