ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

DEPARTMENT OF COMPUTER SCIENCE
AND ENGINEERING

ARTIFICIAL INTELLIGENCE

MASTER THESIS

DEEP LEARNING AND CONSTRAINED OPTIMIZATION FOR EPIDEMIC CONTROL

CANDIDATE
Michele Iannello

SUPERVISOR
Prof. Michele Lombardi

CO-SUPERVISOR
Federico Baldo

Academic year 2020-2021
Session 3rd

To the Ukrainian people.

Contents

Abstract

The SARS-CoV-2 pandemic has galvanized the interest of the scientific community. Particular interest has been posed on methodologies apt at predicting the trend of the epidemiological curve, i.e. the daily number of infected individuals in the population. In this work, we argue for a model capable of producing intervention plans focused on counteracting the negative effects of an outbreak, with real applications on the ongoing pandemic. To do so, we relied on the use of Machine Learning models and Combinatorial Optimization approaches.

The project entails the development of a new predictive model capable of forecasting the number of infected individuals depending on non-pharmaceutical interventions. The development of the model required the use of state-of-the-art techniques, which relied on prior knowledge injection to guide the training process. The model is then used to boost a combinatorial process effectively producing the intervention plan. The ultimate result is a working prototype of a Decision Support System capable of assisting policy-makers during a virus outbreak.

Introduction

The SARS-CoV-2 (also known as COVID-19) virus was firstly identified in late 2019 in Wuhan, China: the resulting outbreak quickly faced a worldwide spread, becoming a pandemic and affecting people all across the globe. Outside of the Asian continent, the first country to encounter severe issues was Italy: the first wave started in late February, quickly leading to a major health emergency. Authorities quickly reacted by enforcing a set of containment measures, including social distancing, face mask-wearing and a generalised lockdown. However, clear intervention plans required months to be investigated, evaluated and adopted, meanwhile leaving room for week-by-week — if not day-by-day — decisions. This situation highlighted the need for effective methods to assist policy-makers in such unprecedented scenarios. In the context of an emerging infectious disease outbreak, a major role is played by the ability to forecast the trend of the epidemic in order to determine its socio-economical impact on the population and to plan effective control policies aimed at limiting its adverse effects. In this work, we precisely address these tasks through the development of a predictive and prescriptive system with the purpose of providing both accurate predictions of the epidemic evolution and effective policies to assist policy-makers in the complex task of counteracting the negative impacts on society. We focus on *Non-Pharmaceutical-Interventions* (**NPIs**), namely actions and measures that can be taken by people and enforced by authorities to help the spread of infectious diseases — other than vaccines and medicines.

The recent successes of Machine Learning (ML) and Deep Learning (DL) has already promoted several research works to investigate the application of ML methods within the field of predictive epidemiology. However, prescriptive analytics often remains a major concern, given the complex relationships between the containment measures and the outbreak evolution.

To address the predictive task, we make use of knowledge injection to

enrich the typical data-driven approach of ML models with model-driven information, namely domain-specific background theories. In particular, we exploit the Universal Differential Equations approach introduced in [16], which enables an integration of universal approximators such as ML models into differential equations. Regarding the prescriptive task, we explore the Empirical Decision Model Learning (EML) framework introduced in [13], which combines ML and Combinatorial Optimization techniques. Specifically, the approach differentiates from traditional techniques — often referred to as black-box optimization — since it introduces the encoding of a formal representation of the ML predictive model into a prescriptive optimization process. As opposed to ”generate and test” methods, this approach allows providing the prescriptor with insights about the predictive model, boosting the optimization process due to a dynamic reduction of the search space during execution. As a result, it brings benefits in terms of robustness, accuracy and computational times with respect to traditional black-box optimization.

Recent work in predictive and prescriptive epidemiology has been encouraged by the insurgency of the COVID-19 pandemic, which promoted several research articles [15] [14] [11]. To the best of our knowledge, however, this is the first application of both Universal Differential Equations and Empirical Decision Model Learning respectively to predictive and prescriptive epidemiology.

Chapter 1

Problem formulation

As already mentioned, the aim of this work is the development of an effective Decision Support System (DSS) designed to recommend the best political **interventions** to assist policy-makers within an epidemic scenario. In particular, we focus on *Non-Pharmaceutical-Interventions* (**NPIs**) and we provide a real-world case application by employing publicly available data regarding the SARS-CoV-2 pandemic diffusion in Italy.

1.1 Outline

Generally speaking, such a task could be addressed in a Combinatorial Optimization fashion, provided that the relationship between the interventions and the epidemiological curve is known and expressible in a mathematical form: the task would amount to formalizing the problem into a constraint modelling language exploiting Constrained Programming (CP), Satisfaction (SAT) or SAT-Modulo Theories (SMT), and deploying solvers to find feasible solutions. However, a formal definition of said relationship is hard to provide, because of both the complex nature of the underlying interactions between the variables and the high variability in the spread of epidemics. A common approach to the task involves its decoupling into two sub-problems, namely a predictive and a prescriptive problem. The former amounts to building a **predictor** to effectively model a relationship between two variables: in our case, we strive to assess how *interventions* affect the epidemiological curve (see Chapter 3). The latter aims at exploiting the resulting information to build a **prescriptor**, which determines the *best* possible interventions apt at obtaining the desired outcome, for instance, the minimization of the said curve (see

Chapter 4). As already mentioned, the second component can be modelled in a Combinatorial Optimization fashion, while the first one is more challenging. A well-established practise involves employing a simulator to forecast the evolution of the target variables depending on input ones, which however allows for a poor exploration of the input space. Another viable approach is to encode the relationship through the deployment of Machine Learning (ML) and Deep Learning (DL) techniques — for example by means of an Artificial Neural Network (ANN) mapping NPIs into the number of infected people.

1.2 Integration

Following the existing literature, the integration of a DL predictive component into an Optimization prescriptive problem can mainly follow two techniques [1].

Black-Box Optimization A first simple approach is the use of the DL model as a predictor whose internal structure is not known (a *black-box*): although advantages typically include a speedup in the optimization process due to the relatively scarce computational complexity, the lack of structural knowledge of the DL model can lead to inefficient exploration of the input space, since the optimization solver has no (direct) access to first- and second-order derivatives of the predictor.

Model Embedding A more advanced approach involves the explicit embedding of the DL model into the optimization process in the form of variables and constraints encoding the neurons of the network along with their interactions. Introduced by [13] as Empirical Decision Model Learning (EML), this framework has the advantage of providing helpful insights to the optimization process, allowing it to perform a guided exploration of the input space. However, limitations include the restriction of DL model to Feed-Forward Neural Networks — thus excluding Recurrent Neural Networks (RNNs), for example — and higher computational costs and times when it comes to encoding large DNNs.

1.3 The UODE framework

Employing domain-specific knowledge to guide the learning phase of the predictor can provide great benefits to DL models both in terms of accuracy and explainability [2] [17]. Universal Ordinary Differential Equations (UODEs) [16] can enable this hybrid approach allowing ODEs to incorporate a Universal approximator, i.e. *a parameterized object capable of representing any possible function in some parameter size limit* [16] — for example a Neural Network. Such an object can be formalized as follows:

$$\frac{d}{dt}y(t) = f(y, t, U(y, t)) \tag{1.1}$$

where the first-order derivative of y, specified by the function f, explicitly depends on the approximator $U(y, t)$. In other words, this hybrid framework combines model-driven and data-driven approaches, allowing differential equations to include *learnable* components, which provides a twofold advantage. On the one hand, a dynamic model of the system can be exploited to make use of domain-specific knowledge, thus contributing to interpretability and (hopefully) accuracy. On the other hand, certain terms of the model do not need to be explicitly modelled by an expert in an analytical fashion, which can be particularly helpful in case of complex real-world interactions. As a consequence, this approach allows learning non-measurable model parameters — for which historical data is not available — addressing the key issue of parameters tuning, which is typically critical when dealing with parametric models. **Training** a Neural Network in this framework is performed as follows. Learning targets (y) are retrieved from historical data, while predictions are obtained by inserting the network outputs into a numerical integration of the ODE: regression loss functions such as the Mean Squared Error can be employed to perform the backward pass.

However, the very formulation of the technique raises a possible drawback. As just mentioned, learning targets do not correspond to the actual model's predictions — except for the trivial case $f(y, t, U(y, t)) = U(y, t)$. Accordingly, training will promote the network's predictions towards the best fitting curve independently of the *correctness* of the parameters. In other words, this kind of integration does not explicitly provide the network with prior knowledge about the nature of the underlying physical phenomenon, thus possibly resulting in unrealistic deviations of the predicted parameters

from the feasible ones. In this context, a critical role is played by the accuracy of the numerical approximation method: for instance, when employing explicit Runge-Kutta iterative methods, the accuracy of the approximation can be enhanced by relying on higher-order instances. Besides, the proper use of regularization throughout the training phase can help mitigate these effects through the injection of semantic constraints.

Chapter 2

Related Works

2.1 Epidemiological Models

A major issue in the case of epidemics is the availability of trustworthy predictive systems that can be successfully deployed to forecast the spreading dynamics of the virus. In this regard, a central point within the field of epidemiology is the study of epidemiological models. Approaches include Mathematical models, Complex network models, (more recently) Agent–based models and Deep Learning–based models [5] [1].

2.1.1 Mathematical models

Mathematical models of epidemics are the earliest methods used to formulate epidemic spread, and they are usually classified into two categories: deterministic epidemic models and stochastic epidemic models.

Compartmental models are among the most effective methods used to describe the spread of infectious diseases. In particular, they rely on **partitioning** the population into categories called *compartments*, the interactions among which are modelled as differential equations and describe the population flow from one compartment to another. There are several instances of this approach differing in the partitioning of the population as well as in the equations governing the flow across compartments, named after the compartments they introduce. A further distinction can be made between *deterministic* and *stochastic* compartmental models. The main characteristic of the former kind is that, given a certain initial state for the variables, along with

the parameter values, the evolution of the variables is fully determined by the differential equations used to describe the model. On the contrary, the latter kind is characterized by the deployment of *probabilities* (e.g. of infection as a consequence of a contact): as a result, the behaviour of the system cannot be exactly determined in advance, rather can be only observed through simulations. Stochastic processes, such as Markov Chains and the Monte Carlo method, are usually deployed to build *stochastic epidemic models* in order to represent uncertainty or randomness in epidemic models. This approach typically involves the introduction of another parameter, namely the infection probability within a contact between a susceptible individual and an infectious individual. As an example, the Reed-Frost model — which is the most widely used stochastic epidemic model — assumes that an infection event in contact between two individuals can be modelled as a binomial stochastic process.

Strengths of mathematical models of epidemics include the capability to perform a large-scale theoretical analysis of epidemic diffusions, such as the epidemic threshold and final epidemic size.

However, mathematical models rely on some assumptions and approximations that limit their deployment, in particular precluding small-scale and detailed insights on the epidemic. For instance, the determining factor of an epidemic is human behaviour, since the spread is ultimately conveyed through human contact. Unfortunately, human behaviour is characterized by high variance both in space and time, meaning that people behave both variously — i.e. differently with respect to one another — and variably — i.e. in different ways in the course of the epidemic. Mathematical models have troubles with this kind of detailed representation due to several factors: the small set of variables that are included in the models limit their complexity; variables parameterized with average quantities and mean values, such as average infection rate and average recovery rate, cannot be used to describe the heterogeneous nature of epidemic spread; the assumptions of homogeneous and well-mixed population and of full connection — i.e. that all individuals make a contact with each other in a time step — fail to represent the individual human behaviour.

2.1.2 Complex networks models

Complex networks can be used to model epidemics, with nodes representing individuals and links representing interactions among individuals. They can

be classified into two categories: spreading dynamics in complex networks and numerical simulations of epidemics in complex networks.

Models of spreading dynamics divide individuals into different groups according to their health states, as in compartmental models. In homogeneous networks, the infectivity of infectious nodes is a function of average node degree, while nodes are further partitioned according to degrees in heterogeneous networks.

Numerical simulations of epidemics usually integrate with the Reed-Frost model and use individual–based models to represent contact patterns between individuals and infection probability on links.

Compared to mathematical models, complex networks can provide more detailed representations, since they allow to represent heterogeneous population structure and interaction patterns among individuals, for instance through node degrees and edge weights.

However, the diversified nature of human behaviour and interactions result in difficulties for complex networks, especially when it comes to modelling daily activities, mobility, ages, and occupations of individuals. A possible improvement in this direction can be achieved by extending the static network topology with the temporal dimension, in order to allow the representation of dynamic patterns.

2.1.3 Agent–based models

Agent–based modelling is a promising computational approach that allows a detailed depiction of the reality, since it provides heterogeneity in individual attributes and behaviours, incorporating the stochastic nature of epidemic spread. The previously discussed techniques can be combined in order to take advantage of the corresponding strengths: complex networks can be exploited to represent agent mobility patterns and contact patterns, while mathematical methods can be used to describe agent behaviours, such as stochastic processes.

However, the higher resolution of agent–based models comes at the cost of data availability and computational complexity. Moreover, data related to human behaviour is typically difficult to collect and formalize into agent algorithms.

2.1.4 Deep Learning–based models

The recent success of Machine Learning (ML) and Deep Learning (DL) has encouraged the development of epidemic models based on such techniques. In particular, DL typical advantages include the relatively simple modelling phase, due to the ability of Deep Neural Networks (DNNs) to easily approximate any given function without any prior knowledge on the physical phenomenon. However, DL–based methods suffer several drawbacks, including the need for large amounts of data and a general lack of explainability.

Traditional approaches to predictive epidemiology include Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs). RNNs were repeatedly proven to work well on sequential data, becoming the standard neural model of choice when it comes to time series forecasting, finding successful applications in the field of predictive epidemiology as well [1]. On the other hand, CNNs are especially well known due to their strengths in handling spatial proximity correlations, which can be translated to temporal correlations in the case of sequential temporal data [1].

Composed approaches aim at performance gains by combining the advantages of multiple traditional DL methods.

Hybrid approaches aim at improving the interpretability and accuracy of the models by adopting domain-specific techniques to guide the learning process. A recent trend in predictive epidemiology is the exploitation of compartmental models (see Section 2.1.1) as domain-specific knowledge to rely upon.

2.2 Decision-Focused Learning

Another critical task within predictive epidemiology consists in developing effective models capable of determining the impact of containment measures on the virus spread.Thus, determining the actual relationship between the interventions and the evolution of the epidemiological curve is a key predictive issue, that can be addressed through DL. A common approach going under the name of ”Predict, then Optimize” basically relies on the quite intuitive assumption that — within an optimization process — the higher the accuracy of the predictive component, the better the results achieved by the prescriptive one. As first pointed out in [4], however, the criteria used to train ML

models frequently differ from the ones eventually employed to assess them, especially when they are deployed as components of larger processes, possibly resulting in an overestimation of the effective performance of the predictive model within the process. Accordingly, the authors propose a form of end-to-end approach through the deployment of a proper loss function, aiming at validating the predictive model directly on the ultimate task. Such work paved the way to the entire field of Decision-Focused Learning, including works in which ML models are directly trained in conjunction with the optimization algorithm [18]. Following the literature, DL approaches to address the predictive task include **Surrogate Models**, **Recurrent Neural Networks** and **Knowledge Injection**.

Surrogate Models. Whenever supervised data is not available, and yet local interactions between the variables can be described — e.g. by a simulator — surrogate models represent an effective way to develop a predictive DL model. Specifically, a Feed-Forward Neural Network can be trained to learn the mapping between the variables by firstly sampling in the input space and then running the simulator on the input samples to obtain the learning targets. Drawbacks of this type of model include computational efficiency and bias injection.

(Recurrent) Neural Networks. Whenever supervised data is available, the problem could be directly tackled via DL. As already mentioned in 2.1.4, RNNs are very well-known to work well on sequential data. Long-Short Term Memory Networks (LSTM) and Gated Recurrent Unit (GRU) are especially suitable for such a task due to their typical capability to capture temporal patterns and develop long-term memory.

Knowledge Injection. Another viable alternative in the presence of supervised data is the exploitation of a mathematical model that frames historical data in a background theory. The idea underlying this approach is to guide the training phase with some form of prior knowledge on the phenomenon. Specifically, the Universal Differential Equations framework introduced in [16] can be exploited in order to effectively estimate the parameters of differential equations via DL from historical data. As already discussed in Section 2.1, *compartmental models* are an effective way to describe the spread of a

virus in a mathematical framework through parametric differential equations. They can bring great benefits to Deep Learning models in terms of explainability, thanks to the introduction of a form of causal knowledge on the underlying mechanisms of the epidemics. However, they are also prone to excessive approximations, since they provide a simplified description of complex phenomena, which are often driven by several interconnected causes.

Chapter 3

Deep Learning for epidemic forecasting

This chapter aims to address the predictive task of the decision system, namely to build a DL predictor mapping the NPIs in the epidemiological curve. Instead of approaching the task in a traditional way, i.e. trying to directly map the NPIs in the number of infected people, we explore a form of neuro-symbolic integration. Specifically, we are going to exploit the compartmental SIR model as a mathematical background theory to frame the epidemic spread and the UODE framework as a learning paradigm (see Section 1.3). The underlying idea is to combine formal and implicit knowledge: the former can be achieved by building a deterministic, parametric model of the epidemic based on differential equations, so as to be able to express the spread of the epidemic through certain parameters. The latter amounts to learning a relationship between the NPIs and said parameters, which can be accomplished through the UODE framework by employing a DNN as approximator: in fact, ANNs are a suitable system for implementing the approximator, due to their typical capability to learn high-dimensional complex mappings. We should emphasize here that our design choices are constrained from the requirements of the model needed to approach the prescriptive task: for instance, RNNs are precluded from EML-based methods, since the encoding of the network would result into very complex optimization problems. From an implementation point of view, this task can be accomplished in two ways, both explored in the following sections. In short, the first one relies on two separate models to firstly learn the parameters of the SIR model from the epidemiological curve and then map the NPIs in such parameters, while the second one learns the relationship

NPIs-infected in an end-to-end fashion.

3.1 SIR-NPI

In this section we aim to approach the task employing two separate models. We develop a first neural network to map the epidemiological curve into the parameters of the SIR model: the resulting estimation can be regarded as learning target for a second neural network with the aim of mapping the NPIs into such parameters. As we discussed in chapter 2, **compartmental models** are an effective method, especially when applied to predictive epidemiology. In particular, the simplest one is the **SIR model**, which is nonetheless very popular since it provides a basic theory background, still offering good local approximations of the epidemic progression.

3.1.1 Modelling

The deterministic **SIR model** without vital dynamics — i.e. in which birth and death rates are neglected — divides the population N into three compartments: S for **S**usceptible (healthy), I for **I**nfectious (diseased and infective) and R for **R**ecovered (either healed and immunized or deceased): since demography is not taken into account, a key property of this model is the constant total population $N = S+I+R$. The following set of ordinary differential equations (ODEs) describes the flow of the population across the compartments:

$$\begin{cases} \frac{d}{dt}S(t) = -\frac{\beta S(t)}{N} \\ \frac{d}{dt}I(t) = \frac{\beta I(t)S(t)}{N} - \gamma I(t) \\ \frac{d}{dt}R(t) = \gamma I(t) \end{cases} \tag{3.1}$$

where β is the **infection rate** — defined as the *average number of contacts* per person per time, multiplied by the *probability of disease transmission* in a contact between a susceptible and an infectious subject — and γ is the **recovery rate** — defined as the inverse of the *recovery period*, i.e. the time that an individual stays in the I compartment (before moving to R). These parameters allow to compute the basic reproduction number $R_0 = \beta/\gamma$, quantifying the average new cases generated by an infected individual in the population

(also referred to as *secondary infections*).

In the most general formulation, both parameters of the SIR model are time-dependent, but we can apply a simplification by arguing that γ tends to keep a roughly constant value during an epidemic — since the average recovery period is affected only by clinical and treatment issues — which eventually become stable whenever a practice is established — rather than depending on possibly variable external factors. Still, β is highly variable during an epidemic, since it does not depend only on biological characteristics of the epidemic, rather it is also influenced by continuously floating variables such as environmental conditions, containment measures, and ultimately people behaviour (see Section 2.1.1). Accordingly, we can assign a fixed value to the former — which can be easily retrieved by historical data and earlier estimates — while we express the latter as a time-dependent variable $\beta(t)$. Taking into account that data generically consist of daily recordings, we will apply discretization and more appropriately refer to it as a **vector of parameters** $\boldsymbol{\beta} = (\beta_0, ..., \beta_{T-1})$ — one for each day t where $\beta_t \equiv \beta(t)$ with T the total number of days covered by available data.

3.1.2 Estimation

The above formulation allows us to formally define the β *estimation problem*, i.e. that of finding the optimal parameters $\boldsymbol{\beta}$ to fit the historical SIR curves[1], given the initial state and a fixed value for γ:

$$\boldsymbol{\beta}^* = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left\{ \sum_y \sum_t L(\hat{y}_t, y_{t,\beta(t)}) \,\middle|\, y(0) = \hat{y}_0 \right\} \tag{3.2}$$

where y ranges over S, I and R and follows Equations 3.1, and L is a loss function to measure the error of approximating $\hat{y}$ with y. Estimating the $\boldsymbol{\beta}$ parameters will give us access to target data to build a DNN mapping their relationship with the NPIs. The optimal parameters can be obtained through a shallow ANN (Artificial Neural Network): the network receives as **input** the series of historical data relative to the three compartments, while its **output** is precisely the vector $\boldsymbol{\beta}$, randomly initialized at the beginning. The network thus consists of a single layer formed by as many neurons as the days on the

[1] Specifically the I compartment, since it represents the most critical variable, causing significant social and economic impact.

historical series.

Training

Training consists in *minimizing* the prediction error — represented by an objective function that we can variously choose — by repeatedly performing two steps. The *forward phase* consists of simulating the evolution of the SIR model compartments employing a numerical iterative method — specifically the Euler's method — on Equations 3.1. Due to the inaccuracy of such simple methods, higher order methods such as the Runge-Kutta 4th-order method (RK4) were tested as well, leading however to comparable results whilst significantly increasing training time due to the higher computational time required to perform the simulation. Nevertheless, the accuracy of the Euler method can be

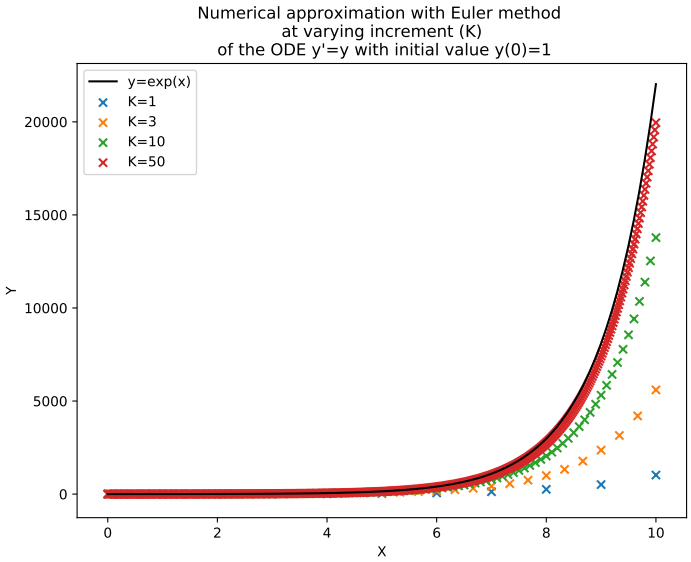


Figure 3.1: In reference to the example ODE $y' = y$ with initial condition $y(0) = 1$, the graph reports the exact solution $y = e^x$ (black curve) and numerical approximations obtained employing the Euler Method with different increments.

enhanced by employing a sub-unitarian increment h, accordingly performing $K = \frac{1}{h}$ simulation steps (see Figure 3.1). The *backward* phase consists in updating the network weights by applying gradient descent on the objective function, initially defined as a simple Mean Squared Error (MSE) on the SIR compartments:

$$L(\boldsymbol{\beta}) = \frac{1}{3} \sum_{X \in \{S,I,R\}} \frac{1}{T} \sum_{t=0}^{T-1} \left(X_t - \hat{X}_t(\boldsymbol{\beta}) \right)^2 \tag{3.3}$$

where T is the number of recorded days, i.e. the length of the historical time series. However, such a simple formulation does not take two issues into account. On the one hand, the I compartment is typically both the least relevant in terms of numerical proportions and the most relevant in terms of physical interest. In other words, while it tends to provide the smallest contribution to the MSE, it is also the compartment of highest interest, since it provides the major proxy of the virus spread. On the other hand, the parameter $\boldsymbol{\beta}$ represents a physical phenomenon, namely the infectivity rate of the epidemic: as such, it is not realistic to expect large and quick fluctuations from its trend. We can therefore revise the objective function accordingly:

$$\begin{gathered} L(\boldsymbol{\beta}) = \mathbf{MSE}(\hat{I}(\boldsymbol{\beta})) + \epsilon \left(\mathbf{MSE}(\hat{S}(\boldsymbol{\beta})) + \mathbf{MSE}(\hat{R}(\boldsymbol{\beta}))\right) + \lambda \Theta_{\boldsymbol{\beta}} \\ \text{with} \\ \Theta_{\boldsymbol{\beta}} = \frac{1}{T-1} \sum_{t=1}^{T-1} (\beta_{t+1} - \beta_t)^2 \end{gathered} \tag{3.4}$$

where we introduced two hyperparameters: ϵ and λ. The former can be tuned to adjust the contribution of the S and R compartments to the MSE: since we are mainly interested in the I compartment, it should take values $0 \leq \epsilon < 1$. The latter weights the relevance of the term $\Theta_{\boldsymbol{\beta}}$ with respect to the whole objective function: in particular, $\lambda\Theta_{\boldsymbol{\beta}}$ acts as a **regularization term**, as it helps enforcing smoothness on $\boldsymbol{\beta}$ by penalizing abrupt variations.

Regularization , however, is a possible source of issues: semantical constraints introduce a model-driven component into a data-driven approach, thus typically sacrificing fitting accuracy in order to comply with desired behaviours. Accordingly, the scale of the regularization term — namely the parameter λ used to control its relevance with respect to the whole loss function — plays a critical role: a proper trade-off allows to reach both high quality fitting of the epidemiological curve and a smooth trend for the estimated parameters. We therefore decided to exploit a recent approach introduced in [7] to perform optimization over λ, which relieves us from the burden of tuning the parameter, allows it to be *learned* and thus automatically provides us with an optimal solution. Formally, the presence of a regularization term (see Equation 3.4) in the optimization problem of Equation 3.2 amounts to the minimization of a Lagrangian Relaxation of the original loss function. As pointed out in the

reference paper [7], the relaxation is by construction a lower bound for the original problem: $L(LR_{\boldsymbol{\lambda}}) \leq L(\mathcal{O})$, where $\mathcal{O}$ is the optimal solution to the original problem, $LR_{\boldsymbol{\lambda}}$ is the optimal solution to the relaxed form and λ is a (vector of) Lagrangian multiplier(s). The best multipliers can be found by obtaining the *strongest* Lagrangian Relaxation through the Lagrangian Dual Framework:

$$LD = \underset{\lambda_i \geq 0}{\operatorname{argmax}} f\left(LR_{\boldsymbol{\lambda}}\right). \tag{3.5}$$

From an implementation point of view, this can be easily achieved in an iterative way: at each step k, firstly usual training is performed, seeking for a minimization of the loss function with respect to the training parameters $\boldsymbol{\beta}$; then, the Lagrangian multiplier λ_k is updated through Gradient Ascent on the same loss function, now seeking for its maximization.

3.1.3 Prediction

The purpose of the second step is to address the β *prediction problem*, namely creating a DL predictor to map the NPIs into $\boldsymbol{\beta}$. In this context we cannot rely on any theoretical model due to the peculiarity of the problem: *NPI* is a broad term and can refer to a variety of interventions that can be undertaken in different areas. Accordingly, we deploy a simple Multi-Layer Perceptron (MLP), experimenting with different combinations of hyperparameters such as the learning rate, the number of epochs, hidden layers and neurons per layer.

3.2 End-to-End

Since the UODE framework allows to integrate arbitrary approximators in ODEs (see Section 1.3), we can develop an end-to-end approach by building a single neural network and directly providing NPIs as input for the SIR model parameters. Within this approach, we explored the provision of the initial state of the epidemic as input to the model, represented by the (normalized) number of infected: the underlying idea is to provide the model with a proxy of the epidemic status at time t_0, and requiring it to output the state at time t_1. It is worth to point out that the development of a predictive model to time-series involves a common pitfall, which is straightforward in our example: the newtork takes as input I_t and is supposed to output I_{t+1}. Actually, as common to all non-independent time-series, a very good prediction for I_{t+1} would be

precisely I_t, namely simply requiring the network to approximate the identity function: the number of infected of each day is closely related to that of the next one. Hence, the deployment of the UODE framework plays a key role in this setup: the number of infected I_t is indeed given as input alongside NPI_t, but the actual network predictions represent the β_{t+1} parameters, which are later employed to numerically approximate the output I_{t+1} (see below). This represents a great benefit of this approach, since it allows us to neglect the possibility to learn trivial mappings. Otherwise, we should have explicitly addressed this issue, for example by enforcing the output to exhibit a minimum deviation from the input through regularization terms.

3.2.1 Modelling

We can accordingly rewrite Equations 3.1 to emphasize this paradigm shift:

$$\begin{cases} \frac{d}{dt}S(t) = -\frac{\beta(NPI(t))S(t)}{N} \\ \frac{d}{dt}I(t) = \frac{\beta(NPI(t))I(t)S(t)}{N} - \gamma I(t) \\ \frac{d}{dt}R(t) = \gamma I(t) \end{cases} \tag{3.6}$$

where we expressed the implicit dependency of β on time — which allows us to identify β with the universal approximator $U(y, t)$ — through the explicit dependency on the NPIs, which represents the actual input to the network. The peculiarity of this approach is to directly map the NPIs into the epidemiological curve, exploiting the SIR model as a theoretical framework but without striving to explicitly fit the real β parameters. Although this method has the advantage of assessing the quality of predictions directly on the degree of fitting of the epidemiological curve, a potential drawback relies in the already mentioned *adjustment* that an inaccurate numerical approximation algorithm such as the Euler method tends to produce in the parameters. However, as long as a good fit is achieved, we could reasonably neglect the correctness of the estimated β parameters, considering that our actual goal is to correctly forecast the evolution of the epidemic. Besides, such an implementation leverages a DNN mapping NPIs into β parameters, which allows us to compare the results with the predictive model described in Section 3.1.3.

3.2.2 Training

The actual implementation of the presented model can be done by wrapping the neural network as a layer of a bigger model. Specifically, the layer consists of an MLP taking both I_t and NPI_t as input and giving a prediction of β_{t+1} as output: since the output is supposed to be non-negative, an exponential activation function was used for the output layer, while the ReLU was employed within the hidden ones. The forward phase involves getting said predictions and employing them together with the Euler method in order to obtain the actual epidemiological curve as described by the SIR compartments. Accordingly, the backward phase consists in performing gradient descent on the loss function, namely the MSE, computed on such values. Regarding the predictor, tunable hyperparameters include the number of hidden layers and neurons per layer, while for the whole model we also include the number of training epochs and simulation steps (K).

Moreover, this kind of end-to-end approach lends itself to a further paradigm shift, suggested by the actual nature of data: specifically, measures taken by the government are typically revised not more often than weekly. This particular layout suggests the reduction of input data, by providing weekly rather than daily NPIs: accordingly, we explored a different model setting by iteratively performing the prediction and simulation steps before backpropagating. In other words, β_{t+1} is predicted 7 times, employing each time weekly NPIs alongside an updating I_t as input: the resulting estimate for β_{t+1} is then used to produce a numerical approximation of I_{t+1}, which serves as input for the following iteration.

Chapter 4

Optimal Policies for Epidemic Control

The prescriptive task amounts to determining the best possible interventions to carry out in order to achieve a given desired outcome of the epidemic. In other words, we can regard the NPIs as *decision variables* within a Combinatorial Optimization (see Chapter 1) problem and choose a certain *objective* to represent our goal — for instance, the minimization of the number of infected. This approach is enabled by the work described in Chapter 3, since the availability of a *predictor* able to effectively forecast the evolution of the epidemiological curve based on given NPIs allows for the assessment of a deterministic relationship between decision variables and the objective function.

We can formally describe the problem as follows:

$$\begin{aligned} &\mathbf{x}^* = \underset{\mathbf{x}}{\operatorname{argmin}} f(y_{T+1}, \mathbf{x}) \\ &\text{s.t. } h(\mathbf{x}) = y_{t+1} \qquad \forall i \in 1, ..., T \\ &\pi_t(x_t) \qquad\qquad\quad \forall i \in 1, ..., T \\ &x_t \in D \qquad\qquad\quad \forall i \in 1, ..., T \\ &\mathbf{x} \in F \end{aligned} \tag{4.1}$$

where T represents the temporal interval over which we are considering the decision process (i.e., the temporal span of the intervention plan). The variable y_{T+1} represents a proxy for the evolution of the epidemic (e.g., the rate of infected people in the population); while $\mathbf{x} = (x_1, .., x_T)$ are the NPIs in

time, which take value in the domain D; π_t are the predicate representing the constraints on the interventions (e.g., a reduced throughput in the production of vaccines can be expressed as a constraint on the distribution of shots in the population) and F is the feasible set containing all the admissible solution for the optimization problem. Finally, the function h is the predictive model approximating the relation NPI–infected, while f is the objective function: due to the socio-economical costs derived by the deployment of strict NPIs, f might involve indices relative to different aspects of the epidemic, apart from the number of infected, such as stringency terms (measuring the restrictiveness of the policies) or economic impact estimations.

4.1 Empirical Model Learning

An effective embedding approach (see 1.2) going under the name of Empirical Model Learning (EML) was introduced in [13] with the goal of enabling the integration of ML models within Optimization processes. As already mentioned in 1.2, this paradigm aims at enhancing the optimization process by providing the solver with explicit knowledge about the relationship between the input and output variables. In the words of the authors, EML is a methodology *that: (1) learns relations between decidables and observables from data, and (2) encapsulates these relations into components of an optimization model, namely objective functions or constraints.* As opposed to a "generate and test" mechanism, this technique has the advantage of boosting the optimization process by providing a dynamic reduction of the input space during execution.

4.2 Modelling

The implementation of this approach requires to encode the ML predictor in a constrained modeling language, which is enabled by `EMLlib`, a Python library directly provided by the authors of the reference paper and available at [6]. The library allows the encoding of different kinds of ML models — such as Decision Trees and Feed-Forward Neural Networks — in an internal format, which is needed to enable an interface with supported optimization solvers. Specifically, the ML model's input variables are encoded as decision variables, while a cost function needs to be defined in terms of the output, in order to drive the search.

In particular, we formalized the problem using Mixed-Integer-Linear Programming (MILP) and deployed the CBC solver [8] to perform the search. As required by the EML approach, input features to the ML model, i.e. the NPIs, were encoded as decision variables, while the minimization of the output variable β was chosen as objective function.

From an implementation point of view, both decision and output variables were constrained to belong in the range $[0, 1]$: as described in Section , this range ensures physical values for beta β; as for the NPIs, actual values take integer discrete values, but we performed normalization, as common practice when designing DL models. The solving process was wrapped into a loop to iteratively perform the optimization process in the following way. As a first thing, the ML model weights are loaded and the EML encoding is performed as described above. Secondly, the SIR initial state is provided to the model, in order to be able to create the epidemiological curve. Thirdly, the solver is used to obtain prescriptions and resulting predictions: striving to obtain the lowest possible value for the β parameters, the optimization process gives a solution in the form of NPIs. These are then fed to the ML model to obtain corresponding β predictions, which are in turn employed to obtain SIR predictions through numerical approximation, as usual. Depending on the instance of the End-to-end model (see Section 5.2.2), the loop thus creates either daily or weekly prescriptions aimed at minimizing the β parameters at each step, along with the number of infected people. Keeping track of the predictions regarding the evolution of the epidemic throughout the optimization process allows to assess the good functioning of the model, i.e. to establish whether the desired minimization is achieved. It is worth pointing out a possible pitfall of this approach: whenever the input space is not uniformly represented in training data, the ML model can behave in quite unpredictable ways, which can lead to performance deterioration. This is likely to be the case for historical data as well as for synthetic data, as will become clearer after the description of the generation system. We decided to address this eventuality by constraining the possible deviation of β_{t+1} with respect to β_t at each optimization step: although pretty rough, this approach allows to avoid edge cases in particular, for instance prescriptions corresponding to the trivial prediction $\beta = 0$. This task was accomplished through an hyperparameter c, corresponding to the maximum allowed deviation: formally, a constraint of the form $\beta_{t+1} \geq \beta_t - c$ was employed, which required several tests to tune the hyperparameter.

Chapter 5

Experimental results

To test the methods described in Chapters 3 and 4, several experiments were conducted to test both the validity and the effectiveness of such methods.

Reliably assessing model performance within an epidemic scenario is a hard challenge, mainly for two reasons. On the one hand, the amount of available data is both scarce and inconsistent — especially in the early stages of the outbreak, when the practices still need to be established: since DL models are extremely data-hungry, this can easily lead to overfitting on currently available data, rather than capturing the actual trend and learning the spread evolution mechanism. Whenever we rely on a mathematical model, on the other hand, it is hard to correctly evaluate a certain parameter configuration, apart from hard physical boundaries: we already mentioned the risk of employing loss functions striving for curve-fitting along with numerical approximation methods, namely that optimal parameters are not necessarily the most reasonable nor likely, rather those that more accurately comply with the expected outcome (see Section 3.2.1). Despite the already discussed reasonable possibility to neglect this problem, we would definitely prefer to address it in order to keep the benefits from our design choices: indeed, a predictive model using correct parameters arguably leads to results that are both more robust and interpretable.

To mitigate and possibly overcome the just mentioned issues, we decided to rely on synthetic data generation to establish a more solid grounding of the performance evaluation of the model: indeed, a great variety of realistic data — complying with the phenomenological interactions that the model is meant to detect — is an effective indicator of its ability to capture meaningful insights, instead of forcing parameters to adapt to historical data.

5.1 Data

In this section, we describe experimental data that was used to test the methods described in Chapters 3 and 4. As already mentioned, we firstly give a brief description of publicly available data concerning a real-world case, namely the COVID-19 global pandemic.

5.1.1 Historical data

Official data related to the COVID-19 pandemic situation in Italy is maintained to date by the *Protezione Civile* department of the *Presidenza del Consiglio dei Ministri* [3]. Among other information, it stores the daily number of infectious (as resulting by swab testing operations), recovered and deceased people: when paired with the records on the total population provided by Istat [10], this information allows to fully determine the epidemiological curve, i.e. to retrieve daily data regarding all three compartments of the SIR model.

The NPIs were collected from the OxCGRT dataset, created and maintained by the Blavatnik School of Government of Oxford University [9]. Quoting from the official website, "The Oxford Covid-19 Government Response Tracker (OxCGRT) collects systematic information on policy measures that governments have taken to tackle COVID-19". Specifically, data consists of 23 indicators, divided by areas: containment and closure, economic, health system, and vaccination policies: we focused in particular on the first group, consisting of 13 different measures. Indicators are recorded in integer ordinal scales of different length, such that a value of 0 represents the complete relaxation of the intervention, while the highest value (up to 5) represents the most strict intervention.

Historical data regarding the metropolitan area of the city of Bologna (Italy) provides an average recovery period from COVID-19 of 17 days, thus resulting in $\gamma = 1/17$, which we decided to employ within the simulations.

5.1.2 Synthetic data

Synthetic data was obtained by starting with the generation of β as an independent variable, from which both the SIR and the NPI values were then obtained.

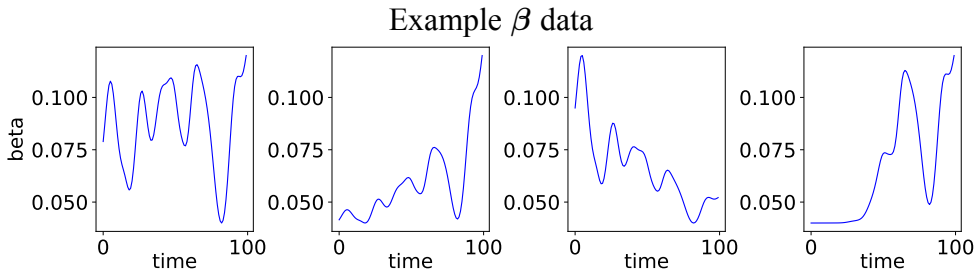


Figure 5.1: A few examples of synthetic data representing the β parameters.

β values were produced according to the following observations and specifications. Firstly, the distributions for β should be **realistic**, starting with the *domain*: since the infectivity rate is a probability, physical values should comply with $0 < \beta < 1$. However, historical values for $\boldsymbol{R_0}$ range in the interval $[1.9, 2.6]$ with a 95% C.I. [12]: hence, feasible values for β should accordingly belong in the range $[0.11, 0.15]$. Secondly, the distributions for β should be **diverse**, meaning that we would like to account for as many different evolution scenarios as possible, for instance: different starting points (β_0) indicate different stages at which the pandemic was detected (higher values indicate a later detection, lower values an earlier detection); on the other hand, a sort of periodicity (as opposed to monotonicity) indicates the occurrence of different infection waves. To account for these variety of possible scenarios, we generated 18 different curves combining *base* shapes, including Gaussian mixtures and sine-squared distributions, and *modifier* shapes, including (inverse-) sigmoids and (inverse-) exponentials. The latter are employed to enforce increasing or decreasing trends, along with different detection stages, while the former have the task of providing a ground realistic shape: Gaussian mixtures help modeling random behaviours with no specific trends, and consist of the sum of Gaussian distributions, each having a random $\boldsymbol{\sigma}$, with $\frac{1}{2} < \sigma < 1$ and a random binary weight to randomly exclude some distributions from the sum. Sine-squared ditributions, defined as $f(x) = sin^2(\nu x)$ with a random frequency $0.25 < \nu < 0.5$, help modeling infection waves by injecting a period in the distribution. To further enlarge the dataset, each of said 18 combinations was generated in 5 differently-parameterized instances, thus resulting in 90 total distributions: some examples are reported on Figure 5.1, while a more comprehensive overview is presented in Appendix A on Figure A.1.

A formal definition of the modifier shapes is quite straightforward:

$$\begin{aligned} \beta_\sigma(x,k) &= \frac{1}{1+e^{kx}} \\ \beta_{exp}(x,k) &= e^{kx/10} \end{aligned} \quad \text{with } k = \pm 1 \tag{5.1}$$

where k is used to tune the kind of general trend, namely increasing or decreasing. On the other hand, the base distributions can be described as follows:

$$\beta_{base}(x) = \begin{cases} \sum_{t=0}^{T-1} w_t \frac{1}{\sigma_t\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-x_t}{\sigma_t}\right)^2} & \text{with } w_t \in \{0,1\}, \sigma_t \in [0,1] \\ sin^2(\nu x) & \text{with } \nu \in [0.25, 0.5] \end{cases} \tag{5.2}$$

where the first case represents the Gaussian mixture, namely the sum of a Gaussian distribution for each sample t, with mean x_t, weight σ_t and a binary weight w_t, used to randomly suppress some distributions; the second case describes the sine-squared distribution, where ν is used to vary the period frequency.

Formally, the β distribution is simply obtained as the composition of base and modifier shapes:

$$\beta(x) = \beta_{base}(x)\beta_\sigma(x)\beta_{exp}(x) \tag{5.3}$$

Afterwards, the resulting distributions were rescaled to approximately comply with the feasible domain for β: the actual range was slightly enlarged in order to take into account a wider set of scenarios.

SIR variables were straightforwardly obtained from β through a numerical approximation method — specifically, the fourth-order Runge-Kutta method (RK4) — employing Equations 3.1, fixing $N = 6 \times 10^7$ (a rough estimate for Italy's population) and $\gamma = 1/17$ (see Section 5.1.1) and choosing an initial state for the SIR variables of $SIR(0) = (S(0), I(0), R(0)) = (6 \times 10^7 - 1050, 1000, 50)$, i.e. an epidemic in quite an early stage.

NPIs generation involved a more complex design: contrary to the SIR variables, we did not make any theoretical assumptions on the connection between β and the NPIs. Since we need to generate data according to unknown relationships, we firstly deal with taking into account a variety of possible links

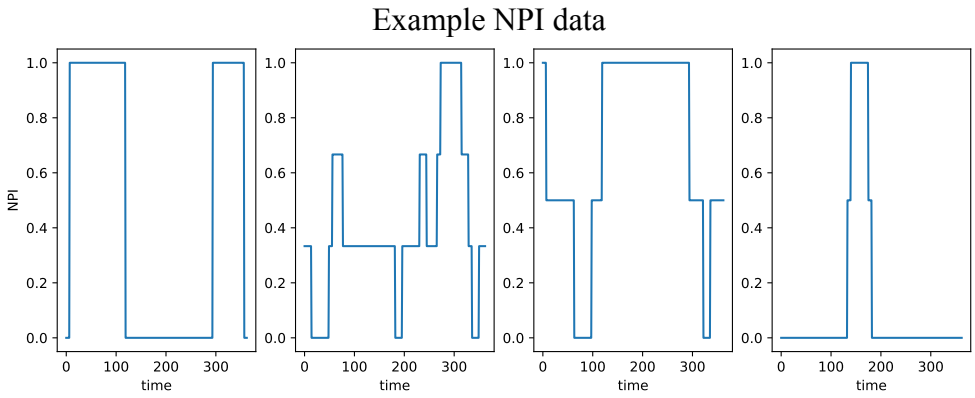


Figure 5.2: A few examples of synthetic data representing the NPIs.

between β and the NPIs by employing several different common functions, such as polynomials, exponentials and trigonometric functions, along with their combinations. Examples include $f(x) = x\log(x)$ and $f(x) = \sin^2(x)$. To achieve more realistic distributions, we inject random gaussian noise into the generation as well, to several different extents. The actual values were obtained by performing a rebinning on such distributions, since real NPIs from the OxCGRT dataset take integer values between 0 and a variable upper bound (up to 5) (see Section 5.1.1): to obtain such a staircase distribution, we assigned a random number of bins between 2 and 5 to each distribution, and then accordingly resampled the NPI to take discrete values. In the end, normalization was applied in order to obtain values belonging in the range $[0, 1]$. Some examples are reported in Figure 5.2.

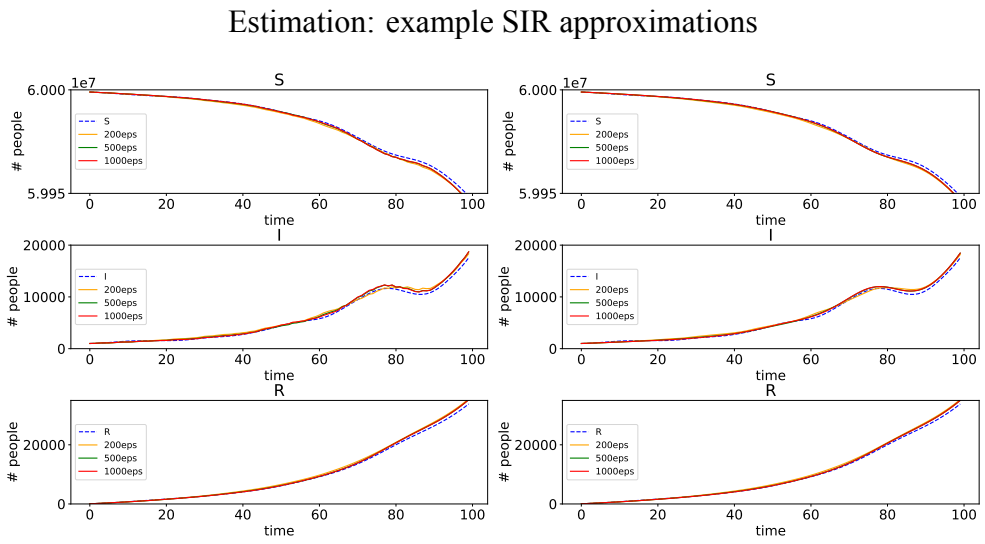
After some experiments, we increased the complexity of the hypothesized dependency by introducing a further variable u in the generation of the NPIs, which we produced relying on the same techniques employed for β. Accordingly, the generation of the NPIs was modified to take into account multiple independent variables rather than a single one, which we achieved in different ways: examples include $f(x, y) = x\log(xy)$ and $f(x, y) = \sin(x + y)$. The variable u was left unknown within the predictive phase of the first experimental setting (see Section 3.1.3), thus not being provided as target along training. A more comprehensive overview of the employed shapes is given in Appendix A in Figure A.2

5.2 Results

The experimental phase was carried out by deploying models firstly on synthetic data and then on real data, in several different configurations of hyper-parameters (see Chapter 3). As already mentioned in the introduction to the present Chapter, this approach has the purpose of providing a more solid validation of the model performance so as to more effectively estimate the quality of the results on real data. Accordingly, we firstly present the results on the synthetic dataset, and then move to historical data.

5.2.1 Synthetic data

In this section, we describe the results and corresponding evaluations on synthetic data regarding both approaches: we firstly address the *SIR-NPI* approach (see Section 3.1), then moving to the *End-to-End* approach (see Section 3.2).

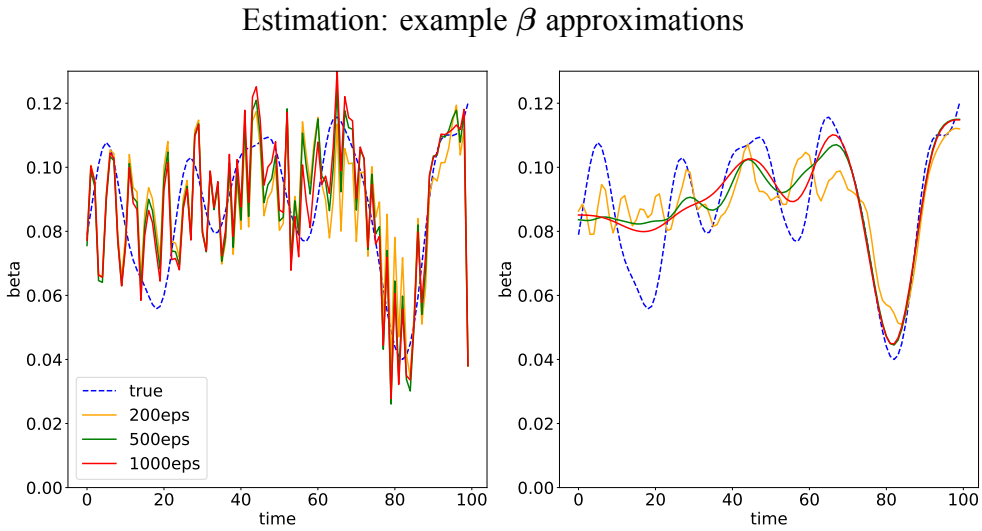


(a) Original loss function **(b)** Regularized loss function

Figure 5.3: Example of fitting results for SIR compartments employing 200, 500, 1000 training epochs.

SIR-NPI

As already discussed, this first method relies on a partitioning of the predictive task: we firstly seek for a curve-fitting *estimate* of the β parameters relying



(a) Original loss function **(b)** Regularized loss function

Figure 5.4: Example of fitting results for β employing 200, 500, 1000 training epochs.

the epidemiological data, then we try to obtain a predictive mapping of the NPIs into such parameters.

Estimation results are obtained by curve-fitting on epidemiological data (see Section 3.1.2): an example is reported on Figures 5.3 and 5.4, respectively showing results on the SIR compartments and on β for a simple Gaussian mixture. Specifically, we employed the loss function as in Equation 3.4 setting $\epsilon = 10^{-3}$ and an unitarian Euler step size ($K = 1$) for the simulation, training the network on 100-samples time series.

A first observation suggests that an increase in the training epochs provides quite significant improvements on β, though much less relevant ones on the SIR compartments: this also allows us to corroborate our arguments about the possibility to obtain good quality curve-fitting even in case of sensible deviations of the estimated parameters from ”real” ones, likely due to adjustments on behalf of the approximation method.

Similar results can be observed by looking at a more comprehensive review of obtained results, which is reported in Appendix B in Figures B.1 and B.2: in particular, the increase in training epochs appears to be particularly relevant in the approximation of β whenever quick variations are present, which is remarkable in sine-squared distributions. Still, approximations of the SIR compartments are much less prone to variations, again indicating that also non

accurate estimates for β can lead to a good fit for the epidemiological curve.

Overall result metrics are summarised in Figure 5.5, reporting the evaluation of results performed through three different scores, namely Mean Squared Error (MSE), Mean Absolute Percentage Error (MAPE) and R^2-score: we focus on the β parameters and the I compartment, which are the most critical variables.

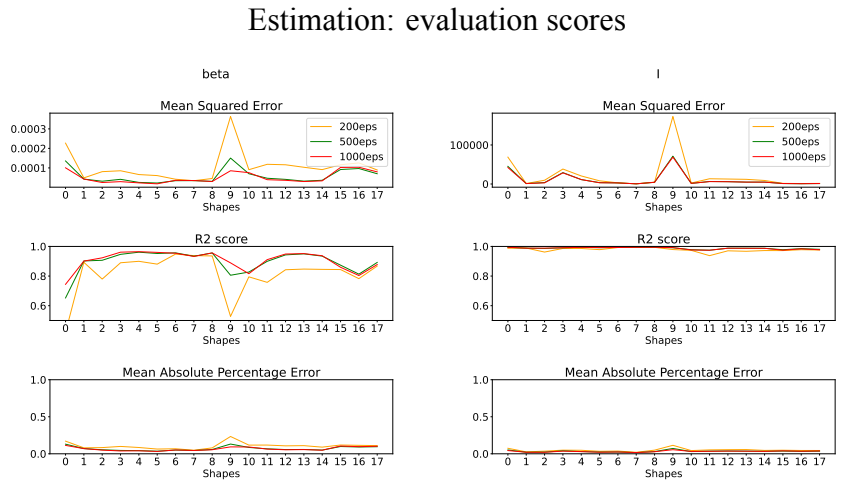


Figure 5.5: Scores over β parameters and the I compartment, varying on different shapes for β.

We firstly point out the correlation between the model performances on β and the I compartment, indicating that the better the estimation for the parameters, the better the fitting of the epidemiological curve. Besides, the quality of I predictions stays very high even when β estimates are not as precise, showing that the Euler simulation can lead to accurate fitting even with non-optimal parameters, as expected (see Sections 1.3 and 3.2.1). We observe a systematical, albeit not always remarkable, enhancement of the model performance alongside the increase in the number of epochs. However, we identify some shapes for β as more challenging: we notice how issues mainly arise in the case of oscillations characterized by both high frequency and large amplitude.

Prediction differs from the previous task, since we are evaluating results directly on the obtained estimates for β, rather than on the actual epidemiological curve (see Section 3.1.3). This setting implies the need for separate splits for training and testing, due to the information flow of ground truth throughout the backward pass. Some training and test results are reported in

Figure 5.6, with focus on β and the I compartment. By inspecting the plots, we generally obtain good fitting results for both variables. Even in some cases in which β is not correctly captured in its oscillations — for instance on the 3rd plot in Figure 5.6 — we notice how the I compartment reaches nonetheless good matching: a possible explanation for this behaviour relies in the low absolute values for the parameters, which results in little to no variations in the evolution of the epidemiological curve.

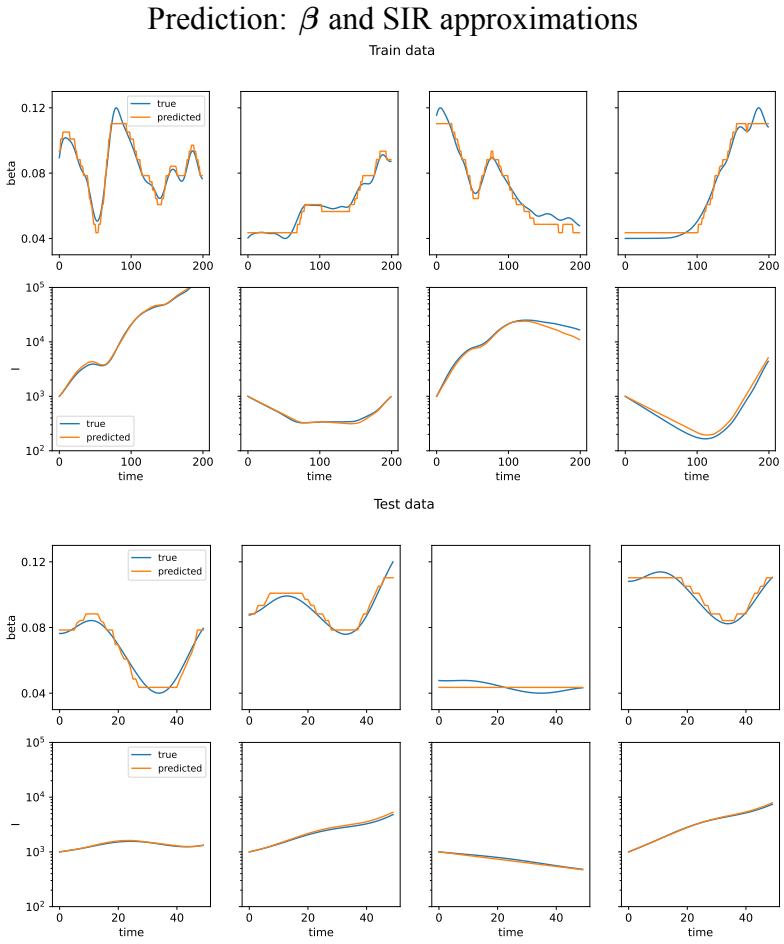
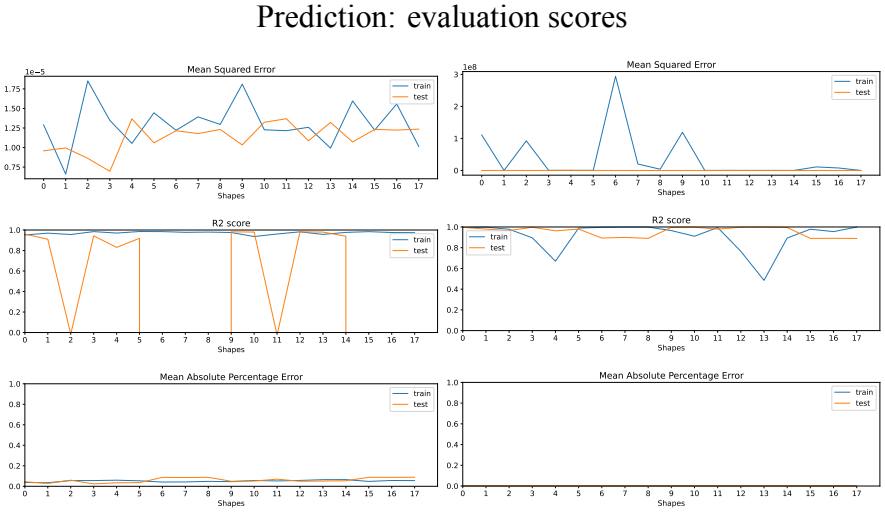


Figure 5.6: Some prediction results for β and the I compartment on both training and test data.

With the purpose of performance analysis, figure 5.7 reports evaluation metrics relative to both training and test data: results regarding the β parameters generally comply with expectations, as the model reaches overall good results on test data, still performing worse than on training data. Actually, the R^2 test score for β shows very bad performances on a few shapes in particular, even with some negative values, still not affecting the performance on

the infected curve. The I compartment exhibits a more peculiar behaviour, in particular the R^2 score is generally higher for test data than for training data.



(a) beta **(b)** I

Figure 5.7: Scores over β parameters and the I compartment, varying on different shapes for β. MAPE results on the I compartment are nearly not visible, since they range in the interval $10^{-7} \div 10^{-5}$.

End-to-End

In this section, we present results obtained through the End-to-End approach described in Section 3.2: we performed a wider set of experiments with respect to the first approach, due to both the simpler setup and accordingly faster computation times, and the encouraging initial results. Specifically, we performed

hyper-parameter	benchmark values
β upper bound	0.12, 0.15, 0.2, 0.3
noise(%)	0, 20, 50, 100
hidden layers	16, 32, 16+16, 32+32
K	1, 5, 10

Table 5.1: The different parameters used to perform the benchmark of the End-to-End model. The lower bound for β was set to 0.04. Values reported for hidden layers indicate the number of neurons for each layer.

a benchmark on several different combinations for the following parameters: the range of β, the amount of noise injected in the NPI generation, the number K of Euler increment, the number and size of hidden layers of the neural network, alongside the usual 18 different shapes for β: values are reported in

Table 5.1. We tested every possible combination of the parameters, namely 3456 different training runs: accordingly, we are forced to present aggregated results.

Some test results are reported in Figure 5.8: it is immediately noticeable how fitting results for β are less accurate with respect to the previous approach, while the epidemiological curve exhibits good matches.

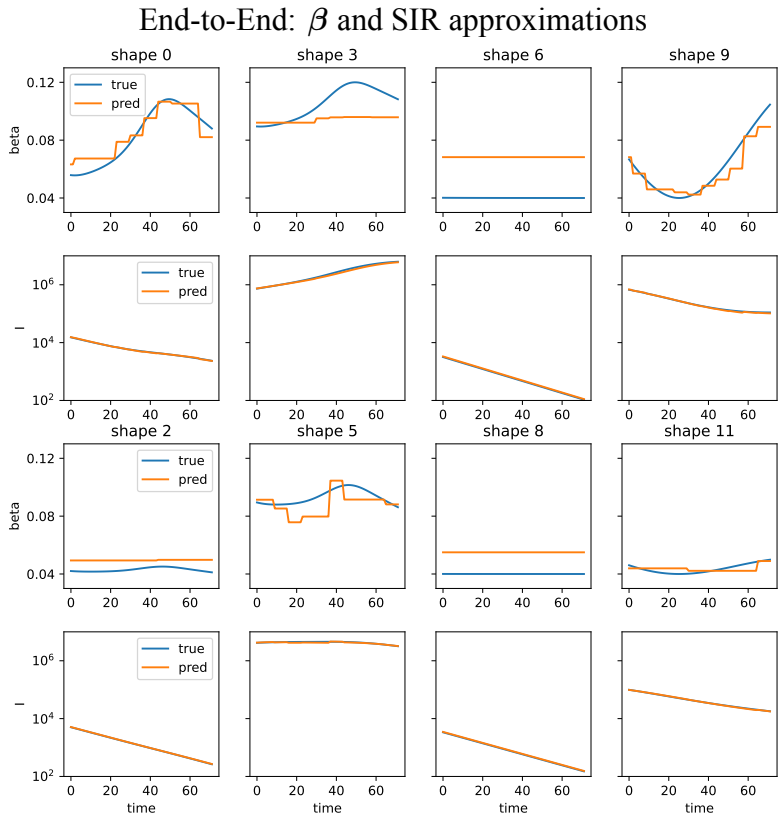


Figure 5.8: Some prediction results for β and the I compartment on test data.

Averaged evaluation results are presented in Figure 5.9, which reports the usual metrics for both the β parameters and the I compartment. It is immediately noticeable how the parameters achieve quite bad performances, still not affecting a generally satisfying fit for the epidemiological curve. More comprehensive results are reported in Appendix B in Figures B.3 and B.4, separately aggregated by different values of K, upper bound for β and noise ratio.

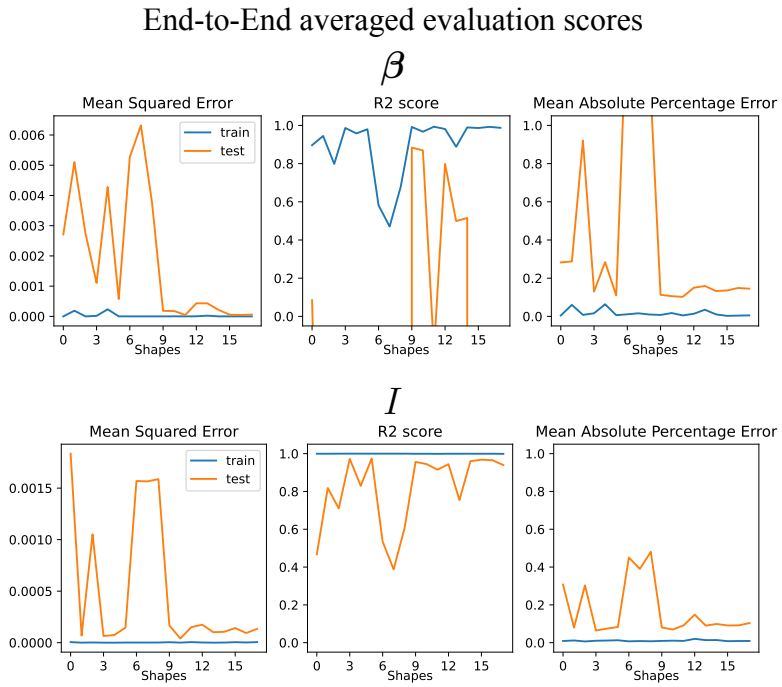
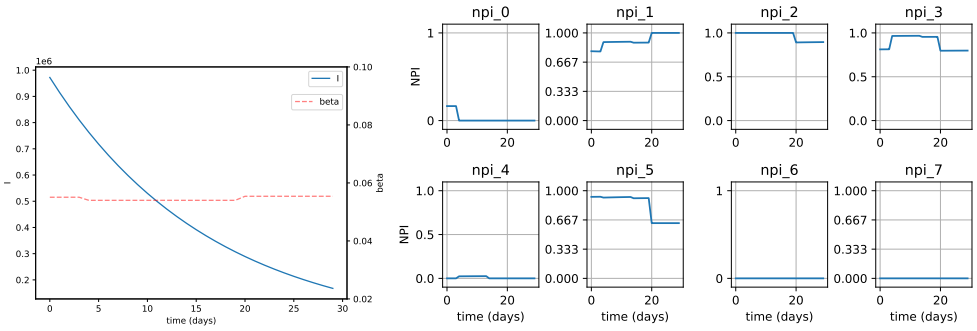


Figure 5.9: Scores for β parameters and the I compartment, varying on different shapes for β and averaged over all performed test runs.

Prescription

Eventually, we performed an exploration of the Empirical Model Learning approach, employing the End-to-End model as predictive component and the CBC solver for the optimization process (see Section 4.2).

Prescription: example NPIs prescriptions, β and SIR approximations



(a) Predicted β and I **(b)** Prescripted NPIs

Figure 5.10: Example of daily prescription and prediction results for 30 days.

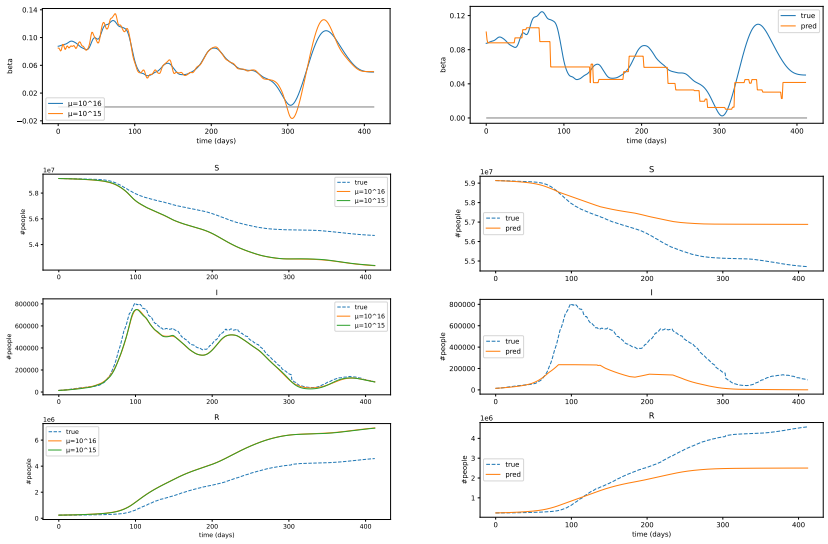
Example results are reported in Figure 5.10: the grids represent the allowable discrete values for the NPIs. Although values for β are nearly constant, we can appreciate a monotonic lowering of the infected people, which well

reflects our goal. Since we have no ground truth for *best interventions*, we do not provide a formal evaluation of prescripted NPIs: however, we can notice how variations in some NPIs generally correlate with even slight variations in the output values, i.e. of the β parameters. Constraints on β were imposed through an hyperparameter $c = 10^{-1}$ (see Section 4.2).

5.2.2 Historical data

In this section, we describe the results and corresponding evaluations on historical data regarding both approaches: as already done for synthetic data, we firstly address the *SIR-NPI* approach (see Section 3.1), then moving to the *End-to-End* approach (see Section 3.2). Data employed to perform the experiments refers to the period starting on August 8th, 2020 and ending on October 1st, 2021: the total number of days is 413, sequentially divided between training and test data with a test ratio of 20%.

Estimation and prediction: β and SIR approximations



(a) Estimation: 500 training epochs, different regularization learning rates μ. The grey line represents the x-axis $\beta = 0$, which is useful to identify negative unfeasible results.

(b) Prediction results: 5000 training epochs, 2 hidden layers of size 32 with L2 regularization.

Figure 5.11: Fitting results for β and corresponding SIR compartments.

SIR-NPI

Estimation results are reported in Figure 5.11, which shows two instances differing in the parameter μ, i.e. the regularization learning rate used to perform Lagrangian Dual optimization on the λ parameter (see Section 3.1.2). While we can observe little to no differences between the two estimates within the epidemiological fitting curves, we can appreciate a smoother behaviour for β relative to a stronger regularization. Moreover, it is worth pointing out how the I compartment exhibits the most accurate approximation, due to its higher relevance within the loss function.

Prediction was performed by employing β estimates obtained with $\mu = 10^{16}$. Results show significant deviations with respect to both the β parameters and the epidemiological curve: Figure 5.11 shows quite noisy predictions for the parameters and accordingly not accurate predictions for the SIR compartments. Actually, the main issue here appears to be the scale, since the trend shape of all compartments closely resembles real data.

	β		SIR	
	train	test	train	test
MSE	$2.58 \times 10{-}4$	$2.28 \times 10{-}3$	6.08×1012	2.12×1011
$R2$	0.68	-3.96	0.18	-3.06
$MAPE$	2.62×10^{-1}	4.75×10^{-1}	2.94×10^{-1}	2.57×10^{-1}

Table 5.2: SIR-NPI: Prediction evaluation scores for β and SIR compartments on both training and test data.

Table 5.2 summarizes the evaluation scores for the predictive task in terms of usual metrics. We don't appreciate good results, especially through the inspection of the R^2 score:

End-to-End

Prediction results obtained through the End-to-End approach are reported on Figure 5.12: to evaluate the predictions for the β parameters, we employed the estimates obtained with the SIR-NPI model with Lagrangian Dual learning rate $\mu = 10^{16}$ as target data. Both *daily* and *weekly* results are reported, in reference to Section : in the former case, predictions are obtained through a single simulation step, while in the latter case the simulation is performed 7

times in a row, each time employing the updated estimated curve as input for the next prediction step. Remarkably, daily results achieve systematic but not overwhelming better performance, indicating that weekly predictions can be provided with reasonable accuracy as well. The main difference with respect to the results obtained with the SIR-NPI approach consists in the very good evaluation scores achieved by the SIR compartments on both training and test sets, even though the performance on β remains mostly unsatisfactory: in particular, the model seems to exhibit a sort of regression to the mean, settling for an average estimation of the parameters able to correctly reproduce the epidemiological curve, rather than trying to capture its oscillations.

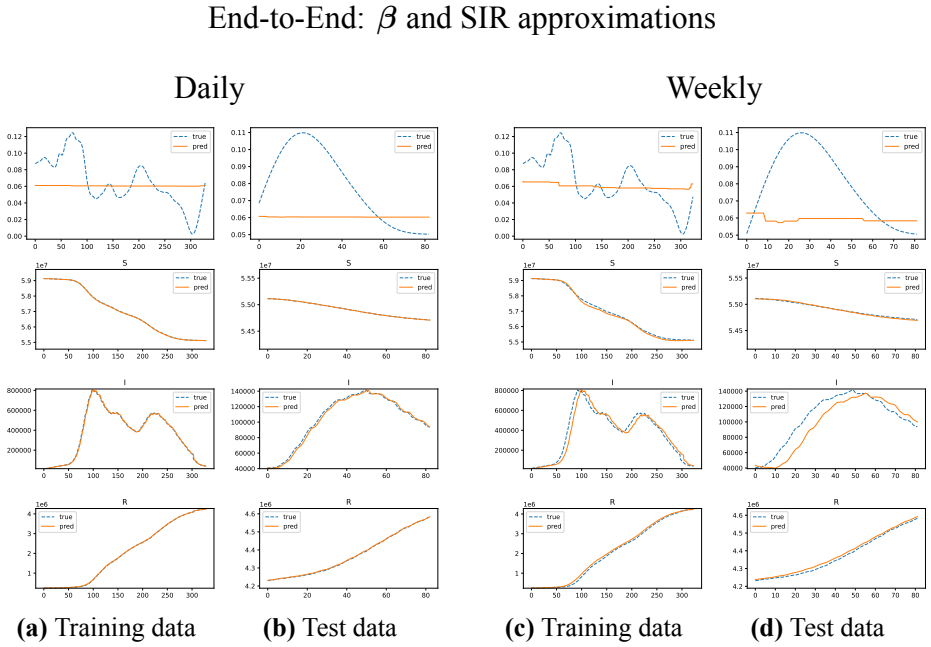


Figure 5.12: Daily and weekly prediction results for β and corresponding SIR compartments employing 500 training epochs and 2 hidden layers of size 32 with L2 regularization. Notice that training and test data are reported on different scales in order to make the results easier to inspect.

	β		SIR	
	train	test	train	test
MSE	8.59×10^{-4}	8.04×10^{-4}	1.29×10^{9}	4.70×10^{7}
$R2$	-0.01	-0.75	0.999	0.998
$MAPE$	9.58×10^{-1}	2.58×10^{-1}	$1.12 \times 10-2$	$7.27 \times 10-3$

Table 5.3: End-to-end: daily prediction evaluation scores for β and SIR compartments on both training and test data.

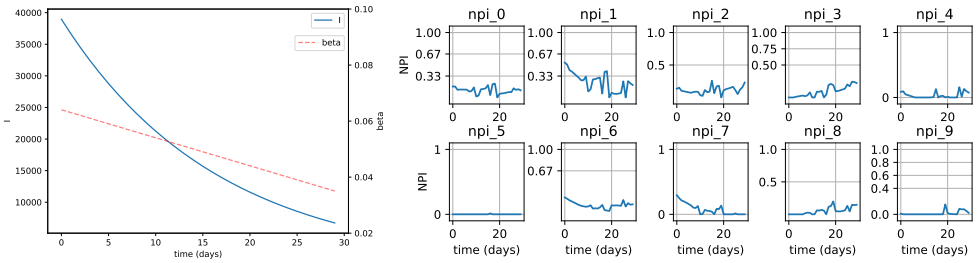
	β		SIR	
	train	test	train	test
MSE	7.80×10^{-4}	8.56×10^{-4}	5.53×10^{10}	1.62×10^{9}
$R2$	0.09	-0.97	0.98	0.92
$MAPE$	9.01×10^{-1}	2.54×10^{-1}	6.80×10^{-2}	4.26×10^{-2}

Table 5.4: End-to-end: weekly prediction evaluation scores for β and SIR compartments on both training and test data.

Prescription

Eventually, we performed an exploration of the Empirical Model Learning approach, employing the End-to-End model as predictive component and the CBC solver for the optimization process (see Section 4.2), as we already did for synthetic data.

Prescription: example NPIs prescriptions, β and SIR approximations



(a) Predicted β and I **(b)** Prescripted NPIs

Figure 5.13: Example of daily prescription and prediction results for 30 days.

Example results are reported in Figure 5.13: the grids represent the allowable discrete values for the NPIs. We can appreciate a monotonic lowering in both the β parameters and the I compartment, which correctly reflects our goal. However, we should highlight here how β tends to always take the minimum allowable value, which is given by the constraint (see Section 4.2), tuned to $c = 10^{-3}$ after some experiments. As for synthetic data, we have no ground truth for *best interventions* for historical data as well: still, we can observe a systematic propensity of NPIs to take low values, which is physically counter-intuitive, since it represents less restricting measures. A possible explaination for this unexpected behaviour could be researched into the input space being scarcely represented across training data, leading to unpredictable output for the ML model when it comes to edge cases such as values close to 0.

Conclusion

The present work explored an application of Empirical Decision Model Learning to epidemic scenarios, setting up a working approach to effectively forecast the evolution of outbreaks and accordingly suggest proper interventions plan. We validated our models through synthetic data generation, which provided a robust assessment of the capabilities of the models with respect to simulated data and paved the way to their application to real-world cases. We were able to obtain good results regarding the predictive task, namely the forecast of epidemiological evolution based on current NPIs. However, we could not simultaneously achieve good fittings for both the β parameters and the compartments of the SIR model, likely due to the simplifications entailed by the epidemiological background theory. As already discussed, this kind of situation is not ideal when it comes to interpretability, since either the epidemic is not correctly forecasted or the predictions for β fail to represent a reliable proxy of the outbreak evolution. Nevertheless, within a Decision-Focused Learning setup (see Section 2.2) the major interest resides in the capability to obtain high-quality mapping of decision variables into observables: this perspective allows us to reasonably neglect poor results regarding the β parameters, as long as a high-quality forecasting of the epidemiological curve is achieved.

As for the prescriptive task, we obtained promising results as long as synthetic data was concerned, while the application to real data posed a harder challenge. In particular, we identified the tendency of the prescriptor to produce NPIs close to their lower bound, which is scarcely represented within training data. This behaviour could be caused by the optimization process trying to *exploit* unexplored regions of the input space. In other words: the prescriptor seeks for the minimization of the β parameters as a function of the NPIs, which is encoded as a ML model. However, a non-uniform sampling of the input space within training data is reflected into unpredictable behaviour

when it comes to model forecasts. As a consequence, the optimization process will search for whatever input features lead to a minimization of the output, regardless of the robustness of the prediction.

Even though there is still much work to do, we have introduced a method to combine model-driven and data-driven approaches to predictive epidemiology through knowledge injection and Universal Differential Equations, as long as setting up an approach to prescriptive epidemiology through an hybrid strategy which exploits Machine Learning and Combinatorial Optimization techniques.

Bibliography

[1] F. Baldo, L. Dall'Olio, M. Ceccarelli, R. Scheda, M. Lombardi, A. Borghesi, S. Diciotti, and M. Milano. Deep learning for virus-spreading forecasting: a brief survey. *CoRR*, abs/2103.02346, 2021. arXiv: 2103.02346. URL: https://arxiv.org/abs/2103.02346.

[2] A. Borghesi, F. Baldo, and M. Milano. Improving deep learning models via constraint-based domain knowledge: a brief survey. *CoRR*, abs/2005.10691, 2020. arXiv: 2005.10691. URL: https://arxiv.org/abs/2005.10691.

[3] D. della Protezione Civile. Italian COVID-19 Data, 2020. URL: https://github.com/pcm-dpc/COVID-19.

[4] P. L. Donti, B. Amos, and J. Z. Kolter. Task-based end-to-end model learning in stochastic optimization, 2019. arXiv: 1703.04529 [cs.LG].

[5] W. Duan, Z. Fan, P. Zhang, G. Guo, and X. Qiu. Mathematical and computational approaches to epidemic modeling: a comprehensive review. *Frontiers of Computer Science*, 9:806–826, 2014.

[6] EMLlib - An Empirical Model Learning library.

[7] F. Fioretto, T. W.K. Mak, F. Baldo, M. Lombardi, and P. Van Hentenryck. A Lagrangian Dual Framework for Deep Neural Networks with Constraints Optimization, 2020. arXiv: 2001.09394 [cs.LG].

[8] J. Forrest, T. Ralphs, H. G. Santos, S. Vigerske, L. Hafer, J. Forrest, B. Kristjansson, jpfasano, EdwinStraver, M. Lubin, rlougee, jpgoncal1, Jan-Willem, h-i-gassmann, S. Brito, Cristina, M. Saltzman, tosttost, F. MATSUSHIMA, and to-st. Coin-or/cbc: release releases/2.10.7, version releases/2.10.7, January 2022. DOI: 10.5281/zenodo.5904374. URL: https://doi.org/10.5281/zenodo.5904374.

[9] T. Hale, N. Angrist, R. Goldszmidt, B. Kira, A. Petherick, T. Phillips, S. Webster, E. Cameron-Blake, L. Hallas, S. Majumdar, and H. Tatlow. A global panel database of pandemic policies (Oxford COVID-19 Government Response Tracker). *Nature Human Behaviour*, 5, April 2021. DOI: 10.1038/s41562-021-01079-8.

[10] Istat. Bilancio Demografico Mensile, 2020. URL: https://demo.istat.it/bilmens/index.php?anno=2020.

[11] V. Janko, N. Reščič, A. Vodopija, D. Susic, C. Masi, T. Tusar, A. Gradišek, S. Vandepitte, D. De Smedt, J. Javornik, M. Gams, and M. Lustrek. Optimizing non-pharmaceutical intervention strategies against covid-19 using artificial intelligence, November 2021. DOI: 10.21203/rs.3.rs-1047502/v1.

[12] I. Locatelli, B. Trächsel, and V. Rousson. Estimating the basic reproduction number for COVID-19 in Western Europe. *PLOS ONE*, 16(3):1–9, March 2021. DOI: 10.1371/journal.pone.0248731. URL: https://doi.org/10.1371/journal.pone.0248731.

[13] M. Lombardi, M. Milano, and A. Bartolini. Empirical decision model learning. *Artificial Intelligence*, 244:343–367, 2017. ISSN: 0004-3702. DOI: https://doi.org/10.1016/j.artint.2016.01.005. URL: https://www.sciencedirect.com/science/article/pii/S0004370216000126. Combining Constraint Solving with Mining and Learning.

[14] M. A. Lozano, Ò. G. i. Orts, E. Piñol, M. Rebollo, K. Polotskaya, M. A. Garcia-March, J. A. Conejero, F. Escolano, and N. Oliver. Open data science to fight covid-19: winning the 500k xprize pandemic response challenge. In Y. Dong, N. Kourtellis, B. Hammer, and J. A. Lozano, editors, *Machine Learning and Knowledge Discovery in Databases. Applied Data Science Track*, pages 384–399, Cham. Springer International Publishing, 2021. ISBN: 978-3-030-86514-6.

[15] R. Miikkulainen, O. Francon, E. Meyerson, X. Qiu, D. Sargent, E. Canzani, and B. Hodjat. From prediction to prescription: evolutionary optimization of non-pharmaceutical interventions in the covid-19 pandemic. *IEEE Transactions on Evolutionary Computation*, PP:1–1, March 2021. DOI: 10.1109/TEVC.2021.3063217.

[16] C. Rackauckas, Y. Ma, J. Martensen, C. Warner, K. Zubov, R. Supekar, D. Skinner, and A. J. Ramadhan. Universal Differential Equations for Scientific Machine Learning. *CoRR*, abs/2001.04385, 2020. arXiv: 2001.04385. URL: https://arxiv.org/abs/2001.04385.

[17] L. von Rueden, S. Mayer, K. Beckh, B. Georgiev, S. Giesselbach, R. Heese, B. Kirsch, M. Walczak, J. Pfrommer, A. Pick, R. Ramamurthy, J. Garcke, C. Bauckhage, and J. Schuecker. Informed machine learning - a taxonomy and survey of integrating prior knowledge into learning systems. *IEEE Transactions on Knowledge and Data Engineering*:1–1, 2021. ISSN: 2326-3865. DOI: 10.1109/tkde.2021.3079836. URL: http://dx.doi.org/10.1109/TKDE.2021.3079836.

[18] B. Wilder, B. Dilkina, and M. Tambe. Melding the data-decisions pipeline: decision-focused learning for combinatorial optimization, 2018. arXiv: 1809.05504 [cs.LG].

Appendix A

Synthetic data - generation

0	gaussian	9	sin
1	gaussian_exp	10	sin_exp
2	gaussian_inverse_exp	11	sin_inverse_exp
3	gaussian_sigmoid	12	sin_sigmoid
4	gaussian_sigmoid_exp	13	sin_sigmoid_exp
5	gaussian_sigmoid_inverse_exp	14	sin_sigmoid_inverse_exp
6	gaussian_inverse_sigmoid	15	sin_inverse_sigmoid
7	gaussian_inverse_sigmoid_exp	16	sin_inverse_sigmoid_exp
8	gaussian_inverse_sigmoid_inverse_exp	17	sin_inverse_sigmoid_inverse_exp

Table A.1: β shapes used to generate synthetic data.

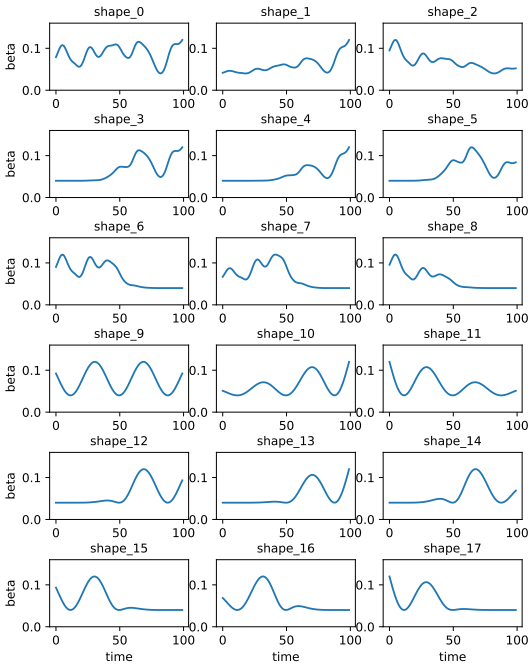


Figure A.1: Graphical representation of β generated shapes.

0	$f(\beta) = \exp(\beta)$	$f(\beta, u) = \exp(\beta + u)$
1	$f(\beta) = \frac{1}{1+\exp(\beta)}$	$f(\beta, u) = \frac{1}{1+\exp(\beta - u)}$
2	$f(\beta) = \exp(-\beta)$	$f(\beta, u) = \exp(-\beta * u)$
3	$f(\beta) = \frac{1}{1-\exp(-\beta)+10^{-5}}$	$f(\beta, u) = \frac{1}{1-\exp(-\beta * u)+10^{-5}}$
4	$f(\beta) = \log(\beta + 10^{-5})$	$f(\beta, u) = \log(\beta + 2 * u + 10^{-5})$
5	$f(\beta) = \beta * \log(\beta + 10^{-5})$	$f(\beta, u) = x * \log(\beta * u + 10^{-5})$
6	$f(\beta) = \sin(\beta)$	$f(\beta, u) = \sin(\beta + u)$
7	$f(\beta) = \sin^2(\beta)$	$f(\beta, u) = \sin^2(\beta * u)$

Table A.2: Functions used to generate NPI synthetic data as a function of β either with or without the hidden variable u. The small constant 10^{-5} is added to ensure that functions are well-defined for $\beta >= 0$.

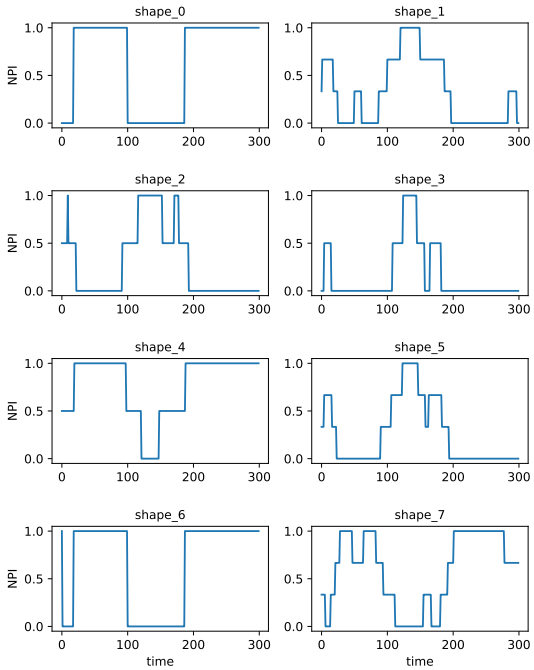


Figure A.2: Graphical representation of NPI generated shapes.

Appendix B

Synthetic data - results

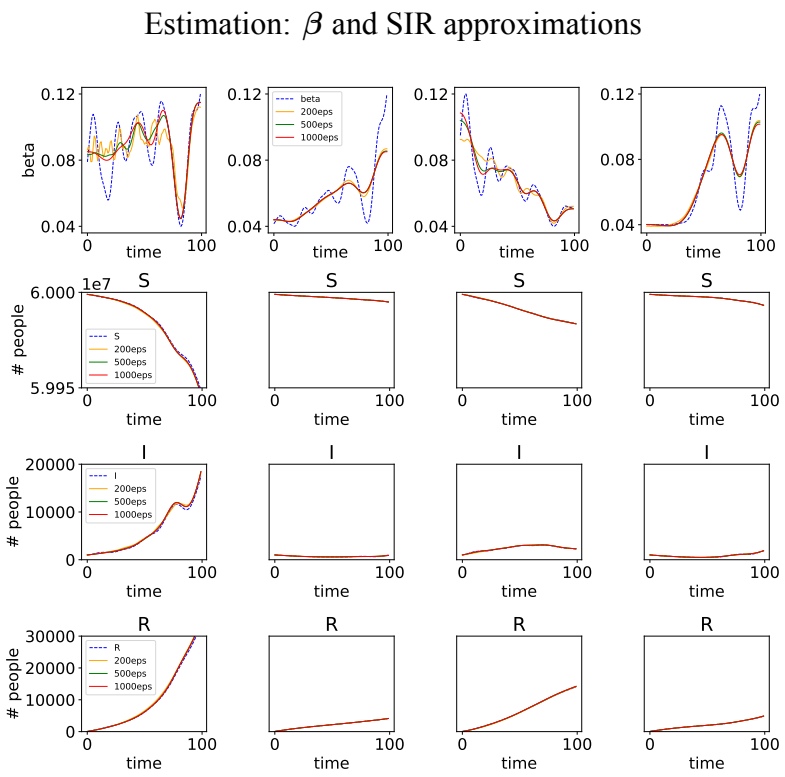


Figure B.1: Fitting results for some Gaussian mixtures-shaped β and corresponding SIR compartments employing 200, 500, 1000 training epochs.

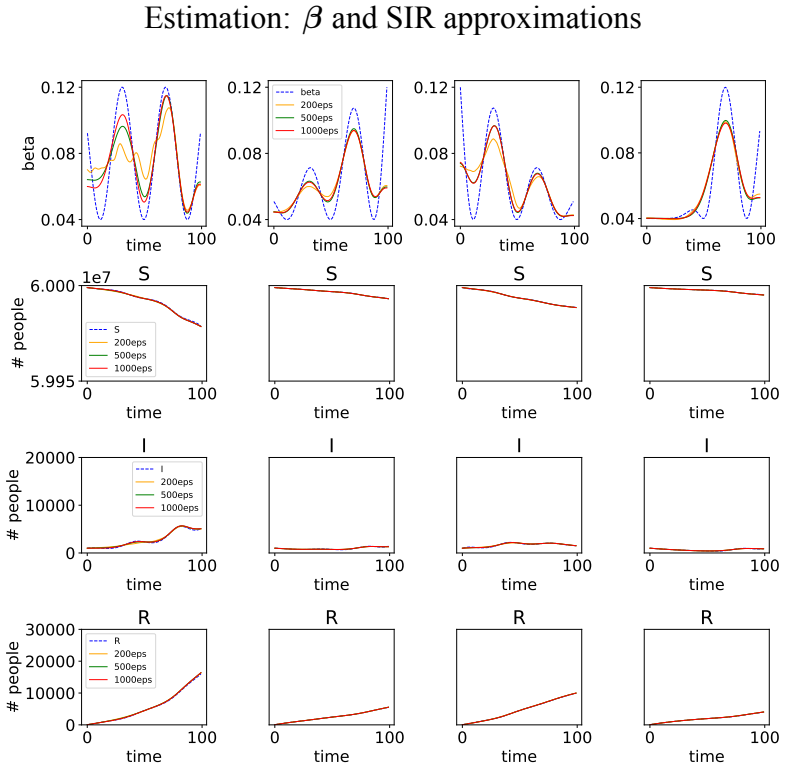
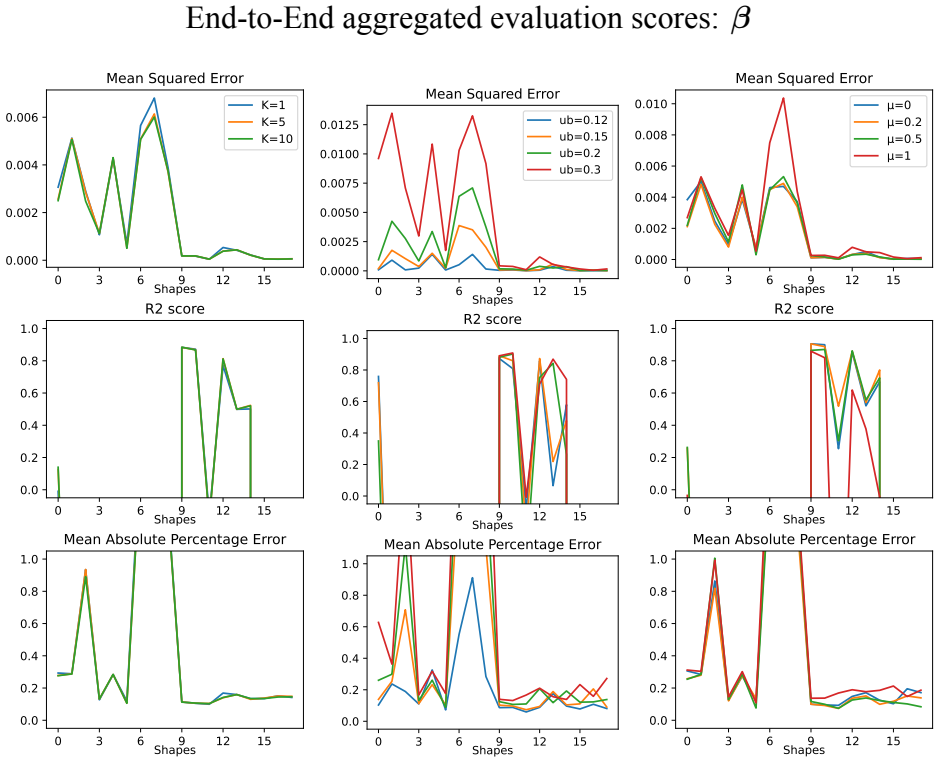
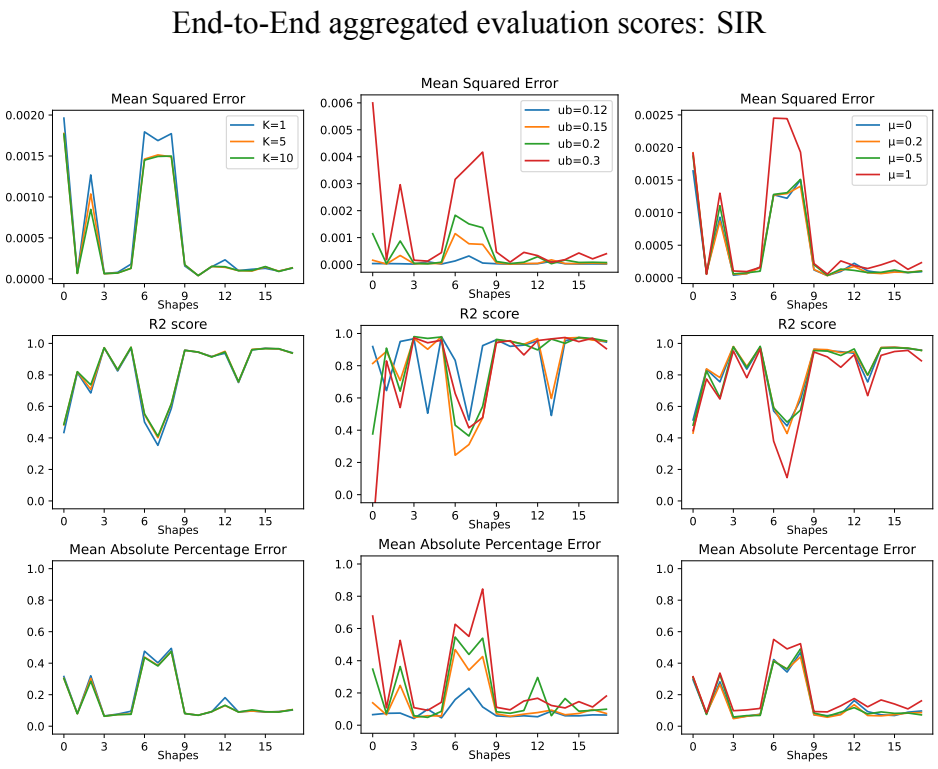


Figure B.2: Fitting results for some sine-squared-shaped β and corresponding SIR compartments employing 200, 500, 1000 training epochs.



(a) K **(b)** upper bound **(c)** noise

Figure B.3: Scores for β parameters over varying K, upper bound and noise amount.



(a) K **(b)** upper bound **(c)** noise

Figure B.4: Scores for the I compartment over varying Euler increment (K), β range (in particular, the upper bound) and noise in the generation of NPIs (μ.