Methods and Applications

Development and Comparison of Time Series Models in Predicting Severe Fever with Thrombocytopenia Syndrome Cases — Hubei Province, China, 2013–2020

Zixu Wang[1,2,&]; Jinwei Zhang[3,&]; Wenyi Zhang[4,&]; Nianhong Lu[1]; Qiong Chen[1]; Junhu Wang[1]; Yingqing Mao[1]; Haiming Yi[1]; Yixin Ge[1]; Hongming Wang[1]; Chao Chen[1]; Wei Guo[1]; Xin Qi[5]; Yuexi Li[7,#]; Ming Yue[6,#]; Yong Qi[1,#]

ABSTRACT

Introduction: Severe fever with thrombocytopenia syndrome (SFTS) is an emerging infectious disease caused by the SFTS virus, which has a high mortality rate. Predicting the number of SFTS cases is essential for early outbreak warning and can offer valuable insights for establishing prevention and control measures.

Methods: In this study, data on monthly SFTS cases in Hubei Province, China, from 2013 to 2020 were collected. Various time series models based on seasonal auto-regressive integrated moving average (SARIMA), Prophet, eXtreme Gradient Boosting (XGBoost), and long short-term memory (LSTM) were developed using these historical data to predict SFTS cases. The established models were evaluated and compared using mean absolute error (MAE) and root mean squared error (RMSE).

Results: Four models were developed and performed well in predicting the trend of SFTS cases. The XGBoost model outperformed the others, yielding the closest fit to the actual case numbers and exhibiting the smallest MAE (2.54) and RMSE (2.89) in capturing the seasonal trend and predicting the monthly number of SFTS cases in Hubei Province.

Conclusion: The developed XGBoost model represents a promising and valuable tool for SFTS prediction and early warning in Hubei Province, China.

Severe fever with thrombocytopenia syndrome (SFTS) is an emerging infectious disease caused by the SFTS virus. Since the first confirmed case was reported in 2009 (*1*), most cases have been reported in northern and central China (*2*–*3*). The number of reported SFTS cases continues to rise, and the areas affected by the disease are expanding (*4*–*5*). Due to its high case-fatality rate and the possibility of pandemic spread, the World Health Organization included SFTS on its list of the top 10 infectious diseases needing immediate research attention (*6*). Although China has established a valuable infectious disease surveillance system to monitor and assess disease burden, the system cannot predict future trends or provide early warnings of outbreaks. Furthermore, the monitoring data obtained are often delayed. Consequently, there is an urgent need for a model to predict the number of SFTS cases in endemic regions.

As a tick-borne disease, the incidence of SFTS exhibits distinct time-series characteristics, referring to data points collected and recorded chronologically, typically at regular intervals. Specialized time-series analysis techniques are likely suitable for effectively modeling and forecasting SFTS incidence.

In this study, we utilized various time series algorithms based on historical data to predict the occurrence of SFTS in Hubei Province, one of the first provinces to report SFTS cases and a province with a high incidence of the disease in China (*7*). Predicting the number of SFTS cases in this region will provide important insights for developing prevention and control interventions.

METHODS

Data Collection

The monthly number of SFTS cases in Hubei Province was obtained from the Public Health Science Data Center (https://www.phsciencedata.cn/Share/). Data reported between January 2013 and December 2019 (84 data points total) were used for model training and development, while the remaining data from January to December 2020 (12 data points total)

were used for external validity assessment.

Model Constructions

SARIMA model: Seasonal autoregressive integrated moving average (SARIMA) is an extension of autoregressive integrated moving average (ARIMA) that requires selecting hyperparameters for both the trend and seasonal elements of the time series. The formula for SARIMA is as follows:

$$(1-B)^d\left(1-B^s\right)^D Y_t = \theta_0 + \frac{\theta(B)\theta_s(B^s)}{\phi(B)\phi_s(B^s)}\varepsilon_t \quad (1)$$

where Y_t refers to the value of the time series at time t, θ_0 is constant, ε_t is the white noise value at period t, and the parameters d and D represent the difference number and seasonal difference number, respectively. B is the backshift operator, $\phi(B)$ is the autoregressive operator, and $\theta(B)$ is the moving average operator. $\phi_s(B^s)$ and $\theta_s(B^s)$are the seasonal operators.

Prophet model: The Prophet model provides a versatile treatment of trends, seasonality, and holiday effects. The trend component, g(t), is engineered to capture non-periodic changes in the time series. The foundational equation of the Prophet model is expressed as:

$$y_t = g(t) + s(t) + h(t) + \varepsilon_t \quad (2)$$

Where y_t denotes the predicted value at time t, $s(t)$ is the seasonality component, $h(t)$ represents the impact of holidays or specific events on the time series, and ε_t is the error term accounting for aspects of the data not explained by the model.

XGBoost: eXtreme Gradient Boosting (XGBoost) iteratively constructs a series of short, basic decision trees. For a dataset with n examples and m features, a tree ensemble model in XGBoost predicts the output using K additive functions:

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i), f_k \in F \quad (3)$$

Here, $\hat{y}$ represents the predicted value for the ith sample, f_k is a function corresponding to the kth tree, and F denotes the space of regression tree functions, with x_i being the feature vector for the ith sample.

To learn the set of functions used in the model, XGBoost minimizes the following regularized objective:

$$L(\varphi) = \sum_{i=1}^{n} l\left(y_i, \hat{y}_i\right) + \sum_{k=1}^{K} \Omega\left(f_k\right) \quad (4)$$

In this equation, $l\left(y_i, \hat{y}_i\right)$ is the loss function that quantifies the error between the observed and predicted data, and $\Omega(f_k)$ is the regularization term that helps smooth the learned weights. This smoothing prevents overfitting and encourages the model to select simpler, more predictive functions. The regularization term is defined as:

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda\sum_{j=1}^{T} w_j^2 \quad (5)$$

Where γ and λ are regularization parameters, T is the number of leaves in the tree, and w_j represents the score on each leaf.

Bayesian optimization was used to select the optimal hyperparameters, with the objective function defined to maximize R^2.

LSTM Networks: Long short-term memory (LSTM) networks incorporate a cell state that acts as a form of memory. The key feature of LSTM networks lies in their gating mechanism, which comprises three types of gates.

The input gate regulates the flow of new information into the cell state through a two-step process. First, a sigmoid function determines the necessary update values, represented by the equation:

$$i_t = \sigma\left(W_{x_i}x_t + W_h h_{t-1} + b_i\right) \quad (6)$$

The second step employs a tanh function to generate a vector of new candidate values that may be added to the state, given by:

$$C_t = tanh\left(W_{x_C}x_t + W_{h_C}h_{t-1} + b_C\right) \quad (7)$$

Here, it is the activation of the input gate, and C_t is the candidate vector for the cell state update.

The forget gate determines which information from the cell state to retain or discard. It uses a sigmoid function to evaluate the importance of existing information in the cell state, defined by:

$$f_t = \sigma\left(W_{x_f}x_t + W_{h_f}h_{t-1} + b_f\right) \quad (8)$$

The activation vector f_t indicates the extent to which past information should be forgotten or retained.

The output gate regulates the information sent to the subsequent layer. This gate functions in two stages. First, a sigmoid function determines which parts of the cell state are outputted, as shown by:

$$o_t = \sigma\left(W_{x_0}x_t + W_{h_g}h_{t-1} + b_0\right) \quad (9)$$

Then, the final output is calculated by multiplying this activation o_t with the tanh of the cell state, resulting in:

$$h_t = o_t \times \tanh(C_t) \quad (10)$$

The output vector h_t represents the information transmitted to subsequent layers or units in the network.

In the models described above, clipping, a data post-processing technique, was used to address unrealistic negative values in the results. A detailed explanation of each model is provided in Supplementary Material (available at https://weekly.chinacdc.cn/).

Performance Evaluations

The predictive performance of the models was assessed using two indices: mean absolute error (MAE) and root mean squared error (RMSE), defined as follows:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \quad (11)$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \quad (12)$$

Software

Descriptive statistics and time series modeling were conducted using Python (version 3.7; Python Software Foundation, Beaverton, OR, USA). The SARIMA, Prophet, XGBoost, and LSTM models were implemented using the statsmodels, fbprophet, scikit-learn, and Keras packages, respectively. A $P<0.05$ was considered statistically significant.

RESULTS

General Analysis

A total of 1,695 SFTS cases were reported in Hubei Province from January 2013 to December 2020, exhibiting clear seasonal characteristics. More cases were reported from April to August each year and fewer from December to February of the following year. Interestingly, a prominent peak occurred in June and a smaller peak in October (Figure 1).

Models

In the SARIMA model construction, the augmented Dickey-Fuller (ADF) test indicated that the time series data were unstable with a $P>0.05$ (Dickey-Fuller=−1.339, $P=0.611$). After the first difference, the original sequence tended to become stationary. The parameters p and q were determined from the autocorrelation function (ACF) and partial autocorrelation function (PACF) plots (Figure 2), and the final model parameters were determined as SARIMA (1,1,1), (0,1,1)12 based on the minimum Akaike information criterion (AIC) (AIC=543.302). All parameters were significant with $P<0.01$ (Supplementary Table S1, available at https://weekly.chinacdc.cn/). The residual autocorrelation test (Ljung-box test) indicated that the residual was not significantly different from a white noise series (Q-statistic=0.32, $P=0.57$), suggesting that the model was acceptable.

The optimized parameters of the other three models are summarized in Table 1.

Model Evaluation and Comparison

The trained SARIMA, Prophet, XGBoost, and LSTM models were used to predict the number of reported SFTS cases in 2020 and were compared with real external validation data (Figure 3). All four models performed well in predicting the trends of SFTS cases;

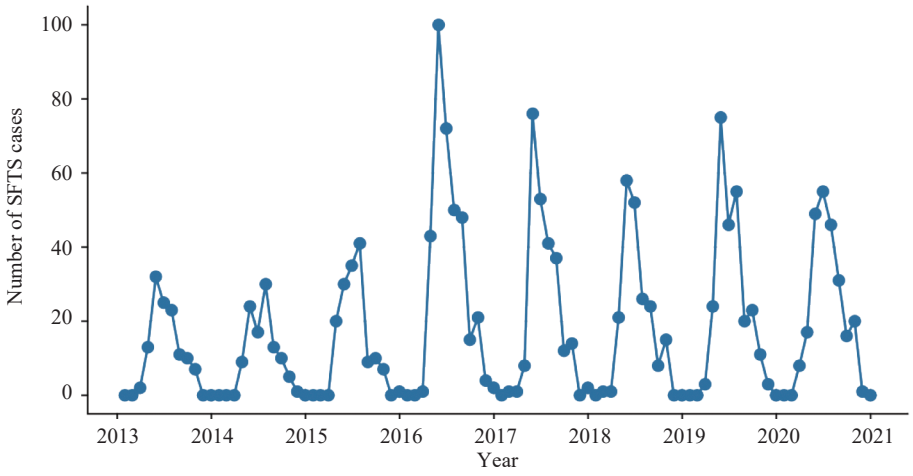

FIGURE 1. Trends of the actual number of SFTS cases from January 2013 to December 2020 in Hubei Province, China.
Abbreviation: SFTS=Severe Fever with Thrombocytopenia Syndrome.

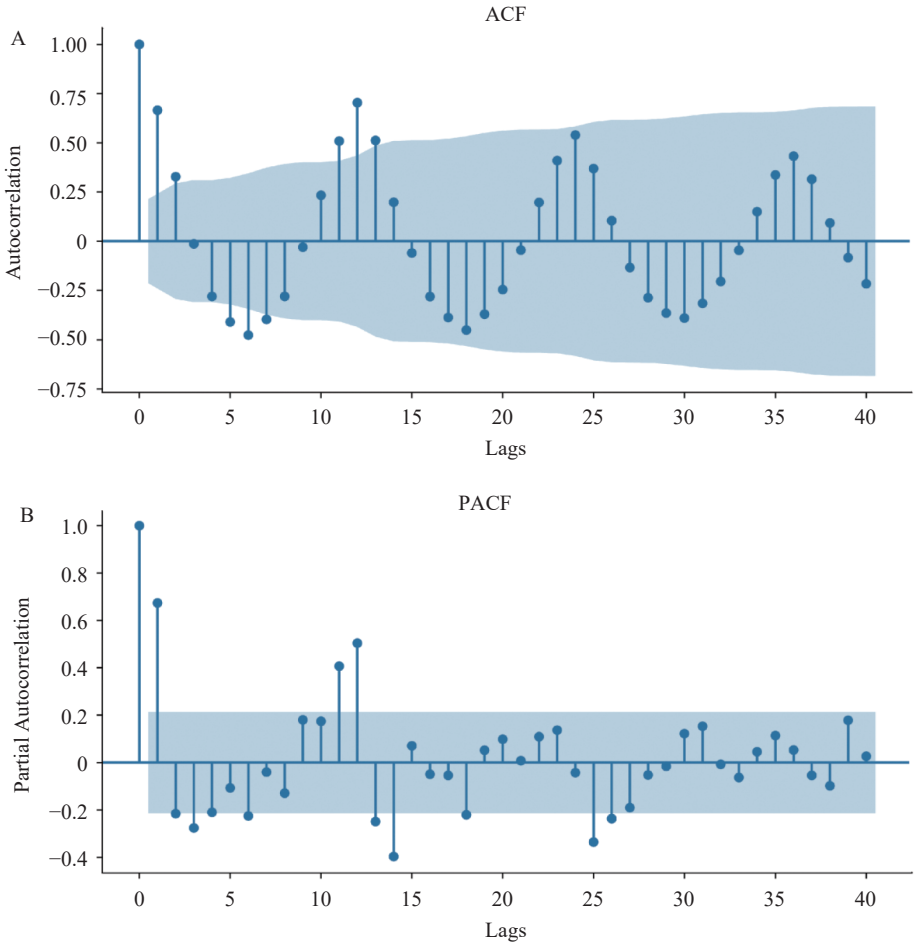

FIGURE 2. ACF and PACF charts after the first-order difference. (A) ACF; (B) PACF.
Abbreviation: ACF=autocorrelation function; PACF=partial autocorrelation function

however, the XGBoost model yielded the closest fit to the actual case numbers (Figure 3). The prediction performances of the models were then compared using error indices, including MAE and RMSE. As shown in Supplementary Table S2 (available at https://weekly.chinacdc.cn/), the MAE and RMSE of XGBoost were lower than those of the other three models, indicating that XGBoost performed best in predicting SFTS cases, followed by Prophet, LSTM, and SARIMA, respectively.

DISCUSSION

Previous studies have conducted multivariate modeling analyses to examine the risk factors associated with SFTS incidence in Hubei Province (*7*). However, to our knowledge, this study is the first to construct predictive models for the number of SFTS cases in Hubei Province.

In this study, we developed four models based on different algorithms to predict SFTS cases in Hubei Province. Each algorithm has advantages and disadvantages. SARIMA models are relatively simple, linear models capable of uncovering dynamic relationships between historical and predicted data. However, they require the original sequence to be stable before modeling and struggle to capture nonlinear relationships in the data. This limitation becomes evident when abrupt changes or nonlinear trends are present in the data, as SARIMA is less flexible in adapting to these complexities.

In contrast, the Prophet model does not require consideration of time series data stationarity and offers greater parameter adjustability, enhancing its flexibility. This model can automatically detect and handle outliers in the data, making it suitable for noisy or irregular datasets. It demonstrates rapid computation, making it appropriate for large datasets and real-time forecasting applications. Prophet has shown excellent performance in predicting various

TABLE 1. Parameters of the optimized Prophet, XGBoost, and LSTM models.

Models	Parameters	Values
Prophet	Growth	linear
	Seasonality mode	additive
	Interval width	0.8
	Changepoints	24
	Changepoint prior scale	0.3
XGBoost	Min_child_weight	9
	Estimators	54
	Learning rate	0.407
	Max depth	6
LSTM	No. of neurons	201
	Layers	1
	Learning rate	0.003
	Activation	tanh
	Recurrent activation	sigmoid
	Dropout	0
	Loss	mse
	Optimizer	Adam
	Batch size	1
	Epochs	100

infectious diseases, including coronavirus disease 2019 (COVID-19) and hand, foot, and mouth disease (*8–10*).

XGBoost displays robustness in handling nonlinear time series data, excelling at forecasting extreme values. This is likely due to its ability to model complex relationships through boosting. LSTM features a memory unit for storing information across time steps, which is advantageous for modeling long-term dependencies. It accommodates varying input and output dimensions for both univariate and multivariate data. However, LSTM may struggle with predicting sudden changes due to its reliance on past data patterns, as seen in our study with the surge in cases from April to May 2020.

All four models performed well in predicting SFTS cases and exhibited similar trends to the actual case counts. XGBoost demonstrated the closest predictions to the actual values, with the lowest MAE and RMSE values. Notably, SARIMA, Prophet, and LSTM did not accurately predict the May case counts (Figure 3). Additionally, SARIMA and Prophet failed to predict the peak month, possibly due to the sharp increase in actual cases from April to May 2020, which may have introduced challenges in predicting such volatile data. XGBoost displayed excellent performance in

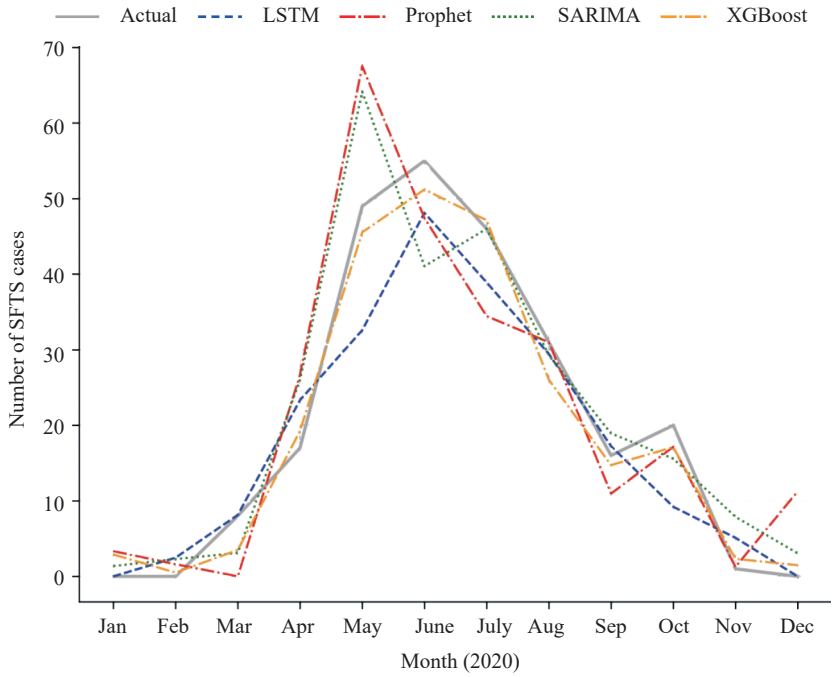

FIGURE 3. Comparison of the actual SFTS cases with the predicted cases from January to December 2020 by the four models.
Abbreviation: LSTM=long short-term memory; SFTS=severe fever with thrombocytopenia syndrome; SARIMA=seasonal auto-regressive integrated moving average; XGBoost=eXtreme Gradient Boosting.

forecasting extreme values (such as the prominent June peak and the smaller October peak) and capturing the overall trend.

Considering that meteorological, geographical, and human activity factors are considered risk factors for SFTS (*11*–*13*), incorporating additional related external variables could enhance the predictive model's performance. Furthermore, studies have indicated that combining linear and nonlinear models may yield superior predictive performance compared to single models, such as SARIMA-Prophet (*14*) and SARIMA-LSTM (*15*), representing a potential avenue for improvement.

In addition, the best model in the present study was developed based on data from Hubei Province, so it may not be suitable for other regions. This limits the model's general applicability. However, the study provides a feasible scheme for other regions to predict the disease.

In conclusion, we established and evaluated various time series models. The XGBoost model demonstrated the best predictive performance for forecasting monthly confirmed SFTS cases in Hubei Province. This model holds promise for providing valuable information and data for the early assessment of potential SFTS risks, which is crucial for developing early warning systems and formulating effective prevention and control measures.

Conflicts of interest: No conflicts of interest.

Funding: Supported by Medical Science and Technology Projects (JK2023002), National Natural Science Foundation of China (82273691), and Open Research Fund Program of the State Key Laboratory of Pathogen and Biosecurity (No. SKLPBS2137).

doi: 10.46234/ccdcw2024.200

[#] Corresponding authors: Yuexi Li, liyxi2007@126.com; Ming Yue, njym08@163.com; Yong Qi, qslark@126.com.

[1] Huadong Research Institute for Medicine and Biotechniques, Nanjing City, Jiangsu Province, China; [2] Bengbu Medical College, Bengbu City, Anhui Province, China; [3] Department of Anesthesiology, Nanjing Drum Tower Hospital, Affiliated Hospital of Medical School, Nanjing University, Nanjing City, Jiangsu Province, China; [4] Chinese PLA Center for Disease Control and Prevention, Beijing, China; [5] The Second People's Hospital of Yiyuan County, Zibo City, Shandong Province, China; [6] Department of Infectious Diseases, The First Affiliated Hospital of Nanjing Medical University, Nanjing City, Jiangsu Province, China; [7] School of Public Health, Nanjing Medical University, Nanjing City, Jiangsu Province, China.
[&] Joint first authors.

Submitted: June 21, 2023; Accepted: September 06, 2024

REFERENCES

1. Yu XJ, Liang MF, Zhang SY, Liu Y, Li JD, Sun YL, et al. Fever with thrombocytopenia associated with a novel bunyavirus in China. N Engl J Med 2011;364(16):1523 – 32. https://doi.org/10.1056/NEJMoa1010095.
2. Li H, Lu QB, Xing B, Zhang SF, Liu K, Du J, et al. Epidemiological and clinical features of laboratory-diagnosed severe fever with thrombocytopenia syndrome in China, 2011-17: a prospective observational study. Lancet Infect Dis 2018;18(10):1127 – 37. https://doi.org/10.1016/S1473-3099(18)30293-7.
3. Sun JM, Lu L, Liu KK, Yang J, Wu HX, Liu QY. Forecast of severe fever with thrombocytopenia syndrome incidence with meteorological factors. Sci Total Environ 2018;626:1188 – 92. https://doi.org/10.1016/j.scitotenv.2018.01.196.
4. Sun JM, Lu L, Wu HX, Yang J, Ren JP, Liu QY. The changing epidemiological characteristics of severe fever with thrombocytopenia syndrome in China, 2011-2016. Sci Rep 2017;7(1):9236. https://doi.org/10.1038/s41598-017-08042-6.
5. Li JC, Zhao J, Li H, Fang LQ, Liu W. Epidemiology, clinical characteristics, and treatment of severe fever with thrombocytopenia syndrome. Infect Med 2022;1(1):40 – 9. https://doi.org/10.1016/j.imj.2021.10.001.
6. Mehand MS, Millett P, Al-Shorbaji F, Roth C, Kieny MP, Murgue B. World health organization methodology to prioritize emerging infectious diseases in need of research and development. Emerg Infect Dis 2018;24(9):e171427. https://doi.org/10.3201/eid2409.171427.
7. Wang T, Li XL, Liu M, Song XJ, Zhang H, Wang YB, et al. Epidemiological characteristics and environmental risk factors of severe fever with thrombocytopenia syndrome in Hubei province, China, from 2011 to 2016. Front Microbiol 2017;8:387. https://doi.org/10.3389/fmicb.2017.00387.
8. Battineni G, Chintalapudi N, Amenta F. Forecasting of COVID-19 epidemic size in four high hitting nations (USA, Brazil, India and Russia) by Fb-Prophet machine learning model. Appl Comput Inf 2020;6(1):1 – 10. https://doi.org/10.1108/ACI-09-2020-0059.
9. Lv CR, Guo WQ, Yin XY, Liu L, Huang XL, Li SM, et al. Innovative applications of artificial intelligence during the COVID-19 pandemic. Infect Med 2024;3(1):100095. https://doi.org/10.1016/j.imj.2024.100095.
10. Xie C, Wen HY, Yang WW, Cai J, Zhang P, Wu R, et al. Trend analysis and forecast of daily reported incidence of hand, foot and mouth disease in Hubei, China by Prophet model. Sci Rep 2021;11(1):1445. https://doi.org/10.1038/s41598-021-81100-2.
11. Deng B, Rui J, Liang SY, Li ZF, Li KG, Lin SN, et al. Meteorological factors and tick density affect the dynamics of SFTS in jiangsu province, China. PLoS Negl Trop Dis 2022;16(5):e0010432. https://doi.org/10.1371/journal.pntd.0010432.
12. Jiang XL, Wang YG, Zhang XM, Pang B, Yao MX, Tian XY, et al. Factors associated with severe fever with thrombocytopenia syndrome in endemic areas of China. Front Public Health 2022;10:844220. https://doi.org/10.3389/fpubh.2022.844220.
13. Wang ZJ, Yang ST, Luo L, Guo XH, Deng B, Zhao ZY, et al. Epidemiological characteristics of severe fever with thrombocytopenia syndrome and its relationship with meteorological factors in Liaoning province, China. Parasit Vectors 2022;15(1):283. https://doi.org/10.1186/s13071-022-05395-4.
14. Luo ZX, Jia XC, Bao JZ, Song ZJ, Zhu HL, Liu MY, et al. A combined model of SARIMA and prophet models in forecasting AIDS incidence in Henan province, China. Int J Environ Res Public Health 2022;19(10):5910. https://doi.org/10.3390/ijerph19105910.
15. Huang D, Grifoll M, Sanchez-Espigares JA, Zheng PJ, Feng HX. Hybrid approaches for container traffic forecasting in the context of anomalous events: the case of the Yangtze River Delta region in the COVID-19 pandemic. Transp Policy (Oxf) 2022;128:1 – 12. https://doi.org/10.1016/j.tranpol.2022.08.019.

SUPPLEMENTARY MATERIAL

Data Preprocessing

Ensuring that all timestamps in the dataset have a consistent format is crucial for accurate data processing and time series analysis. All timestamps were converted to the YYYY-MM format. This uniformity lays the foundation for applying various time series models effectively. The data were divided into training data and prediction data. Data starting from January 2020 to December 2020 were used as prediction data, while the rest as training data.

SARIMA Model

The autoregressive integrated moving average (ARIMA) and seasonal ARIMA (SARIMA) models are the most general class of models for forecasting a time series in theory (*1–2*). The ARIMA model aims to describe the autocorrelations in the data by outlining its components, including Autoregression (AR), Integrated (I), and Moving Average (MA). SARIMA is an extension of ARIMA that explicitly supports univariate time series data with a seasonal component, adding three new hyperparameters to specify the seasonal component. The SARIMA model requires selecting hyperparameters for both the trend elements (trend autoregression order p, trend difference order d, and trend moving average order q) and seasonal elements (seasonal autoregressive order P, seasonal difference order D, and seasonal moving average order Q) of the series. The formula for SARIMA is as follows:

$$(1-B)^d\left(1-B^s\right)^D Y_t = \theta_0 + \frac{\theta(B)\theta_s(B^s)}{\phi(B)\phi_s(B^s)}\sigma_t \quad (1)$$

where Y_t refers to the value of the time series at time t, θ_0 is constant, ε_t is the white noise value at period t, and the parameters d and D represent the difference number and seasonal difference number, respectively. B is the backshift operator, $\phi(B)$ is the autoregressive operator and $\theta(B)$ is the moving average operator. $\phi_s(B^s)$ and $\theta_s(B^s)$ are the seasonal operators.

The construction process of the SARIMA model is as follows.

Grid Search: The model starts by defining possible combinations of parameters for the seasonal aspects of the time series. It uses a grid search approach where p, d, and q values (representing autoregressive, differencing, and moving average terms, respectively) are tested along with seasonal counterparts.

AIC Evaluation: For each combination, a SARIMA model is fitted, and the Akaike Information Criterion (AIC) is calculated to assess model fit. The combination with the lowest AIC is considered optimal as it suggests a model that best explains the data with minimal complexity.

Best parameters selection: Use the best parameters determined through grid search, and then fit the SARIMA model to data before the specified date (in this case, January 2020).

Diagnostic check: Perform the Ljung Box test on the residuals to check for white noise, which indicates that the model's residuals have no autocorrelation and the model has fully captured the information in the data.

Prediction: The model performs a step-by-step (single-step-ahead) forecast using the trained model, constantly updating with actual data as it becomes available. This simulates a real-world scenario where predictions are made as new data comes in.

Evaluation: As indicated in part "Performance Evaluations" below.

Prophet Model

The Prophet model is a sophisticated method for forecasting time series, particularly tailored for business data and adept at navigating through complex trends and habitual seasonal variations. It provides a versatile treatment of trends, seasonality, and holiday influences. The trend component *g(t)* is engineered to capture non-periodic changes in the time series. It can employ either a logistic growth model for data with saturation limits or a piecewise linear model for data without clear saturation points.

The foundational equation of the Prophet model is expressed as:

$$y_t = g(t) + s(t) + h(t) + \varepsilon_t \quad (2)$$

Where yt denotes the predicted value at time t, s(t) is the seasonality component, h(t) represents the impact of holidays or specific events on the time series, and ε t is the error term accounting for aspects of the data not

explained by the model. The model also incorporates an advanced feature for automatically detecting change points in trends.

The construction process for the Prophet model is as follows:

Initiation: Prophet model is initialized with a specific configuration, including Growth Trend, Seasonality Components, Seasonality Mode, and Regularization Parameters.

Model Training: The Prophet model is fitted on the training data.

Prediction: Future dates are generated (future) for 12 months (periods=12) with monthly frequency (freq='M'). Forecasting is performed [forecast=m.predict(future)], and predictions for the year 2020 are extracted (forc).

Evaluation: As indicated in part "Performance Evaluations" below.

XGBoost

eXtreme Gradient Boosting (XGBoost) is an advanced optimization technique based on Gradient-boosting decision trees (GBDT). It operates by iteratively constructing a series of short, basic decision trees, each termed as a "weak learner". The process begins with the construction of an initial tree that exhibits subpar performance. Subsequent trees are then trained to correct the errors of their predecessors. This sequence of producing weaker learners continues until a stopping condition is met, such as reaching a predetermined number of trees. This method has been demonstrated to be effective in predicting human brucellosis (*3*) and renal hemorrhagic fever syndrome (*4*).

For a dataset with n examples and m features, a tree ensemble model in XGBoost predicts the output using K additive functions:

$$\hat{y}_k = \sum_{k=1}^{K} f_k(x_t), f_k \in F \tag{3}$$

Here, $\hat{y}$represents the predicted value for the i-th sample, fk is a function corresponding to the k-th tree, and F denotes the space of regression tree functions, with xi being the feature vector for the i-th sample.

To learn the set of functions used in the model, XGBoost minimizes the following regularized objective:

$$L(\varphi) = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k) \tag{4}$$

In this equation, $l(yi, \hat{yi})$ is the loss function quantifying the error between the observed and predicted data, and Ω(fk) is the regularization term that aids in smoothing the learned weights to prevent overfitting and encourage the model to select simpler yet predictive functions. The regularization term is defined as:

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 \tag{5}$$

Where γ and λ are regularization parameters, T is the number of leaves in the tree, and wj represents the score on each leaf.

SUPPLEMENTARY TABLE S1. Parameters of the optimized SARIMA model.

Parameters	Coefficient	Standard Errors	Z	*P*
ar.L1	0.424	0.133	3.190	0.001
ma.L1	−0.923	0.101	−9.136	<0.001
ma.S.L12	−0.878	0.242	−3.621	<0.001
sigma2	118.976	25.138	4.733	<0.001

SUPPLEMENTARY TABLE S2. Comparison of four models using MAE and RMSE.

Error metrics	SARIMA	Prophet	XGBoost	LSTM
MAE	5.47	6.64	2.54	4.75
RMSE	7.21	8.50	2.89	6.77

Abbreviation: MAE=mean absolute error; RMSE=root mean square error.

The construction process of the XGBoost model is as follows:

Parameter optimization: Use methods like Bayesian optimization to optimize the parameters.

Model Training: The XGBoost model is fitted on the training data.

Prediction: The model performs a step-by-step (single-step ahead) forecast using the trained model, constantly updating with actual data as it becomes available. This simulates a real-world scenario where predictions are made as new data comes in.

Evaluation: As indicated in part "Performance Evaluations" below.

LSTM Networks

Long Short-Term Memory (LSTM) networks, a specialized type of Recurrent Neural Networks (RNNs), excel at capturing both short-term and long-term dependencies in sequential data. This is primarily due to the unique architecture of the LSTM, which incorporates a cell state acting as a form of memory. This cell state is crucial for retaining information across various time steps, addressing the limitations of traditional RNNs. Additionally, LSTMs effectively alleviate the vanishing and exploding gradient issues commonly encountered in standard RNNs, particularly in lengthy sequences. The key feature of LSTM networks lies in their gating mechanism, comprising three types of gates, each with specific roles and formulas.

The input gate in an LSTM is pivotal in regulating the influx of new information into the cell state. This gate operates in two steps. The first step entails a sigmoid function that determines the essential update values, represented by the equation:

$$i_t = \sigma\left(W_4 x_t + W_4 h_{4-1} + b_t\right) \tag{6}$$

The second step employs a tanh function to generate a vector of new candidate values that may be added to the state, given by:

$$C_t = tanh\left(W_{x_C} x_t + W_{h_C} h_{t-1} + b_C\right) \tag{7}$$

Here, it is the activation of the input gate, and C_t is the candidate vector for the cell state update.

The forget gate in an LSTM decides which information from the cell state should be retained or discarded. It operates using a sigmoid function that evaluates the importance of the existing information in the cell state, defined by:

$$f_t = \sigma\left(W_{x_f} x_t + W_{h_f} h_{t-1} + b_f\right) \tag{8}$$

The activation vector ft indicates the extent to which past information is to be forgotten or retained.

The output gate in an LSTM manages the output sent to the next layer. This gate operates in two stages. Initially, a sigmoid function determines which parts of the cell state will be outputted, as shown by:

$$o_t = \sigma\left(W_{x_t} x_t + W_{A_t} h_{t-1} + b_0\right) \tag{9}$$

Then, the final output is calculated by multiplying this activation o_t with the tanh of the cell state, resulting in:

$$h_t = o_t \times \tanh\left(C_t\right) \tag{10}$$

The output vector ht represents the information transmitted to subsequent layers or units in the network.

The construction process of the LSTM model is as follows:

Network architecture: Build a neural network containing an LSTM layer, which extracts features from the input sequence, followed by a fully connected layer that outputs prediction results.

Loss function and model training: Use mean squared error as the loss function and use Adam optimizer for model training.

Early Stopping: To avoid overfitting, the training is stopped when the loss on the validation set no longer improves.

Parameter optimization: Use methods like Bayesian optimization to optimize the parameters.

Model Training: The LSTM model is fitted on the training data.

Prediction: Rolling prediction of future time points involves using a model to gradually predict future values, and after each prediction step, the results are fed back into the input data for the next prediction step.

Evaluation: As indicated in part "Performance Evaluations" below.

In the models mentioned above, a data post-processing technique known as clipping is utilized to handle unrealistic negative values in the results. This involves adjusting all negative forecast values to zero, ensuring data consistency and interpretability, and preventing potential analytical errors stemming from impractical negative predictions.

Performance Evaluations

The predictive performance of the models was assessed using two indexes: mean absolute error (MAE) and root mean squared error (RMSE).

MAE is a metric used to measure the average absolute errors between actual and predicted values in a dataset. It is calculated by taking the average of the absolute differences between the actual values and the predicted values. MAE is often used in regression analysis to evaluate the accuracy of a regression model. A lower MAE indicates better accuracy of the model, as it means that the model's predictions are closer to the actual values.

RMSE is another metric used to measure the accuracy of a regression model by calculating the square root of the average of the squared differences between actual and predicted values in a dataset. RMSE penalizes larger errors more heavily compared to MAE because it squares the errors before taking the square root. This means that outliers or large errors have a bigger impact on the RMSE compared to the MAE. Similar to MAE, a lower RMSE indicates better accuracy of the model. RMSE is often preferred when a small number of large errors are more significant than a large number of small errors.

The formulas are defined as follows:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \hat{y}_i\right| \tag{11}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{j=1}^{n}\left(y_t - \hat{y}_j\right)^2} \tag{12}$$

Software

The descriptive statistics and time series modeling were conducted using Python 3.7. The SARIMA model, Prophet model, XGBoost model, and LSTM model were implemented using the statsmodels package, fbprophet package, scikit-learn package, and Keras package, respectively. In the analysis, a $P<0.05$ was considered significant.

REFERENCES

1. Kim KR, Park JE, Jang IT. Outpatient forecasting model in spine hospital using ARIMA and SARIMA methods. J Hosp Manage Health Policy 2020;4:20. https://doi.org/10.21037/jhmhp-20-29.
2. ArunKumar KE, Kalaga DV, Sai Kumar CM, Chilkoor G, Kawaji M, Brenza TM. Forecasting the dynamics of cumulative COVID-19 cases (confirmed, recovered and deaths) for top-16 countries using statistical machine learning models: Auto-Regressive Integrated Moving Average (ARIMA) and Seasonal Auto-Regressive Integrated Moving Average (SARIMA). Appl Soft Comput 2021;103:107161. https://doi.org/10.1016/j.asoc.2021.107161.
3. Alim M, Ye GH, Guan P, Huang DS, Zhou BS, Wu W. Comparison of ARIMA model and XGBoost model for prediction of human brucellosis in mainland China: a time-series study. BMJ Open 2020;10(12):e039676. https://doi.org/10.1136/bmjopen-2020-039676.
4. Lv CX, An SY, Qiao BJ, Wu W. Time series analysis of hemorrhagic fever with renal syndrome in mainland China by using an XGBoost forecasting model. BMC Infect Dis 2021;21(1):839. https://doi.org/10.1186/s12879-021-06503-y.