Supplementary Materials: Temporal trends analysis of tuberculosis morbidity in mainland China from 1997 to 2025 using a new SARIMA-NARNNX hybrid model

Yongbin Wang, Chunjie Xu, Shengkui Zhang, Zhende Wang, Li Yang, Ying Zhu, Juxiang Yuan

Steps of the SARIMA, SARIMA-GRNN and SARIMA-NARNN models are as below:

Establishing the SARIMA model

Owing to the characteristics of TB morbidity sequences showing apparent periodicity and seasonality,[1] a classical SARIMA(p, d, q) $(P, D, Q)_s$ method is quite appropriate to understand the future long-trajectory patterns.[2] Where, d and D are the non-seasonal and seasonal differenced orders, respectively. p, q, P and Q represent the orders of models including the autoregressive(AR), moving average(MA), seasonal autoregressive(SAR) and moving average(SMA) methods, respectively. In this approach, the seasonal component of TB morbidity sequences was used for the independent variable, the actual monthly incidence values were taken for the dependent variable. Considering the requirement of stationary incidence time series during the process of SARIMA model construction,[3] in this work, an augmented Dickey-Fuller (ADF) test was hence employed to identify whether the actual reported TB cases and processed data adopting a transformation technique or differencing met the requirement of stationarity.[4] Subsequently, the autocorrelation function (ACF) and partial autocorrelation function(PACF) plots, schwarz bayesian criterion (SBC), akaike information criterion (AIC) and bias-corrected AIC (AICc), coupled with the Lagrangian multiplier (LM) and Ljung-Box Q tests were utilized to be in search for the possible candidate models.[5 6] Then, providing a further estimation and diagnosis for the residual errors series generated by these tentative models until the best-undertaking SARIMA model was discovered. Finally, using this identified model to calculate forecasts.

Erecting the SARIMA-GRNN combined model

As noted before, the SARIMA method has a limited approximate ability to the nonlinear property of the complex time series because of its essence being a linear model, which may result in an unwanted degeneration of simulating performance.[7] Whereas ANNs, essentially a class of

nonlinear models, are able to capture the nonlinear property hidden behind the complex series well owing to their powerful nonlinear modelling capacity.[8] Of all ANNs, GRNN, as a variant of the radial basis artificial network (RBF), has extensively been used for function approximation as it has swifter training speed and more flexible nonlinear mapping capability than other neural networks, such as RBF and back propagation network (BP).[8 9] In applications, the GRNN model can derive the underlying function with high accuracy by analyzing the relationship between the input variables and the expected output variables.[8] Driven by this trait of GRNN technique, in the SARIMA-GRNN hybrid technique, the estimates of the SARIMA approach were viewed as the input values, while the actual observed values were deemed as the output values. As such, the regression of outputs on inputs can be used to further approximate the information unable to be exploited by the SARIMA model, i.e., this hybrid method simultaneously consider both linear clues and non-linear trait of TB incidence data. Its mathematical expression can be written as[10]

$$E\left[y|X\right]=\frac{\int_{-\infty}^{\infty} yf(X,y)dy}{\int_{-\infty}^{\infty} f(X,y)dy} \qquad (1)$$

Here y denotes the observed values, X represents the estimates of the SARIMA approach, $E\left[y|X\right]$ refers to the expected output values of y given the estimates of the SARIMA approach, $f(X,y)$ signifies the joint function of the estimates of the SARIMA approach and the observed values.

Since the forecasting accuracy relies upon the smoothing factor that is the only parameter to be evaluated in the process of establishing a GRNN technique.[11] Consequently, in order to achieve a satisfactory forecasting, the preferred smoothing factor was sought based on the procedure proposed by Specht in the present study.[11] Firstly, two data points out of the training dataset were randomly chosen as testing samples and the rest were employed to mimic the GRNN model. Then, a series of smoothing factors were tested. Among which once the lowest RMSE value was observed, the approximation of GRNN network to the testing data achieved the best fit. Finally, the model with the optimal smoothing factor yielded was used to forecast.

Building the SARIMA-NARNN hybrid model

Generally, the incidence sequences of infectious diseases encompass both linear and nonlinear elements. The SARIMA model often gives priority to the linear trend, the nonlinear information is thus stored in the residual errors.[12] While the NARNN model based on the theory of nonlinear regression analysis has a penetrating sight to find what hides in the residual errors with the help of time-varying state of interconnected neurons able to dynamically relate current inputs to earlier observations, and therefore it has been deemed as a function approximator to solve nonlinear problems in the forecasting domains.[10 12 13] Inspired by the desirable trait of this model, hence a SARIMA-NARNN hybrid technique was erected to boost the estimated performance by exploiting their individual strength of both models.[13 14] In this way, the underlying relationship (both linear and nonlinear correlation structures) between the input variables and the output variables can be extracted by simulation using this combined method. The specified equation of the SARIMA-NARNN combined model can be written as

$$\hat{e}(t) = f(e(t-1), e(t-2), \cdots, e(t-d)) \tag{2}$$

$$\hat{y} = \hat{a}_t + \hat{e}_t \tag{3}$$

Here f denotes a function relied on the structure and connection weights of the NARNN model, $\hat{y}$ is the values modeled and projected by this combined model, $\hat{a}_t$ refers to the estimates of the SARIMA approach, $\hat{e}_t$ stands for the fitted and predicted points based only on the SARIMA errors series at lagged period d.

In this project, the modelling procedures of SARIMA-NARNN hybrid method were as below: Initially, the residuals from the SARIMA model were processed between 0 and 1 using a normalized approach,[15] which could give a facilitation for further analysis. Subsequently, the random *dividerand* function was applied to classify the in-sample points in the errors into three portions containing a training dataset, a validation dataset, and a testing dataset, among them, their corresponding ratios were set to 80%, 10% and 10%, respectively. Then, the number of hidden neurons and delays d were adjusted by trial and error using Levenberg-Marquardt algorithm in an open feedback loop mode. Meanwhile, the response plot of outputs and targets, residual ACF plot, together with the mean square error (MSE) and correlation coefficient (R) were offered to find the best-performing NARNN model. Next, the training open-loop architecture was transformed to be a closed loop mode to take multi-step-ahead forecasts (**figure S1**). Here, bearing in mind that the

estimated values should be undertaken back-transformation. Finally, the combination method's results were generated by summing the simulations and predictions from the SARIMA model and the NARNN model.

Figure and Table captions

Table S1 Time series of monthly TB observed series in mainland China from January 1997 to March 2018.

Table S2 The estimated parameters and performance indexes of selected candidate models based on original observations.

Table S3 The in-sample fitting and out-of-sample predicted performances from the SARIMA-NARNN hybrid model built with dataset from January 1997 to December 2012.

Table S4 The ARCH effects and Ljung-Box Q test of the residuals from the SARIMA-NARNNX hybrid model developed based on the additional testing dataset from January 1997 to December 2012 at different lags.

Table S5 The ARCH effects and Ljung-Box Q test of the residuals from the SARIMA-NARNNX hybrid model for the whole data at different lags.

Table S6 Comparison of goodness of fit for the candidate ETS models constructed with the dataset from January 1997 to December 2016.

Table S7 Initial parameters of ETS(M,A,A) model.

Table S8 Comparison of in-sample fitting and out-of-sample predicting performances among the best-performing approaches chosen.

Figure S1 The layer architecture of the SARINA-NARNN hybrid model. (a) The opened loop form; (b) The closed loop form. It can be observed that the SARIMA-NARNN hybrid model with 32 neurons and 6 delays and an output layer with 1 neuron should be considered as the best-fitting in our experiments. This is a two-layer feed forward network, with a sigmoid transfer function in the hidden layer and a linear transfer function in the output layer. A feedback connection from the model output is taken for one input of this model. After the model is trained, this feedback connection should be closed for multi-step-ahead-forecasting. For each of these inputs, there exists

a tapped delay line to store the past values.

Figure S2 The layer architecture of the SARIMA-NARNNX hybrid model. (a) The opened loop mode; (b) The closed loop mode. It can be found that the SARIMA-NARNNX hybrid model with 17 neurons and 4 delays and an output layer with 1 neuron should be considered as the best-fitting in our experiments. This is also a two-layer feed forward network, with a sigmoid transfer function in the hidden layer and a linear transfer function in the output layer. A feedback connection from the model output is taken for one input of this model. After the model is trained, this feedback connection should be closed for multi-step-ahead-forecasting. For each of these inputs, there exists a tapped delay line to store the past values. In this network, the inputs include the simulations and forecasts from the SARIMA model and the time factor, the expected outputs are the actual TB incidence values.

Figure S3 Time series of monthly TB incidence cases in mainland China from January 1997 to March 2018. As shown above, a downward trend was observed since 2006.

Figure S4 ACF and PACF graph with first-order non-seasonal difference (d=1) of monthly TB incidence series in mainland China from January 1997 to December 2016.

Figure S5 ACF and PACF graph with first-order seasonal difference (D=1) of monthly TB incidence series in mainland China from January 1997 to December 2016.

Figure S6 The root mean square error (RMSE) values corresponding to different smoothing factors for the SARIMA-GRNN combined technique. The smoothing factor between 0.001 and 1 with an interval of 0.001, it can be seen that when smoothing factor is 0.006, the lowest RMSE value is 0.0024.

Figure S7 The linear regression between the network outputs and the corresponding targets for the traditional SARIMA-NARNN hybrid model developed based on the dataset from January 1997 to December 2016. The R value is an indication of the relationship between the outputs and targets. If R=1, this indicates that there is an exact linear relationship between outputs and targets. If R is close to zero, then there is no linear relationship between outputs and targets. Generally, all the R values are greater than 0.8, the training data indicate a good fit. However, depending on our experience and referring to the related literature[10 14], we find that it is quite difficult to simultaneously train the R value of greater than 0.8 in the training, validation, testing and all data.

Thus, the overall R value of greater than 0.8 is often regarded as a good fitting in the ARIMA-NARNN hybrid model.

Figure S8 The linear regression between the network outputs and the corresponding targets for the SARIMA-NARNNX hybrid model developed based on the dataset from January 1997 to December 2016. The R value is an indication of the relationship between the outputs and targets. If R=1, this indicates that there is an exact linear relationship between outputs and targets. If R is close to zero, then there is no linear relationship between outputs and targets. Generally, all the R values are greater than 0.8, the training data indicate a good fit. In this SARIMA-NARNNX hybrid model, all the R values are greater than 0.98, showing a perfect modeling.

Figure S9 The layer architecture of the SARIMA-NARNNX hybrid model for testing the long-term forecasting performance to the TB incidence data. (a) The opened loop mode; (b) The closed loop mode. It can be found that the SARIMA-NARNNX hybrid model with 16 neurons and 5 delays and an output layer with 1 neuron should be considered as the best-fitting in our experiments. This is also a two-layer feed forward network, with a sigmoid transfer function in the hidden layer and a linear transfer function in the output layer. A feedback connection from the model output is taken for one input of this model. After the model is trained, this feedback connection should be closed for multi-step-ahead-forecasting. For each of these inputs, there exists a tapped delay line to store the past values. In this network, the inputs include the simulations and forecasts from the SARIMA model and the time factor, the expected outputs are the actual TB incidence values.

Figure S10 The linear regression plot between the network outputs and the corresponding targets for the SARIMA-NARNNX hybrid model developed based on the additional testing dataset from January 1997 to December 2012. The R value is an indication of the relationship between the outputs and targets. If R=1, this indicates that there is an exact linear relationship between outputs and targets. If R is close to zero, then there is no linear relationship between outputs and targets. Generally, all the R values are greater than 0.8, the training data indicate a good fit. In this SARIMA-NARNNX hybrid model, all the R values are greater than 0.97, displaying a perfect modeling.

Figure S11 The corresponding time series response of outputs and targets at various time points for the SARIMA-NARNNX hybrid model developed based on the additional testing dataset from

January 1997 to December 2012. This plot shows the inputs, targets and errors versus time. It also indicates which time points were selected for training, testing and validation. Overall in the training, validation and testing samples the errors were small. We confirm that the model is adequate for TB incidence data modeling.

Figure S12 The resulting error autocorrelation plot for the SARIMA-NARNNX hybrid model developed based on the additional testing dataset from January 1997 to December 2012. There are no significant correlations in the residual series, so the model provides a reasonable approximation to the TB incidence data.

Figure S13 The linear regression between the network outputs and the corresponding targets for the SARIMA-NARNNX hybrid model developed based on the dataset from January 1997 to March 2018. The R value is an indication of the relationship between the outputs and targets. If R=1, this indicates that there is an exact linear relationship between outputs and targets. If R is close to zero, then there is no linear relationship between outputs and targets. Generally, all the R values are greater than 0.8, the training data indicate a good fit. In this SARIMA-NARNNX hybrid model, all the R values are greater than 0.98, showing a perfect modeling.

Figure S14 The resulting error autocorrelation plot for the SARIMA-NARNNX hybrid model developed based on the dataset from January 1997 to March 2018. The correlation coefficients fall within the 95% confidence limits apart from the one at zero lag, it seems that the model is suitable for TB incidence data forecasting.

Figure S15 The corresponding time series response of outputs and targets for the SARIMA-NARNNX hybrid model developed based on the dataset from January 1997 to March 2018 at various time points. This plot shows the inputs, targets and errors versus time. It also indicates which time points were selected for training, testing and validation. Overall in the training, validation and testing samples the errors were small. We confirm that the model is adequate for TB incidence data modeling.

Figure S16 The corresponding time series response of outputs and targets at various time points for the basic NARNN model developed based on the dataset from January 1997 to December 2016. This plot shows the inputs, targets and errors versus time. It also indicates which time points were selected for training, testing and validation. Overall in the training, validation and testing samples the errors were small. We confirm that the model is adequate for TB incidence data

modeling.

Figure S17 The linear regression plot between the network outputs and the corresponding targets for the basic NARNN model developed based on the dataset from January 1997 to December 2016. The R value is an indication of the relationship between the outputs and targets. If R=1, this indicates that there is an exact linear relationship between outputs and targets. If R is close to zero, then there is no linear relationship between outputs and targets. Generally, all the R values are greater than 0.8, the training data indicate a good fit. In this SARIMA-NARNNX hybrid model, all the R values are greater than 0.96, displaying a perfect modeling.

Figure S18 The resulting error autocorrelation plot for the basic NARNN model developed based on the dataset from January 1997 to December 2016. None of autocorrelation coefficients are beyond the estimated 95% confidence limits; this identified method seems to have captured the essence of the dependence in the TB incidence series.

Figure S19 The AIC values of the 30 candidate models. Based on this plot, it can be found the ETS (M, A, A) with the lowest values among the 30 candidate models. Thus this model is selected to conduct out-of-sample forecasting.

Table S1 Time series of monthly TB observed series in mainland China from January 1997 to March 2018.

Time	Cases	Time	Cases	Time	Cases	Time	Cases	Time	Cases
01-1997	14515	04-2001	52094	07-2005	133807	10-2009	109343	01-2014	96067
02-1997	17623	05-2001	52617	08-2005	132729	11-2009	104782	02-2014	88549
03-1997	30961	06-2001	52563	09-2005	123062	12-2009	120341	03-2014	111975
04-1997	35122	07-2001	51394	10-2005	104851	01-2010	105877	04-2014	118849
05-1997	37257	08-2001	49367	11-2005	117468	02-2010	88759	05-2014	106557
06-1997	36691	09-2001	45830	12-2005	116859	03-2010	138574	06-2014	101463
07-1997	38655	10-2001	42801	01-2006	89436	04-2010	133833	07-2014	105782
08-1997	38246	11-2001	43320	02-2006	103823	05-2010	128598	08-2014	97039
09-1997	35498	12-2001	44073	03-2006	147996	06-2010	127545	09-2014	95893
10-1997	32967	01-2002	29018	04-2006	145029	07-2010	122602	10-2014	88991
11-1997	31801	02-2002	28990	05-2006	132367	08-2010	117221	11-2014	86081
12-1997	31873	03-2002	48085	06-2006	134879	09-2010	112288	12-2014	93000
01-1998	16330	04-2002	51178	07-2006	123829	10-2010	101463	01-2015	95151
02-1998	27208	05-2002	51107	08-2006	127509	11-2010	110414	02-2015	75541
03-1998	38467	06-2002	53200	09-2006	115540	12-2010	105036	03-2015	113242
04-1998	43856	07-2002	56408	10-2006	108177	01-2011	99617	04-2015	107997
05-1998	44869	08-2002	52355	11-2006	111384	02-2011	98157	05-2015	102067
06-1998	44479	09-2002	49588	12-2006	114262	03-2011	135848	06-2015	102388
07-1998	45433	10-2002	46739	01-2007	115457	04-2011	129351	07-2015	102986
08-1998	43112	11-2002	49011	02-2007	92080	05-2011	125129	08-2015	95746
09-1998	39580	12-2002	50528	03-2007	141508	06-2011	119344	09-2015	95081
10-1998	38760	01-2003	33148	04-2007	148930	07-2011	112647	10-2015	86024
11-1998	37212	02-2003	40014	05-2007	135933	08-2011	115140	11-2015	88115
12-1998	37271	03-2003	27119	06-2007	134775	09-2011	106925	12-2015	89818
01-1999	26566	04-2003	62066	07-2007	134695	10-2011	100392	01-2016	86181
02-1999	22579	05-2003	60369	08-2007	130404	11-2011	110662	02-2016	83527
03-1999	38784	06-2003	73756	09-2007	119078	12-2011	104710	03-2016	114627
04-1999	43626	07-2003	68798	10-2007	111325	01-2012	87893	04-2016	103397
05-1999	47754	08-2003	64889	11-2007	116503	02-2012	124005	05-2016	100500
06-1999	46912	09-2003	65234	12-2007	119421	03-2012	138683	06-2016	94064
07-1999	46988	10-2003	57781	01-2008	111688	04-2012	128683	07-2016	92793
08-1999	45324	11-2003	62069	02-2008	101689	05-2012	131257	08-2016	95924
09-1999	40855	12-2003	73375	03-2008	156679	06-2012	118740	09-2016	89342
10-1999	42132	01-2004	28904	04-2008	153978	07-2012	117359	10-2016	83092
11-1999	36449	02-2004	67522	05-2008	142612	08-2012	115019	11-2016	87963
12-1999	39606	03-2004	95713	06-2008	131699	09-2012	106188	12-2016	89609
01-2000	25997	04-2004	108089	07-2008	136378	10-2012	100091	01-2017	80911
02-2000	29967	05-2004	96151	08-2008	122923	11-2012	105264	02-2017	92037
03-2000	45400	06-2004	102495	09-2008	123998	12-2012	98306	03-2017	105633
04-2000	46927	07-2004	105254	10-2008	119821	01-2013	104238	04-2017	97296
05-2000	48230	08-2004	106334	11-2008	110896	02-2013	89582	05-2017	101628

06-2000	49481	09-2004	104815	12-2008	121114	03-2013	125549	06-2017	99001
07-2000	49105	10-2004	89973	01-2009	90160	04-2013	121701	07-2017	96471
08-2000	49222	11-2004	102560	02-2009	127167	05-2013	122172	08-2017	100076
09-2000	43598	12-2004	104038	03-2009	139986	06-2013	106546	09-2017	92494
10-2000	42955	01-2005	85577	04-2009	139915	07-2013	113531	10-2017	81554
11-2000	42594	02-2005	75625	05-2009	128411	08-2013	107051	11-2017	89976
12-2000	42761	03-2005	151935	06-2009	142182	09-2013	105219	12-2017	87630
01-2001	25411	04-2005	165419	07-2009	129537	10-2013	100254	01-2018	96125
02-2001	38182	05-2005	150823	08-2009	126893	11-2013	101595	02-2018	77224
03-2001	48268	06-2005	151330	09-2009	124152	12-2013	100685	03-2018	110124

Table S2 The estimated parameters and performance indexes of selected candidate models based on original observations.

Models	Estimated parameter coefficients of candidate models									Performance indexes of candidate models		
	AR1 (SE)	AR2 (SE)	MA1 (SE)	MA2 (SE)	MA3 (SE)	SAR1 (SE)	SAR2 (SE)	SMA1 (SE)	SMA2 (SE)	AIC	AICc	SBC
SARIMA$(0,1,1)(0,1,1)_{12}$			-0.619* (0.050)					-0.643* (0.064)		4915.24	4915.34	4925.59
SARIMA$(0,1,1)(1,1,1)_{12}$	-0.632* (0.050)					0.220 (0.130)		-0.804* (0.111)		4914.03	4914.21	4927.84
SARIMA$(0,1,1)(0,1,2)_{12}$			-0.633* (0.050)					-0.573* (0.072)	-0.148 (0.084)	4913.86	4914.04	4927.67
SARIMA$(1,1,1)(0,1,1)_{12}$	-0.086 (0.106)		-0.568* (0.086)					-0.632* (0.065)		4916.58	4916.76	4930.39
SARIMA$(0,1,1)(2,1,0)_{12}$			-0.634* (0.048)			-0.493 (0.063)	-0.317 (0.060)			4920.72	4920.89	4934.52
SARIMA$(0,1,2)(0,1,1)_{12}$			-0.655* (0.068)	0.068 (0.067)				-0.632* (0.065)		4916.57	4916.75	4930.38
SARIMA$(2,1,0)(0,1,1)_{12}$	-0.604* (0.065)	-0.262* (0.064)						-0.645* (0.063)		4921.93	4922.10	4935.73
SARIMA$(1,1,0)(1,1,1)_{12}$	-0.476* (0.058)					0.201 (0.137)		-0.834* (0.129)		4935.84	4936.01	4949.64
SARIMA$(1,1,0)(0,1,1)_{12}$	-0.473* (0.059)							-0.678* (0.064)		4936.32	4936.42	4946.67
SARIMA$(1,1,0)(0,1,2)_{12}$	-0.476* (0.058)							-0.627* (0.071)	-0.125 (0.085)	4935.96	4936.13	4949.76
SARIMA$(2,1,3)(0,1,1)_{12}$	1.520* (0.054)	-0.909* (0.046)	-2.161* (0.055)	1.938* (0.084)	-0.633* (0.048)			-0.700* (0.066)		4914.44	4914.94	4938.60

Note: * the estimated parameters are statistical significance at the 5% level; SE stands for standard error; SARIMA, seasonal autoregressive integrated moving average model;AR1, moving average, lag1; AR2, moving average, lag2; MA1, moving average, lag1; MA2, moving average, lag2; MA3, moving average, lag3; SAR1, seasonal moving average, lag1; SAR2, seasonal moving average, lag2; SMA1, seasonal moving average, lag1; SMA2, seasonal moving average, lag2; AIC, akaike information criterion; AICc bias-corrected akaike information criterion; SBC, schwarz bayesian criterion .

Table S3 The in-sample fitting and out-of-sample predicted performances from the SARIMA-NARNN hybrid model built with dataset from January 1997 to December 2012.

Model	Simulating power				Predictive power			
	MAE	**MER**	**MAPE**	**RMSE**	**MAE**	**MER**	**MAPE**	**RMSE**
SARIMA-NARNNX	3341.434	0.037	0.045	4769.682	22324.157	0.228	0.221	26870.360

SARIMA, seasonal autoregressive integrated moving average model; NARNNX, nonlinear autoregressive neural network with external input; SARIMA-NARNNX; integrating a seasonal autoregressive integrated moving model with a nonlinear autoregressive neural network with external input; MAE, mean absolute error; MER, mean error rate; MAPE, mean absolute percentage error; RMSE, root mean square error.

Table S4 The ARCH effects and Ljung-Box Q test of the residuals from the SARIMA-NARNNX hybrid model developed based on the additional testing dataset from January 1997 to December 2012 at different lags.

Lags	Observed values				Residuals of SARIMA-NARNNX model			
	Box-Ljung Q	*P*	LM-test	*P*	Box-Ljung Q	*P*	LM-test	*P*
1	154.855	<0.001	143.570	<0.001	0.231	0.631	0.065	0.798
3	424.975	<0.001	143.810	<0.001	1.034	0.793	0.520	0.915
6	774.306	<0.001	141.190	<0.001	2.080	0.912	0.798	0.992
9	1119.289	<0.001	142.050	<0.001	8.043	0.530	1.848	0.994
12	1504.353	<0.001	146.320	<0.001	9.851	0.629	4.198	0.980
15	1816.048	<0.001	148.190	<0.001	13.979	0.527	8.902	0.883
18	2051.319	<0.001	145.290	<0.001	19.279	0.375	10.705	0.906
21	2276.548	<0.001	142.240	<0.001	19.433	0.557	11.386	0.955
24	2519.234	<0.001	139.640	<0.001	25.456	0.381	11.800	0.982
27	2696.775	<0.001	136.540	<0.001	26.950	0.466	12.125	0.994
30	2819.505	<0.001	133.400	<0.001	40.214	0.101	19.711	0.924
33	2932.674	<0.001	130.120	<0.001	46.041	0.065	20.963	0.948
36	3054.331	<0.001	126.850	<0.001	46.129	0.120	22.068	0.967

SARIMA, seasonal autoregressive integrated moving average model; NARNNX, nonlinear autoregressive neural network with external input; SARIMA-NARNNX; integrating a seasonal autoregressive integrated moving model with a nonlinear autoregressive neural network with external input; LM test, Lagrangian multiplier test, which is used to verify the conditional heteroskedastic behaviour and volatility (ARCH effect); Box-Ljung Q is used to verify the stochasticity (white noise).

Table S5 The ARCH effects and Ljung-Box Q test of the residuals from the SARIMA-NARNNX hybrid model for the whole data at different lags.

Lags	Observed values				Residuals of SARIMA-NARNNX model			
	Box-Ljung Q	*P*	LM-test	*P*	Box-Ljung Q	*P*	LM-test	*P*
1	217.950	<0.001	202.830	<0.001	0.018	0.894	0.528	0.467
3	599.170	<0.001	203.600	<0.001	0.486	0.922	0.771	0.856
6	1093.400	<0.001	201.210	<0.001	2.871	0.825	1.342	0.969
9	1586.900	<0.001	204.170	<0.001	5.578	0.781	3.265	0.953
12	2145.100	<0.001	211.240	<0.001	6.143	0.909	5.997	0.916
15	2606.100	<0.001	215.920	<0.001	7.317	0.948	6.312	0.974
18	2962.300	<0.001	212.960	<0.001	10.608	0.910	6.769	0.992
21	3311.500	<0.001	209.780	<0.001	10.781	0.967	7.349	0.997
24	3699.300	<0.001	207.410	<0.001	16.345	0.875	8.909	0.998
27	3996.500	<0.001	204.460	<0.001	26.664	0.482	9.344	0.999
30	4212.600	<0.001	201.390	<0.001	28.392	0.550	10.041	1.000
33	4420.800	<0.001	198.160	<0.001	32.833	0.475	10.861	1.000
36	4655.000	<0.001	195.530	<0.001	33.551	0.586	11.540	1.000

SARIMA, seasonal autoregressive integrated moving average model; NARNNX, nonlinear autoregressive neural network with external input; SARIMA-NARNNX; integrating a seasonal autoregressive integrated moving model with a nonlinear autoregressive neural network with external input; LM test, Lagrangian multiplier test, which is used to verify the conditional heteroskedastic behaviour and volatility (ARCH effect); Box-Ljung Q is used to verify the stochasticity (white noise).

Table S6 Comparison of goodness of fit for the candidate ETS models constructed with the dataset from January 1997 to December 2016.

Models	Compact LL	Likelihood	AIC*	BIC	HQ	AMSE
M,A,A	-2813.028	-2495.897	5658.057	5713.747	5680.496	9.10E+07
M,MD,A	-2812.057	-2494.925	5658.114	5717.284	5681.955	9.00E+07
M,M,A	-2813.810	-2496.678	5659.619	5715.310	5682.058	9.90E+07
M,AD,A	-2812.900	-2495.768	5659.800	5718.971	5683.641	NA
A,MD,M	-2817.507	-2500.376	5669.015	5728.186	5692.856	7.80E+07
A,A,M	-2818.734	-2501.603	5669.468	5725.158	5691.907	8.10E+07
A,AD,M	-2818.314	-2501.182	5670.628	5729.798	5694.469	NA
M,N,A	-2823.424	-2506.292	5674.847	5723.576	5694.481	8.90E+07
A,M,M	-2821.850	-2504.719	5675.700	5731.390	5698.139	8.50E+07
A,N,A	-2832.521	-2515.390	5693.043	5741.771	5712.677	8.60E+07
A,MD,A	-2830.218	-2513.086	5694.436	5753.606	5718.277	8.30E+07
A,AD,A	-2830.325	-2513.193	5694.649	5753.820	5718.491	NA
A,A,A	-2831.748	-2514.617	5695.496	5751.187	5717.935	8.50E+07
A,M,A	-2832.399	-2515.267	5696.798	5752.488	5719.237	8.60E+07
A,N,M	-2837.027	-2519.896	5702.054	5750.783	5721.688	8.80E+07
M,MD,M	-2834.050	-2516.919	5702.100	5761.271	5725.942	8.80E+07
M,A,M	-2835.959	-2518.828	5703.918	5759.608	5726.357	9.30E+07
M,AD,M	-2835.276	-2518.144	5704.552	5763.722	5728.393	NA
M,M,M	-2836.673	-2519.541	5705.345	5761.036	5727.785	1.00E+08
M,N,M	-2851.263	-2534.132	5730.526	5779.255	5750.161	9.50E+07
A,N,N	-2938.591	-2621.459	5881.181	5888.143	5883.986	2.60E+08
A,A,N	-2938.463	-2621.332	5884.927	5898.849	5890.537	2.60E+08
A,M,N	-2938.476	-2621.344	5884.951	5898.874	5890.561	2.60E+08
A,MD,N	-2937.941	-2620.809	5885.881	5903.285	5892.894	2.60E+08
A,AD,N	-2937.963	-2620.831	5885.926	5903.329	5892.938	NA
M,AD,N	-2955.040	-2637.909	5920.080	5937.483	5927.092	2.60E+08
M,A,N	-2956.395	-2639.263	5920.789	5934.712	5926.399	2.60E+08
M,MD,N	-2958.721	-2641.590	5927.443	5944.846	5934.455	2.50E+08
M,M,N	-2960.107	-2642.976	5928.214	5942.137	5933.824	2.70E+08
M,N,N	-2962.530	-2645.398	5929.059	5936.021	5931.864	2.50E+08

*1 model failed to converge; Compact LL, Compact Log-likelihood; AIC, Akaike Information Criterion; BIC, Schwarz Criterion; HQ, Hannan-Quinn Criterion; AMSE, Average Mean Squared Error. Selection among ETS processes can be done by the performance indices including the Compact LL, AIC, BIC, HQ and AMSE, among these measures, the ETS model with the smallest AIC, BIC, HQ and AMSE, along with the largest Compact LL and Likelihood should be identified as the best-mimicking. Based on the results presented above, the ETS(M,A,A) is the best-performing model for the TB incidence data from January 1997 to December 2016. (**Note**: A: Additive, N: None, M: Multiplicative, AD: Additive damped, MD: Multiplicative damped). Besides, the detail descriptions regarding the ETS model can be found in the reference [16].

Table S7 Initial parameters of ETS(M,A,A) model.

Parameters	**Values**
Alpha:	0.214
Beta:	0.044
Gamma:	0.155
Initial Parameters	
Initial level:	29349.008
Initial trend:	350.134
Initial state 1:	-2096.173
Initial state 2:	-2769.307
Initial state 3:	-1326.864
Initial state 4:	865.241
Initial state 5:	4565.916
Initial state 6:	6465.163
Initial state 7:	6734.249
Initial state 8:	6410.058
Initial state 9:	6252.016
Initial state 10:	2141.260
Initial state 11:	-11021.883
Initial state 12:	-16219.675
Performance	
Compact Log-likelihood	-2813.028
Log-likelihood	-2495.897
Akaike Information Criterion	5658.057
Schwarz Criterion	5713.747
Hannan-Quinn Criterion	5680.496
Sum of Squared Residuals	2.468
Root Mean Squared Error	0.101
Average Mean Squared Error	91409903.199

Note: the detail descriptions regarding the ETS model can be found in the reference [16].

Table S8 Comparison of in-sample fitting and out-of-sample predicting performances among the best-performing approaches chosen.

Models	Simulating power				Predictive power			
	MAE	**MER**	**MAPE**	**RMSE**	**MAE**	**MER**	**MAPE**	**RMSE**
SARIMA	5636.303	0.062	0.067	8781.186	5726.262	0.061	0.060	7104.34
SARIMA-GRNN	4437.958	0.049	0.054	6939.078	5415.985	0.058	0.059	6155.964
SARIMA-NARNN	3283.274	0.035	0.043	5265.82	5259.556	0.056	0.055	6418.445
SARIMA-NARNNX	2878.484	0.031	0.038	4468.578	3563.179	0.038	0.038	4917.829
NARNN	4041.963	0.045	0.056	5953. 684	7394.040	0.079	0.080	8503.905
ETS(M,A,A)	5545.661	0.063	0.067	8874.912	4981.129	0.053	0.053	7278.845
Percentage reduction (%)								
D vs. A	48.930	50.000	43.284	49.112	37.775	37.705	36.667	30.777
D vs. B	27.668	29.032	23.881	28.134	32.356	32.787	35.000	17.428
D vs. C	7.182	6.452	7.463	9.079	29.625	29.508	28.333	21.123
D vs. E	28.785	31.111	32.143	24.944	51.810	51.899	52.500	42.170
D vs. F	65.987	71.111	51.786	74.010	19.177	18.987	18.750	27.764

A refers to the SARIMA model; B stands for the SARIMA-GRNN hybrid model; C signifies the SARIMA-NARNN hybrid model; D represents the SARIMA-NARNNX the hybrid model; E is the basic NARNN model; F refers to the ETS model; SARIMA, seasonal autoregressive integrated moving average model; GRNN, generalized regression neural network; NARNN, nonlinear auto-regressive neural network model; NARNNX, nonlinear autoregressive neural network with external input; ETS, Error-Trend-Seasonal approach; SARIMA-GRNN; integrating a seasonal autoregressive integrated moving model with a generalized regression neural network; SARIMA-NARNN; integrating a seasonal autoregressive integrated moving model with a nonlinear autoregressive neural network; SARIMA-NARNNX; integrating a seasonal autoregressive integrated moving model with a nonlinear autoregressive neural network with external input; MAE, mean absolute error; MER, mean error rate; MAPE, mean absolute percentage error; RMSE, root mean square error.

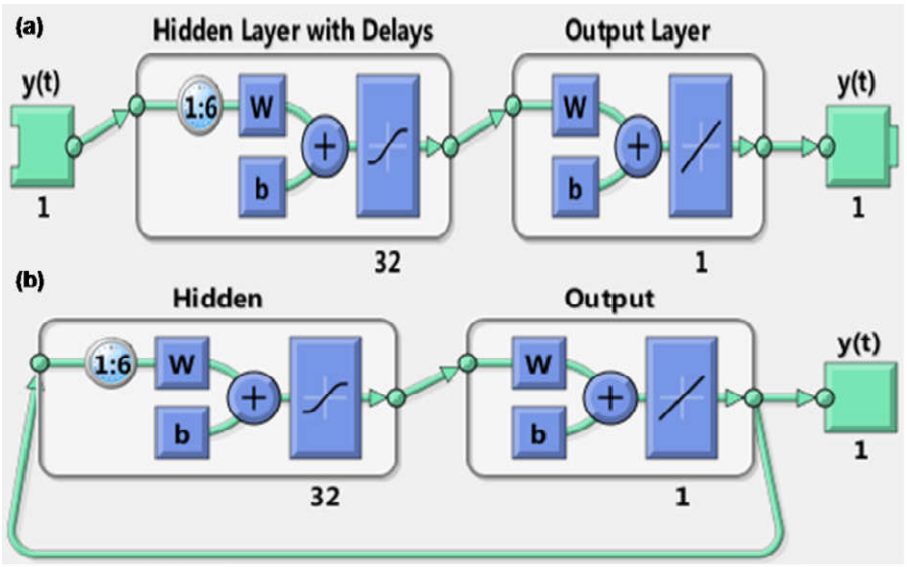

Figure S1 The layer architecture of the SARINA-NARNN hybrid model. (a) The opened loop form; (b) The closed loop form. It can be observed that the SARIMA-NARNN hybrid model with 32 neurons and 6 delays and an output layer with 1 neuron should be considered as the best-fitting in our experiments. This is a two-layer feed forward network, with a sigmoid transfer function in the hidden layer and a linear transfer function in the output layer. A feedback connection from the model output is taken for one input of this model. After the model is trained, this feedback connection should be closed for multi-step-ahead-forecasting. For each of these inputs, there exists a tapped delay line to store the past values.

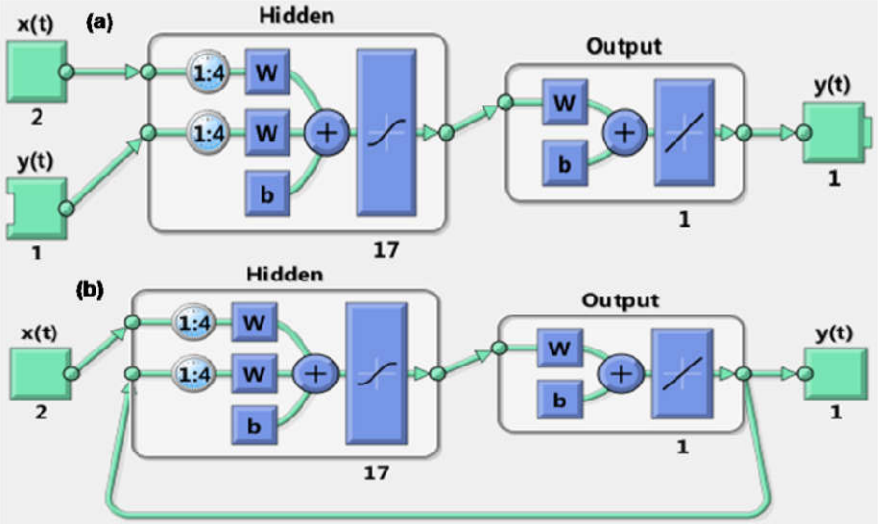

Figure S2 The layer architecture of the SARIMA-NARNNX hybrid model. (a) The opened loop mode; (b) The closed loop mode. It can be found that the SARIMA-NARNNX hybrid model with 17 neurons and 4 delays and an output layer with 1 neuron should be considered as the best-fitting in our experiments. This is also a two-layer feed forward network, with a sigmoid transfer function in the hidden layer and a linear transfer function in the output layer. A feedback connection from the model output is taken for one input of this model. After the model is trained, this feedback connection should be closed for multi-step-ahead-forecasting. For each of these inputs, there exists a tapped delay line to store the past values. In this network, the inputs include the simulations and forecasts from the SARIMA model and the time factor, the expected outputs are the actual TB incidence values.

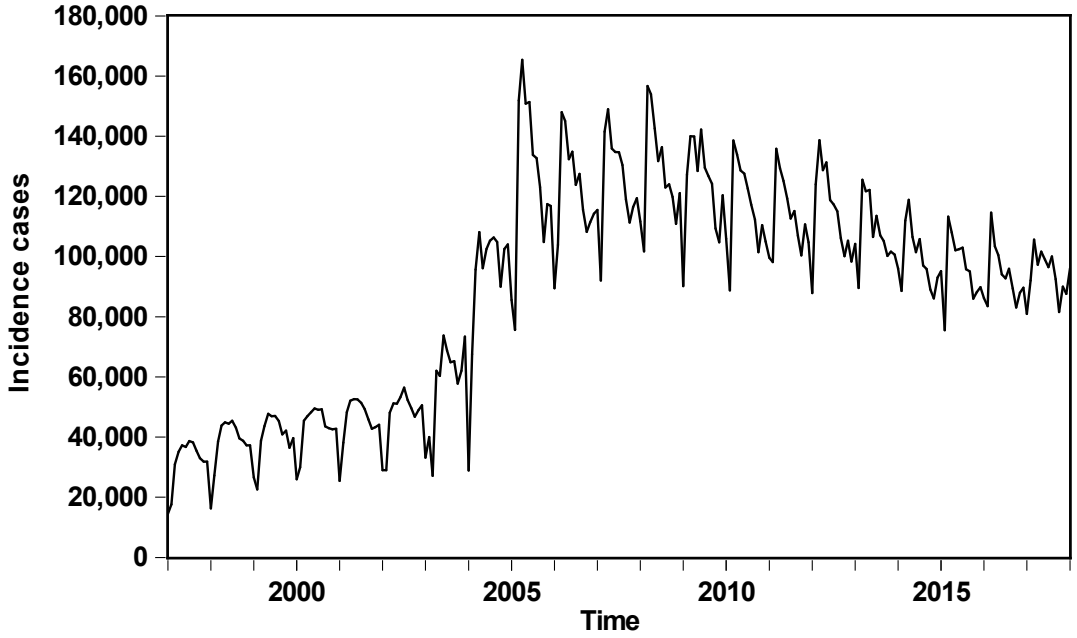

Figure S3 Time series of monthly TB incidence cases in mainland China from January 1997 to March 2018. As shown above, a downward trend was observed since 2006.

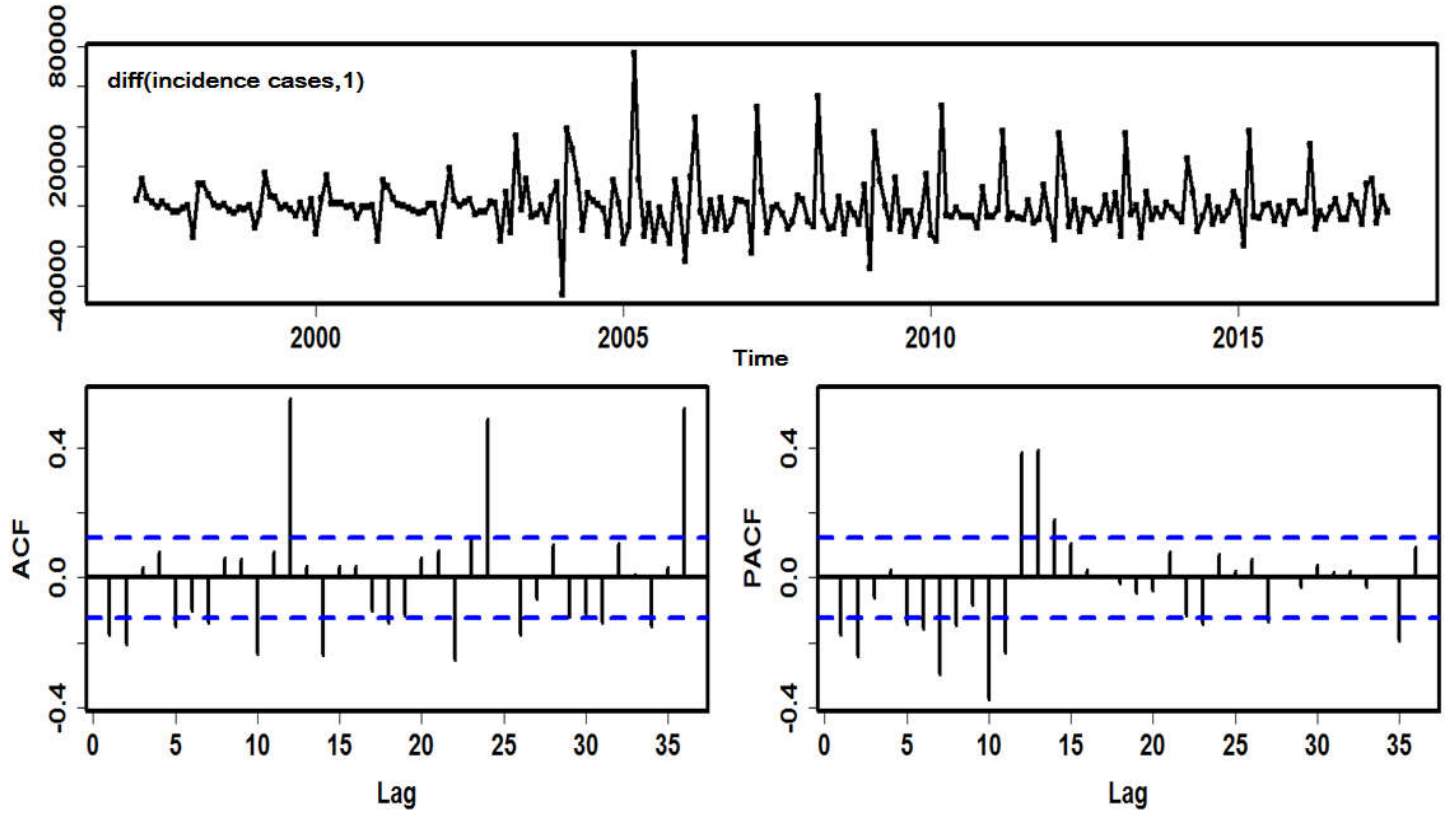

Figure S4 ACF and PACF graph with first-order non-seasonal difference (d=1) of monthly TB incidence series in mainland China from January 1997 to December 2016.

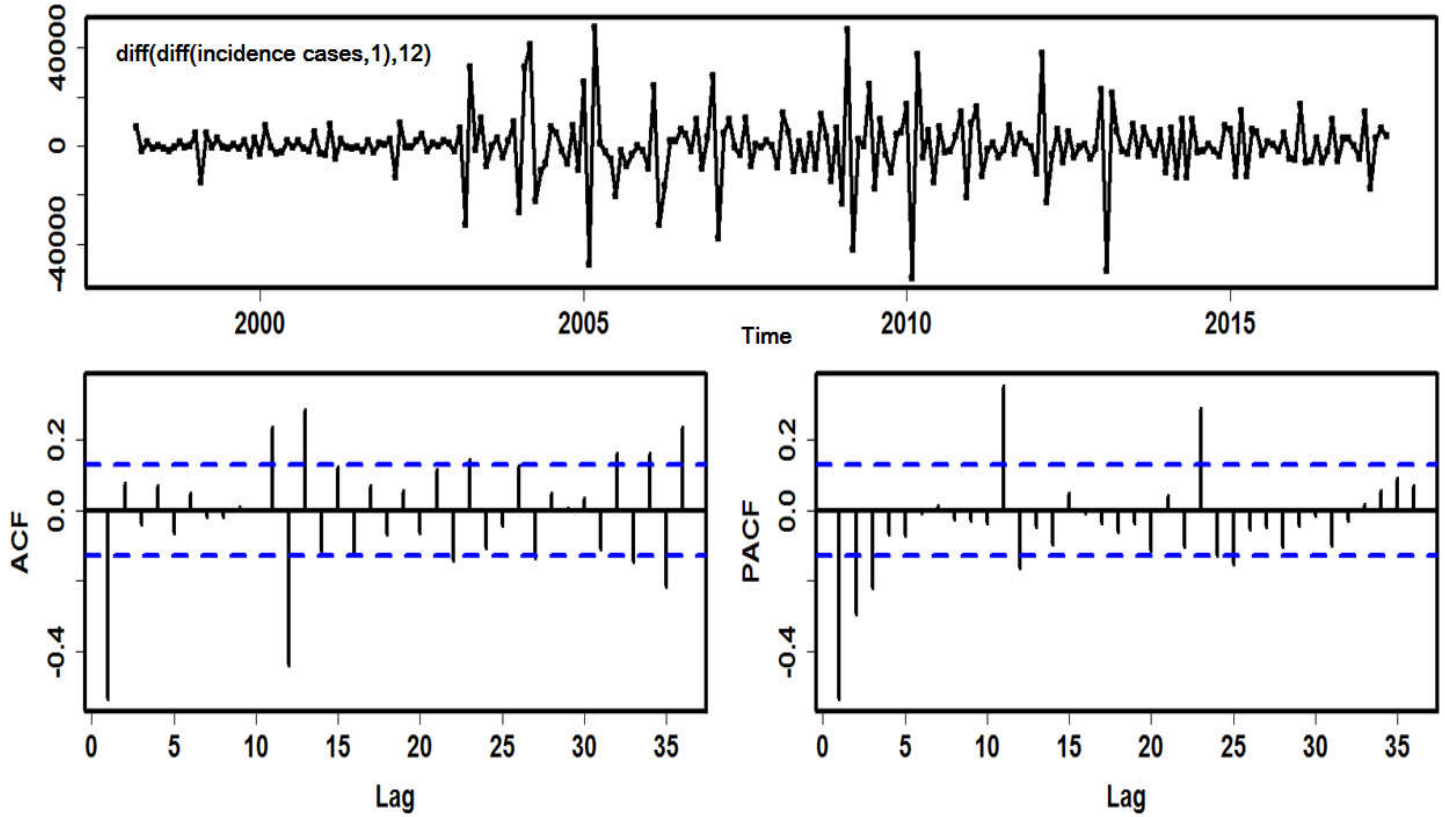

Figure S5 ACF and PACF graph with first-order seasonal difference (D=1) of monthly TB incidence series in mainland China from January 1997 to December 2016.

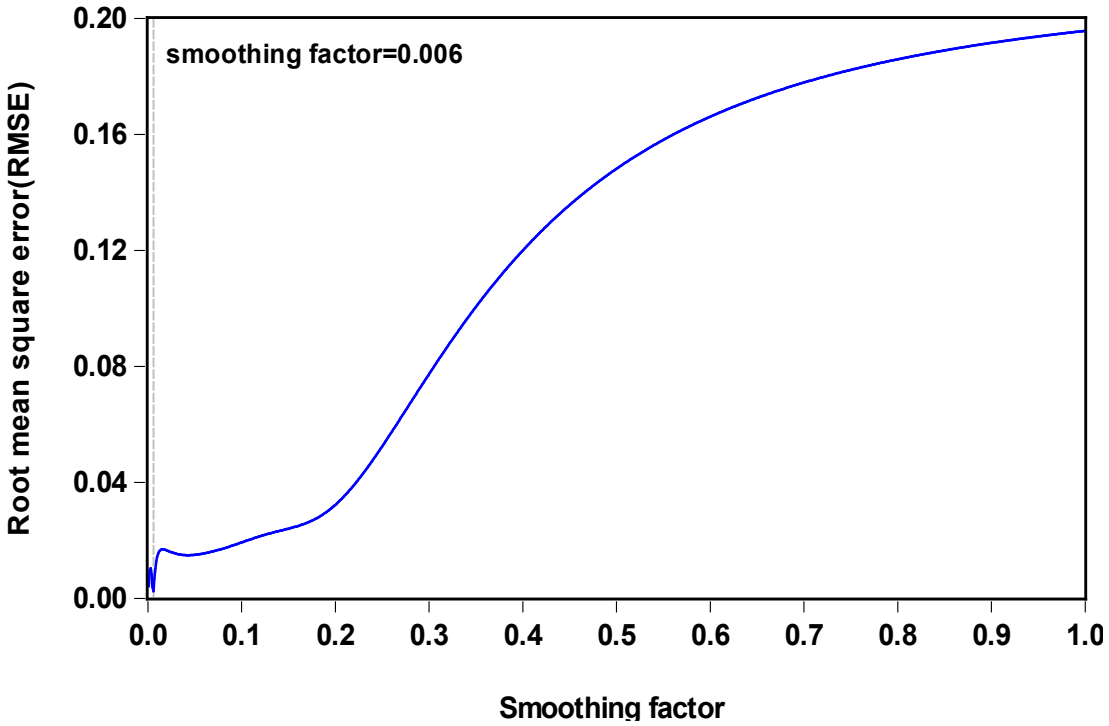

Figure S6 The root mean square error (RMSE) values corresponding to different smoothing factors for the SARIMA-GRNN combined technique. The smoothing factor between 0.001 and 1 with an interval of 0.001, it can be seen that when smoothing factor is 0.006, the lowest RMSE value is 0.0024.

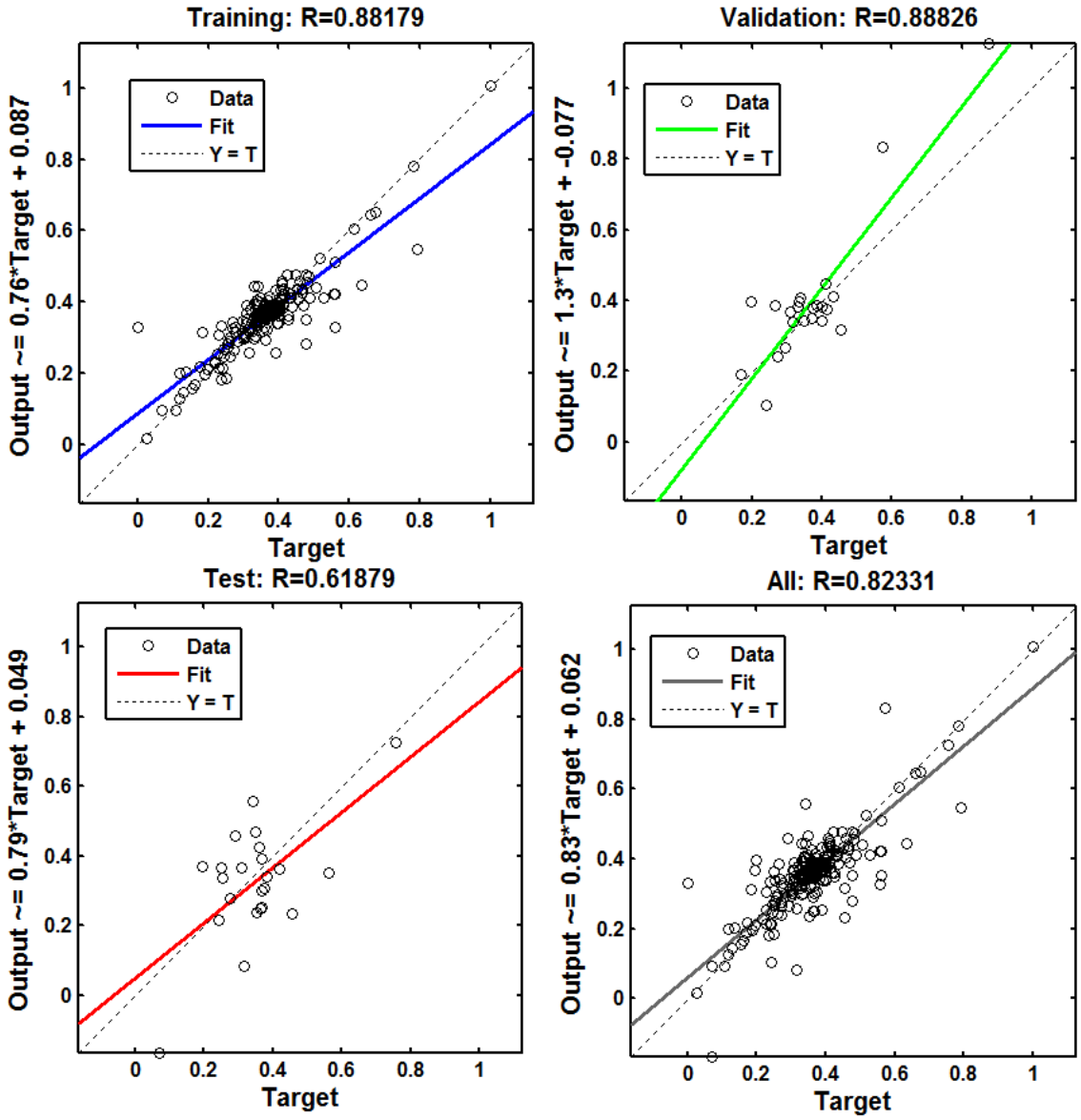

Figure S7 The linear regression between the network outputs and the corresponding targets for the traditional SARIMA-NARNN hybrid model developed based on the dataset from January 1997 to December 2016. The R value is an indication of the relationship between the outputs and targets. If R=1, this indicates that there is an exact linear relationship between outputs and targets. If R is close to zero, then there is no linear relationship between outputs and targets. Generally, all the R values are greater than 0.8, the training data indicate a good fit. However, depending on our experience and referring to the related literature[10 14], we find that it is quite difficult to simultaneously train the R value of greater than 0.8 in the training, validation, testing and all data. Thus, the overall R value of greater than 0.8 is often regarded as a good fitting in the ARIMA-NARNN hybrid model.

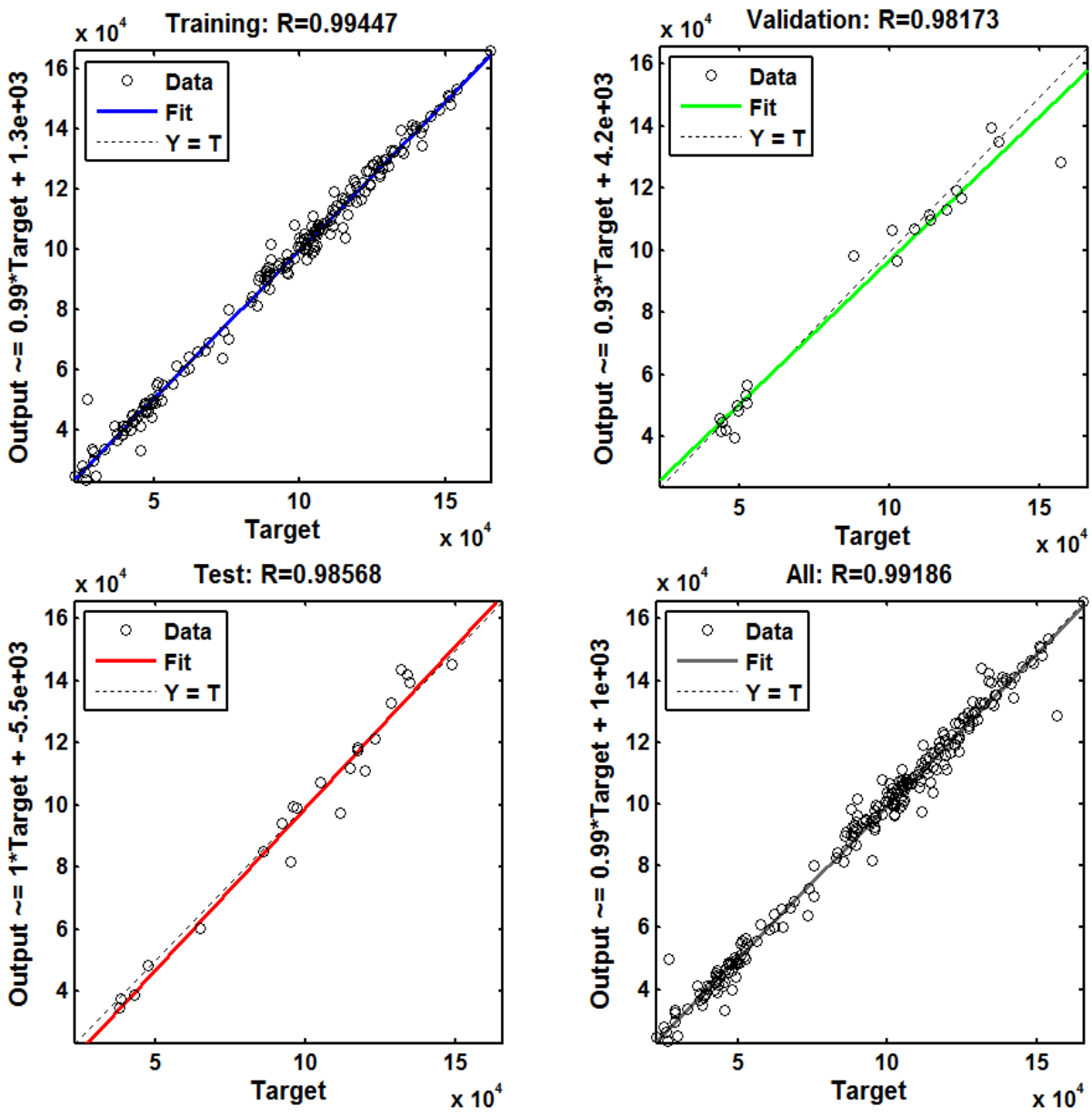

Figure S8 The linear regression between the network outputs and the corresponding targets for the SARIMA-NARNNX hybrid model developed based on the dataset from January 1997 to December 2016. The R value is an indication of the relationship between the outputs and targets. If R=1, this indicates that there is an exact linear relationship between outputs and targets. If R is close to zero, then there is no linear relationship between outputs and targets. Generally, all the R values are greater than 0.8, the training data indicate a good fit. In this SARIMA-NARNNX hybrid model, all the R values are greater than 0.98, showing a perfect modeling.

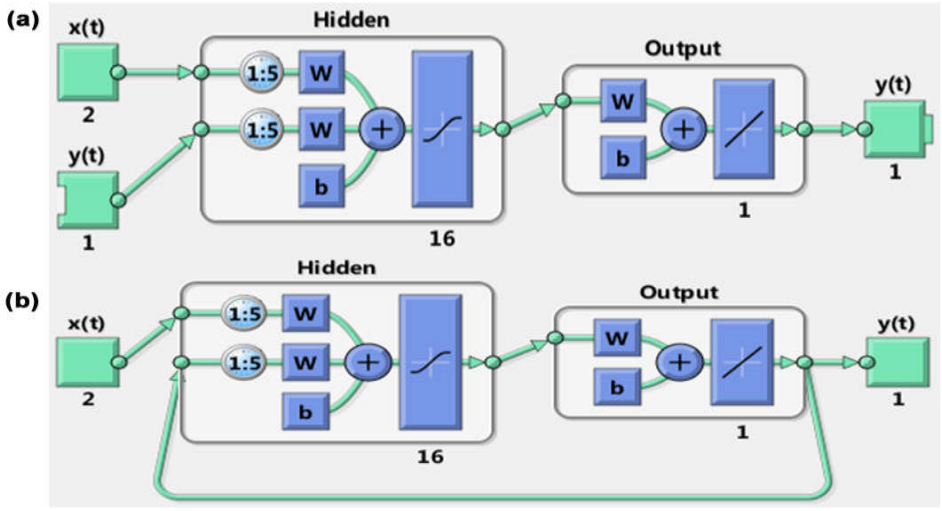

Figure S9 The layer architecture of the SARIMA-NARNNX hybrid model for testing the long-term forecasting performance to the TB incidence data. (a) The opened loop mode; (b) The closed loop mode. It can be found that the SARIMA-NARNNX hybrid model with 16 neurons and 5 delays and an output layer with 1 neuron should be considered as the best-fitting in our experiments. This is also a two-layer feed forward network, with a sigmoid transfer function in the hidden layer and a linear transfer function in the output layer. A feedback connection from the model output is taken for one input of this model. After the model is trained, this feedback connection should be closed for multi-step-ahead-forecasting. For each of these inputs, there exists a tapped delay line to store the past values. In this network, the inputs include the simulations and forecasts from the SARIMA model and the time factor, the expected outputs are the actual TB incidence values.

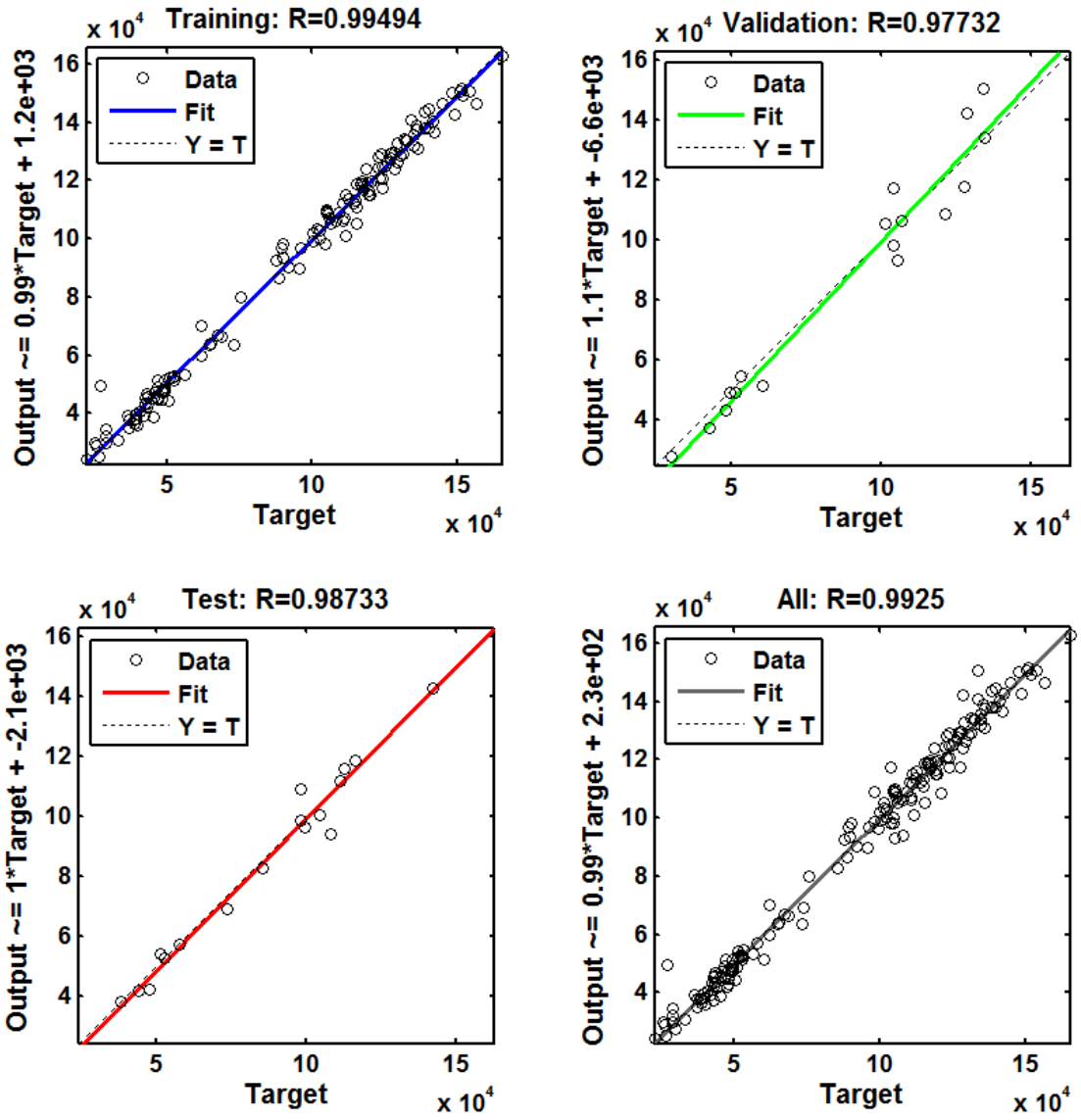

Figure S10 The linear regression plot between the network outputs and the corresponding targets for the SARIMA-NARNNX hybrid model developed based on the additional testing dataset from January 1997 to December 2012. The R value is an indication of the relationship between the outputs and targets. If R=1, this indicates that there is an exact linear relationship between outputs and targets. If R is close to zero, then there is no linear relationship between outputs and targets. Generally, all the R values are greater than 0.8, the training data indicate a good fit. In this SARIMA-NARNNX hybrid model, all the R values are greater than 0.97, displaying a perfect modeling.

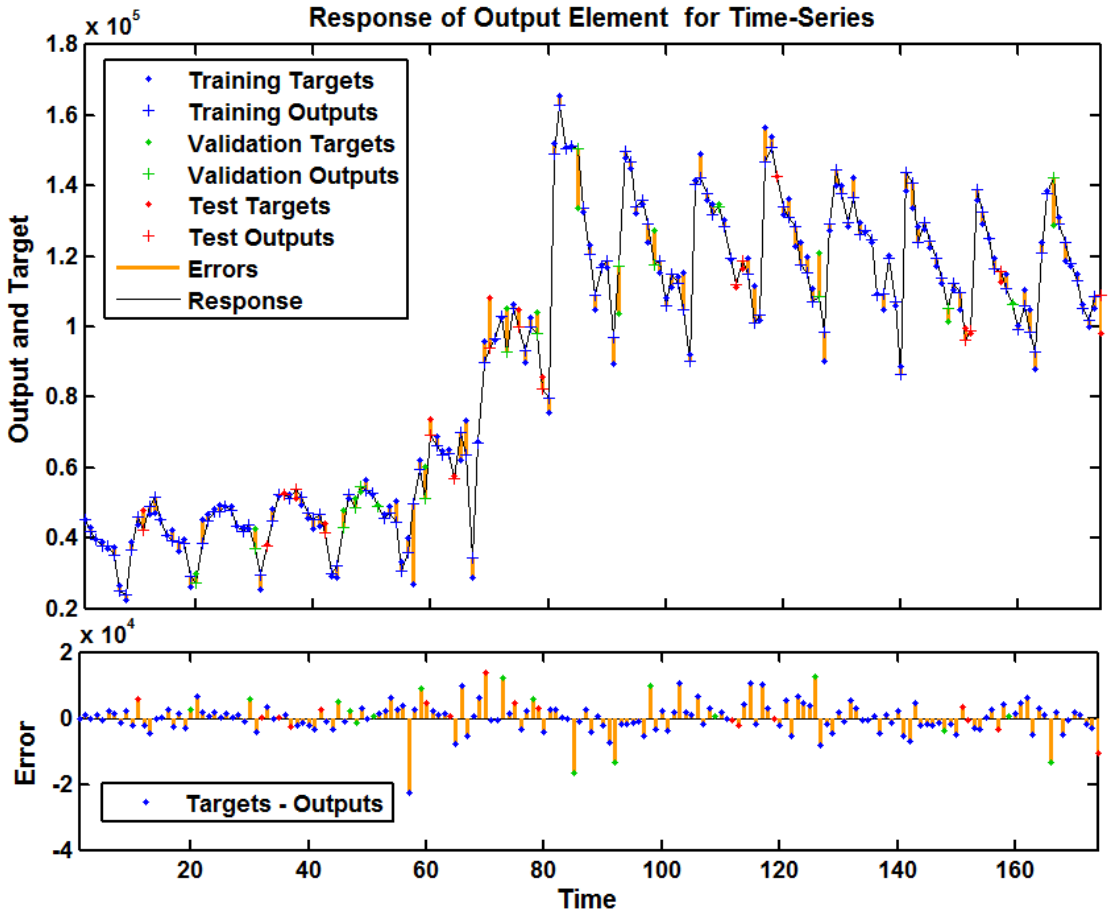

Figure S11 The corresponding time series response of outputs and targets at various time points for the SARIMA-NARNNX hybrid model developed based on the additional testing dataset from January 1997 to December 2012. This plot shows the inputs, targets and errors versus time. It also indicates which time points were selected for training, testing and validation. Overall in the training, validation and testing samples the errors were small. We confirm that the model is adequate for TB incidence data modeling.

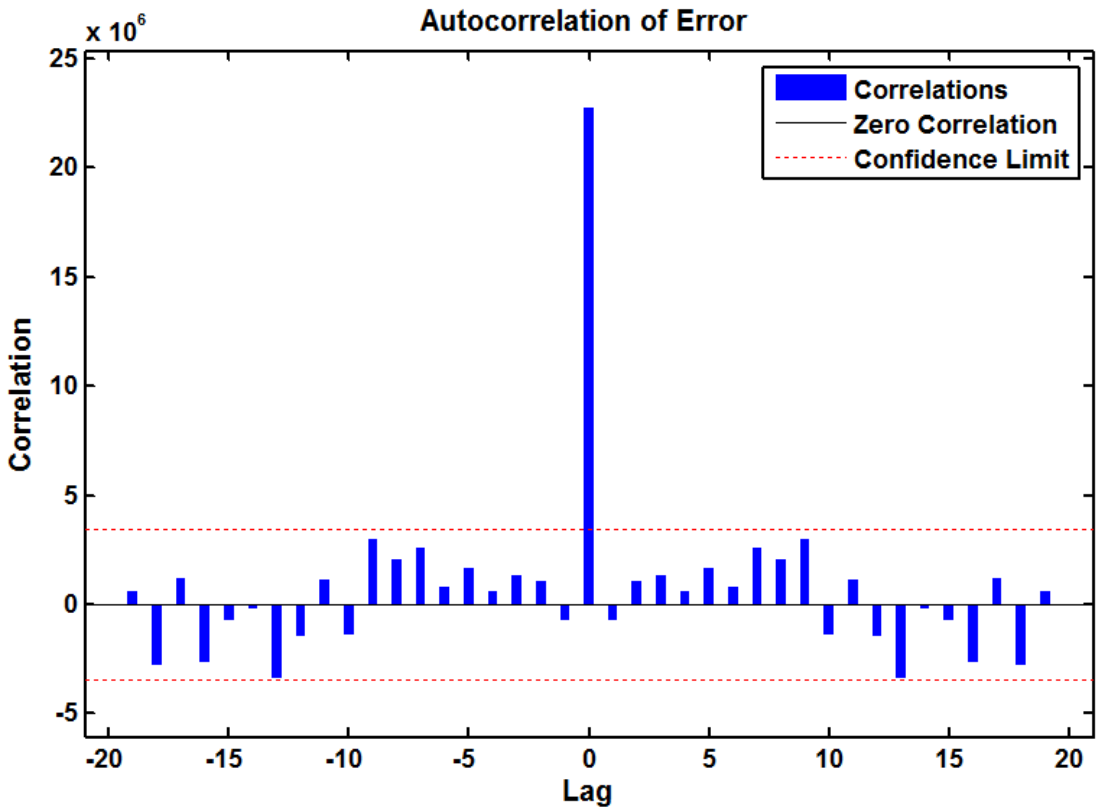

Figure S12 The resulting error autocorrelation plot for the SARIMA-NARNNX hybrid model developed based on the additional testing dataset from January 1997 to December 2012. There are no significant correlations in the residual series, so the model provides a reasonable approximation to the TB incidence data.

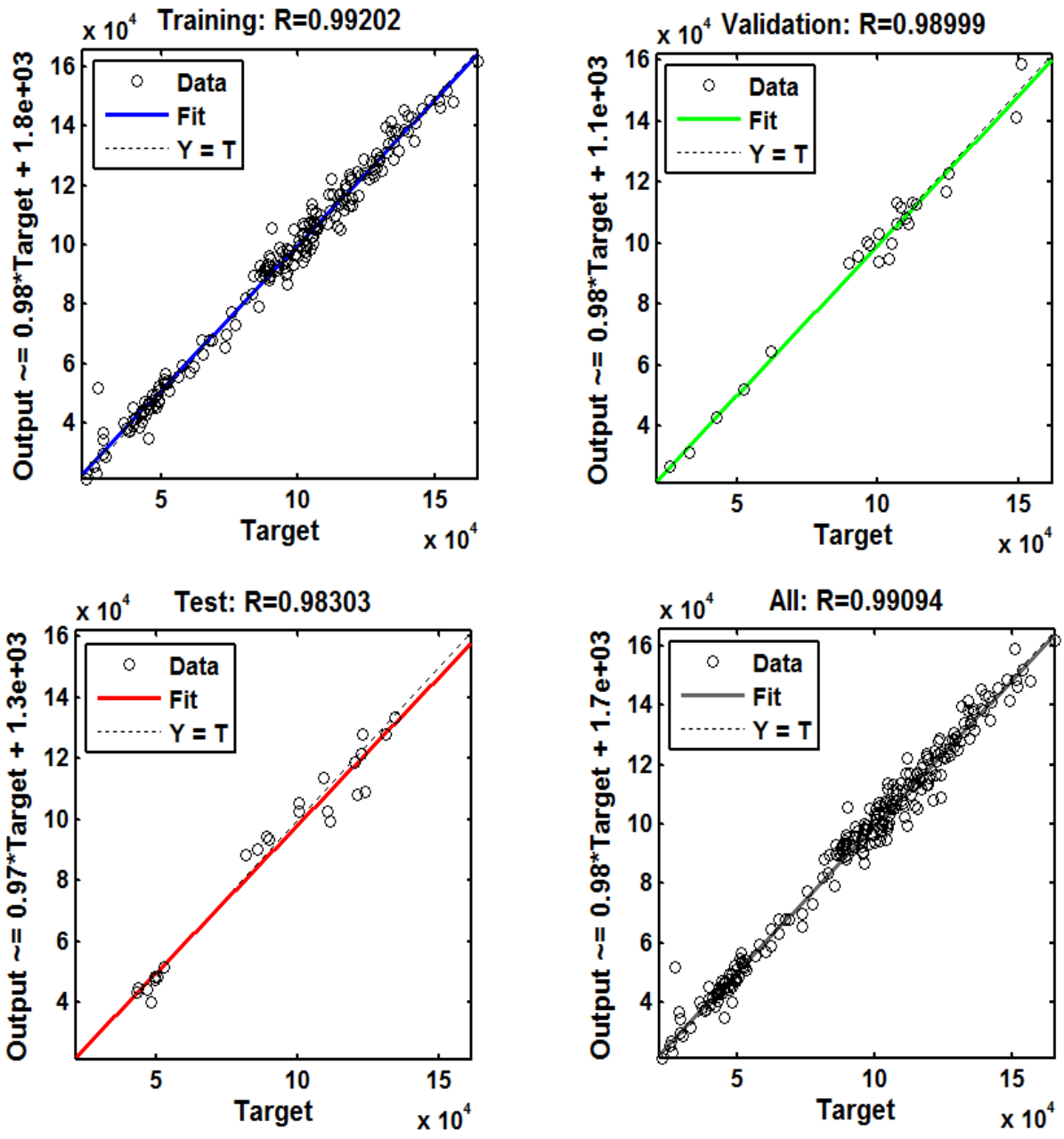

Figure S13 The linear regression between the network outputs and the corresponding targets for the SARIMA-NARNNX hybrid model developed based on the dataset from January 1997 to March 2018. The R value is an indication of the relationship between the outputs and targets. If R=1, this indicates that there is an exact linear relationship between outputs and targets. If R is close to zero, then there is no linear relationship between outputs and targets. Generally, all the R values are greater than 0.8, the training data indicate a good fit. In this SARIMA-NARNNX hybrid model, all the R values are greater than 0.98, showing a perfect modeling.

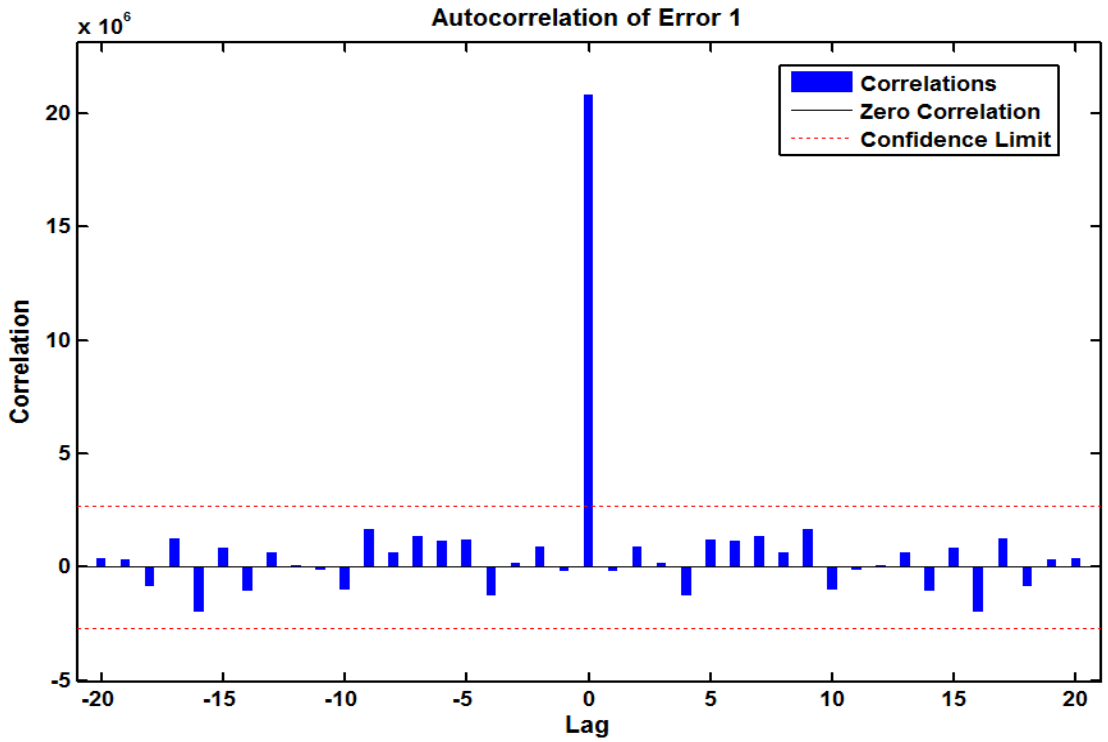

Figure S14 The resulting error autocorrelation plot for the SARIMA-NARNNX hybrid model developed based on the dataset from January 1997 to March 2018. The correlation coefficients fall within the 95% confidence limits apart from the one at zero lag, it seems that the model is suitable for TB incidence data forecasting.

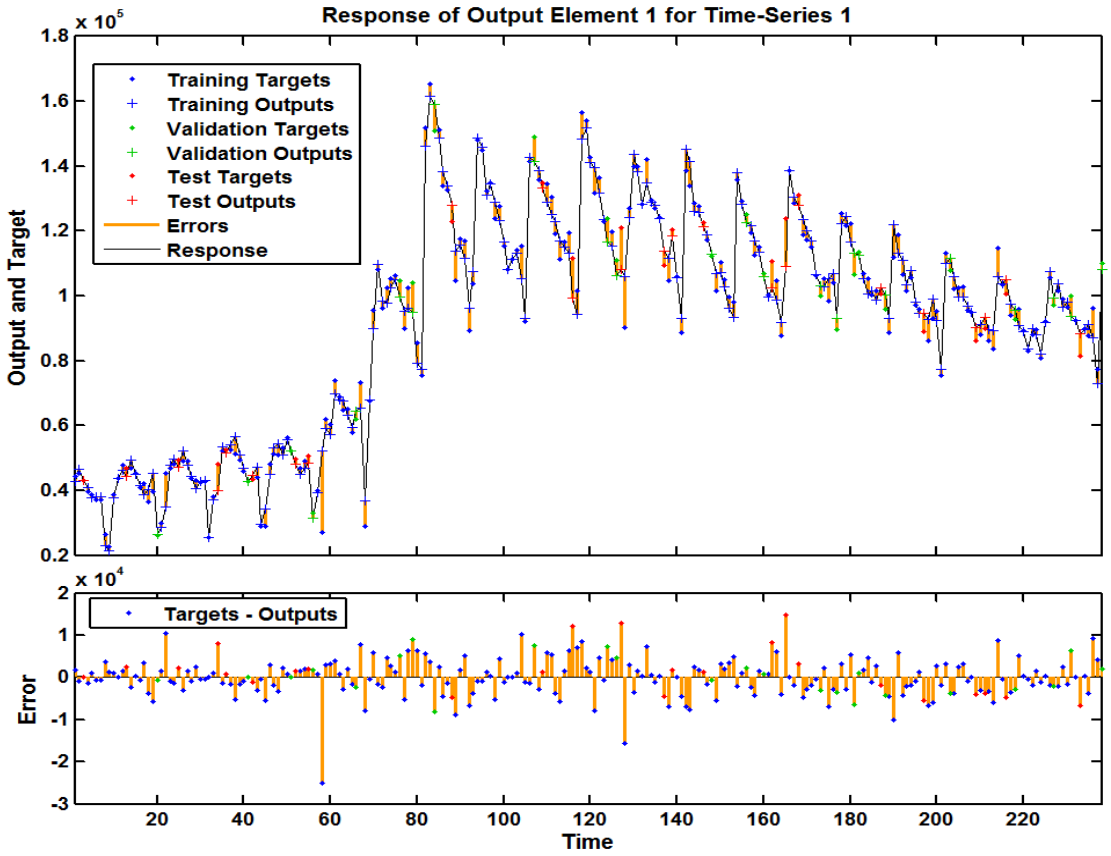

Figure S15 The corresponding time series response of outputs and targets for the SARIMA-NARNNX hybrid model developed based on the dataset from January 1997 to March 2018 at various time points. This plot shows the inputs, targets and errors versus time. It also indicates which time points were selected for training, testing and validation. Overall in the training, validation and testing samples the errors were small. We confirm that the model is adequate for TB incidence data modeling.

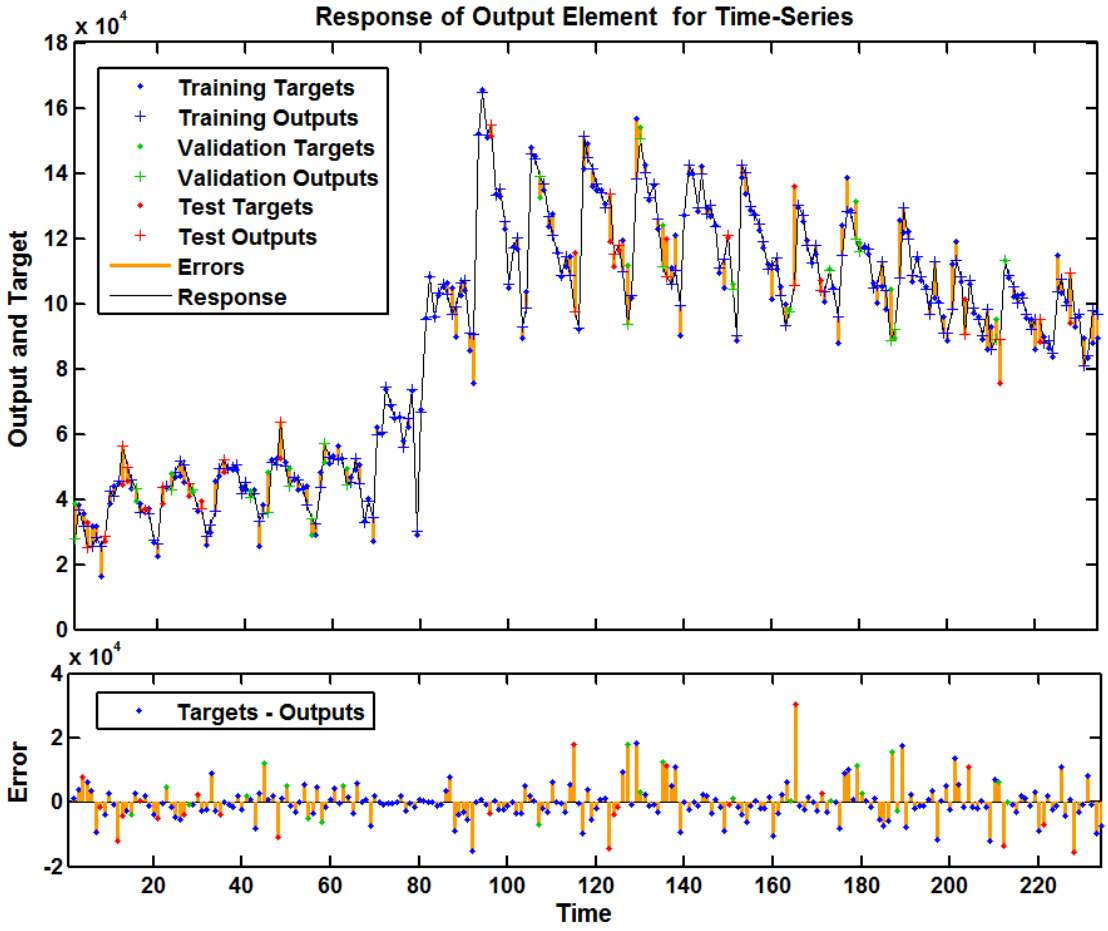

Figure S16 The corresponding time series response of outputs and targets at various time points for the basic NARNN model developed based on the dataset from January 1997 to December 2016. This plot shows the inputs, targets and errors versus time. It also indicates which time points were selected for training, testing and validation. Overall in the training, validation and testing samples the errors were small. We confirm that the model is adequate for TB incidence data modeling.

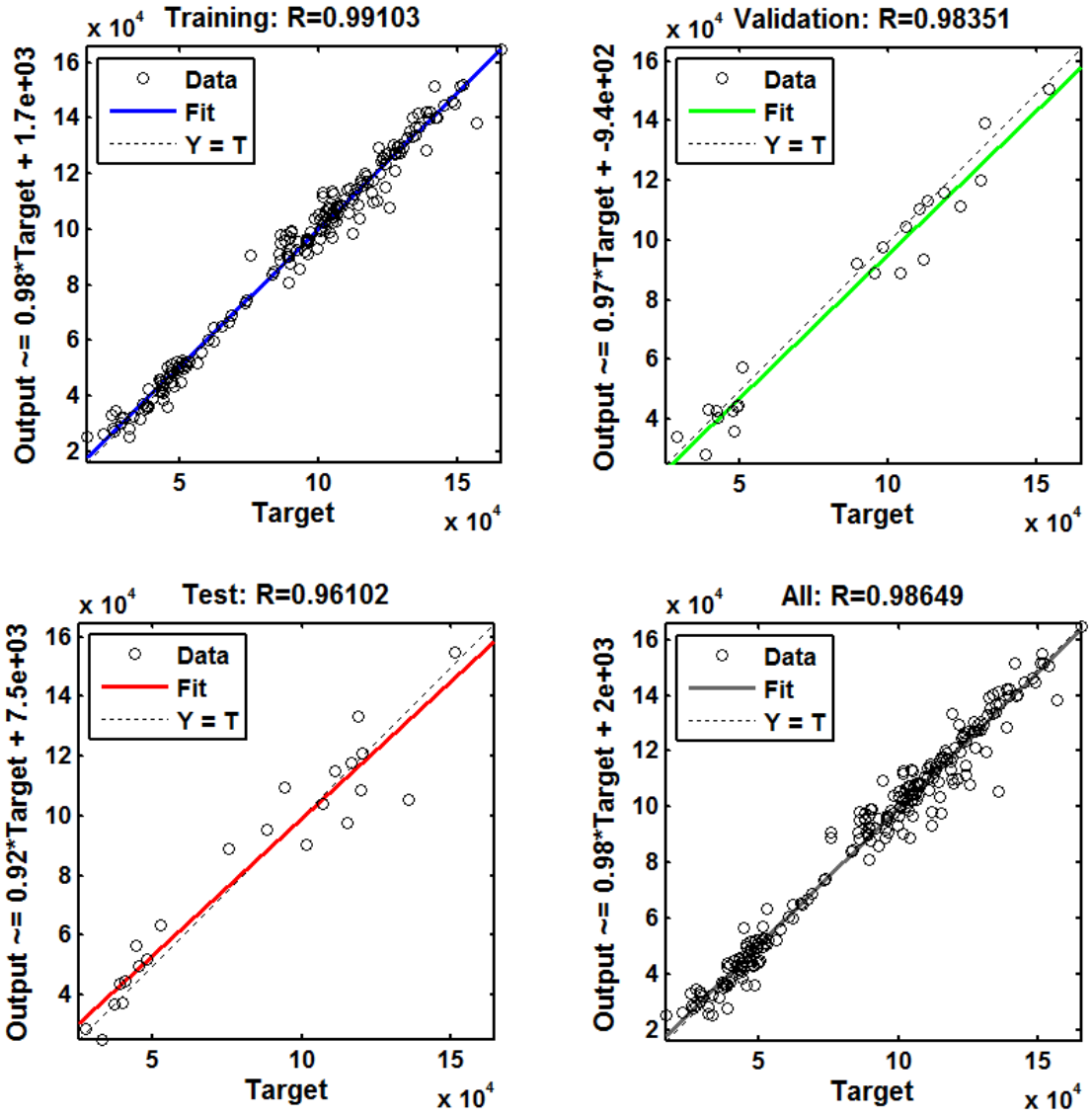

Figure S17 The linear regression plot between the network outputs and the corresponding targets for the basic NARNN model developed based on the dataset from January 1997 to December 2016. The R value is an indication of the relationship between the outputs and targets. If R=1, this indicates that there is an exact linear relationship between outputs and targets. If R is close to zero, then there is no linear relationship between outputs and targets. Generally, all the R values are greater than 0.8, the training data indicate a good fit. In this SARIMA-NARNNX hybrid model, all the R values are greater than 0.96, displaying a perfect modeling.

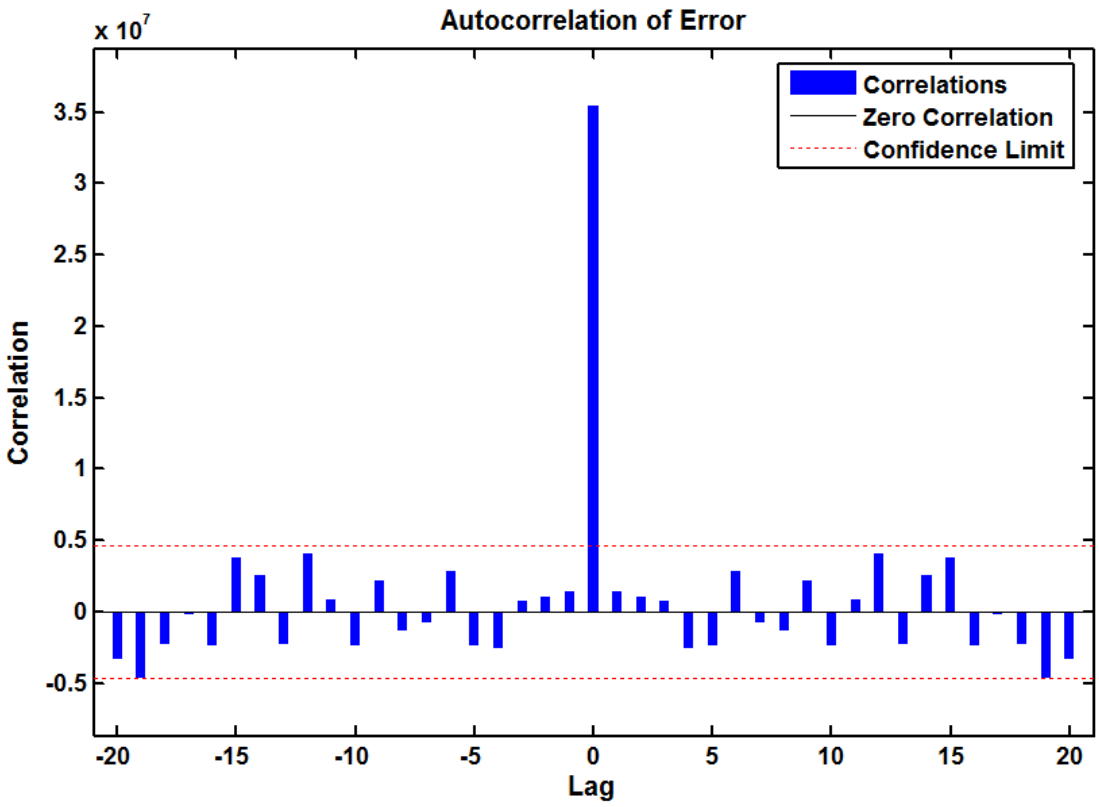

Figure S18 The resulting error autocorrelation plot for the basic NARNN model developed based on the dataset from January 1997 to December 2016. None of autocorrelation coefficients are beyond the estimated 95% confidence limits; this identified method seems to have captured the essence of the dependence in the TB incidence series.

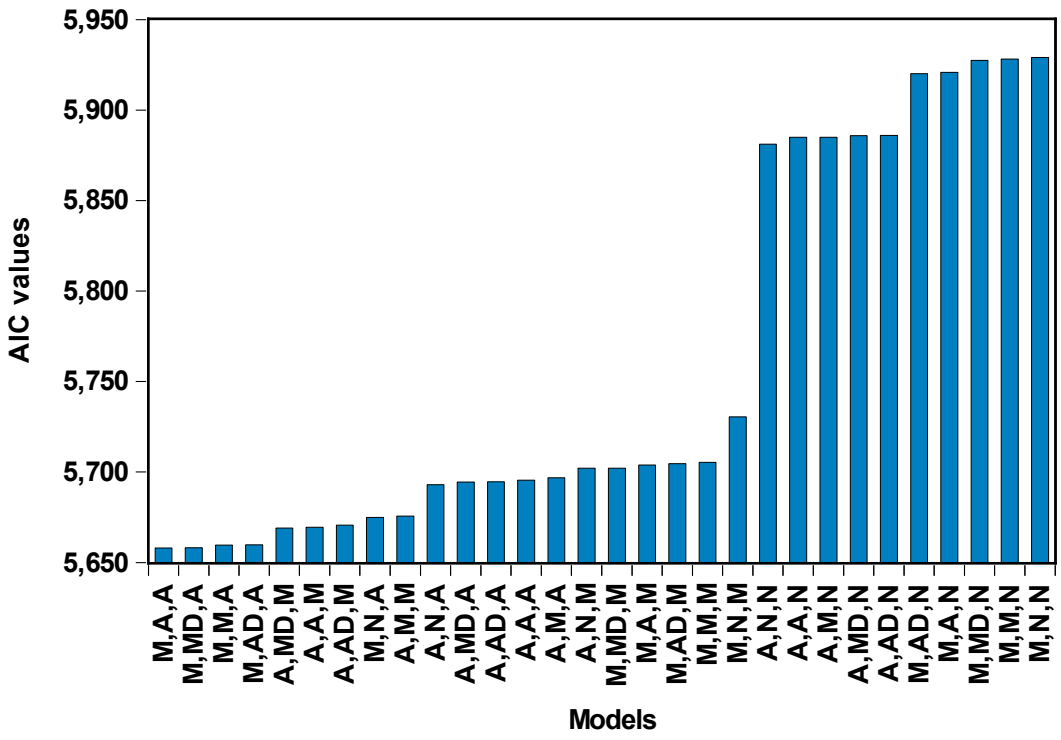

Figure S19 The AIC values of the 30 candidate models. Based on this plot, it can be found the ETS (M, A, A) with the lowest values among the 30 candidate models. Thus this model is selected to conduct out-of-sample forecasting.

References

1. Wang H, Tian CW, Wang WM, et al. Time-series analysis of tuberculosis from 2005 to 2017 in China. *Epidemiol Infect* 2018:1-5. doi: 10.1017/S0950268818001115 [published Online First: 2018/05/01]
2. Liu L, Luan RS, Yin F, et al. Predicting the incidence of hand, foot and mouth disease in Sichuan province, China using the ARIMA model. *Epidemiol Infect* 2016;144(1):144-51. doi: 10.1017/s0950268815001144 [published Online First: 2015/06/02]
3. Yu L, Zhou L, Tan L, et al. Application of a new hybrid model with seasonal auto-regressive integrated moving average (ARIMA) and nonlinear auto-regressive neural network (NARNN) in forecasting incidence cases of HFMD in Shenzhen, China. *PLoS One* 2014;9(6):e98241. doi: 10.1371/journal.pone.0098241 [published Online First: 2014/06/04]
4. Wang C, Li Y, Feng W, et al. Epidemiological Features and Forecast Model Analysis for the Morbidity of Influenza in Ningbo, China, 2006-2014. *Int J Environ Res Public Health* 2017;14(6) doi: 10.3390/ijerph14060559 [published Online First: 2017/06/08]
5. Peng Y, Yu B, Wang P, et al. Application of seasonal auto-regressive integrated moving average model in forecasting the incidence of hand-foot-mouth disease in Wuhan, China. *J Huazhong Univ Sci Technolog Med Sci* 2017;37(6):842-48. doi: 10.1007/s11596-017-1815-8
6. Xu Q, Li R, Liu Y, et al. Forecasting the Incidence of Mumps in Zibo City Based on a SARIMA Model. *Int J Environ Res Public Health* 2017;14(8):925. doi: 10.3390/ijerph14080925
7. Zhang GP. Time series forecasting using a hybrid ARIMA and neural network model. *Neurocomputing* 2003;50(1):159-75.
8. Zhang G, Huang S, Duan Q, et al. Application of a hybrid model for predicting the incidence of tuberculosis in Hubei, China. *PLoS One* 2013;8(11):e80969. doi: 10.1371/journal.pone.0080969
9. He F, Hu ZJ, Zhang WC, et al. Construction and evaluation of two computational models for predicting the

incidence of influenza in Nagasaki Prefecture, Japan. *Sci Rep* 2017;7(1):7192. doi: 10.1038/s41598-017-07475-3 [published Online First: 2017/08/05]

10. Wu W, Guo J, An S, et al. Comparison of Two Hybrid Models for Forecasting the Incidence of Hemorrhagic Fever with Renal Syndrome in Jiangsu Province, China. *PLoS One* 2015;10(8):e0135492. doi: 10.1371/journal.pone.0135492 [published Online First: 2015/08/14]
11. Specht DF. A general regression neural network. *IEEE Trans Neural Netw* 1991;2(6):568-76. doi: 10.1109/72.97934 [published Online First: 1991/01/01]
12. Zhou L, Yu L, Wang Y, et al. A hybrid model for predicting the prevalence of schistosomiasis in humans of Qianjiang City, China. *PLoS One* 2014;9(8):e104875. doi: 10.1371/journal.pone.0104875 [published Online First: 2014/08/15]
13. Zhou L, Jing X, Yu L, et al. Using a Hybrid Model to Forecast the Prevalence of Schistosomiasis in Humans. *Int J Environ Res Public Health* 2016;13(4):355. doi: 10.3390/ijerph13040355.
14. Wang KW, Deng C, Li JP, et al. Hybrid methodology for tuberculosis incidence time-series forecasting based on ARIMA and a NAR neural network. *Epidemiol Infect* 2017;145(6):1118-29. doi: 10.1017/S0950268816003216
15. Shafaei M, Adamowski J, Fakherifard A, et al. A wavelet-SARIMA-ANN hybrid model for precipitation forecasting. *Journal of Water & Land Development* 2016;28(1):27-36. doi: 10.1515/jwld-2016-0003
16. Hyndman RJ, Khandakar Y. Automatic Time Series Forecasting: The forecast Package for R. *Journal of Statistical Software* 2008;27(3):1-22.

The code used in our experiments

```
% Solve an Autoregression Problem with External Input with a NARX Neural Network
% This script assumes these variables are defined:
%       A- input time series.
%       B - feedback time series.
%       C- input time series(whole data).


X = tonndata(A,true,false);
T = tonndata(B,true,false);
% G= tonndata(C,true,false);


% Choose a Training Function
% For a list of all training functions type: help nntrain
% 'trainlm' is usually fastest.
% 'trainbr' takes longer but may be better for challenging problems.
% 'trainscg' uses less memory. NTSTOOL falls back to this in low memory situations.
trainFcn = 'trainlm';    % Levenberg-Marquardt

% Create a Nonlinear Autoregressive Network with External Input
inputDelays = 1:4;
feedbackDelays = 1:4;
hiddenLayerSize = 17;
net = narxnet(inputDelays,feedbackDelays,hiddenLayerSize,'open',trainFcn);

% Choose Input and Feedback Pre/Post-Processing Functions
% Settings for feedback input are automatically applied to feedback output
% For a list of all processing functions type: help nnprocess
% Customize input parameters at: net.inputs{i}.processParam
% Customize output parameters at: net.outputs{i}.processParam
net.inputs{1}.processFcns = {'removeconstantrows','mapminmax'};
net.inputs{2}.processFcns = {'removeconstantrows','mapminmax'};

% Prepare the Data for Training and Simulation
% The function PREPARETS prepares timeseries data for a particular network,
% shifting time by the minimum amount to fill input states and layer states.
% Using PREPARETS allows you to keep your original time series data unchanged, while easily customizing
it for networks with differing numbers of delays, with open loop or closed loop feedback modes.
[x,xi,ai,t] = preparets(net,X,{},T);

% Setup Division of Data for Training, Validation, Testing
```

```
% The function DIVIDERAND randomly assigns target values to training,
% validation and test sets during training.
% For a list of all data division functions type: help nndivide
net.divideFcn = 'dividerand';    % Divide data randomly
% The property DIVIDEMODE set to TIMESTEP means that targets are divided into training, validation and
test sets according to timesteps.
% For a list of data division modes type: help nntype_data_division_mode
net.divideMode = 'value';    % Divide up every value
net.divideParam.trainRatio = 80/100;
net.divideParam.valRatio = 10/100;
net.divideParam.testRatio = 10/100;

% Choose a Performance Function
% For a list of all performance functions type: help nnperformance
% Customize performance parameters at: net.performParam
net.performFcn = 'mse';    % Mean squared error

% Choose Plot Functions
% For a list of all plot functions type: help nnplot
% Customize plot parameters at: net.plotParam
net.plotFcns = {'plotperform','plottrainstate','plotresponse', 'plotregression', ...'ploterrcorr', 'plotinerrcorr'};

% Train the Network
[net,tr] = train(net,x,t,xi,ai);

% Test the Network
y = net(x,xi,ai);
e = gsubtract(t,y);
performance = perform(net,t,y)

% Recalculate Training, Validation and Test Performance
trainTargets = gmultiply(t,tr.trainMask);
valTargets = gmultiply(t,tr.valMask);
testTargets = gmultiply(t,tr.testMask);
trainPerformance = perform(net,trainTargets,y)
valPerformance = perform(net,valTargets,y)
testPerformance = perform(net,testTargets,y)

% View the Network
view(net)

% Plots
% Uncomment these lines to enable various plots.
%figure, plotperform(tr)
```

```
%figure, plottrainstate(tr)
%figure, plotregression(t,y)
%figure, plotresponse(t,y)
%figure, ploterrcorr(e)
%figure, plotinerrcorr(x,e)

% Closed Loop Network
% Use this network to do multi-step prediction.
% The function CLOSELOOP replaces the feedback input with a direct connection from the output layer.
netc = closeloop(net);
netc.name = [net.name ' - Closed Loop'];
view(netc)
[xc,xic,aic,tc] = preparets(netc,X,{},T);
yc = netc(xc,xic,aic);
closedLoopPerformance = perform(netc,tc,yc)
% Multi-step Prediction
% Sometimes it is useful to simulate a network in open-loop form for as long as there is known output data, and then switch to closed-loop form to perform multistep prediction while providing only the external input.
% Here all but 5 timesteps of the input series and target series are used to simulate the network in open-loop form, taking advantage of the higher accuracy that providing the target series produces:
numTimesteps = size(x,2);
knownOutputTimesteps = 1:(numTimesteps-5);
predictOutputTimesteps = (numTimesteps-4):numTimesteps;
X1 = X(:,knownOutputTimesteps);
T1 = T(:,knownOutputTimesteps);
[x1,xio,aio] = preparets(net,X1,{},T1);
[y1,xfo,afo] = net(x1,xio,aio);
% Next the the network and its final states will be converted to closed-loop form to make five predictions with only the five inputs provided.
x2 = X(1,predictOutputTimesteps);
[netc,xic,aic] = closeloop(net,xfo,afo);
[y2,xfc,afc] = netc(x2,xic,aic);
multiStepPerformance = perform(net,T(1,predictOutputTimesteps),y2)
% Alternate predictions can be made for different values of x2, or further predictions can be made by continuing simulation with additional external inputs and the last closed-loop states xfc and afc.


% Step-Ahead Prediction Network
% For some applications it helps to get the prediction a timestep early.
% The original network returns predicted y(t+1) at the same time it is given y(t+1).
% For some applications such as decision making, it would help to have predicted y(t+1) once y(t) is available, but before the actual y(t+1) occurs.
% The network can be made to return its output a timestep early by removing one delay so that its minimal tap delay is now 0 instead of 1.
```

```
% The new network returns the same outputs as the original network, but outputs are shifted left one timestep.
nets = removedelay(net);
nets.name = [net.name ' - Predict One Step Ahead'];
view(nets)
[xs,xis,ais,ts] = preparets(nets,X,{},T);
ys = nets(xs,xis,ais);
stepAheadPerformance = perform(nets,ts,ys)

% Deployment
% Change the (false) values to (true) to enable the following code blocks.
% See the help for each generation function for more information.
if (false)
% Generate MATLAB function for neural network for application deployment in MATLAB scripts or with
   MATLAB Compiler and Builder tools, or simply to examine the calculations your trained neural network
   performs.
   genFunction(net,'myNeuralNetworkFunction');
   y = myNeuralNetworkFunction(x,xi,ai);
end
if (false)
   % Generate a matrix-only MATLAB function for neural network code generation with MATLAB Coder
tools.
   genFunction(net,'myNeuralNetworkFunction','MatrixOnly','yes');
   x1 = cell2mat(x(1,:));
   x2 = cell2mat(x(2,:));
   xi1 = cell2mat(xi(1,:));
   xi2 = cell2mat(xi(2,:));
   y = myNeuralNetworkFunction(x1,x2,xi1,xi2);
end
if (false)
   % Generate a Simulink diagram for simulation or deployment with.
   % Simulink Coder tools.
   gensim(net);
end
```