Bergische Universität Wuppertal

Fakultät für Mathematik und Naturwissenschaften

Institute of Mathematical Modelling, Analysis and Computational Mathematics (IMACM)

Preprint BUW-IMACM 21/35

Sarah Treibert and Matthias Ehrhardt

An Unsupervised Physics-Informed Neural Network to Model COVID-19 Infection and Hospitalization Scenarios

December 9, 2021

http://www.imacm.uni-wuppertal.de

Noname manuscript No.
(will be inserted by the editor)

An Unsupervised Physics-Informed Neural Network to Model COVID-19 Infection and Hospitalization Scenarios

Sarah Treibert · Matthias Ehrhardt

Received: date / Accepted: date

Abstract In this paper we replace the standard numerical approach of estimating parameters in a mathematical model using numerical solvers for differential equations with an unsupervised physics-informed neural network (PINN). This neural network requires only an undetermined vector of time instances as input to learn the underlying parameters of the model, which are used for the loss calculations.

The underlying model is an extended susceptible-infected-recovered (SIR) model in which the transitions between disease-related population groups, called compartments, and the physical laws of epidemic transmission dynamics are expressed by a system of ordinary differential equations (ODEs). The system of ODEs and its time derivative are included in the residual loss function of the PINN, additional to the data error between the current network output and the time series data of the compartment sizes. Further, we illustrate how this PINN approach can also be used for differential equation-based models such as the proposed extended SIR model, called SVIHDR model.

In a validation process, we investigate the performances of the numerical technique of non-standard finite differences (NSFD) as well as the exclusively data-driven recurrent neural network method, more specifically the long short term memory (LSTM), in generating future COVID-19 scenarios based on the parameters identified by the PINN.

Most importantly, we obtain a two-step or hybrid approach, as the PINN is then used to generate future COVID-19 outbreak scenarios. The week in which

S. Treibert
Bergische Universität Wuppertal
Angewandte Mathematik und Numerische Analysis
Tel.: +49-202-439-5299
E-mail: sarah.treibert@uni-wuppertal.de

M. Ehrhardt
Bergische Universität Wuppertal
Angewandte Mathematik und Numerische Analysis
E-mail: ehrhardt@uni-wuppertal.de

the predictions begin is chosen in this work as the second week of November 2021. In addition, the predictions of infection and hospitalization rates obtained from the NSFD and LSTM are compared to the PINN predictions.

Keywords physics-informed neural networks · compartment models · COVID-19 · SARS-CoV-2 · epidemiology · Long-Short-Term-Memory

1 Introduction

To pursue the goal of developing a hybrid method for predicting future epidemiological trends, different mathematical and data-driven approaches are combined or compared here in order to generate COVID-19 scenarios. All methods are applied in this work to data on COVID-19-related population group sizes in Germany, but are also applicable to other countries for which data are available.

The COVID-19 pandemic is currently one of the most discussed topics around the world. The first cases of severe acute respiratory syndrome coronavirus type 2 (SARS-CoV-2) occurred in Asia in December 2019, but were not reliably identifiable at that time. The People's Republic of China experienced a peak of about 4,600 cases per day in mid-February 2020, but by March 2020, the epidemic was largely contained in China and other Asian parts of the world. Europe experienced the first wave of the pandemic in March and April 2020, with, for example, about 5,840 new daily infections in Germany in late March and 13,260 new daily infections in France in mid-April 2020.

While infection numbers in Europe were generally low in summer 2020, peaks were observed in the United States ($\sim$ 67,000 infections/day), Brazil ($\sim$ 46,000 infections/day), and India ($\sim$ 93,000 infections/day) at certain times between July and September 2020. The third wave was characterized by approximately 35,000 new daily infections in Italy in mid-November 2020, 25,000 new daily infections in Germany around Christmas, 60,000 new daily infections in the United Kingdom in early January 2021, and severe lockdowns within Europe in fall 2020 and winter 2020/2021. The summer of 2021 was characterized by a relaxation of intervention measures in Europe. However, some countries experienced catastrophic COVID-19 events, such as India with approximately 390,000 new daily infections in early May 2021 [18].

In November 2021, the fourth wave of the pandemic reached Europe despite a fully vaccinated proportion of 67.6 % in Germany, 68.9 % in the United Kingdom, 74.8 % in Italy, 77.7 % in France, and 80.6 % in Portugal [26]. Nearly 40,000 new daily infections were observed in Germany on November 14th 2021, as well as in the United Kingdom. Achieving even higher vaccination rates and providing booster vaccinations for all to maintain a high level of infection protection are policy issues of concern to all countries. The dangers posed by mutant virus variants such as the delta variant (B.1.617.2), which was first discovered in India in October 2020 and is now the dominant variant infecting people in several countries such as Germany, or the omicron variant (B.1.1.529) discovered in the autumn of 2021, are also being discussed in medicine and the literature. According to the Robert Koch-Institute (RKI), the mRNA vaccines from BioNTech/Pfizer, Moderna, and AstraZeneca are expected to have a protective effect of approximately 90 % against severe infection with the alpha (B.1.1.7) variant and 75 % against symptomatic infection with the delta (B.1.617.2) variant [22].

The mathematical model used in this work to describe the population dynamics of COVID-19 is a SVIHDR model. It is based on a system of ordinary differential equations (ODEs). Most mathematical models describing the spread of the disease employ classical compartments, which the Susceptible-Infected-Recovered (SIR) structure is the most basic form of [1]. Over the past almost two years, a variety of compartmental models have been introduced as enhanced SIR models to study various aspects of the COVID-19 pandemic.

Here, we do not establish the simplest version of compartment model. Instead, we develop some kind of extended model, which complements the basic SIR model by a vaccinated, a hospitalized and a deceased class. The hospitalized compartment is added due to the high significance of hospitalization number predictions for hospital capacity planning and the assessment of the number of severely diseased individuals at pandemic times. We do not include an exposed compartment, that usually is the first enhancement to the SIR model and incorporates infected people who are not (yet) infectious, so pre-symptomatic and potentially asymptomatic individuals.

In our model, pre-symptomatic individuals are condensed with symptomatic people in the infected compartment, so that we have a single infected compartment of people not hospitalized. Since determining the proportion of asymptomatic individuals in the total infected population is not our goal at this point, we do not include a class of infected individuals who are asymptomatic, but assume at least very mild symptoms in infected individuals. The degree of infectivity of infected individuals can be controlled by adjusting the transmission rate in the model.

Thus, our model includes a vaccination rate and the proportion of the population vaccinated each week. Therefore, the model is adaptable to different vaccination scenarios. In addition, the general transmissibilities of SARS-CoV-2 and its variants, which are constantly changing, lead to altered protective effects of available vaccines. The established model includes a transmission rate explained in section 2.

The first mathematical method used in this work, called *Physics-Informed Neural Network* (PINN), explained in the subsection 3.1, is used to estimate the transmission rate based on the data available in Germany. The PINN itself combines a data-driven method (here based on compartment size data, e.g., number of infections) with the developed ODE system so that it incorporates physical laws. In other words, this approach trades off between the data-based and physical loss functions in the training process. This reduces the space of feasible solutions to those that satisfy a 'physical law' to some degree, i.e., an SVIHDR compartmental model in this case. The ODE system corresponding to the model serves as an additional constraint in the training phase, which is encoded by an appropriate additional residual loss term. More specifically, the

PINN loss function consists of the two weighted terms *data loss* and *residual loss*. The data loss is calculated as the difference between the current network output in terms of infection or hospitalization numbers and the reported 2019 coronavirus pandemic (COVID-19) data covering weeks between March 2020 and November 2021. The residual loss is based on a mathematical model with a system of ordinary differential equations that describes the main population dynamics observed during the COVID-19 pandemic. A PINN approach for the simple SIRD model was proposed by Malinzi et al. [14] and a PINN approach for a SIR based vaccination model was described by Torku et al. [28]. In contrast to this, Zeroual et al. [30] compared different pure deep learning models for forecasting COVID-19 cases and found the Variational AutoEncoder (VAE) algorithm to be superior.

Raissi et al. [19] explain that PINNs are neural networks that embed physics as a regularization term in the loss function. They say that given a sufficient number of data points and an expressive neural network architecture, they can achieve good approximation accuracy if the given differential equation is well-posed and has a unique solution. PINNs can also be viewed as a *surrogate model* for solving differential equations by incorporating additional data or as a data-driven correction (or even discovery) of the underlying physical system. One motivation for this hybrid approach can be seen in the observed non-compliance of some of the individuals with social distancing (or physical distancing) and hygiene rules. This type of behavior is difficult to formulate in ODEs, but is included in the neural network training data.

Olumoyin et al. [17] use the term *Epidemiology-Informed Neural Network (EINN)*, which describes a type of feedforward neural network that incorporates epidemiological dynamics such as lockdown into its loss function. Their EINN learns solutions for the so-called asymptomatic SIR model, i.e., the proportion of asymptomatic infected individuals to of the total number of infected individuals.

Shaier et al. [25] use the term *Disease-Informed Neural Networks (DINN)* to refer to a type of PINN-based neural network that can be applied to increasingly complex systems of differential equations describing various known infectious diseases. The DINN formulation does not require a training, validation, or testing data set, as is the case with most neural networks; instead, the model learns the infectious disease models and predicts the parameters that generated them.

Because our PINN operates based on transmission and transition dynamics in a population affected by COVID-19, estimates transmission rate parameters, and incorporates a transmission rate that can incorporate remedial measures such as quarantine and contact restrictions, our PINN can be described as a special type of EINN designed to predict COVID-19 incidence. Because our PINN uses a system of differential equations to learn the parameters that generate it, this PINN can also be considered a DINN. Our approach of using PINN-identified parameters of an ODE system to predict infection and hospitalization rates by using the PINN itself in a slightly modified form, an LSTM method, and a numerical method of NSFD is innovative. The uniqueness of our approach lies in the fact that we use our PINN for parameter identification and give it a second input of initial compartment size data to generate accurate future compartment size scenarios, and then apply a purely data-driven method and a purely numerical method, both of which are well suited for epidemic predictions and validated by us, so that we can compare the predictions of the data- and ODE-based PINN with their predictions.

Long et al. [12] use a PINN to identify weekly and daily time-varying parameters in a system of ODEs, and then use the LSTM method to make future predictions for parameter values over the next four weeks. In contrast, we use PINN-identified parameters to generate future scenarios of infection and hospitalization rates using PINN, LSTM method, and numerical NSFD scheme.

The data used consist of infection and hospitalization rates as well as vaccination, death, and cure rates for Germany obtained from the RKI [20,21]. The PINN also works on the basis of the established dynamic ODE system, which forms the core of the model and is developed in section 2. The transmission rate is one of the most important parameters affecting the occurrence of infections and thus the established ODE system. Therefore, changes in transmissibility due to mutations or altered susceptibility of the underlying population are part of the model-based predictions.

The exact procedures used in this work are described in subsection 3.2 in section 3. In this work, the estimation of certain model-specific parameters is performed by the PINN mentioned above. A standard solver for ODEs (explicit embedded Runge-Kutta (4,5) scheme [5]) and a model-specific non-standard finite difference scheme (NSFD) serve as numerical integration methods, and a long-term memory (LSTM)-based neural network serves as a non-numerical approach, both of which are used for comparison and validation. Here, these three methods were applied to data from calendar weeks 10 in 2020 to 28 in 2021 to make predictions so that differences from the actual available data from calendar weeks 29 to 45 in 2021 became apparent. Predictions of infection and hospitalization rates were then made using all approaches as well as PINN. Different weights for the loss terms are used as examples in the prediction section to analyze the impact of weight modification.

2 Model Structure

In this work, a SVIHDR compartmental model was developed based on the basic SIR model introduced by Kermack and McKendrick in 1927. The SIR model consists of three compartments of susceptible (S), infected (I), and recovered (R) individuals. Susceptible individuals have not yet become infected but may become ill. Infected individuals have already become infected. In the basic SIR model, they are also capable of infecting susceptible persons. Therefore, they are assumed to be infectious and may or may not have symptoms. Recovered individuals have overcome the disease and are no longer ill.

2.1 The SIR Model in Epidemiology

The basic SIR model assumes that no births or deaths enter the system, that the population is closed so that no one enters or leaves a compartment from the outside, and that recovered individuals are completely immune so that they can never be reinfected. The total size of the population at a time t is denoted by $N(t)$. The satisfaction of the equation

$$N(t) = S(t) + I(t) + R(t) \qquad \text{with } N\colon [0,T] \to \mathbb{N},$$

means that the number of individuals in the system is the sum of the compartment sizes at each time point considered $t \in [0,T]$. The system must have initial conditions $S(0)$, $I(0)$, $R(0)$ to be well-defined [15, p. 11]. The population size $N(t)$ is constant if the derivative of $N(t)$ is zero. If there is no natural death rate and no recruitment or birth rate in the system, this constancy is given. The individuals in the system are infected, i.e., they migrate from compartment S to I at a rate $\theta(t)$, which is defined as

$$\theta(t) := \beta \cdot \gamma(t) \cdot \bigl(1-q\bigr) \cdot I(t)\,, \tag{1}$$

where β is the transmission risk and $\gamma(t)$ is a time-dependent contact rate. The parameter q symbolizes the degree of strength of intervention, quarantine, and isolation measures implemented. For example, when more infectious individuals are isolated, fewer further infections occur. The rate

$$\Theta(t) := \theta(t) \cdot \frac{S(t)}{N(t)} \tag{2}$$

is called a standard incidence rate, and $\beta \cdot \gamma(t) \cdot I(t)$ is the force of infection.

2.2 The SVIHDR Model

The basic model is extended in this work to include a vaccinated compartment, a hospitalized compartment, and a deceased compartment. Infected individuals remain infected for T_I days until they recover, when a proportion ξ of all transiting individuals are hospitalized. Thus, the rate ω_1 at which persons per unit time (week) pass from compartment I to R is given by

$$\omega_1 = \frac{1-\xi}{T_I}, \tag{3}$$

and the rate η at which individuals are reach the compartment H per unit of time is defined as

$$\eta = \frac{\xi}{T_I}. \tag{4}$$

It is assumed that hospitalized individuals cannot infect susceptible individuals because of their isolated state. They remain infected T_H days from the time of their hospitalization. A proportion $\mathcal{M}$ of all transiting persons die from disease-related causes rather than recover. Consequently, the rate ω_2 at which persons per unit time pass from compartment H to R is given by

$$\omega_2 = \frac{1 - \mathcal{M}}{T_H}, \tag{5}$$

and the rate λ at which individuals reach the deceased compartment D per unit of time is

$$\lambda = \frac{\mathcal{M}}{T_H}. \tag{6}$$

The vaccinated compartment V contains all susceptible individuals who have received a COVID-19 vaccination. It is reached from compartment S at a rate $\mathcal{V}$. Since vaccination does not guarantee complete immunity to infection, i.e., we speak of a *leaky vaccination*, it is assumed that vaccinated individuals in the system may contract the infection with a small probability. The respective rate at which vaccinated individuals pass into the infected compartment I is $\kappa \cdot \theta(t)$, where κ denotes the residual probability of infection after vaccination.

BioNTech/Pfizer's Comirnaty and Moderna's Spikevax vaccines are about 95 % effective, AstraZeneca's Vaxzevria vaccine is about 80 % effective, and Johnson & Johnson's Janssen vaccine is about 65 % effective. Thus, a leaky-vaccinated compartment is assumed, rather than an all-or-nothing vaccinated compartment. Because leakiness was assumed, all vaccinated individuals have a lower probability of contracting the infection than susceptible individuals in compartment S. When an all-or-nothing vaccine was assumed, vaccination provided complete protection from infection to a portion $\mathcal{V}$ of the susceptible class per unit time t, whereas the $1 - \mathcal{V}$ portion received no protection.

The corresponding system of ordinary differential equations (ODEs) has the following form:

$$\begin{aligned}
\frac{dS(t)}{dt} &= -\theta(t)\,\frac{S(t)}{N(t)} - \mathcal{V}\,S(t),\\
\frac{dV(t)}{dt} &= \mathcal{V}\,S(t) - \theta(t)\,\kappa\,\frac{S(t)}{N(t)},\\
\frac{dI(t)}{dt} &= \theta_I(t)\,\big(1+\kappa\big)\frac{S(t)}{N(t)} - \big(\eta + \omega_1\big)\,I(t),\\
\frac{dH(t)}{dt} &= \eta\,I(t) - \big(\omega_2 + \lambda\big)\,H(t),\\
\frac{dD(t)}{dt} &= \lambda\,H(t),\\
\frac{dR(t)}{dt} &= \omega_1\,I(t) + \omega_2\,H(t)
\end{aligned} \tag{7}$$

The dynamical system described by equation (7) is depicted in Figure 1. Blue arrows from one compartment to another indicate a transition, where the compartment from which a red dashed arrow emanates can infect susceptibles.

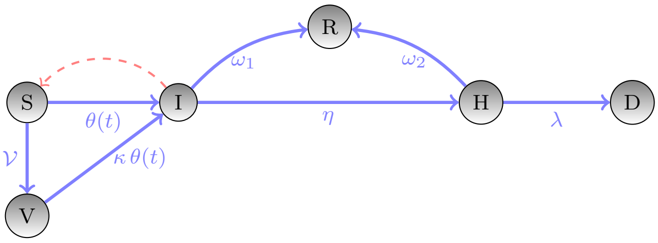

Fig. 1 Compartment model for the SVIHDR model

3 Methods

Data were obtained from the Robert Koch-Institute (RKI) [20, 21] and the German COVID-19 vaccination dashboard [4]. They refer to the calendar week 10 in 2020 to 45 in 2021. Weekly case-hospitalization, case-fatality and vaccination rates were computed on the basis of the given data sets. The RKI registers deceased individuals, in whom the SARS-CoV-2 pathogen was detected, as people who died from COVID-19. In Subsec. 3.1, the approach of PINN is explained. Subsec. 3.2 explains the technical procedure of building the PINN. In Subsec. 3.3 the technique of Nonstandard Finite Difference Schemes (NSFD) is explained and Subsec. 3.4 gives a short explanation of implementation of the Long-Short-Term-Memory (LSTM) method, which are used in the validation and prediction part.

3.1 Physics-informed Neural Networks for Compartment Models

The basic concept of *physics-informed neural networks* (PINN) is to incorporate the laws of dynamical systems modeled by ordinary or partial differential equations into a deep learning framework. The loss function of the corresponding neural network includes not solely the so-called *loss error* related to the difference between the output of the network and the reported data used, but also the so-called *residual error* related to the ODEs or PDEs. The sum of these two errors is then minimized in the least squares sense. The PINN does not require any data on the predicted parameters so that it belongs to unsupervised learning methods.

The vector ϑ of all parameters included in (7) is given by

$$\vartheta(t) = [\beta, \gamma, q, \mathcal{V}, \kappa, \xi, T_I, T_H, \mathcal{M}, T_H]^\top . \tag{8}$$

The parameters in ϑ can be partitioned into fixed parameters p_f and trainable parameters p_t, that we select as follows:

$$\begin{aligned} p_f &:= [\gamma, q, \mathcal{V}, \xi, T_I, T_H, \mathcal{M}, T_H]^\top , \\ p_t(t) &:= [\beta, \kappa]^\top . \end{aligned}$$

A neural network

$$\mathcal{PINN}_p^W : \mathbb{R} \to \mathbb{R}^6$$

has to be defined in order to be able to discretize the system of ODEs (7). The superscript W stands for the weights used during the forward and backward propagation in the neural network. We are initially given l points in time $T = [t_1, \ldots, t_l]^\top$, that are the obligatory input to the neural network. A first version of the PINN with exclusively T as the input vector was implemented. The vector of all $n = 6$ compartment sizes is given by

$$\mathcal{K}_p(t) = [\mathcal{K}_p^1(t), \ldots, \mathcal{K}_p^n(t)]^\top .$$

We are given reported compartment size data

$$\hat{\mathcal{K}}_p(t) = [\hat{\mathcal{K}}_p^1(t), \ldots, \hat{\mathcal{K}}_p^n(t)]^\top .$$

As the ODEs in the system (7) can be described by $\frac{d\mathcal{K}(t)}{dt} = -F_p(\mathcal{K})$ for $t \in [t_1, t_l]$ it can be defined that

$$F_p(\mathcal{K}) = [F_p^1(\mathcal{K}), \ldots, F_p^n(\mathcal{K})]^\top ,$$

where $\mathcal{K}_p^j(t) \in \mathcal{C}^1(\mathbb{R})$ and $F_p^j \in \mathcal{C}(\mathbb{R})$ $\forall j \in \{1, \ldots, n\}$. We want to approximate the solution $\mathcal{K}_p = [S, V, I, H, D, R]^\top : \mathbb{R} \to \mathbb{R}^6$ using the PINN performing error minimization [7].

The parameters W and $p_t(t)$ are optimized during the backpropagation process of the neural network such that $\mathcal{PINN}_p^W$ fits the reported data $\hat{\mathcal{K}}$ in a least-squares sense [7]. Doing so, we obtain the *loss error* defined by

$$MSE_{\mathcal{U}} := \frac{1}{l} \sum\nolimits_{j=1}^{l} ||\mathcal{PINN}_p^W(t_j) - \hat{\mathcal{K}}_j||^2 . \tag{9}$$

Moreover, the *residual error*

$$MSE_{\mathcal{F}} := \frac{1}{l} \sum\nolimits_{j=1}^{l} ||\mathcal{F}_p(\mathcal{PINN}_p^W, t_j)||^2 \tag{10}$$

is added to the loss error in the training loop of the PINN, in which the weights and trainable parameters are updated per step. It holds that

$$\mathcal{F}_p(NN_p^{W,b}, t_j) := \frac{d\mathcal{PINN}_p^W(t)}{dt} - F_p\big(\mathcal{PINN}_p^W(t)\big) . \tag{11}$$

The residual error symbolizes the physics-informed part of the loss function $\mathcal{L}_\alpha$ since it incorporates the system of ODEs. Let $\alpha \in [0, 1]$ be a weighting factor that can be applied to the residual error in the following backward propagation with loss optimization performed during training. We define that

$$\mathcal{L}_\alpha := \alpha \, MSE_{\mathcal{U}} + (1 - \alpha) \, MSE_{\mathcal{F}} \tag{12}$$

such that the final minimization problem of the neural network becomes

$$\underset{W, p_t}{\operatorname{argmin}} \big(\mathcal{L}_\alpha\big) .$$

The code was enhanced by the option of having the initial sizes of the six compartments corresponding to a specific selected point in time as an additional input to the PINN. The aim of this was the more precise generation of infection scenarios that stronger depend on the used starting point in time. This enhanced PINN was later used for predictions.

3.2 Procedures of Building the PINNs

A single feed-forward PINN was used for each of the compartments I and H, which are those compartments appearing in equations (7) apart from S. This was done in order that separate parameter vectors $p_t(t) := [\beta, \kappa]^\top$ were estimated per run of the neural network for I and H. Among all model parameters, it is most difficult to assign realistic values to β and κ from raw data. Values for the case-fatality, the case-hospitalization and the vaccination rate were computed from the available RKI data [20, 21]. The fixed model parameters were computed as $\mathcal{V} = 0.013517486$, $\xi = 0.079718848$ and $\mathcal{M} = 0.026720524$ from the given data sets [20, 21]. According to the RKI, contagiosity strongly recedes after a mean of 10 days of infectedness [23]. In a paper concerning the hospitalization of COVID-19 cases compared to flu epidemics, the mean duration of COVID-19-induced hospitalization in Germany was 10 days, whereby the length of hospital stay of people transferred to an intensive care unit was 16 days and the hospital sojourn time for ventilated individuals was 18 days on average [27]. We set the parameter concerning the length of stay in the infected state to $T_I = 1.42$ weeks, and used a slightly higher value of $T_H = 1.5$ weeks for the hospital sojourn time in our implementations. We selected the transmission rate β and the transmission variation coefficient for the vaccinated κ as trainable parameters. In further implementations or using a different model, other or more model parameters could be selected as trainable.

We used three hidden linear layers for the PINN. Moreover, a linear output layer was applied to obtain a compartment size output, and a ReLU output layer was used for the trained parameter vector. 87 points in time (weeks) were the obligatory input and the initial sizes of the 6 compartments were a second optional input for more exact infection and hospitalization number forecasts. The output size is 87 for each compartment size as compartment sizes for the 87 points in time were wanted. The output size was 5 for the parameter vector because the vector contains 5 parameters that were to be predicted. The ReLu function was used as activation functions per layer. The Adam algorithm was selected as optimizer. Different layers and activation function were tested and compared with respect to output compartment size curves. The selected ones yielded the most reasonable size ranges. The ReLU output layer was used for the parameter vector prediction particularly because non-negative values were wanted. The number of conducted steps per training session was 20,000.

The PINN for later compartment size scenario generation received the additional input $\mathcal{K}^0 := [S^0, V^0, I^0, H^0, D^0, R^0]^\top$. Here, the activation function used for the second layer of the infection or hospitalization number part was changed into the tanh-function. In order to compute the derivative $d\mathcal{PINN}_p^W(t)/dt$ the PyTorch automatic differentiation package `torch.autograd.grad` was used. It computes and returns the sum of gradients of the respective compartment size tensor $\mathcal{PINN}_p^W(t)$ with respect to the input time tensor t. Results of the compartment size and trainable parameter vector predictions of the PINN are described in Subsection 4.1.

Subsequently, the prognosticated parameters β and κ were used as the inputs to an ODE integrator and additionally a nonstandard finite difference (NSFD) scheme in a validation process. NSFD schemes are explained in Subsec. 3.3. They preserve certain properties like the positivity or the asymptotic behaviour of the analytic solution of differential equations on the discrete level. Their most important characteristic is, in many cases, the complete absence of the elementary numerical instabilities which plague common finite difference schemes [16].

The long-short-term memory (LSTM) technique was used as another method of forecast. It is explained in Subsection 3.4. In the whole validation procedure, the errors between the predictions obtained through these two methods and the actual available data were analyzed and compared. The corresponding results can be found in Subsection 4.2. Future scenarios drawn with the aid of the two numerical methods of standard ODE integration, NSFD scheme and PINN are given in Subsection 4.3 For both the PINN and the LSTM implementations the PyTorch Library was used.

3.3 Nonstandard Finite Difference Schemes

NSFD methods for the numerical integration of differential equations had their origin in a paper by Mickens published in 1989 [16]. In [29], we established an NSFD scheme for a similar compartment model as here. We implemented a simultaneous parameter estimation using a nonlinear least squares minimization of the error between time series compartment size data and the result of the NSFD-based integration of the respective system of ODEs. This does not equal the data or residual loss of our PINN approach that we use in this paper. With the optimized parameters and the NSFD scheme, we generated future COVID-19 scenarios. Here, we are now able to compare NSFD results to the results obtained using neural networks. A numerical scheme for a system of first-order differential equations is called NSFD scheme if at least one of the following conditions described in [16] is satisfied:

- The first-order derivatives in the system are approximated by the generalized forward difference method (forward Euler method) $\frac{du_n}{dt} \approx \frac{u_{n+1}-u_n}{\phi(h)}$, where $u_n = u(t_n)$ and $\phi \equiv \phi(h)$ is the so-called *denominator function* such that $\phi(h) = h + \mathcal{O}(h^2)$.
- The nonlinear terms are approximated in a non-local way, for instance by a suitable function of several points of a mesh, like $u^2(t_n) \approx u_n u_{n+1}$ or $u^3(t_n) \approx u_n^2 u_{n+1}^2$.

If we define $\tilde{N} = N - D = S + V + I + H + R$ and additionally add the recruitment or system-inflow rate ψ equalling the mortality rate μ, i.e. $\psi = \mu$, and describing the recruitment e.g. birth of new individuals that can get infected, we obtain the differential equation

$$\frac{d\tilde{N}(t)}{dt} = \mu \cdot \Big(1 - \tilde{N}(t)\Big). \tag{13}$$

It is solved by

$$\tilde{N}(t) = 1 + \Big(\tilde{N}^0 - 1\Big) \cdot e^{-\mu \cdot t} = \tilde{N}^0 + (N^0 - 1) \cdot (e^{-\mu \cdot t} - 1). \tag{14}$$

with $\tilde{N}^0 = S(0) + V(0) + I(0) + H(0) + R(0)$. Adding the equations in Eq. (19) yields

$$\frac{\tilde{N}^{n+1} - \tilde{N}^n}{\phi(h)} = \mu \cdot \Big(1 - \tilde{N}^{n+1}\Big). \tag{15}$$

The denominator function can be derived by comparing Equation (15) with the discrete version of Equation (14), that is

$$\tilde{N}^{n+1} = \tilde{N}^n + (\tilde{N}^n - 1) \cdot (e^{-\mu \cdot t} - 1), \quad h = \Delta t, \tag{16}$$

such that the (positive) denominator function is defined by

$$\phi(h) = \frac{e^{-\mu \cdot h} - 1}{-\mu} = \frac{1 - \mu \cdot h + \frac{1}{2} \cdot \mu^2 \cdot h^2 + \ldots - 1}{-\mu} = h - \frac{\mu \cdot h^2}{2} = h + \mathcal{O}(h^2)\ . \tag{17}$$

An even more accurate way to compute the denominator function would take into account the transition rate Υ_i at which the i^{th} compartment is entered by individuals for all model compartments $\mathcal{K}_i$, $i = 1, 2, \ldots$ [6]. In this case the parameter μ occurring in the denominator function in Equation (17) would be replaced by a parameter T^*. T^* could be determined as the minimum of the inverse transition parameters:

$$T^* = \min_{i=1,2,\ldots} \Big\{\frac{1}{\Upsilon_i}\Big\}.$$

With the aid of this and the denominator function, the NSFD discretization can be established, which is provided in Eq. (18).

$$
\begin{aligned}
\frac{S^{n+1}-S^{n}}{\phi(h)} &= -\beta \cdot I^{n} \cdot S^{n+1} - \mu \cdot S^{n+1},\\
\frac{V^{n+1}-V^{n}}{\phi(h)} &= \mathcal{V} \cdot S^{n+1} - \beta \cdot \kappa \cdot I^{n} \cdot S^{n+1} - \mu \cdot V^{n+1},\\
\frac{I^{n+1}-I^{n}}{\phi(h)} &= \beta \cdot (1+\kappa) \cdot I^{n+1} \cdot S^{n+1} - (\eta + \omega_1 + \mu) \cdot I^{n+1},\\
\frac{H^{n+1}-H^{n}}{\phi(h)} &= \eta \cdot I^{n+1} - (\omega_2 + \lambda_1 + \mu) \cdot H^{n+1},\\
\frac{D^{n+1}-D^{n}}{\phi(h)} &= \lambda_1 \cdot H^{n+1},\\
\frac{R^{n+1}-R^{n}}{\phi(h)} &= \omega_1 \cdot In + 1 + \omega_2 \cdot H^{n+1} - \mu \cdot R^{n+1}.
\end{aligned}
\tag{18}
$$

The explicit form of the NSFD scheme corresponding to Eq. (18) with an added natural mortality rate μ is given by

$$
\begin{aligned}
S^{n+1} &= \frac{S^{n}}{1 + \phi(h) \cdot (\beta \cdot I^{n} + \mu)},\\
V^{n+1} &= \frac{V^{n} + \phi(h) \cdot S^{n+1} \cdot (\mathcal{V} - \beta \cdot \kappa \cdot I^{n})}{1 + \phi(h) \cdot \mu},\\
I^{n+1} &= \frac{I^{n}}{1 + \phi \cdot (\eta + \omega_1 + \mu - \beta \cdot (1+\kappa) \cdot S^{n+1})},\\
H^{n+1} &= \frac{\phi(h) \cdot \eta \cdot I^{n+1} + H^{n}}{1 + \phi \cdot (\omega_2 + \lambda_1 + \mu)},\\
D^{n+1} &= \lambda_1 \cdot H^{n+1} \cdot \phi(h) + D^{n},\\
R^{n+1} &= \frac{R^{n} + \phi(h) \cdot (\omega_1 \cdot I^{n+1} + \omega_2 \cdot H^{n+1})}{1 + \phi(h) \cdot \mu}.
\end{aligned}
\tag{19}
$$

3.4 Long-Short-Term Memory for Time Series Data

The LSTM is a special type of a *recurrent artificial neural network* (RNN) that is capable of learning long-term dependencies. The main problem of conventional RNN is the difficulty of learning to preserve information over multiple time steps, which is due to the *vanishing gradient problem.* In ordinary RNN, the gradients are computed using the backpropagation algorithm. The weights are updated proportionally to the partial derivative of the loss function with respect to the current weight per training iteration. The derivatives of the network are computed by successive shifting layer by layer so that they are multiplied from the final to the starting layer. The respective gradient can thus become very small, preventing effective updates. For example, when using sigmoid activation functions, m derivatives of m hidden states are multiplied successively, such that the gradient decreases exponentially.

The results of several implemented deep learning COVID-19 predictions show that the LSTM method is well-suited for COVID-19 epidemic forecasts owing to its aforesaid long-term learning capacity [2,3,11,13]. It is suggested that the prediction accuracy of LSTM models increases with the increase of training data, so that they can overcome the problems of the vanishing gradient and gradient explosion problems and have a good memory [13].

This particular type of RNN was implemented here to obtain purely data-driven validations and predictions of infection and hospitalization rates. It does not operate on the basis of model parameters ϑ, but solely on a given compartment size dataset. It learns long-term dependencies between a sequence of time series data to predict the value for the next time step. LSTMs use *memory cells* to store values of previous data. So-called *gates* are used to learn the significance of certain inputs and decide on updates and data rejection.

In the training step, an input sequence passes through an LSTM layer whose output consists of the so-called hidden and cell state and is passed to a linear layer. The predicted number of infected or hospitalized individuals is stored in the last element of a prediction list that is returned to the function. For the hidden layer, 100 neurons were selected. An output has size 1, since it represents the number of infected or hospitalized persons per week. We denote by n and d the number of input features and entries of the hidden unit, respectively. Let $x_t \in \mathbb{R}^n$ be an input feature at time t, $W_k \in \mathbb{R}^{d\times n}$, $k \in \{i,f,o,c\}$ be a weight, and $b_k \in \mathbb{R}^d, k \in \{i,f,o,c\}$ be a bias. An LSTM layer contains a memory cell with, first, an input gate

$$i_t = \sigma\bigl(W_i[h_{t-1}, x_t] + b_i]\bigr) \in (0,1)^d, \tag{20}$$

that decides which data is updated, secondly a forget gate

$$f_t = \sigma\bigl(W_f[h_{t-1}, x_t] + b_f]\bigr) \in (0,1)^d, \tag{21}$$

which determines which previous layer is discarded and which remains in the current state, and thirdly, an output gate

$$o_t = \sigma\bigl(W_o[h_{t-1}, x_t] + b_o]\bigr) \in (0,1)^d, \tag{22}$$

that decides which parts of the cell state are updated. The current cell memory C_t with d cell units is computed as

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t \in \mathbb{R}^d, \quad C_0 = 0, \tag{23}$$

where

$$\tilde{C}_t = \tanh\bigl(W_c[h_{t-1}, x_t] + b_c\bigr). \tag{24}$$

The cell state C_t can be regarded as the global memory over all time steps or long-term memory capability of the network. Finally, the output of the LSTM cell is given by

$$h_t = o_t \cdot \tanh(C_t) \in (0,1)^d, \quad h_0 = 0. \tag{25}$$

It is the so-called hidden state, which encodes the most recent time-step.

Technically, the number of people infected or hospitalized in the following 18 weeks was predicted based on the numbers of the first 69 known weeks in the validation section. Thus, the data from the first 69 weeks were used to train the LSTM model. The performance of the model was evaluated using the values from the last 18 weeks. Therefore, the size of the test window was set to 18 [weeks]. In the prediction part, the incidence in the following 18 weeks was predicted.

In a for-loop when applying the trained network in the code, the first 18 elements of the test data set are used to obtain the first element of the predicted set. These are the last 18 elements of the entire data set, which contains a total of 87 elements. The loop is run 18 times, which is also the number of weeks for which predictions are sought. Then, the respective predicted element is appended to the test data set each time.

4 Results

The results are separated into the parameter estimation with the PINN, the validation process using the LSTM approach and the numerical methods of standard ODE integration and NSFD, as well as the part of scenario prognostication using all four presented methods.

4.1 Parameter Estimation with the Physics-informed Neural Network

The trained unsupervised PINN outputted the parameters β and κ by exclusively taking a vector of time instances as input. The data set $\hat{\mathcal{K}} = [\hat{S}, \hat{V}, \hat{I}, \hat{H}, \hat{D}, \hat{R}]^\top$ was used in the loss computation but not as a network input. The wanted vector of trainable parameters obtained from the trained neural network slightly varied between different test runs. The size of the respective compartment I or H, the course of which was obtained as a second output of the PINN, depended on the layer architecture. The average values of parameter vectors obtained from two PINN, which were the ones predicting the sizes of I and H, were used as the inputs to further Python codes used for validation and formation of future SARS-CoV-2 scenarios. They are explained in Subsec. 4.2 and Subsec. 4.3.

Figure 2 shows the loss obtained in a run of the PINN for the estimation of the trainable parameters and the size of the compartment I. It can be seen that the data loss MSE_U, computed as the mean squared error between reported and network-generated compartment size data using the Python function $MSELoss$, decreases by more than 90 % between the first and the 640^{th} training iteration. This means that the updated sizes of the compartment I approaches the reported infection numbers by weight updates of the neural network during training in these iterations. Then it remains on a level of approximately $0.17 \cdot 10^9$. The residual loss MSE_F increases from 1.079 to a

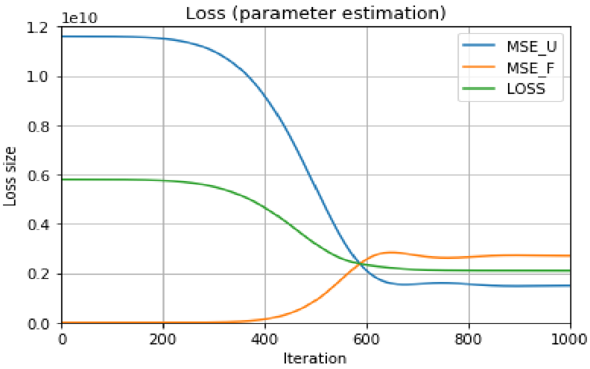

Fig. 2 Trends of the errors $MSE_{\mathcal{U}}$, $MSE_{\mathcal{F}}$ and the loss $\mathcal{L}_{0.5} = \frac{1}{2}MSE_{\mathcal{U}} + \frac{1}{2}MSE_{\mathcal{F}}$ in 1,000 observed training iterations of the PINN for the estimation of β and κ.

maximal value of $2.838 \cdot 10^9$ during the same number of iterations, is then very marginally reduced to around $2.710 \cdot 10^9$ and remains on this level. As a consequence of this, the training loss decreases from $5.8 \cdot 10^9$ in the beginning to $2.2 \cdot 10^9$ in the 640^{th} training pass, remaining on this level, and was thus reduced by 62 %.

We computed the average of all values of β and κ output by the PINN for the prediction of the size of the hospitalized compartment and the PINN for the prediction of the size of the infected compartment. Resulting values for the parameters β and κ obtained in 100 program runs were lying in the interval $[0, 1]$. Thus, we found that $\kappa \in [0.005, 0.0015]$ and $\beta \in [0.1 \cdot 10^{-9}, 0.1 \cdot 10^{-7}]$ were the final parameter intervals to be used in the validation and prediction step.

4.2 Validation: Specificity Analysis

First, an LSTM was used to plot scenarios for calendar weeks 29 through 45 in 2021. Therefore, the forecast basis was the compartment size data for calendar weeks 10 in 2020 to 28 in 2021 (beginning of March 2020 to mid-July 2021). On the other hand, we applied the numerical integration technique to the ODE system in Eq. (7) with PINN-trained parameters β and κ to generate scenarios for calendar weeks 29 to 45 in 2021 based on compartment size data from calendar weeks 10 in 2020 to 28 in 2021. As a third prediction approach, we used the NSFD scheme to create similar scenarios.

The specificity analysis consisted in calculating the difference between the reported data and the method-dependent predictions of infection or hospitalization rates for calendar weeks 29 to 45 in 2021 and observing the trends of the resulting curves. Running the respective code 100 times, we validated the performance of the LSTM method by adapting the structure of the hidden and LSTM layer by trial, e.g. the number of neurons, as well as the number of predicted time steps in the way that we finally obtained sensible results. This firstly means that the outputs of the LSTM method were reasonable and showed similar trends as the true reported data in multiple further program runs, and secondly the Euclidean distance regarding all time steps in the predicted calendar weeks was decreased.

For the implementation of the NSFD, we used values for β and κ lying in the ranges identified by the PINN (c.f. 4.1). In order to generate Figs. 3 to 7, we set $\beta = 0.000000018$ and $\kappa = 0.001$ for the ODE solver and NSFD scheme, respectively, based on the results in Subsec. 4.1. In the validation process, we adapted these two parameters only very slightly to obtain the most realistic predictions. These values lead to reasonable incidence curves, which can be seen below. Fig. 3 shows an example output of the infection number prediction of the LSTM method in the validation step. It was created after the most significant modifications.

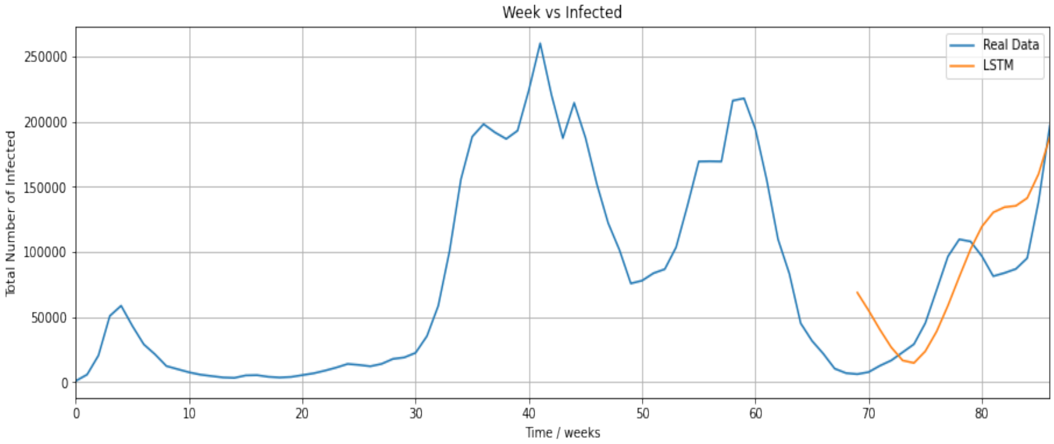

Fig. 3 The infection numbers in Germany reported to the RKI [20,21] between the calendar weeks 10 in 2020 and 45 in 2021 (*blue*) and the LSTM prediction made for the calendar weeks 29 to 45 in 2021 (*orange*) based on the data of the calendar weeks 10 in 2020 to 28 in 2021.

In Fig. 3, it can be seen that the selected LSTM prediction approaches 200,000 infected in the 87^{th} week and thus the reported data. The prediction starts in the 69^{th} week for which data were available. We see that the predicted curve begins to rise in the 75^{th} week, which is 1-2 weeks later than the curve in the reported data. The slope of the incidence curve is slightly lower than that of the real data. In addition, the LSTM predicts an increase beginning in the 78^{th} week considered, when the real data show a decrease of about 30,000 infections. In the 83^{rd} week, the LSTM curve shows a smaller and later decline than was observed in reality. Aside from that, we can see the predictions for calendar weeks 29 to 45 using data from the previous 69 weeks in the case of

LSTM or compartment size data from calendar week 29 and PINN-estimated parameters β and κ in the case of the ODE integrator and NSFD in Fig. 4.

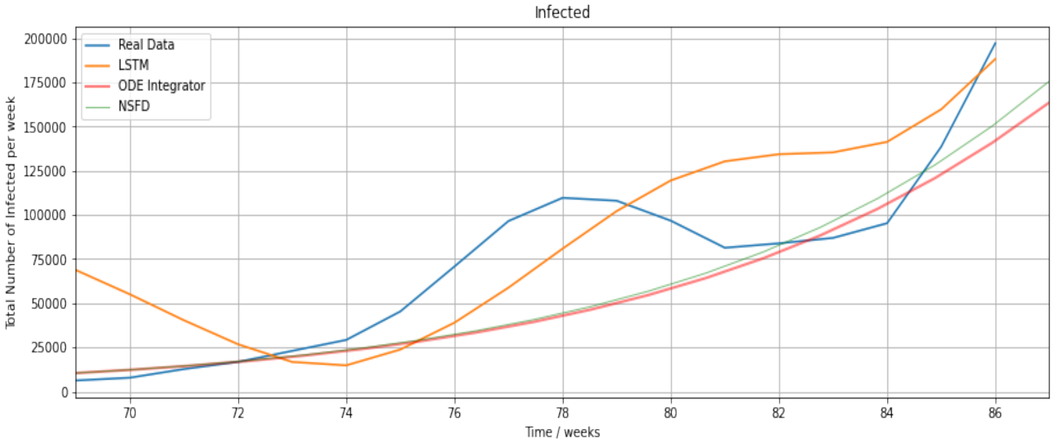

Fig. 4 The prediction of infection numbers in Germany for the calendar weeks 29 to 45 in 2021 using three different methods (solver for ordinary differential equations, numerical Nonstandard Finite Difference Scheme, data-driven Long-Short-Term-Memory) based on reported data of the calendar weeks 10 in 2021 to 28 in 2021.

In terms of specificity of the methods, Fig. 5 shows the errors i.e. differences between the reported data and the outputs of the three methods for the 18 weeks. While the LSTM starts its prediction at 7,000 infections, the

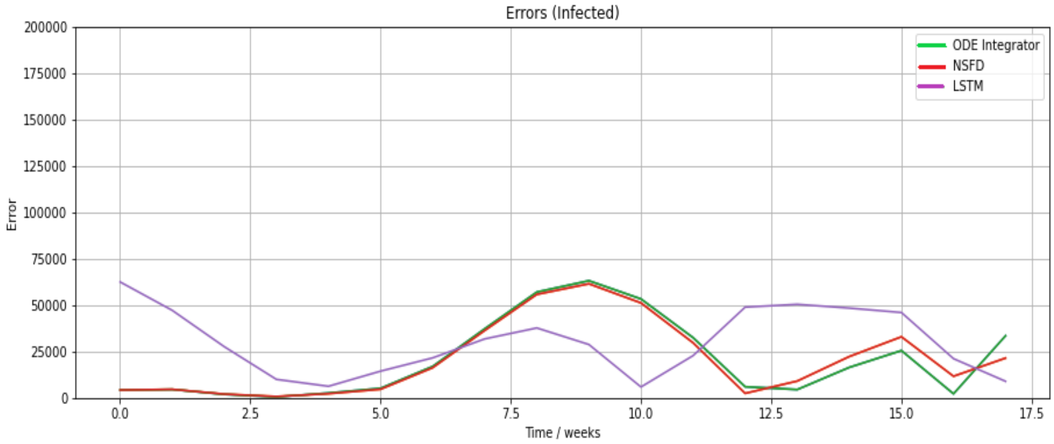

Fig. 5 The errors between the reported and predicted data in the calendar weeks 29 to 45 in 2021 using the three methods, corresponding to the predictions of Fig. 4

.

two numerical methods and the reported data start at less than 2,000 infected individuals. Consequently, the error between the prediction and the real data is 6,000 infected individuals at the beginning in Fig. 5. This deviation of the LSTM from the actual starting data is justified by its purely data-driven nature. The scenarios obtained by the ODE solver and NSFD in Fig. 4 are monotonically decreasing functions and do not exhibit oscillations or local extrema. They do not diverge noticeably at the beginning, diverge slowly from then on, and differ by 10,000-15,000 infections in the 87^{th} week considered. This pattern of divergence among them is also reflected in the errors to the reported data in Fig. 5.

The LSTM intersects the other three curves at the 73^{rd} week in Fig. 4, shows the smallest numbers at the 74^{th} week, and increases more sharply from then on than the curves predicted by the two numerical methods. The local maximum in the reported data at the 78^{th} week is not captured by the numerical methods. The LSTM prediction approaches the slope of the reported curve, although it has lower infection numbers until the 79^{th} observed week when it passes the reported curve at 110,000 infections. When the real data curve drops back to the level of the numerically generated curves (85,000 infections at week 83), the LSTM prediction shows 135,000 infections. The large increase in reported infection numbers between weeks 84 and 87 (100,000 infections) is not captured as strongly by all three methods. However, because of its higher level beforehand, the LSTM best approximates the number of infections in the 87^{th} week under consideration. This whole scenario confirms that the LSTM, as a deep-learning method, is able to learn the long-term behavior of the given time series dataset and simulate outliers that also occur in the underlying dataset. Nevertheless, the LSTM prediction is not always closest to the trend of the reported data (as here between weeks 69-72, 81-84) and the NSFD with optimized trainable parameters proves to be well suited for predicting epidemiological trends. The error between the NSFD and the reported data is smaller than the error between the LSTM and the reported data in all weeks considered, except in weeks 75-80 and in the last week, as shown in Fig. 5.

With respect to the 100 test runs we performed, meaning 100 complete training iterations with outputted curves generated via the LSTM, NSFD and ODE solver, the Euclidean distance between the ODE solver prediction and reported data was $56,185$ if the assignments $\beta = 0.000000018$ and $\kappa = 0.001$ were used. It was $74,744$ for the NSFD. We could certainly reduce the error between the NSFD prediction and the real data by further parameter modifications. However, we maintained the same parameters for the NSFD scheme as we used for the standard ODE solver in order to keep comparability. The Euclidean distance between the LSTM prediction and the reported data fluctuated between 40,000 and 100,000 between the individual programs. In many cases, it was between 60,000 and 70,000 i.e. the performance of all three methods was comparable. Depending on the progress of the validation process, we later obtained smaller distances of 40,000-60,000.

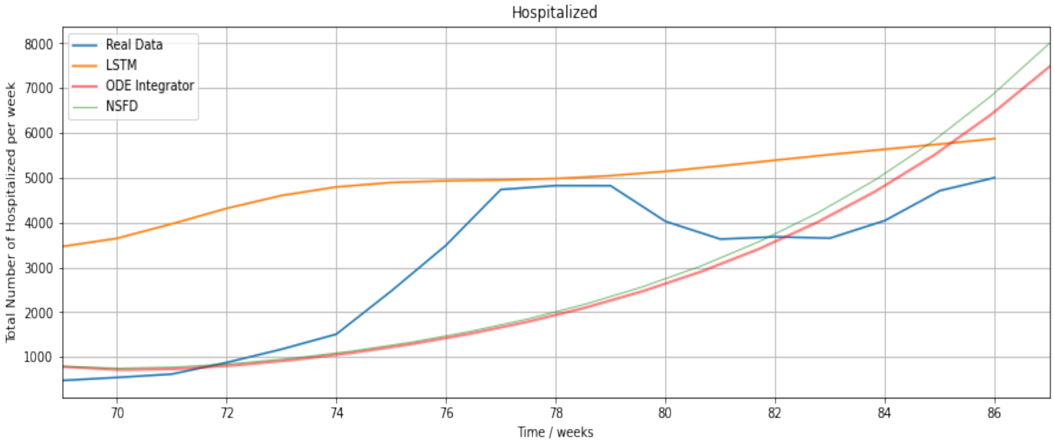

Fig. 6 The real curve of hospitalization numbers in Germany between the calendar weeks 10 in 2020 and 45 in 2021 (*blue*) and the corresponding LSTM prediction made for the calendar weeks 29 to 45 in 2021 (*orange*) based on the data of the calendar weeks 10 in 2020 to 28 in 2021.

As we can see in Fig. 6, the courses of the curves predicted by the two numerical methods are similar to the courses in Fig. 4. All three methods meet the increasing trend of the reported data, although the LSTM reflects the real slope between the 82^{nd} and 87^{th} week best, whereas the two numerical methods show a much stronger increase such that 7,500 or 8,000 hospitalizations are obtained in the 87^{th} week by using them.

Until the 82^{nd} week observed, where the NSFD and ODE integrator cut off the curve of real data, these two numerical methods make good approximations of the general real slope. Nonetheless, they do not show a peak, as is observable in the reported data between the 77^{th} and 79^{th} week concerned. A difference of almost 3,000 hospitalizations is observable between the two numerically generated curves and the reported data in the 77^{th} and 78^{th} week. This also the biggest error to be noted in this example, regarding the numerical approaches.

The LSTM shows a large deviation from the reported data in the beginning as in Fig. 4 (3,000 hospitalizations), but then approximated the real data. In the 77^{th} and 79^{th} week, only a difference of around 100 hospitalizations can be observed. The difference between the LSTM and reported data curves becomes larger again, since the LSTM curve slightly increases and the real data curve declines. As the reported data re-start increasing in the 83^{rd} week, the difference becomes smaller again and is 900 in the 87^{th} week.

We can analyze that the LSTM method is able to predict the increases or outliers of the reported data well and learns them from long-term data, but it tends to precede them and thus exhibits larger errors than the numerical methods at pre-peak times. In contrast to that, the NSFD scheme is a good choice if we are interested in the intermediate-term behaviour of the curve, and less in oscillations or outliers.

4.3 Infection and Hospitalization Number Scenarios

Finally, future scenarios were predicted by applying the NSFD method and the standard Python ODE solver to initial reported data

$$\mathcal{K}^0 = [22,570,182; 55,858,178; 197,090; 5,000; 95,729; 4,373,822]^\top$$

which corresponds to calendar week 45 in 2021. It should be noted that approximately two-thirds of the German population of 83.1 million had was vaccinated by this time. In addition, the LSTM method was used for predictions based on the entire 87-week data set, starting with calendar week 10 in 2020.

The PINN previously used for parameter estimation was also used to predict infection and hospitalization rates. This time, the vector $\mathcal{K}_0$ was used as a second input to the PINN to obtain more accurate output curves. Its implementation was based on the same calendar weeks as the LSTM method.

Fig. 7 shows the loss and its parts $MSE_{\mathcal{U}}$ and $MSE_{\mathcal{F}}$ of the modified PINN for predicting the infection number in $2,500$ training steps using $\alpha = 0.5$, while Fig. 8 shows the same for $\alpha \in \{0.7, 0.3, 0.9, 0.1\}$.

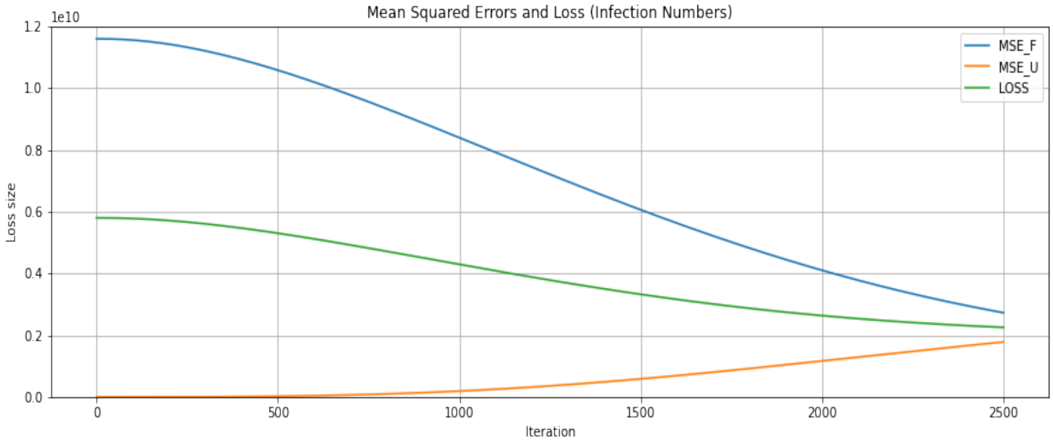

Fig. 7 Trends of the errors $MSE_{\mathcal{U}}$, $MSE_{\mathcal{F}}$ and the loss $\mathcal{L}_{0.5} = \frac{1}{2}MSE_{\mathcal{U}} + \frac{1}{2}MSE_{\mathcal{F}}$ in 2,500 observed training iterations of the PINN for the prediction of infection numbers.

Similarly to Fig. 2 can be observed that the function of the mean squared error $MSE_{\mathcal{U}}$ is monotonically decreasing, whereas the function of the mean squared error $MSE_{\mathcal{F}}$ is monotonically increasing in Fig. 7 and the four loss scenarios in Fig. 8. Except for the case $\alpha = 0.1$ in Fig. 8, where $MSE_{\mathcal{U}}$ reaches a value of $0.8 \cdot 10^{10}$ and then only declines insignificantly, the data loss still decreases after the 1700^{th} training step with $\alpha \in \{0.3, 0.5, 0.7, 0.9\}$. The loss part $MSE_{\mathcal{F}}$ approaches a value of $0.2 \cdot 10^{10}$ after 2,500 iterations for $\alpha \in \{0,3, 0.5, 0.7\}$. With $\alpha = 0.9$ or $\alpha = 0.1$ however, we obtain the much smaller final values of $MSE_{\mathcal{F}}$ of $0.2 \cdot 10^{10}$ or $0.01 \cdot 10^{10}$, respectively.

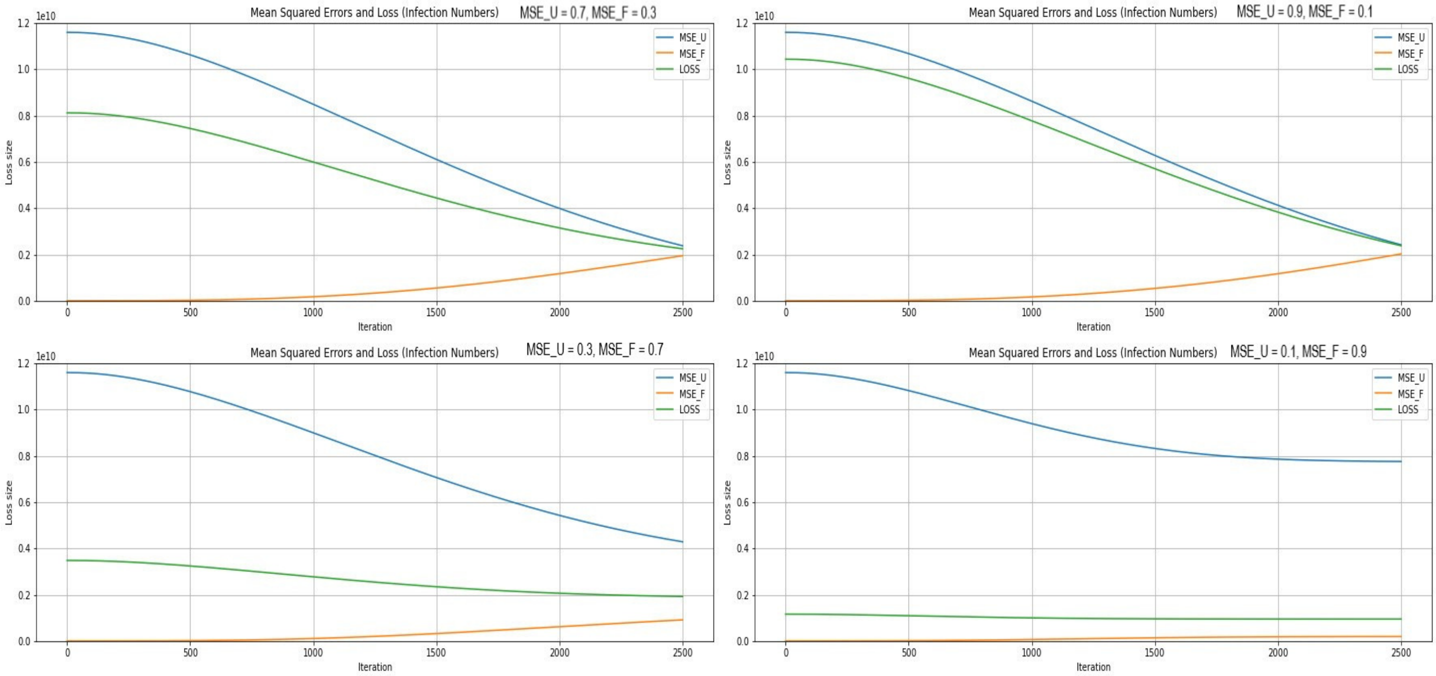

Fig. 8 Trends of the errors $MSE_{\mathcal{U}}$, $MSE_{\mathcal{F}}$ and the loss $\mathcal{L}_\alpha = \alpha\, MSE_{\mathcal{U}} + (1-\alpha)\, MSE_{\mathcal{F}}$ in 2,500 observed training iterations of the PINN for the prediction of infection numbers, for $\alpha = 0.7, 0.3, 0.9, 0.1$.

Moreover, the composition of the loss curve changes as α is modified. We note that the trajectories of the two mean square errors also change during the training process when α is modified. The loss curve is monotonically decreasing in all five cases (cf. Figures 7 and 8). It exhibits a value of $0.22 \cdot 10^{10}$ for $\alpha \in \{0.5, 0.7\}$, of $0.23 \cdot 10^{10}$ for $\alpha = 0.9$, of $0.2 \cdot 10^{10}$ for $\alpha = 0.3$, and of $0.1 \cdot 10^{10}$ for $\alpha = 0.1$ after 2,500 iterations. This implies that $\alpha < 0.5$ yields loss curves at smaller levels, but the data loss decreases less sharply with smaller α.

In the following, examples of predicted infection scenarios are shown. The scenarios predicted by the LSTM differ slightly, of course, because the neural network is exclusively data-driven. The predictions made with the NSFD or ODE solver give the same curve in all diagrams, since the same parameters were used in the corresponding program runs. Fig. 9 conveys incidence scenarios created by applying PINN, LSTM, NSFD, and standard ODE solvers using different values for α in PINN. Thus, the sensitivity of PINN predictions with respect to the α parameter can be investigated. In the prediction part, different assignments of the α parameter produce different PINN prediction scenarios. In Figs. 7 and 8 in the validation part, we noticed that the loss curve became lower and flatter the smaller we choose α, although the mean square error $MSE_{\mathcal{U}}$ remained at a higher level.

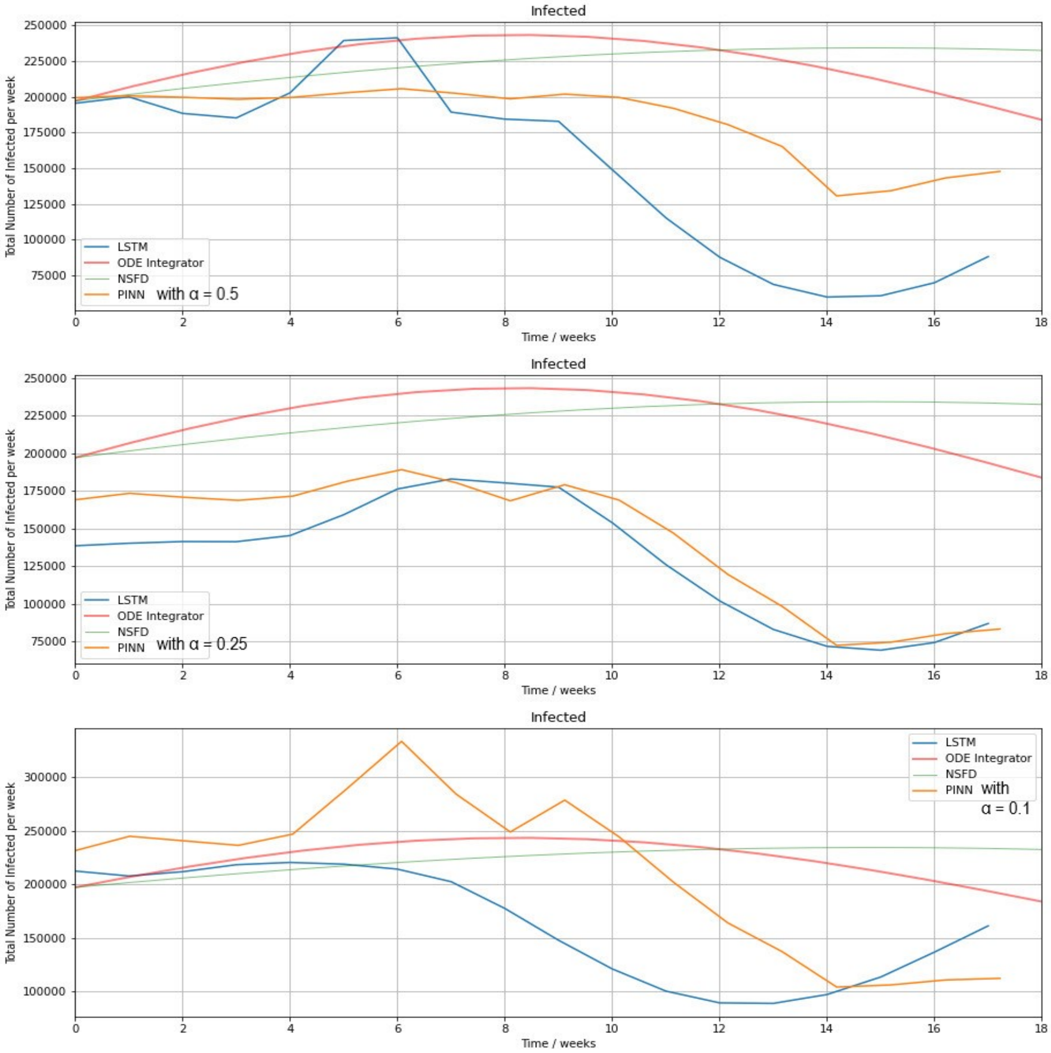

Fig. 9 The prediction of infection numbers over 18 weeks starting in calendar week 46 in 2021 on the basis of data covering the calendar weeks 10 in 2020 to 45 in 2021 using four different approaches (solver for ordinary differential equations, Nonstandard Finite Difference Scheme, Physics-informed Neural Network, Long-Short-Term-Memory) and with $\alpha \in \{0.5, 0.25, 0.1\}$.

In Fig. 9 we notice that smaller α lead to curves with sharper local extrema in the incidence prediction. The local maxima in the 52^{nd} calendar week of 2020 and the 3^{rd} calendar week of 2022 are 370,000 and 275,000 weekly infections, respectively, at $\alpha = 0.1$. The incidence level at $\alpha = 0.1$ in calendar weeks 46 in 2021 through 4 in 2022 is generally much higher than in the $\alpha = 0.25$ and $\alpha = 0.5$ cases. In the graph generated with $\alpha = 0.5$, local maxima are barely discernible and the curve remains at about 200,000 infections until calendar week 4. In the graph generated with $\alpha = 0.25$, the peaks are somewhat more evident, but the incidence level here hovers around 175,000 infections.

For all three assignments of the α parameter, the predicted infection count curve declines starting at calendar week 4 in 2022. In the $\alpha = 0.5$ case, a value of 150,000 infections is reached at calendar week 11 in 2022. In the cases $\alpha = 0.25$ or $\alpha = 0.1$, there are 80,000 and 110,000, respectively.

Both the NSFD and ODE solver forecast functions increase monotonically from 200,000 to 245,000 infections between calendar week 46 in 2021 and calendar week 2 in 2022, with a much weaker increase in the NSFD curve. From calendar week 2, the curve predicted by the ODE solver drops to less than 190,000 infections in calendar week 12, while the curve generated by the NSFD scheme still rises slightly until calendar week 6 and then remains at about 235,000 infections until calendar week 12. The NSFD and ODE solver predictions also cut off at 235,000 infections in calendar week 6 in 2022.

With the parameters $\beta = 0.000000035$ and $\kappa = 0.001$ used in the two numerical methods, we obtain higher incidence levels in the following 18 calendar weeks when we choose $\alpha = 0.5$ or $\alpha = 0.25$ in the PINN approach. We also obtain lower levels between calendar week 46 in 2021 and 4 in 2022, but higher levels in calendar weeks 5 to 12 in 2022 if we choose $\alpha = 0.1$.

Various of our LSTM-based predictions showed local maxima ranging from 180,000 infections in calendar week 1 to 225,000 infections in calendar week 50 to 235,000 infections in calendar week 52. The LSTM-predicted curve reached a minimum of less than 7,500 infections between calendar weeks 6 and 8 in our program runs.

The trends of the LSTM and PINN curves in Fig. 9 are increasing and not relevantly decreasing, respectively, in calendar weeks 46 through 52. Thereafter, both drop significantly, illustrating their similar behavior despite different predicted infection numbers or curve heights, which also depend on model parameters. The trend of the LSTM and PINN curves in calendar weeks 9 to 12 is upward. The trend of the NSFD prediction is then almost constant, while the curve generated by the standard ODE solver decreases significantly. The similarity between all four approaches is that there is a peak around the turn of the year and a significant decline thereafter.

Fig. 10 illustrates three hospitalization scenarios created by using the same LSTM, NSFD and standard ODE solver outputs, but different values for α in the PINN approach.

Instead of the results of different LSTM method program runs as for Fig. 9, the same LSTM output was used to create Fig. 10. Here, we can observe that the modification of the weight parameter α influences the scenario drawn by the PINN. The smallest assignment $\alpha = 0.1$ leads to the smallest amount of data loss in the loss function among the three compared scenarios. It yields the highest number of hospitalizations among the three, while the course of the three curves is similar with peaks at the same times (calendar weeks 5 and 8 in 2022). Whereas $\alpha = 0.1$ yields 30,000 hospitalizations in calendar week 5 and 26,500 in calendar week 12, the assignment $\alpha = 0.5$ results in 28,000

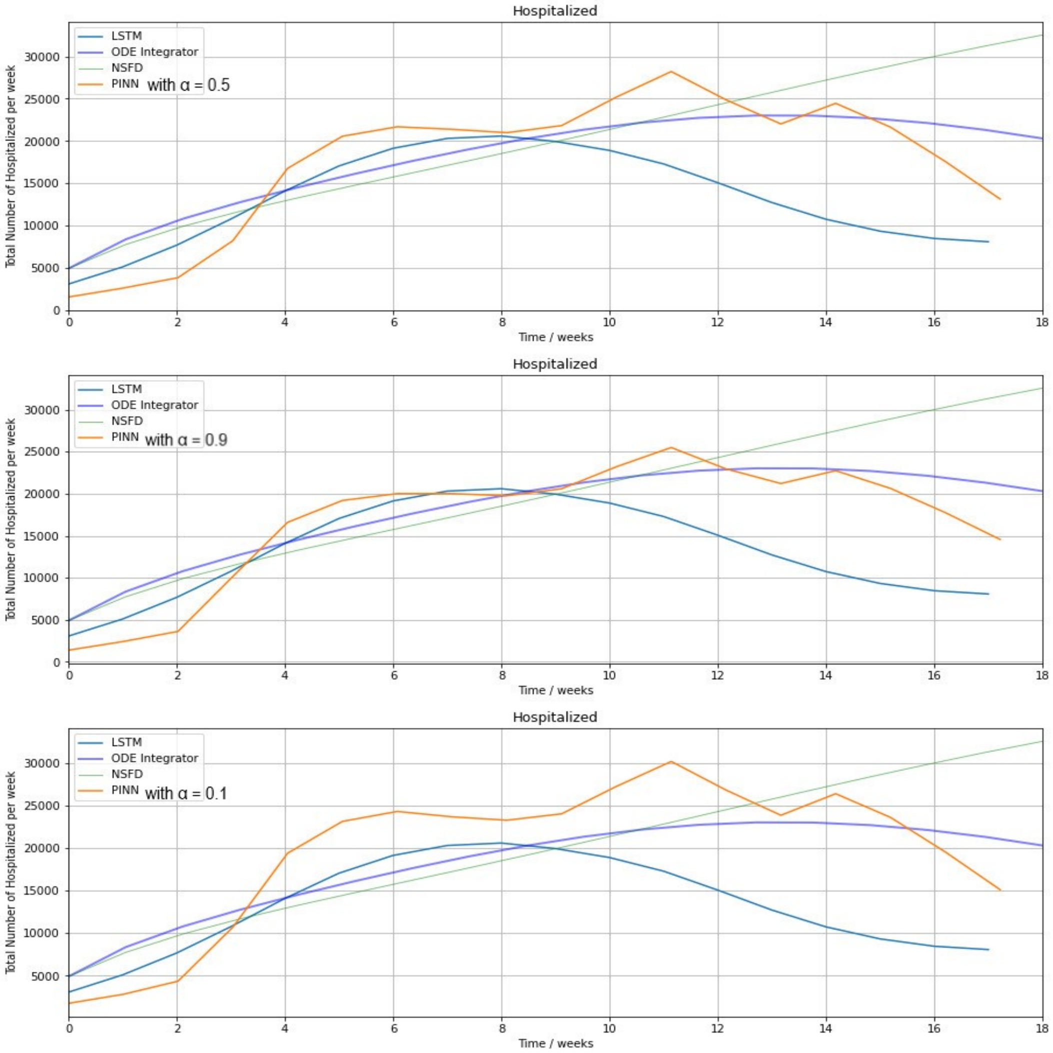

Fig. 10 The prediction of hospitalization numbers over 18 weeks starting in calendar week 46 in 2021 on the basis of data covering the calendar weeks 10 in 2020 to 45 in 2021 using four different approaches (solver for ordinary differential equations, Nonstandard Finite Difference Scheme, Physics-informed Neural Network, Long-Short-Term-Memory) and with $\alpha \in \{0.5, 0.9, 0.1\}$.

hospitalizations in calendar week 5 and 24,500 in calendar week 5. Moreover, setting $\alpha = 0.9$, i.e. increasing the amount of data loss but decreasing the amount of residual error in the loss function, yields 25,500 hospitalizations in calendar week 5 and 22,500 in calendar week 12.

5 Conclusion and Outlook

We have presented a data- and physics-driven deep learning algorithm that identifies the transmission parameter β and the parameter κ, representing the proportion of transmissibility in the vaccinated population, in a system of ordinary differential equations of a compartment model describing population dynamics at pandemic times. For this purpose, we made use of a physics-informed neural network (PINN), the loss function of which combines a weighted data loss with a weighted residual loss, which again is based on the compartment model applied to the respective current network output.

Using COVID-19 infection, hospitalization, vaccination and mortality data from Germany, we calculated the prediction error of an exclusively data-driven Long-Short-Term-Memory (LSTM), i.e., a special type of recurrent neural network that learns the behaviour of time series data, and the two numerical methods of a standard solver for ordinary differential equations [5] and a nonstandard finite difference (NSFD) scheme [16]. During a validation process, we slightly adjusted parameters in the NSFD scheme and the network architecture of the LSTM in order that we obtained resulting plots approximating the trend of the real course of infection numbers in the calendar weeks 29 to 45 in 2021.

The fact that we were able to improve the LSTM scheme in the way that the Euclidean distance between the LSTM predictions and the reported data became smaller when observing various test runs, implies the flexibility of the LSTM approach. We note that the LSTM method is well able to learn long-term dependencies and account for outliers. Also, we found that the prognoses made by the numerical approaches were smooth curves, conveyed correct tendencies and strongly depended on the underlying parameters as substantially the parameters influencing transmission in the regarded system. With all three methods, we were finally able to achieve curves the trends and slopes of which came close to the ones of the real data curves.

Applying the improved versions of the three methods, we were able to compare the predictions concerning the course of the fourth wave with the forecasts of a modified PINN. This PINN had the same structure as the PINN we used for parameter identification, but yielded more precise incidence prognoses as we gave initial compartment size data as additional inputs. Generally, we obtained a clearly decreasing data loss, proving that the PINN approach correctly approximated the reported data, and a slighter increasing residual error, such that we overall obtained a decreasing loss curve of the PINN. Our results imply that the strength of the decline of the loss curve depends on the choice of the weighting parameter of the loss function.

Apart from parameter choices, the predictions made by the PINN depended on the composition of the loss function used in both infection and hospitalization number forecasts. The PINN forecasts exhibited the same trend (upwards or downwards) as the LSTM prediction in all regarded program runs. The trend of the NSFD prognosis was closer to the one of the two data-driven approaches than the trend of the ODE solver forecast in all cases. Both numerical methods exhibited weaker rising or falling curve segments than the neural network approaches. We found strong decreases in infection numbers during the fourth wave after 6-10 weeks from prediction start when using the two data-driven methods. In contrast to that, the curve predicted by the NSFD declined in a more marginal way and the curve predicted using the ODE solver was on an almost constant level after 14 weeks. The prognoses of data-driven and numerical methods deviate since their fundamental frameworks differ. However, we can observe similarities as the approximate point in time when the global maximum of the wave is achieved, or the approximate peak height if parameters are selected deliberately. The NSFD approach is well-suited if we are interested in incidence trends in the medium term, since the predicted curves rise or fall consistently.

The next step will be to apply the validation and prediction procedure of our PINN approach to data from different countries, in particular to estimate the influence of the level of vigilance and intervention and vaccination programs that differ between countries. There is also large interest in model scenarios with differently transmissible variants of the novel coronavirus with our approaches.

Our next PINN implementations will include and identify transmission rates consisting of a time-dependent transmission coefficient $\beta(t)$, a time-dependent contact rate $\gamma(t)$, and a time-dependent mitigation or quarantine rate $q(t)$. We will use these rates to analyze the effects of different lockdown scenarios and contact restrictions.

References

1. F. Brauer, C. Castillo-Chávez (2001) *Basic ideas of mathematical epidemiology*, Mathematical Models in Population Biology and Epidemiology, Springer, 275–337.
2. R. Chandra, A. Jain, D.S. Chauhan (2021) *Deep learning via LSTM models for COVID-19 infection forecasting in India*, arXiv:2101.11881.
3. V.K.R. Chimmula, L. Zhang (2020) *Time series forecasting of COVID-19 transmission in Canada using LSTM networks*, Chaos Solitons Fractals 135, 109864.
4. COVID-19 Vaccination Dashboard (2021) `https://impfdashboard.de/daten`. Last access: July 14, 2021.
5. J.R. Dormand, P.J. Prince (1980) *A family of embedded Runge-Kutta formulae* Journal of Computational and Applied Mathematics, 6(1), 19-26.
6. M. Ehrhardt, R.E. Mickens *A Nonstandard Finite Difference Scheme for Solving a Zika Virus Model*, unpublished manuscript.
7. V. Grimm, A. Heinlein, A. Klawonn, M. Lanser, J. Weber (2020) *Estimating the time-dependent contact rate of SIR and SEIR models in mathematical epidemiology using physics-informed neural networks*, Technical Report CDS-2020-5, Center for Data and Simulation Science, University of Cologne.
8. Infektionsschutz.de: `https://www.infektionsschutz.de/coronavirus/fragen-und-antworten/alles-zu-den-impfstoffen/wirksamkeit-und-sicherheit.html#tab-4802-4`. Last access: October 1^{st} 2021.
9. W.O. Kermack, A.G. McKendrick (1927) *A contribution to the mathematical theory of epidemics*, Proc. Royal Soc. A, Volume 115, pp. 700-721.
10. E. Kharazmi, M. Cai, X. Zheng, Z. Zhang, G. Lin, G.E. Karniadakis, (2021). *Identifiability and predictability of integer-and fractional-order epidemiological models using physics-informed neural networks*, Nature Computational Science, pp.1-10.

11. S. Kumar , R. Sharma, T. Tsunoda et al. (2020) *Forecasting the spread of COVID-19 using LSTM network*, BMC Bioinformatics, 22, 316.
12. J. Long, A.Q.M. Khaliq, K.M. Furati (2021) *Identification and prediction of time-varying parameters of COVID-19 model: a data-driven deep learning approach*, Int. J. Comput. Math. 98, 1617–1632.
13. R. Ma, X. Zheng, P. Wang, H. Liu, C. Zhang (2021) *The prediction and analysis of COVID-19 epidemic trend by combining LSTM and Markov method*, Nature Scientific Reports, 11, 17421.
14. J. Malinzi, S. Gwebu, S. Motsa, (2021) *Determining COVID-19 Dynamics Using Physics Informed Neural Networks*, Preprints 2021, 2021110555.
15. M. Martcheva (2015) *An Introduction to Mathematical Epidemiology*, Springer, New York.
16. R. E. Mickens (2000) *Applications of Nonstandard Finite Difference Schemes*, World Scientific.
17. K.D. Olumoyin, A.Q.M. Khaliq, K.M. Furati, (2021) *Data-Driven Deep-Learning Algorithm for Asymptomatic COVID-19 Model with Varying Mitigation Measures and Transmission Rate*, Epidemiologia, 2(4), 471-489.
18. Ourworldindata.com (2021) `https://ourworldindata.org/covid-cases`.
19. M. Raissi, P. Perdikaris, G.E. Karniadakis (2019) *Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations*, J. Comput. Phys. 378, pp. 686-707.
20. Robert Koch-Institute (2021) `https://www.rki.de/DE/Content/InfAZ/N/Neuartiges_Coronavirus/Daten/Fallzahlen_Kum_Tab.html`. Last access: October 17, 2021.
21. Robert Koch-Institute (2021) `https://www.rki.de/DE/Content/InfAZ/N/Neuartiges_Coronavirus/Daten/Klinische_Aspekte.html`. Last access: October 30, 2021.
22. Robert Koch-Institute (2021) `https://www.rki.de/SharedDocs/FAQ/COVID-Impfen/FAQ_Liste_Wirksamkeit.html`. Last access: November 16, 2021.
23. Robert Koch-Institute (2021) `https://www.rki.de/DE/Content/InfAZ/N/Neuartiges_Coronavirus/Steckbrief.html`. Last access: November 23, 2021.
24. E. Schiassi, M. De Florio, A. D'ambrosio, D. Mortari, R. Furfaro, (2021) *Physics-informed neural networks and functional interpolation for data-driven parameters discovery of epidemiological compartmental models*, Mathematics, 9(17), p. 2069.
25. S. Shaier, M. Raissi (2021) *Disease Informed Neural Networks*, arXiv preprint arXiv:2110.05445.
26. Statista.de (2021) `https://de.statista.com/statistik/daten/studie/1203308/umfrage/impfstoffabdeckung-der-bevoelkerung-gegen-das-coronavirus-nach-laendern/`. Last access: November 16, 2021.
27. K. Tolksdorf, S. Buda, E. Schuler, L. H. Wieler, W. Haas (2020) *Eine höhere Letalität und lange Beatmungsdauer unterscheiden COVID-19 von schwer verlaufenden Atemwegsinfektionen in Grippewellen* 41, 3-10.
28. T.K. Torku, A.Q. Khaliq, K.M. Furati (2021) *Deep-Data-Driven Neural Networks for COVID-19 Vaccine Efficacy*, Epidemiologia, 2(4), 564-586.
29. S. Treibert, H. Brunner, M. Ehrhardt (2022) *A nonstandard finite difference scheme for the SVICDR model to predict COVID-19 dynamics*, Mathematical Biosciences and Engineering 19(2): 1213-1238.
30. A. Zeroual, F. Harrou, A. Dairi, Y. Sun (2020) *Deep learning methods for forecasting COVID-19 time-Series data: A Comparative study*, Chaos Solitons Fractals 140, 110–121.