AMATH 575 FINAL PROJECT

CLAYTON STRAUCH, GAURAV JOSHI, JACKIE LEMMOND, MAXIMUS KOLAVENNU, RICHARD CARINI

Abstract. There are many fields, such as fluid mechanics, climate science, ecology, and epidemiology, where data is copious, but unfortunately noisy. As scientific computing continues to gain advancements, so does our understanding of these fields. One of these advancements is SINDy (Sparse Identification of Nonlinear Dynamics), which utilizes sparsity-promoting techniques and machine learning, combined with nonlinear dynamics, to extract governing equations from noisy data [3]. In this paper, we will focus on the field of epidemiology to demonstrate the robustness of pySINDy (a Python package of SINDy) by applying it to case notification data for the following three diseases: measles, chickenpox, and rubella. This will uncover mechanistic equations to fit the dynamics of these diseases. We will use a second-order function library in pySINDy, as the discovered models have been known to incorporate mass action incidence and seasonally varying transmission rates. Furthermore, third-order function libraries tend to overfit noise, making second-order the more efficient library [2]. Additionally, we will add synthetic noise to the case notification data to further study the effects of noisy data on the results found with pySINDy.

1. Introduction

Traditionally, discovering the governing equations of dynamical systems has posed a substantial challenge—often due to limitations in computational resources, data scarcity, or difficulties in handling high-dimensional systems. However, the convergence of large-scale data availability and advancements in algorithmic methods has made it increasingly feasible to infer such equations directly from time-series observations. A key breakthrough in this space is the Sparse Identification of Nonlinear Dynamics (SINDy) framework, which formulates model discovery as a sparse regression problem. SINDy begins with a comprehensive library of candidate nonlinear terms and systematically prunes insignificant ones, identifying a minimal set of governing dynamics that best explain the observed behavior [2].

Epidemiological systems offer a particularly compelling testbed for data-driven model discovery. These systems are often modeled using compartmental frameworks (e.g., SIR models), and prior studies have demonstrated that SINDy can successfully rediscover known epidemic dynamics from simulated data—even in the presence of noise [2]. However, empirical epidemiological data presents a greater challenge due to its inherent noisiness, irregular sampling, and reporting inconsistencies. Despite these issues, pySINDy has shown promise in reconstructing models with dynamics similar to those found in traditional compartmental approaches.

In this study, we apply pySINDy to both simulated and real-world epidemiological datasets for three diseases: measles, rubella, and chickenpox. We begin by evaluating pySINDy's ability to rediscover an idealized discrete-time SIR model from simulated data with synthetic noise. We then apply the same framework to case notification data from historical outbreaks in Canada and the UK. Finally, as a novel extension, we analyze more recent chickenpox incidence data from Hungary to assess pySINDy's generalization capabilities. Our methodology emphasizes reconstruction of the susceptible and infectious classes from raw incidence data and compares models built from different polynomial libraries to highlight the trade-off between overfitting and underfitting in noisy environments.

Date: June 12, 2025.

2. Reproduction

We will begin our reproduction with a brief explanation how the SINDy algorithm works and the functional library we are choosing to use (2.1 and 2.2), followed by running the model on simulated SIR data (2.3), and finally running SINDy on historical disease data (2.4) for three common childhood virus-caused illnesses: chickenpox, rubella, and measles.

2.1. **SINDy Algorithm:**

Consider an ODE[1] of the following form:

$$\dot{x} = f(x(t), t)$$

Where $x(t) = (x_1(t), (x_2(t), (x_3(t)...(x_n(t))$ represents the n-dimensional state variables with respect to time, and $f = (f_1, f_2, f_3, ...f_n)$ are the sparse functions that govern the dynamical system. Let $t_1, t_2, t_3, ...t_n$ be the time series of the sampled data points for both x and $\dot{x}$. The time series data of state variables and the response are represented by the following matrices:

$$x = \begin{bmatrix} x_1(t_1) & x_2(t_1) & \cdots & x_n(t_1) \\ x_1(t_2) & x_2(t_2) & \cdots & x_n(t_2) \\ \vdots & \vdots & \ddots & \vdots \\ x_1(t_m) & x_2(t_m) & \cdots & x_n(t_m) \end{bmatrix} \qquad \dot{x} = \begin{bmatrix} \dot{x_1}(t_1) & \dot{x_2}(t_1) & \cdots & \dot{x_n}(t_1) \\ \dot{x_1}(t_2) & \dot{x_2}(t_2) & \cdots & \dot{x_n}(t_2) \\ \vdots & \vdots & \ddots & \vdots \\ \dot{x_1}(t_m) & \dot{x_2}(t_m) & \cdots & \dot{x_n}(t_m) \end{bmatrix}$$

Then, we construct a library of linear and nonlinear terms as candidate functions for the model:

$$\Theta(x) = \begin{bmatrix} 1 & x & x^{P_2} & x^{P_3} & \cdots \end{bmatrix}$$

and P_n are all possible polynomials of degree n. Note that any nonlinear functions (trigonometric, exponential, rational, etc.) can be used. However, polynomial and trigonometric functions are often the most commonly used for state variables. Moreover, assuming that $\dot{x}$ can be described by relatively few of the nonlinear terms active in $\Theta(x)$, we can set up the sparse regression

$$\dot{x} = \Theta(x)\Xi$$

where $\Xi = (\xi_1, \xi_2, \cdots, \xi_p)$ is the set of coefficient vectors. Then, we apply the following iterative method for sparse regression:

(1) Perform a least-squares regression on $\dot{x} = \Theta(x)\Xi$
(2) Set all terms in Ξ that are less (in absolute value) than some threshold λ to zero.
(3) Create new library Θ', dropping functions that correspond to the zero entries from Ξ.
(4) Repeat the previous steps (a-c) until there are a max number of iterations that are reached, or until an error threshold is attained between the model data and the training data.

This gives us a set of sparse vectors that provides an approximate solution to $\dot{x} = \Theta(x)\Xi$. Now, we can reconstruct the k^{th} row of the dynamical system by taking

$$\dot{x}_k = \Theta(x_k^T)\xi_k$$

where $\Theta(x_k^T)\xi_k$ is the symbolic representation of the element x. Finally, combining all the rows of the discovered dynamical systems results in the following system of equations.

$$\dot{x} = \Xi^T \Theta(x^T)^T$$

This algorithm came from Ref.[2] and originated from [3]. Pseudo-code for this algorithm can be found in the appendix under Algorithm (1).

[1]Note that we can also consider a discrete timestep mapping, as is done throughout the rest of this paper. This changes nothing of the overall technique besides replacing a potentially numerically reconstructed $\dot{x}$ with the exact value of x_{t+1}

2.2. **Epidemiological library:** The following are the second and third order libraries used for both model rediscovery and model discovery from empirical data:

$$\Theta(x) = \begin{bmatrix} 1 & S & I & S^2 & I^2 & SI & \beta & \beta S & \beta I & \beta S^2 \\ \beta I^2 & \beta SI & & & & & & & & \end{bmatrix}$$

$$\Theta(x) = \begin{bmatrix} 1 & S & I & S^2 & I^2 & SI & S^3 & S^2 I & SI^2 & I^3 \\ \beta & \beta S & \beta I & \beta S^2 & \beta I^2 & \beta SI & \beta S^3 & \beta SI^2 & \beta SI^2 & \beta I^3 \end{bmatrix}$$

using the same variables defined in our SIR model.

2.3. **Using Simulated Data.** First, we will see how pySINDy rediscovers the SIR model. We used a discrete SIR model due to the fact that time dynamics for an ODE require differentiation techniques, which may be extremely noisy in an already noisy system. A discrete time model, where the next iteration is already part of the training data, allows us to circumvent this challenge, meaning pySIDNy is able to extract dynamics from empirical data more accurately with the discrete time model [2]. Therefore, it will benefit us for the sake of comparison to also use a discrete model for our rediscovery. This model is as follows:

$$\begin{aligned} S_{t+1} &= S_t + \nu - \beta(t) S_t I_t - \mu S_t \\ I_{t+1} &= I_t + \beta(t) S_t I_t - \gamma I_t - \mu I_t \\ R_{t+1} &= R_t - \gamma I_t - \mu R_t \end{aligned} \tag{1}$$

where S_t is the susceptible, I_t is the infected, R_t is the removed/recovered, and $\beta(t) = \beta_0(1 + \beta_1 \cos(2\pi t/T - \phi))$ is the time-varying rate of transmission, with period $T = 1$ year. In addition, t is our timestep (in weeks), ν (μ) are birth (death) rates, and γ is the recovery rate [2]. Note that S_t and I_t are not dependent on R_t, making the equation redundant and will not be considered for our simulations. We solved the SIR model numerically and generated a simulated data set by adding variable amounts Gaussian noise to the output state variables. We then apply pySINDy to this dataset with a second-order library of S_t, I_t, as done in [2].

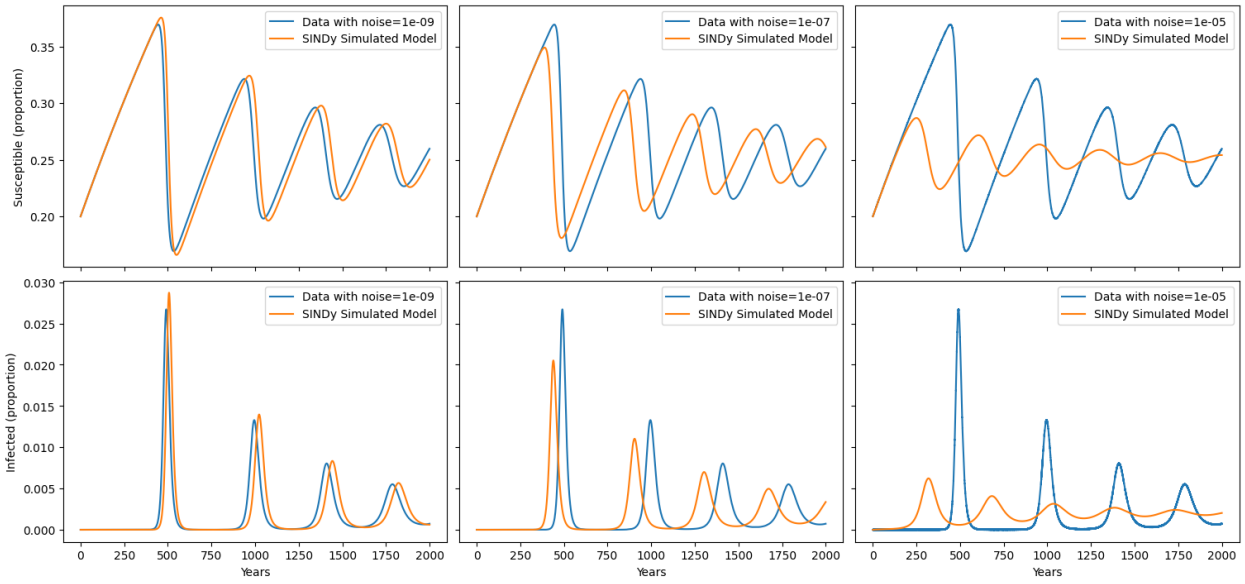


FIGURE 1. Rediscovery of SIR with simulated data using the S and I formulas in 1, with $S_0 = 0.2$, $I_0 = 10^{-10}$, $\beta_0 = 0.8$ week^{-1}, $\beta_1 = 0.025$ week^{-1}, $\gamma = 0.2$ week^{-1} $\phi = 0$, and $\mu = \nu = 5.4795 * 10e - 5$.

As we can see in Fig 1, pySINDy does a very efficient job of rediscovering the SIR model with added noise, but with even noise on the order of 10^{-6}, the noisiness affects the overall dynamics of the system. This was also identified in [2] and emphasizes the sensitivity to noise that even the idealized compartmental SIR model contains. This presents a unique challenge when considering real-world epidemiological data, where data can be extremely susceptible to error and noisiness. To combat this, we utilize various statistical techniques developed by the literature, demonstrated in the "Reconstruction" sections of the Appendix 5.

2.4. **Using Empirical Disease Data.** Next, we evaluate pySINDy's ability to discover models directly from empirical data. Specifically, we analyze case notification records for three infectious diseases: chickenpox, rubella, and measles. These diseases were selected based on the distinct periodicities in their transmission dynamics, $\beta(t)$: chickenpox exhibits an annual cycle, rubella a multi-annual cycle, and measles a biannual cycle [2]. Data sources are detailed in Appendix 5.1.

To prepare the datasets for pySINDy, we applied a standardized preprocessing pipeline across all three diseases, including the Hungarian dataset used in our novel extension. The following steps were taken:

(1) **Data smoothing and completion.** Missing weekly entries in the population and birth rate time series were filled using linear regression.
(2) **Susceptible class reconstruction.** The susceptible population was estimated using both global and local reconstruction methods as outlined in 2.4.1.
(3) **Incidence-to-prevalence conversion.** Because pySINDy requires time series of state variables, we converted incidence data to prevalence using the method described in 2.4.2.
(4) **Transmission rate estimation.** Finally, we reconstructed the time-varying transmission rate $\beta(t)$ following the formulation provided in 2.4.4, and visualized the reconstructed transmission and state variables across one-year cycles.

2.4.1. ***Susceptible reconstruction.*** Due to a lack of historical data on the susceptible population, we need to reconstruct the susceptible class with the following:

$$S_{t+1} = S_t - \alpha C_{t,t+1} + B_{t,t+1}$$

where S_t represents the number of susceptible population at the start of each week t, $C_{t,t+1}$ are the number of new cases in week t, $B_{t,t+1}$ are the number of births in week t, and α is the rate at which cases are reported. This equation shows that the susceptible population shrinks with each infected case reported and grows with each birth. As a result of the inconsistent reporting rates, we adapt the above equation as follows:

$$S_{t+1} = S_t - \alpha C_{t,t+1} + B_{d-t,d-t+1} + u_t \tag{2}$$

where u is additive noise such that $E(u) = 0$ and $V(u) = \sigma_u^2$, and d is the delay to account for the difference between the actual birth time and actually being susceptible to the disease. Now, let's define Z_t to be the deviation from the mean $E(S) = \bar{S}$ at week t.

$$S_t = \bar{S} + Z_t \tag{3}$$

and by taking Eq 3 and substituting it into Eq 2 yields:

$$Z_{t+1} = Z_t - \alpha_t C_{t,t+1} + B_{d-t,d-t+1} + u$$

We can express this equation by iterating it forward:

$$Z_t = Z_0 - \sum_{i=1}^{t} \alpha_i C_{i,i+1} + \sum_{i=1}^{t} B_{d-i,d-i+1} + \sum_{i=1}^{t} u_i \tag{4}$$

To simplify, consider the following redefined variables:

$$Y_t = \sum_{i=1}^{t} B_{i-d,i-d+1}, \qquad R_t = \sum_{i=1}^{t} (\alpha_i - \overline{\alpha}) C_i, \qquad U_t = \sum_{i=1}^{t} u_i, \qquad X_t = \sum_{i=1}^{t} C_{i,i+1}$$

Using these, we have:

$$Z_t = Z_0 + \bar{\alpha} X_t + Y_t + R_t + U_t \tag{5}$$

Assuming that the reporting rate is constant and noise is negligible, we can further reduce this to the following linear relationship:

$$Y_t = \bar{\alpha} X_t + (Z_t - Z_0) \tag{6}$$

Therefore, applying linear regression to Y_t against X_t provides an estimate of $Z_t - Z_0$ and $\bar{\alpha}$.

Eq 6 is known as the global regression method and is still flawed since it does not account for local shifts in mean since α is considered invariant. Hence, this can be corrected by introducing a variation in the reporting rate α (assuming this is the dominant fluctuation in the mean):

$$Y_t = R_t - U_t Z_0 - (\alpha_{t+1} - \bar{\alpha}) X_t + \alpha_{t+1} X_{t+1} + Z_{t+1} - u_{t+1} \tag{7}$$

where local linear regression techniques can be applied[2]. This method of reconstruction was found in Ref. [2].

2.4.2. ***Prevalence conversion.*** The prevalence of the infection is defined by the number (or proportion) of infectious individuals at any given time [2]. Infection prevalence is typically predicted in compartmental epidemic models. However, this presents a problem for pySINDy since the data collected is incidence data. Let C_t be the incidence data, D_i be the duration of infection, L be the mean individual life span, and p be the proportion of people that will contract the infection in their lifetime. Therefore, the infection prevalence, P_t, needs to be calculated by the following:

$$\langle P_t \rangle = \frac{pD_i}{L}$$

Given the relation:

$$\frac{P_t}{\langle P_t \rangle} = \frac{C_t}{\langle C_t \rangle}$$

Then it follows:

$$P_t = \frac{C_t p D_i}{\langle C_t \rangle L}$$

where the values of $D_i = 2$ weeks, $\forall i$; $L = 65$ years, and $p = 0.95$ were chosen for all calculations, following the selection of Ref. [2].

[2]The particular local linear regression technique that was applied in [2] was that of the Gaussian (radial basis function) kernel. It is worth noting that this kernel also requires careful techniques for selection, but our method was to experimentally try different values until our reproduced data resembled that of the original paper

2.4.3. ***Hyperparameter tuning for*** λ. Since λ is a degree of freedom that we can choose, with lower values of λ encouraging sparsity in our model and higher values decreasing error rates, we may perform a grid search over various values of λ in order to find a best fit. Typically this is done by constructing and analyzing a Pareto curve for various models trained with a spread of thresholds λ. As Ref. [2] points out, the Alkaike Information Criterion (AIC) of the model may serve as a strong indicator of a model's usefulness, especially since epidemiological models may wish to place more emphasis on models that correctly identify the frequency of epidemics, without necessarily caring much about the discrepancies between the mean square error (more on this in "Power Spectral Density" below).

As an example of this hyperparameter search, we take the model that was used to generate Fig 1. With the plots depicted in Fig , we can see that the MSE computed for various threshold values λ were not very helpful in narrowing down a choice of the parameter, but including the AIC for these values shows that there is indeed a sweet spot of a low enough threshold that encourages parsimony while still achieving a low (better) AIC score.

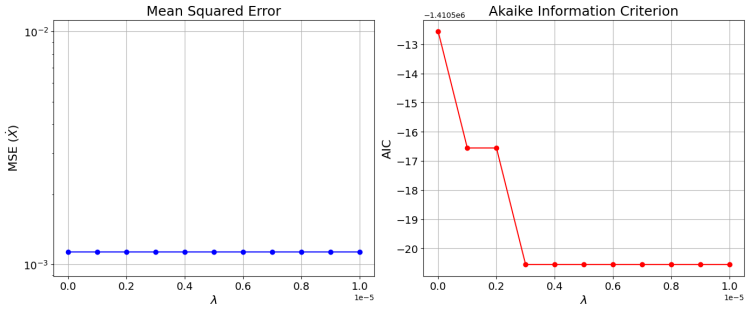


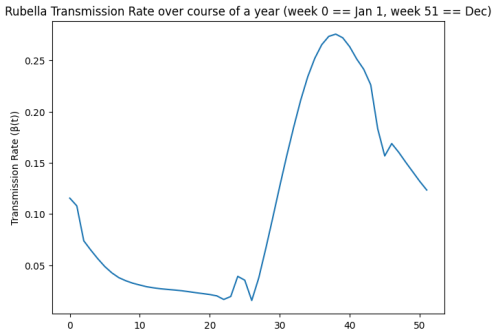
FIGURE 2. Parameter tuning for λ for the simulated SIR model shown in Fig 1 - we can see that changing the value of λ does not appear to impact the MSE in an observable way, but the AIC value suggests a selection of $\lambda = 3.0007e - 06$.

2.4.4. ***Transmission rate reconstruction***. Based off the discrete model, we need recover $\beta(t)$ in a time series model as well. We used the following:

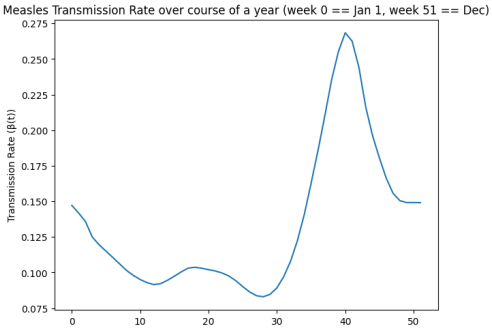
$$\beta(t) = \frac{Z_{k+1}}{S_k I_k \Delta t}$$

where S_k, I_k are the susceptible class and the infectious class, respectively, and Z_k is the incidence reported with time interval $(t - \Delta t, t)$ [1].

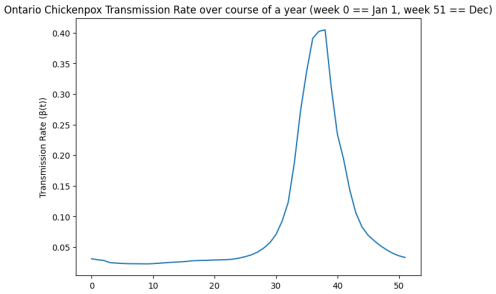
Using this method, we are able to produce the average seasonal transmission rates for each of the diseases, as seen in Fig 4. Interestingly, a sinusoidal model that appears to peak at around the start of the school season does seem to be a reasonable approximation for the seasonal periodic transmission rate.



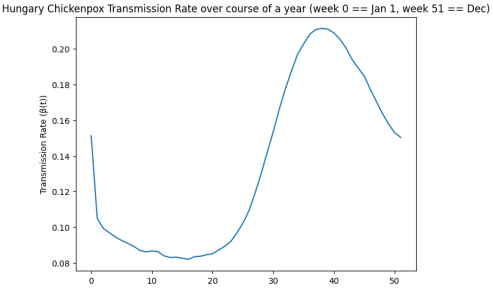
(A) Rubella Average Seasonal Transmission Rate



(B) Measles Average Seasonal Transmission Rate



(A) Chickenpox (Ontario) Average Seasonal Transmission Rate



(B) Chickenpox (Hungary) Average Seasonal Transmission Rate

FIGURE 4. Year-long average reconstructed transmission rate for all three types of diseases considered in this project. All appear sinusoidal with a peak at around the start of the school season, though some diseases (such as Rubella) are noticeably noisier.

2.4.5. ***Reproduction Results***. We will begin the analysis of results found for each disease with Chicken pox (Ontario). We used case notification data from Ontario, Canada, from 1946-1967 (Appendix 5.1). Our learned model was of the form:

$$S_{t+1} = -0.721 + 4.47S + -0.68S^2 + 1.13\beta + -2.28\beta S + 0.06\beta I + -0.06\beta SI + 1.15\beta S^2$$
$$I_{t+1} = -2.011 + 4.04S + -2.01S^2 + -2.45\beta + 4.99\beta S + -2.54\beta S^2$$

where $\lambda = 0.03$ was the most successful thresholding parameter we discovered. This resulted in the simulated vs actual time series plot that can be seen in Fig 5.

It is interesting to compare this to the coefficients learned in the original paper Ref. [2], where the best fit chickenpox model appeared to have strong dependence on cross terms $\beta(t)SI$, similar to that of the so-called "mixing term" in the original SIR model. It appears that the Susceptible class may have been overfit in our model; we see this as a trend in some of the other simulated plots where we performed SINDy without using the PySINDy library. In spite of this, we performed a Power Spectral Decomposition of both the reconstructed real case data and compared that to the SINDy resimulated data for both the Susceptible and Infected classes to see if the frequencies of the learned model resemble that of the known known annual periodicity of the chickenpox disease. Details of why and how we perform this analysis can be found in the Appendix under 5.2. The results of this analysis can be seen in Fig 6.

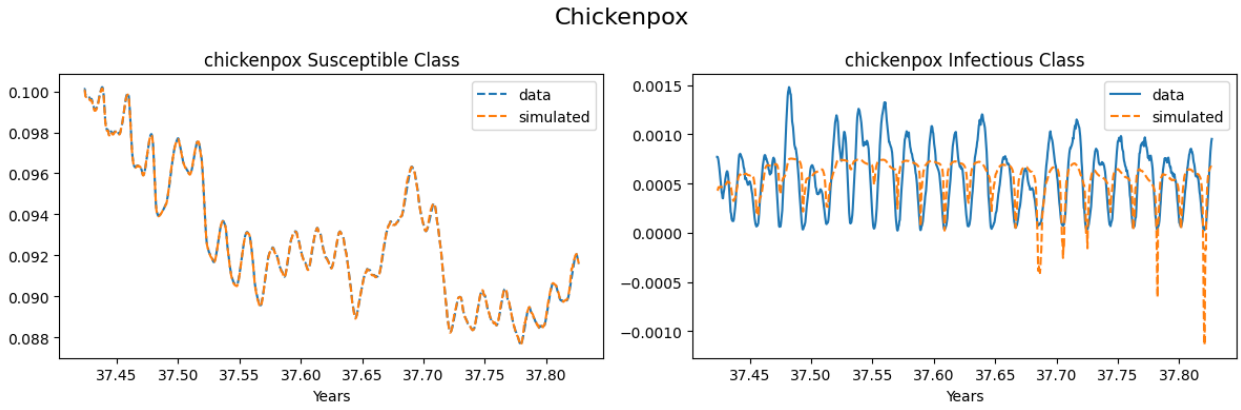


FIGURE 5. Comparison of the actual Ontario chicken pox dynamics to learned SINDy simulated dynamics

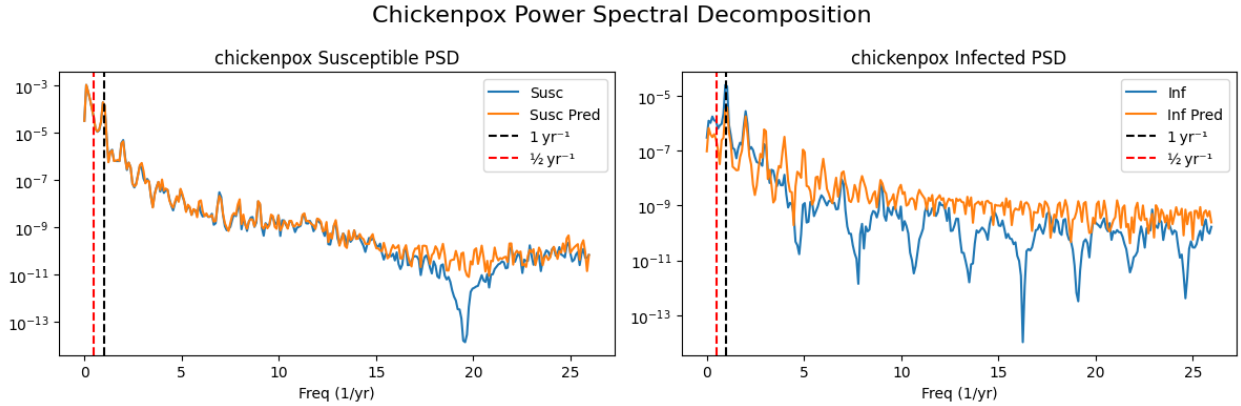


FIGURE 6. Comparison of the reconstructed vs simulated Power Spectral Density for both of the Ontario Chickenpox Susceptible (Susc) and Infected (Inf) classes of the population

From Fig 6, we do see a fairly well-defined spike around the 1 yr Infected rate (indicated by the dotted black line), which is what we would likely expect from a chickenpox annual transmission. As discussed in Ref. [2], in an epidemiological setting, it is often very important for a model to correctly identify the expected frequency of outbreaks, rather than necessarily correctly identifying the exact size of the outbreak. Therefore, our learned model is one that still may have some utility in spite of the likely overfitting to the noisy ground truth data.

Turning our attention now to Rubella, we achieved a model of the following form:

$$S_{t+1} = -0.131 + 3.14S + -0.14S^2 + -0.41\beta + 0.88\beta S + -0.47\beta S^2$$
$$I_{t+1} = 0.10S + -0.08S^2$$

We see in our model for Rubella, that the returned system approximation is quite sparse in the infected population's governing equation which was found to be true in the original paper, but does fall short in the terms that should be present as there are no interacting terms, which is reflected in Figure 7. The main challenge here can be partly explained by the significant difference in the magnitudes of the coefficients that influence the susceptible and infectious populations. We

note that the original authors also struggled the most with Rubella of the three diseases without a thorough power spectral density.

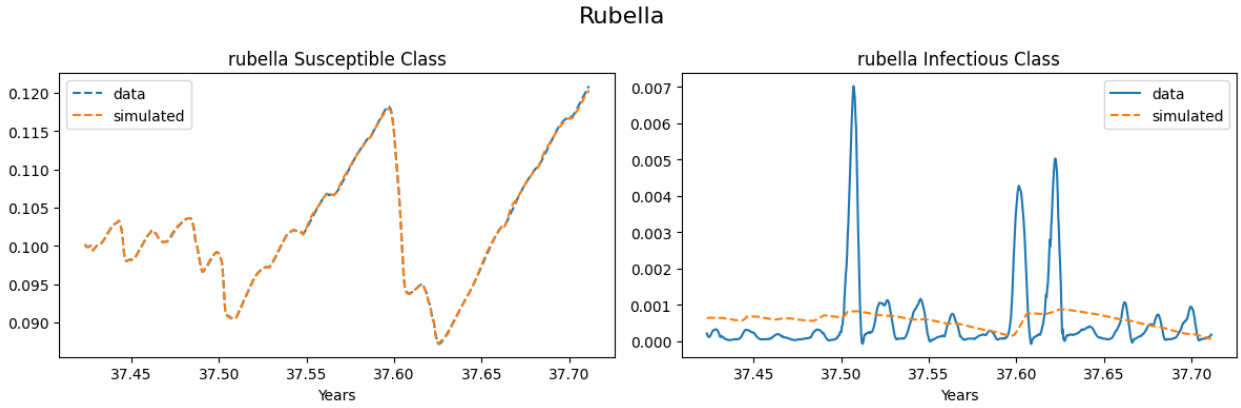


FIGURE 7. Comparison of the actual Rubella dynamics to learned SINDy simulated dynamics

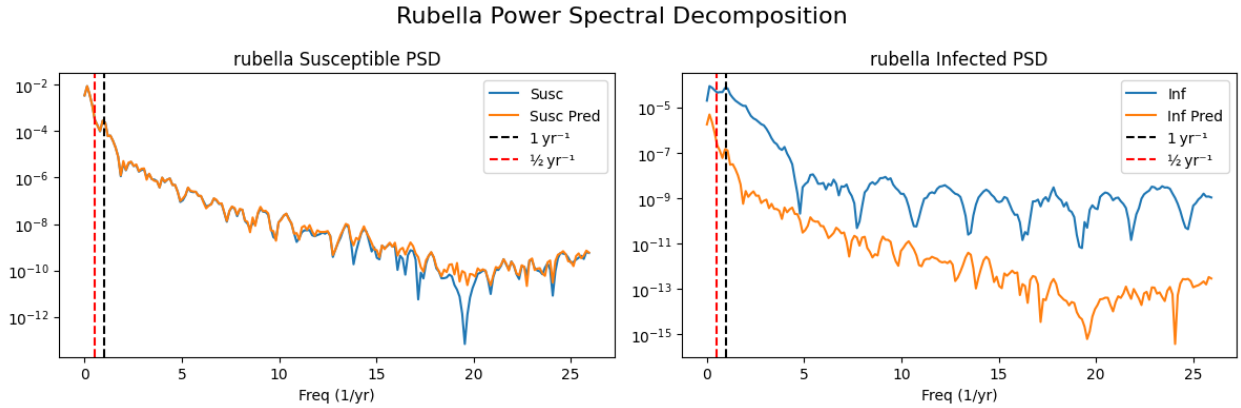


FIGURE 8. Comparison of the reconstructed vs simulated Power Spectral Density for both of the Rubella Susceptible (Susc) and Infected (Inf) classes of the population

Finally, we end the reproduction of the reference paper Ref [2] results in an analysis of Measles data. The learned SINDy model we achieved was:

$$S_{t+1} = 0.341 + 2.68S + 0.21S^2 + -1.61\beta + 3.21\beta S + -0.12\beta I + 0.12\beta SI + -1.60\beta S^2$$
$$I_{t+1} = -0.061 + 0.09S + 1.82\beta + -3.63\beta S + 1.79\beta S^2$$

While less sparse than the other learned models for the diseases above, we did successfully pick up a dependence in the Infectious class on the $\beta(t)$ variable rate of transmission parameter.

Following from Figure 9, we see that our model does pick up on the biennial peaks in the infectious class and similar to our other disease models, we see that the susceptible class S_t is well-modeled. While the scale of the infection outbreak is not picked up on, we do see that our model retains the peaks close to the same points in time that the peaks occur in the data. Like the chickenpox modeling results, we believe determining the frequency of an outbreak may prove a much more desirable feature compared to the size of the outbreak as it allows for efficient and timely preparation.

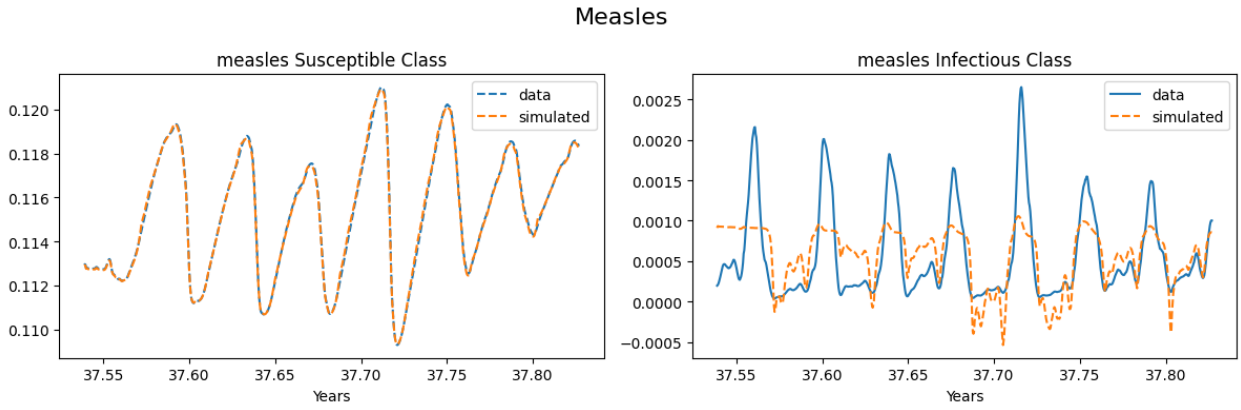


FIGURE 9. Comparison of the actual Measles dynamics to learned SINDy simulated dynamics

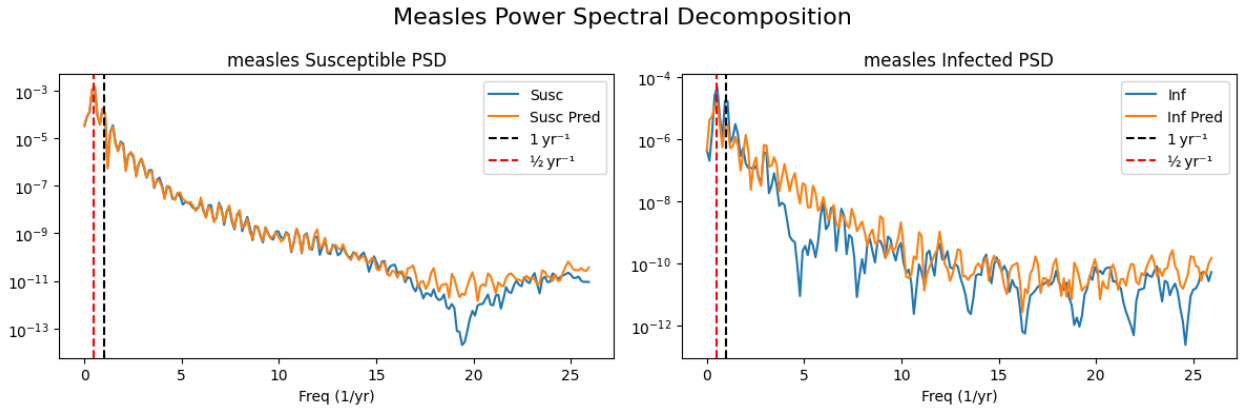


FIGURE 10. Comparison of the reconstructed vs simulated Power Spectral Density for both of the Measles Susceptible (Susc) and Infected (Inf) classes of the population

3. NOVEL RESULTS

As an extension to the results found in Ref. [2], we applied pySINDy to the case notification data from 2005-2014 in Hungary (data sources can be found in Appendix 5.1). Note that the chicken pox case notification data here is more recent, eliminating some of the challenges presented in using data from the pre-digital age seen above. We utilized the same data analytical methods developed and utilized for the diseases analyzed within Ref. [2] in order to reconstruct Susceptible, Infected, and estimated transmission rates. Using the same steps described 2.4, and then applying pySINDy using a third-order polynomial library, we developed the results show in Figure 11.

Although PySINDy accurately captures the dynamics of the susceptible class, closely matching the seasonal oscillations and long-term trends observed in the data, it fails to reproduce the infectious class dynamics with similar fidelity. Instead of the expected periodic behavior, the identified infectious model exhibits a smooth parabolic trajectory, indicating a significant mismatch between the true oscillatory dynamics and the model's representation. Here, pySINDy assigns a significant weight to the term $S_t I_t^2$ and I^3 in the infectious class, suggesting a possible overfit of the data [2]. However, when a second-order library was used, pySINDy was unable to capture the susceptible dynamics with as high of accuracy as the third-order library was able to achieve. Therefore, in this case, there may be another underlying cause for pySINDy's poor fitting of the infectious class.

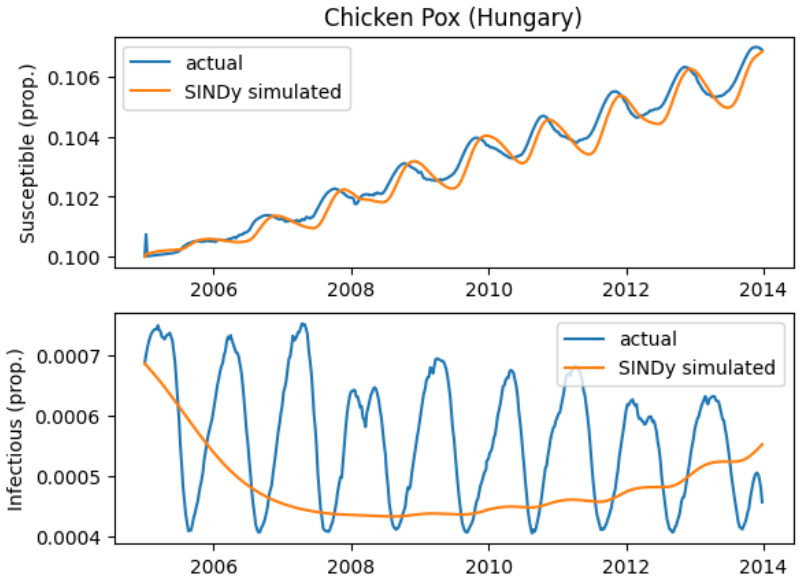


FIGURE 11. Comparison of the actual chicken pox dynamics to learned SINDy simulated dynamics

When performing an analysis on the Power Spectral Decomposition of the Hungarian chickenpox data (Fig 12), we see there there is a similar spike at the annual mark - unfortunately there also appears to be a nonnegligible spike at around $1/4$ yr^{-1} for the predicted infectious value. The most likely explanation for this is again noise, but it is interesting that we were able to achieve some analysis of the frequency distribution of the predicted infectious class in spite of the predicted values being far in MSE from the true data.

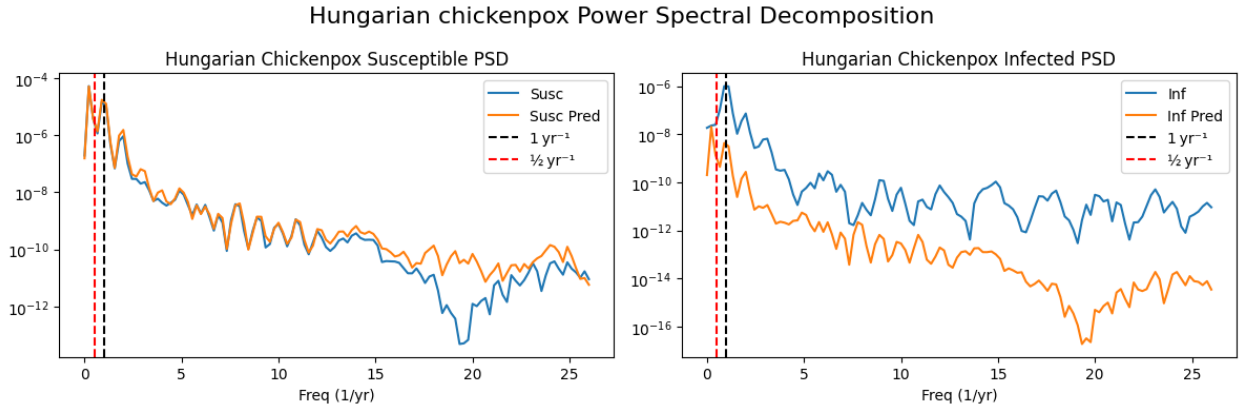


FIGURE 12. Comparison of the reconstructed vs simulated Power Spectral Density for both of the Hungarian Chickenpox Susceptible (Susc) and Infected (Inf) classes of the population

4. CONCLUSION

In this study, we have demonstrated that while pySINDy performs well in rediscovering governing dynamics from simulated data with synthetic noise, its effectiveness diminishes when applied directly to empirical epidemiological data. In particular, the model captures the dynamics of the

susceptible class with reasonable accuracy—as observed in the Hungarian chickenpox case—but struggles to reproduce the infectious class dynamics with comparable fidelity.

We attribute this difficulty, in part, to the disparity in the magnitudes of the coefficients governing the susceptible and infectious classes [2]. This imbalance complicates the selection of an optimal sparsity threshold λ. Although we employed hyperparameter tuning techniques (Appendix 2.4.3) to select λ, the resulting models were still suboptimal, suggesting that even carefully tuned thresholds may not sufficiently mitigate the problem.

Recent extensions to the pySINDy framework aim to overcome the need for manual λ selection and may offer more robust alternatives for future work. Incorporating these approaches—or integrating additional noise-handling techniques and latent-variable modeling—presents a promising direction for improving data-driven model discovery in noisy, real-world settings.

5. Appendix

Algorithm 1 SINDy Algorithm

Require: $\lambda > 0$, max_iter $< \infty$, error_threshold > 0
  $\Xi_0 \leftarrow$ least-squares regression on $\dot{x} = \Theta(x)\Xi$
  error $\leftarrow ||\Theta(x)\Xi_0 - \Theta(x)\Xi||_2$
  iter $\leftarrow 1$
  while error $>$ error_threshold && iter $<$ max_iter **do**
    zeroed_coeff $\leftarrow \{\}$
    for coefficient $\xi_j \in \Xi_i$ **do**
      if $\xi_j < \lambda$ **then**
        $\xi \leftarrow 0$
        zeroed_coeff $\leftarrow$ zeroed_coeff $\cup \{j\}$
      end if
    end for
    $\Theta' \leftarrow \Theta$
    for coefficient $j \in$ zeroed_coeff **do**
      $\Theta'[:, j] \leftarrow \mathbf{0}$
    end for
    error $\leftarrow ||\Theta(x)\Xi_i - \Theta(x)\Xi||_2$
    iter $\leftarrow$ iter $+ 1$
  end while

5.1. **Data Sources.** The case notification data (number of cases reported each week) came from the International Disease Data Archive http://iidda.mcmaster.ca/. These incidents over the time reporting window can be seen in Fig 13. We used measles cases from England and Wales (1948-1967), chicken pox cases from Ontario, Canada (1946-1967), and rubella cases from Ontario, Canada (1946-1960)[2].

Furthermore, case notification data for our extension was found here: https://archive.ics.uci.edu/dataset/580/hungarian+chickenpox+cases and is represented in Fig. 14. Population data was found here: https://tinyurl.com/34dbkprp

5.2. **Power Spectral Density.** When evaluating epidemiological models, it is often more important to capture qualitative features of the dynamics—such as the frequency and periodicity of outbreaks—than to minimize pointwise error metrics like mean squared error (MSE). To this end, we compute the Power Spectral Density (PSD) of both the empirical and simulated time series as an alternative model validation approach.

The PSD quantifies the distribution of signal power across different frequencies, highlighting dominant cycles such as annual, biannual, or multi-annual outbreaks. By comparing the PSD of pySINDy-generated dynamics to that of the empirical data, we assess how well the model captures the underlying periodic structure.

The Power Spectral Density (PSD) of a time series $x(t)$ can be estimated using the periodogram:

$$\text{PSD}(f) = \frac{1}{N}\left|\sum_{t=0}^{N-1} x(t)w(t)e^{-2\pi i f t}\right|^2 \tag{8}$$

where N is the length of the time series, $w(t)$ is a windowing function (we use the Hanning window), and f is the frequency in cycles per time unit (e.g., weeks^{-1}).

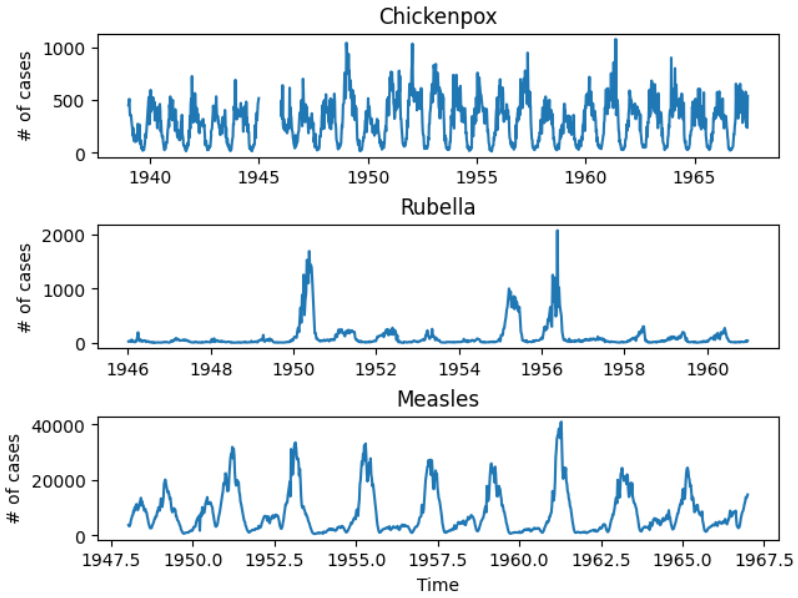


FIGURE 13. Case Notifications for chicken pox (in Ontario, CA), rubella (Ontario, CA), and measles (England & Wales)

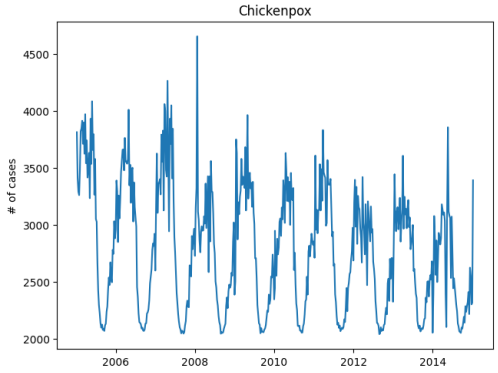


FIGURE 14. Case Notifications for chicken pox in Hungary 2005-2014

This method was applied to the infectious time series for both the empirical and simulated data. As shown in Figure 15, the PSD of the pySINDy-generated dynamics aligns well with that of the empirical data in terms of dominant frequency components—even in cases where the time-domain signal shapes do not perfectly match.

This spectral agreement reinforces that frequency-domain analysis is a valuable complementary metric to MSE for evaluating model fidelity in noisy, real-world epidemiological datasets [2].

Please note that the code used to create figures is attached following this paper. The original paper whose results we reconstructed [2] is attached following the code.

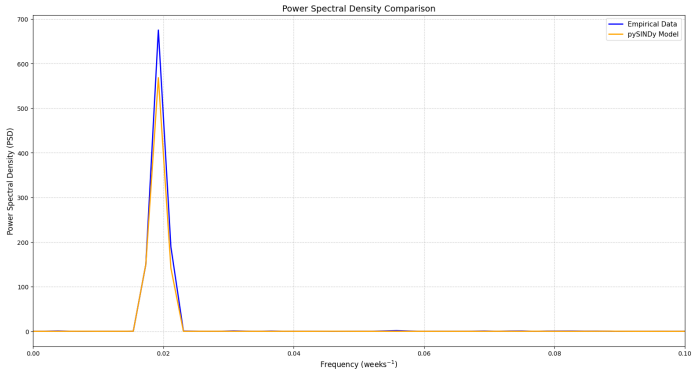


FIGURE 15. Power Spectral Density comparison between empirical infectious data and pySINDy-generated dynamics. Both curves exhibit dominant peaks at similar frequencies, indicating that pySINDy captures the correct oscillatory behavior despite potential mismatches in time-domain trajectories.

REFERENCES

[1] Mikael Jagan Michelle S. deJonge Olga Krylova David J. D. Earn. Fast estimation of time-varying infectious disease transmission rates. *PLoS Comput Biol 16(9)*, 2020.

[2] J Horrocks. Algorithmic discovery of dynamic models from infectious disease data. *Sci Rep 10, 7061*, 2020.

[3] J. Nathan Kutz Steven L. Brunton, Joshua L. Proctor. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *PNAS, 113, 15*, 2016.