A SEASONAL AUTO-REGRESSIVE MODEL BASED SUPPORT VECTOR REGRESSION PREDICTION METHOD FOR H5N1 AVIAN INFLUENZA ANIMAL EVENTS

Jie Zhang, Jie Lu and Guangquan Zhang

Decision Systems & e-Service Intelligence Lab, Centre for Quantum Computation & Intelligent Systems
Faculty of Engineering and Information Technology, University of Technology, Sydney
P.O. Box 123, Broadway, NSW 2007, Australia

{jie.zhang, jie.lu, guangquan.zhang}@uts.edu.au

The time series prediction of avian influenza epidemics is a complex issue, because avian influenza has latent seasonality which is difficult to identify. Although researchers have applied a neural network (NN) model and the Box-Jenkins model for the seasonal epidemic series research area, the results are limited. In this study, we develop a new prediction seasonal auto-regressive-based support vector regression (SAR-SVR) model which combines the seasonal auto-regressive (SAR) model with a support vector regression (SVR) model to address this prediction problem to overcome existing limitations. Fast Fourier transformation is also merged into this method to identify the latent seasonality inside the time series. The experiments demonstrate that the developed SAR-SVR method out-performs SVR, Box-Jenkins models and two layer feed forward NN model both in accuracy and stability in the avian influenza epidemic disease time series prediction.

Keywords: Time series prediction, support vector regression, auto-regressive model, Box-Jenkins model, avian influenza

1. Introduction

Avian influenza epidemic has been, and is still damaging living creatures throughout the world. The most notorious H5N1 highly pathogenic avian influenza (HPAI) virus is the main causative agent of avian influenza, which has caused billions animal deaths by either infection or culling[1, 2]. By the end of 2009, a total of 397 people had been confirmed as infected by the H5N1 virus and of them, 249 died (http://www.who.int/csr/disease/avian_influenza/country/cases_table_2009_01_19/en/index.html). These facts emphasize the need for monitoring and predicting the H5N1 infectious trend.

The most prominent achievement for analyzing epidemic disease transmission is the susceptible-infectious-recovered (SIR) model[3]. The SIR model and its extensions, e.g. the susceptible-exposed-infectious-recovered (SEIR) model, can simulate the transmission dynamics between different sub-groups in a population[3-7]. These models work well in many epidemic situations, but they encounter difficulties in mimicking the practical transmission mechanism[8], especially in the avian influenza domain. One reason is that poultry are often raised on separate farms with no free contact. Another reason is that, if some poultry on a farm are verified as infected by the virus, all poultry on the farm will be culled. Therefore, SIR type models cannot be directly applied to predict the transmission trend of the H5N1 virus.

The other type of model, epidemic time series prediction, has also been introduced into this field to compensate for the gaps encountered by the SIR model. Auto-regressive moving average (ARMA) or auto-regressive integrated moving average (ARIMA) models[9-11] (also known as Box-Jenkins models[12]), amongst others, are the most classic and the most widely applied practical models in time series analysis. The Box-Jenkins model is a parametric model with a cutting edge calculation speed which is considered the benchmark model in time series prediction[13, 14], normally comprising three or four steps to identify an appropriate ARIMA process, estimate the model parameters and prediction. The Box-Jenkins model has also adapted to forecasting seasonal time series[12], known as the SARMA/SARIMA model, which is suitable for epidemic diseases where the time series often follows cyclic behaviour patterns. However, the Box-Jenkins model is based on specific linear or error normality distribution assumptions, while in the real world, such as in an epidemic time series, linearity and stationary assumption conditions are seldom fulfilled. There is also a huge obstacle in applying this model because the textbook model selection methodology is complex and does not work well[15, 16]. In these circumstances, the neural network (NN) and support vector regression (SVR) models have been widely used to alleviate the limitations of the parametric Box-Jenkins model.

The neural network model[17-20] is a widely used non-parametric model with the advantage that it can predict and classify the data without relying too much on assumptions. Researchers have found that the NN model outperforms ARMA/ARIMA in time series prediction accuracy[21-23], and it has been effectively employed in epidemic disease time series analysis[17, 18]. The NN model also has limitations of over-fitness and instability[24].

The support vector regression model is a new type of non-parametric model which outperforms the NN model in time series prediction[23]. Though NN and SVR models are both machine learning models[24] and both can achieve better performance over the classic method when predicting the non-linear time series[23, 24], research demonstrates [23] that SVR significantly outperforms NN in non-stationary time series with no obvious trends or seasonality. Moreover, research has shown that SVR with an RBF kernel function can forecast seasonality with no trends[25]. Compared with the NN model, SVR has two further advantages in overcoming over-fitting and local optimization problems[24], because the structural risk minimization principle in a support vector machine is superior to the empirical risk minimization adopted in NN[26]. Therefore, SVR models have been largely applied in financial time series prediction[27, 28], electricity load prediction[29], biology protein prediction[30], climate[31], and other commercial areas[26, 32]. However, little research has been applied in the avian influenza time series forecasting area until now.

Nonetheless, SVR still encounters seasonality difficulties, because an SVR model performs better when the time series has performed the seasonal adjustment[32]. From a structural time series principle, a time series can be taken as the sum of trend, seasonal and irregular components[33], and it seems reasonable to eliminate the seasonal components before the prediction. Epidemic time series are complex seasonal and non-linear time series[34, 35]; especially the weekly data for animal avian influenza breakout event counts we obtained. However, applying SVR directly on this series without de-seasoning will degrade the model performances.

In this paper, we present a seasonal auto-regressive model based support vector regression (SAR-SVR) method to address this issue. The method combines the seasonal auto-regressive model with SVR together and, in addition, is merged with the latent seasonality identification method. Instead of eliminating the seasonal components, we apply SAR analysis and fast Fourier transformation (FFT) to identify and make full use of the seasonal component in the forecasting. We first applied SVR as the main prediction method in the avian influenza weekly counts time series, and we also compared it with NN and Box-Jenkins models to measure the SVR advantages.

The structure of this paper is: Section 2 introduces the Box-Jenkins and SVR models as preliminaries of the study. Section 3 discusses the characteristics of complex seasonality of epidemic trend time series data and presents the proposed SAR-SVR prediction method. Section 4 illustrates the SAR-SVR method with avian influenza weekly counts time series data, and Section 5 compares the results with the SVR method, Box-Jenkins model and NN methods to demonstrate the advantages of the method. Section 6 discusses the effectiveness of the SAR-SVR method and Section 7 summarizes major insights and outlines future research.

2. Box-Jenkins and SVR time series prediction models (Preliminary)

Box-Jenkins and SVR models have been successfully applied to address time series prediction issues. The Box-Jenkins model is a traditional parametric model while the SVR model is a new non-parametric model.

2.1. *Box-Jenkins model*

The Box-Jenkins model[12] can be applied to both stationary and non-stationary time series. The ARMA model predicts stationary series and the ARIMA model can be applied to a non-stationary model which has an increasing or decreasing trend over time. Also, some time series have obvious seasonal properties, in which circumstances we apply the SARMA/SARIMA model. $SARIMA(p,d,q)(P,D,Q)S$ model is described as follows

$$\phi_p(B)\Phi_P(B^s)(1-B)^d(1-B^s)^D x_t = \theta_q(B)\Theta_Q(B^s)\varepsilon_t \tag{1}$$

where

$\phi_p(B) = 1-\phi_1 B - ... - \phi_p B^p$, $\theta_q(B) = 1-\theta_1 B - ... - \theta_q B^q$
$\Phi_P(B^s) = 1-\Phi_1 B^s - ... - \Phi_P B^{s^P}$, $\Theta_Q(B^s) = 1-\Theta_1 B^s - ... - \Theta_Q B^{s^Q}$
B is the backshift operator, $Bx_t = x_{t-1}$, ε_t is white noise of normal distribution $N(0,\delta)$, and $\phi_{1,...,}\phi_p$, $\Phi_{1,...,}\Phi_p$, $\theta_{1,...,}\theta_p$, $\Theta_{1,...,}\Theta_p$ are all constant parameters.

Application of the Box-Jenkins model contains four steps:

- Model identification: Examine the data to identify the most suitable Box-Jenkins model by the estimation of the auto-correlation function (ACF) and the partial auto-correlation function (PACF).
- Parameters estimation: Apply methods, such as maximum likelihood, to estimate the model parameters of the training data-set.
- Model validation: Verify the adequacy by performing the residual analysis of the prediction error on the testing data-set.
- Prediction: Finally, apply the suitable model to do the prediction.

2.2. *Support vector regression models*

Support vector regression has been widely applied in time series predictions[27, 28]. Here we provide the main proposal of least square SVR (LS-SVR), ε-SVR and ν-SVR. LS-SVR is a very robust method when the noise is Gaussian; it is also very fast because it relies on fewer tuning parameters[36]. ε-SVR[37] is the classic SVR method and ν-SVR[38] is a variation. Both ε-SVR and ν-SVR have more parameters than LS-SVR and can provide more accurate results when ε and ν tune to suitable values.

2.2.1. *LS-SVR model*

The least-square version SVR (LS-SVR) is the SVR method for non-linear function regression to solve the optimization problem in primal weight space. Suppose that $\{ (\mathbf{x}_1, y_1),...,(\mathbf{x}_l, y_l) \}$, that $\mathbf{x}_i \in R^n$ is an input and $y_i \in R^1$ is a target output, and LS-SVR is a programming of :

$$\min_{\omega,b,e} \frac{1}{2}\mathbf{w}^T\mathbf{w} + \frac{1}{2}C\sum_{i=1}^{l} e_i^2 \tag{2}$$

s.t.

$$y_i = \mathbf{w}^T\phi(\mathbf{x}_i) + b + e_i, \quad i = 1,...,l \qquad (3)$$

where $\phi(\cdot): R^n \longrightarrow R^{n_h}$ a function which maps the input space into a higher dimensional feature space, weight vector $\mathbf{w} \in R^{n_h}$, error variables $b, e_i \in R$. After applying Lagrange multipliers and solving the optimal problem, the solution is:

$$\begin{bmatrix} 0 & \mathbf{1}_v^T \\ \mathbf{1}_v & \mathbf{\Omega} + \frac{1}{C}\mathbf{I} \end{bmatrix} \begin{bmatrix} b \\ \boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{y} \end{bmatrix} \tag{4}$$

where $\mathbf{y} = [y_1;...;y_l]$, $\mathbf{1}_v = [1;...;1]$, $\boldsymbol{\alpha} = [\alpha_1;...;\alpha_l]$, $\mathbf{I}$ is identity matrix and $\mathbf{\Omega}_{ij} = \phi(\mathbf{x}_i)^T\phi(\mathbf{x}_j)$ for $i, j = 1,...,l$. If we choose a kernel function $K(\cdot,\cdot)$,

$$K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^T\phi(\mathbf{x}_j), \qquad i, j = 1,...,l \tag{5}$$

then the result of the LS_SVR model for function estimation becomes:

$$y(\mathbf{x}) = \sum_{i=1}^{l}\alpha_i K(\mathbf{x}_i, \mathbf{x}) + b \tag{6}$$

If we choose Gaussian RBF kernel:

$$K(\mathbf{x}_i,\mathbf{x}_j)=\exp(-\frac{\|\mathbf{x}_i-\mathbf{x}_j\|^2}{2\sigma^2}) \tag{7}$$

then the result of the LS-SVR regression is affected by the choice of (c,σ).

2.2.2. *ε-SVR model and ν-SVR model*

There are also ε-Support Vector Regression ($\varepsilon-SVR$) and ν-Support Vector Regression ($\nu-SVR$). These two SVR models have different optimal methods to LS-SVR. Suppose that { $(\mathbf{x}_1,y_1),...,(\mathbf{x}_l,y_l)$ }, that $\mathbf{x}_i\in R^n$ is an input and $y_i\in R^1$ is a target output as previous LS-SVR, the standard form of $\varepsilon-SVR$ is:

$$\min_{w,b,\xi,\xi^*}\frac{1}{2}\mathbf{w}^T\mathbf{w}+C(\sum_{i=1}^{l}\xi_i+\sum_{i=1}^{l}\xi_i^*) \tag{8}$$

s.t.

$$\begin{aligned}\mathbf{w}^T\phi(\mathbf{x}_i)+b-y_i &\le \varepsilon+\xi_i,\ i=1,..,l\\ y_i - \mathbf{w}^T\phi(\mathbf{x}_i)-b &\le \varepsilon+\xi_i^*,\ i=1,...,l\\ \xi_i,\xi_i^* \ge 0,& \qquad i=1,...,l.\end{aligned} \tag{9}$$

This programming problem can be solved by its dual problem as:

$$\min_{\alpha,\alpha^*}\frac{1}{2}(\boldsymbol{\alpha}-\boldsymbol{\alpha}^*)^T\mathbf{Q}(\boldsymbol{\alpha}-\boldsymbol{\alpha}^*)+\varepsilon\sum_{i=1}^{l}(\alpha_i+\alpha_i^*)+\sum_{i=1}^{l}y_i(\alpha_i+\alpha_i^*) \tag{10}$$

s.t.

$$\begin{aligned}&\sum_{i=1}^{l}(\alpha_i-\alpha_i^*)=0,\\ &0\le\alpha_i,\alpha_i^*\le C,\ \ i=1,...,l,\end{aligned} \tag{11}$$

where $Q_{ij}=K(\mathbf{x}_i,\mathbf{x}_j)$. Finally, the approximate function is:

$$\sum_{i=1}^{l}(-\alpha_i+\alpha_i^*)K(\mathbf{x}_i,\mathbf{x})+b. \tag{12}$$

$\nu-SVR$ use a parameter ν to control the number of support vectors. The problem is:

$$\min_{w,b,\xi,\xi^*,\varepsilon}\frac{1}{2}\mathbf{w}^T\mathbf{w}+C(\nu\varepsilon+\frac{1}{l}\sum_{i=1}^{l}(\xi_i+\xi_i^*)) \tag{13}$$

s. t.

$$\begin{aligned}\mathbf{w}^T\phi(\mathbf{x}_i)+b-y_i &\le \varepsilon+\xi_i, i=1,...,l\\ y_i - \mathbf{w}^T\phi(\mathbf{x}_i)-b &\le \varepsilon+\xi_i^*, i=1,...,l\\ \xi_i,\xi_i^* \ge 0 & \qquad i=1,...,l\end{aligned} \tag{14}$$

Where $\varepsilon\ge 0$ and the dual problem is:

$$\min_{\boldsymbol{\alpha},\boldsymbol{\alpha}^*}\ \frac{1}{2}(\boldsymbol{\alpha}-\boldsymbol{\alpha}^*)^T\mathbf{Q}(\boldsymbol{\alpha}-\boldsymbol{\alpha}^*)+\mathbf{y}^T(\boldsymbol{\alpha}-\boldsymbol{\alpha}^*) \tag{15}$$

s. t.

$$\begin{aligned}&\mathbf{1}^T(\boldsymbol{\alpha}-\boldsymbol{\alpha}^*)=0,\\ &\mathbf{1}^T(\boldsymbol{\alpha}+\boldsymbol{\alpha}^*)\le C\nu,\\ &0\le\alpha_i,\alpha_i^*\le C/l,\ \ i=1,...,l,\end{aligned} \tag{16}$$

where $\mathbf{1}=[1;...;1]$. The decision function is the same as (12) of $\varepsilon-SVR$.

3. Seasonal auto-regressive model based support vector regression method

The Box-Jenkins method has the advantage of handling seasonality inside the data and SVR has the ability to deal with non-linear time series. We combine the two models to forge a new model which improves prediction performance.

3.1. *SAR-SVR method necessity and characteristics avian influenza time series*

The first step of predicting the avian influenza time series is identifying a suitable model, because "Essentially, all models are wrong, but some are useful"[39]. Though there is an increasing use of state-of-the-art models, the final result can only be improved by applying a suitable model which narrowly depicts the characteristics of the series. Normally a time series is composed of a trend, seasonality and random error, but in avian influenza time series, there is no obvious trend and no obvious seasonality. In other words, the behaviour of a real data series such as avian influenza data is difficult to identify. However, in time series analysis of epidemic event data, seasonal dynamics analysis[40] is one of the most important issues to be discussed. It is reasonable that epidemic event time series should have seasonality because of the biological characteristics of the virus. Some researchers have found that the outbreaks of H5N1 occur in approximate yearly cycles[41, 42]. But in practice, this seasonality changed subtlety. The outbreak wave in 2006 in Vietnam appeared in November, but should have occurred in January in 2005. One reason for cycle variations is that the virus is biologically strong and can adapt and survive in different environments[43], and the breakouts can be identified in different times in a year. In order to depict the inside dynamic of this phenomenon, we develop a SAR-SVR method do address this issue and we will discuss it in detail in the following section.

3.2. *SAR-SVR method description*

We present a new SAR-SVR method in which we combine the SARMA model with the SVR model to analyze FFT results. The process has four steps: latent seasonality time lag identification, establishment of input vector set, parameter optimization, and final prediction.

Step 1: Identify the latent cycles and auto-correlation in the time series

The objective of this step is to identify a time lag set *S* with the significant auto-correlation values and seasonal cycles. From the analysis in Section 3.1, the auto-correlation strength will answer for the prediction qualities. Also, seasonality of the time series is another different description of the auto-correlation. Therefore, we use the auto-correlation function (ACF) and fast Fourier transformation (FFT) to identify auto-correlation and the periodic behaviors in the time series. Normally, the epidemic breakouts have seasonal cycles, some of which are obvious, and some are not. For avian influenza breakouts it is hard to identify these cycles.

Step 1.1: Identify the time series cycle by FFT

The time lag set *S1* can be obtained from time lags of FFT peak points of training data. If a time series has regular cycles, then the FFT result will show a group of high sharp peak values. Otherwise, the FFT results will show irregular values without very sharp peaks. In this situation, we can still identify the normal peak values from the series. A peak point is a point at time period *t* which has value *fft(t), t>0* and it has larger value than the adjacent point as |fft(t)-fft(t-ε)|>*e* and |fft(t)-fft(t+ε)|>*e*, where *e*>0. If a time lag value is *t*, a real value, and $m < t < m+1$ then we define $m, m+1 \in S1$, where *m* is an integer. In the real application, we can select the *e* values as the system minimum positive value but we limit the significant peak values according to the mean value $mean(fft)$ and the standard error S_E of the FFT series, for example $fft(t) > \theta$, where $\theta > 0$ and $\theta \in \{mean(fft) \pm iS_E\}, i = 0,...,n$.

Step 1.2: Identify the significant auto-correlations by ACF

We then calculate the ACF values of the training data and choose all the significant peak point lags as cyclic time lag set S2. There should be many significant ACF values, but we only choose those which are significant, as well as a peak value. In this step, we choose e>0 to identify all the peak points and we apply the significant interval to exclude the insignificant peak values where the *e* is set to the system minimum positive value. The x axes values, the time lags, of these peaks can then form the set *S2*.

In practice, if we have large number of sample, the ACF value series are normal distribution with 0 mean and variance of approximately $1/N$ (*N*>50). For 95% confidence level, a confidence interval will be given by [$-1.96/\sqrt{N}, 1.96/\sqrt{N}$]. An ACF value can be taken as significant if it outside the confidence interval.

Step 1.3: Union S1 and S2 to obtain the time lag set S

With step 1.1 and 1.2 we can estimate the interested lag set in different methods and unite them to obtain one comprehensive set *S*. We can limit the element *s* to a natural time cycle to decrease the size of set *S*, for example, 52 weeks for a whole year cycle.

Step 2: Apply the SAR model to obtain input vectors for each lag in S

In the SARMA model, if we only consider the auto-correlation mechanism, we can use the SAR part to identify the input vector. A $SAR(p,P)s$ model is:

$$\phi_p(B)\Phi_P(B^s)x_t = \varepsilon_t \tag{17}$$

If we telescope and transform the equation,

$$\phi_p(B)\Phi_P(B^s)x_t = (1-\phi_1 B - ... - \phi_p B^p)(1-\Phi_1 B^s - ... - \Phi_P B^{s^P})x_t = \varepsilon_t$$
$$\Rightarrow x_t - a_1 x_{t-1} - a_2 x_{t-2} - ... - a_p x_{t-p} - a_s x_{t-s} - a_{s+1} x_{t-s-1} - ... - a_{s*P+p} x_{t-s*P-p} = \varepsilon_t$$
$$\Rightarrow x_t = a_1 x_{t-1} + a_2 x_{t-2} + ... + a_p x_{t-p} + a_s x_{t-s} + a_{s+1} x_{t-s-1} - ... + a_{s*P+p} x_{t-s*P-p} + \varepsilon_t$$
$$let \quad x_t = y \quad then\, we\, have$$

$$y = f(\mathbf{x}^*, \varepsilon) \tag{18}$$

Finally, we have:

$$\mathbf{x}^* = \{x_{t-1}, x_{t-2}, ..., x_{t-p}, x_{t-s}, x_{t-s-1}, ..., x_{t-s*P-p}\} \tag{19}$$

For $SAR(1,1)12$ model, the telescoped formula is:

$$(1-\phi_1 B)(1-\Phi_1 B^{12})x_t = (1-\phi_1 B - \Phi_1 B^{12} - \phi_1\Phi_1 B^{13})x_t = \varepsilon_t$$
$$\Rightarrow x_t - \phi_1 x_{t-1} - \Phi_1 x_{t-12} + \phi_1\Phi_1 x_{t-13} = \varepsilon_t$$
$$\Rightarrow x_t = \phi_1 x_{t-1} + \Phi_1 x_{t-12} - \phi_1\Phi_1 x_{t-13} + \varepsilon_t$$

If we let $y = x_t$ then:

$$y = \phi_1 x_{t-1} + \Phi_1 x_{t-12} - \phi_1\Phi_1 x_{t-13} + \varepsilon_t = f(\mathbf{x}^*, \varepsilon)$$

The final $\mathbf{x}^*$ can be taken as the input vector for the next step. If we apply the parameter of $SAR(1,1)12$ then $\mathbf{x}^* = (x_{t-1}, x_{t-12}, x_{t-13})$. This is different from the normal input vector where, if we use 12 as window lags, then $\mathbf{x} =(x_{t-1}, x_{t-2}, x_{t-3}, ..., x_{t-12})$.

Step 2.1: Estimate p^* and P^* value sets with partial auto-correlation function (PACF)

Maximum p values can be identified by the number of partial auto-correlation function (PACF) significant values, but the parsimonious rules for identifying values do not always work well [16], so we apply $p^* = [0, \max\{p,2\}]$ as finally p value set. For P values we estimate the same way as $P^* = [0, \max\{P,2\}]$ for the same reason mentioned in reference [16], where P is identified by PACF.

Step 2.2: For each time lag s in S telescope $SAR(p,P)s$ model to obtain $\mathbf{x}^*$

The final $\mathbf{x}^*$ can be obtained by the telescoping process outlined in (18) and final format is described in (19). $\mathbf{x}^*$ can now be only defined by:

$$\mathbf{x}^* = f(s, p, P) \tag{20}$$

In identifying $\mathbf{x}^*$, if s and P both have very large values, then the maximum back lag operator will be very large. This will lead to the edge effect and will confuse the final result. So we can limit s and P to prevent from occurring. Normally, we limit s to a nature cycle such as year, month or week, as in the following predicting example.

Step 3: Apply SVR for prediction

For each *s* in *S* we apply SVR for every possible input vector $\mathbf{x}^*$ obtained from Step 2. We apply LS-SVR but we use the $\mathbf{x}^*$ in (19) as the input vector. We apply:

$$y_k = \mathbf{w}^T \phi(\mathbf{x}_k^*) + b + e_k, \quad k = 1, ..., N \tag{21}$$

in LS-SVR as constraints. After training, we can use the prediction equation provided, but use the vector in (19) as:

$$y = \sum_{k=1}^{N} \alpha_k K(\mathbf{x}_i^*, \mathrm{x}) + b \tag{22}$$

For $\varepsilon - SVR$ or $\nu - SVR$, we make a similar change by applying $\mathbf{x}^*$ instead of $\mathbf{x}$ in (9), and apply:

$$y = \sum_{i=1}^{l} (-\alpha_i + \alpha_i^*) K(\mathbf{x}^*_{\mathrm{j}\,i}, \mathrm{x}\ ) + b \tag{23}$$

to do the prediction.

LS-SVR is more efficient than $\varepsilon - SVR$ or $\nu - SVR$ because LS-SVR has less parameters to be estimated, while $\varepsilon - SVR$ or $\nu - SVR$ may achieve a more accurate result despite the slower solving time. Usually, the larger the time lags *s*, the shorter the input vector should be compared with the original input window size.

Step 3.1: For each input vector $\mathbf{x}^*$ *apply SVR to train and predict.*

Each input instance is obtained through a slide widow with form $\mathbf{x}^*$ over the time series observations, and we apply SVR to do the training and predicting.

Step 3.2: For each input vector $\mathbf{x}^*$ *calculate and record the predict results.*

By calculating the RMSE (root mean square deviation) value for the testing data for each input vector, record all the parameters and evaluating indicators such as *s, p, P* and RMSE with optimal results.

The final evaluation of different models performances is only based on the test data-set. We apply RMSE to estimate the results for different prediction models. If we use Y_t denote the observation at time $t \in \{1,...,n\}$ and F_t is the forecast, then RMSE is:

$$RMSE(Y,F) = \sqrt{\frac{\sum_{i=1}^{n} (Y_i - F_i)^2}{n}} \tag{24}$$

Step 4: Find the optimal lag s and the predicting vector for the final prediction

In this step, we evaluate each input vector and find the optimal result.

Step 4.1: Estimate result of the entire predicting vector

For each input vector $\mathbf{x}^*$ or, in other words, for each set of s, p, P a result is obtained. After evaluating each result, an optimal input vector is identified. Then the prediction with vector $\mathbf{x}^*$ will produce the optimal result.

Step 4.2: Apply the optimal vector and parameters to predict

After analyzing and evaluating of all the results, we can finally obtain the most suitable parameters for the final prediction.

By the proposed four steps, we can finally have the output of more accurate prediction results because the proposed method can identify the characteristics of the time series. This process is illustrated in Figure 1.

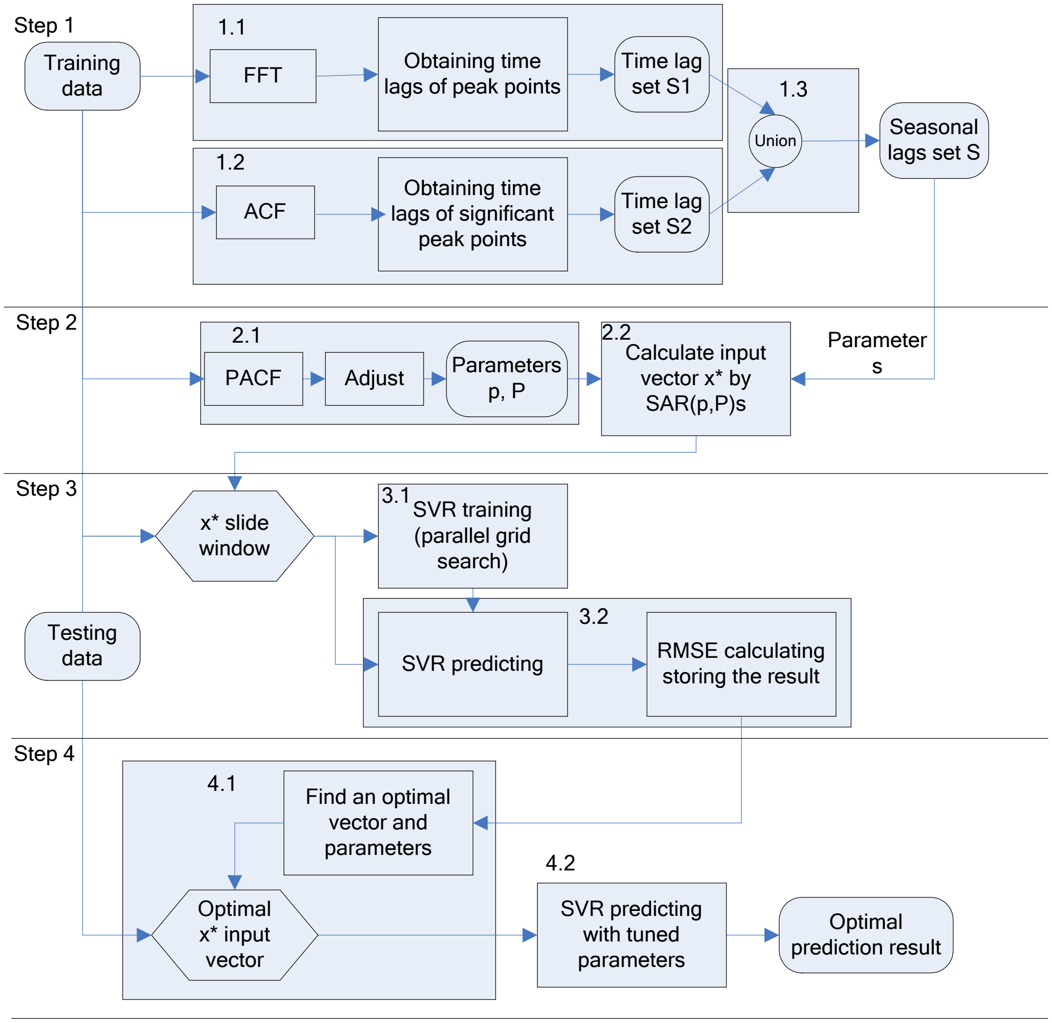

Step 1
Training data
1.1
FFT
Obtaining time lags of peak points
Time lag set S1
1.3
Union
Seasonal lags set S
1.2
ACF
Obtaining time lags of significant peak points
Time lag set S2
Step 2
2.1
PACF
Adjust
Parameters p, P
2.2
Calculate input vector x* by SAR(p,P)s
Parameters
Step 3
x* slide window
3.1
SVR training (parallel grid search)
3.2
SVR predicting
RMSE calculating storing the result
Testing data
Step 4
4.1
Find an optimal vector and parameters
Optimal x* input vector
4.2
SVR predicting with tuned parameters
Optimal prediction result

Figure 1. The SVR-based method process

4. A case study

The avian influenza weekly series is a time series with complex seasonal dynamics. In this section, we first introduce the data source and pre-processing approach. We then illustrate the prediction process by using our SAR-SVR method and comparing it with existing methods.

4.1. *Time series data source of avian influenza*

Data was obtained from reports on the website: "http://www.oie.int/downld/AVIAN%20 INFLUENZA/A_AI-Asia.htm". The reports contain the verified H7 and H5 infectious animal events from different countries around the world dated from December 2003 until now. The first section of data is from data collections from 2005 to end 2009. Because these reports have no uniform format until 2005, only data from 2005 is collected by us from this site. In each report file, there are one or more avian influenza animal breakout event records from a country. Each record contains the outbreak time, outbreak location, infected population, location type, and so on. If the outbreak location is a farm, the number of poultry infected or destroyed is reported. If the outbreak occurred in a wild animal population (often wild birds), the number of population deaths are reported. Also, data from 2003 to 2005 was obtained from *Nature News* reporter Dr Declan Butler (http://www.nature.com/news/author/Declan+Butler/index.html). The two data-sets are combined into one data-set to form a data-set dated from 2003 to 2009. Because the number of animals culled on a farm does not affect other sites, the dead bird numbers or infected numbers on the farm do not affect other farms. For processing purposes, we count this farm event as only one event. Therefore, we have a series of daily events reported, but there are lots of days with no event reports. Finally, we sum up the time series to total 289 weeks but apply only 261 weeks in case of some events which happened but were not reported. Figure 2 displays the data count series.

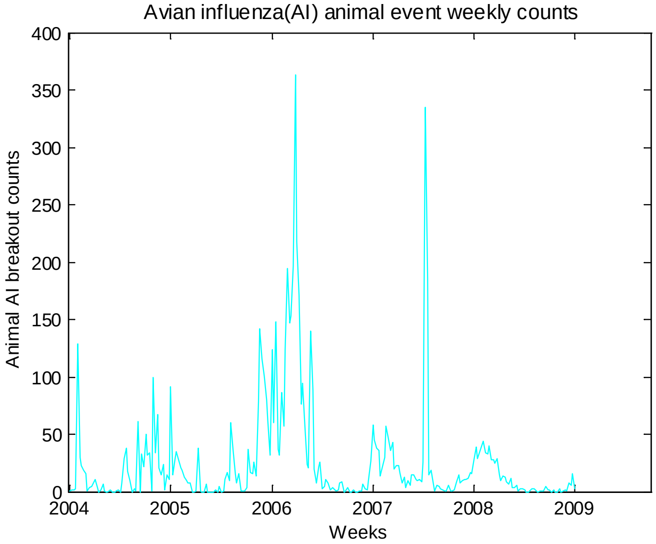


Figure 2. Avian influenza animal event weekly counts data

4.2. *Data pre-processing method*

Before feeding the data into the model, the weekly event count data has to be pre-processed. First, we transform the time series into a matrix according to the input vectors. Each record contains a series of previous week data as an input vector, plus one current week's data. The data matrixes are normalized to [-1, 1] column by column, and then the data is divided into a 75 percent training set and a 25 percent testing set. We apply different models to train and test the results. The data is then reversed back to calculate the evaluation indictors. Calculation of the model performance indictors is based on the testing data-set.

4.3. *Prediction procedures and results*

We now apply the proposed SAR-SVR for the avian influenza data series prediction.

Step1: Identify the latent cycles and auto-correlation in the time series

Step 1.1: Identify the time series cycle by FFT

First, we use FFT results of the data to identify the peak points and to identify the set *S1*. We apply the system minimum positive value as *e* value, for example, in Matlab the value can be obtained by *eps* function. We then mark each peak and apply two integers adjacent to each peak point's x axes as *S1*. Here we only use the integer value less than 100 which we can limit to 52 to make a one-year time interval. We apply 100 to show an edge effect in this instance. The first time lag set *S1* is:

S1= {2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 17, 18, 21, 22, 38, 63, 64}

Step 1.2: Identify the significant auto-correlations by ACF

We apply the x axes time lags of significant peak value of ACF series to obtain set *S2:*

S2= {1, 13, 15, 18, 42, 49, 51, 53, 55, 60, 68, 72, 78, 80, 82, 86}

Step 1.3: Union S1 and S2 to obtain the time lag set S

We combine S with S1 and S2, to finally find:

S= {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 17, 18, 21, 22, 38, 42, 49, 51, 53, 55, 60, 63, 64, 68, 72, 78, 80, 82, 86}

The FFT and ACF results are shown in Figure 3. From the time series FFT result we can not identify the obvious, very abrupt high peaks. There is a high point around 63 weeks, but not around 52 weeks as we expected. In the ACF series we can also obtain the normal peak points.

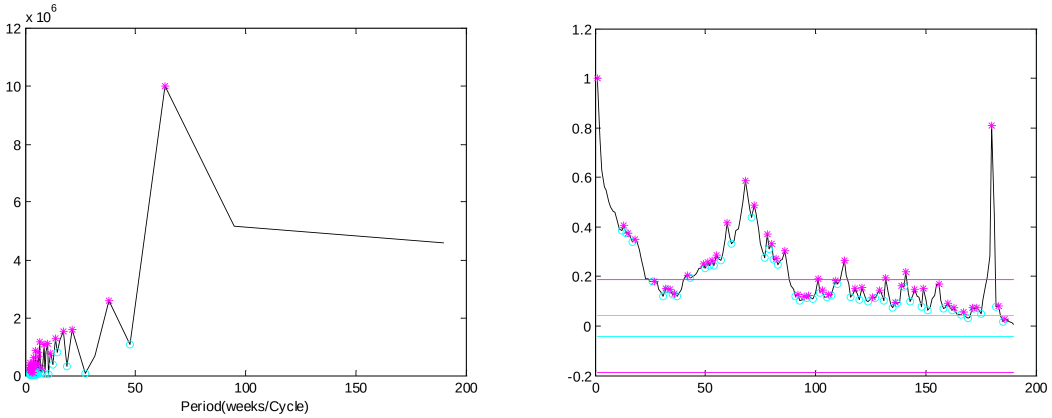
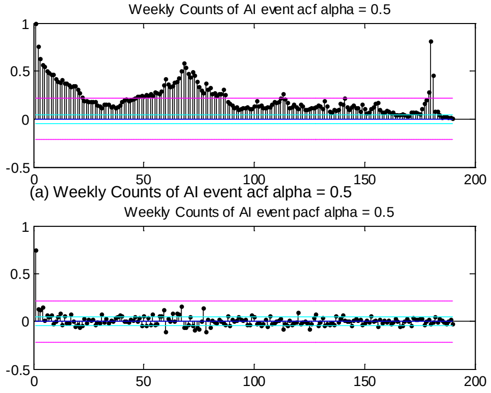


Figure 3 (a) FFT result of time series. (b) Significant result of ACF series

Step 2: Apply the SAR model to obtain the input vectors for each lag in S

Step 2.1: Estimate p^* *and* P^* *value set with PACF*

The PACF values are shown in Figure 4. Only the first is significant, so we choose max $p = 1$. We adjust it to max $p = 2$ to achieve max $p = 2$ and max $P = 2$, so that $p \in [0,1,2]$, $P \in [0,1,2]$



(a) Weekly Counts of AI event acf alpha = 0.5

(b) Weekly Counts of AI event pacf alpha = 0.5

Figure 4 (a) The autocorrelation function (ACF) and (b) partial autocorrelation function (PACF) of data series

Step 2.2: For each time lag s in S telescope $SAR(p,P)s$ *model to obtain* $\mathbf{x}^*$

For each time lag *s* in set *S* and *p* and *P* values we calculate the input vector. For example if *s*=4 and if we apply $SAR(2,1)_4$ then $\mathbf{x} = \{x_{t-1}, x_{t-2}, x_{t-4}, x_{t-5}, x_{t-6}\}$ For *s*=82, this cycle period is too large for our 269 weeks data. For $SAR(2,2)82$, the time **lag**={t-1, t-2, t-82, t-83, t-84, t-164, t-165, t-166}, so we lost 166 data in the prediction. For $SAR(1,2)51$, the input vector time lag set ={ t-1, t-51, t-52, t-102, t-103} also lost nearly half the data. If we do not want to lose a large amount of data, we can choose a time lag of no more than 52 weeks (one year) and lose no more data than a ratio, for example, one third of the total data. If we limited the time lags to, at most, one year, 52 weeks, then *S* is:

S= {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 17, 18, 21, 22, 38, 42, 49, 51}

Step 3: Apply SVR for prediction

Step 3.1: For each input vector $\mathbf{x}^*$ *apply SVR to train and predict*

Each input instance is obtained through a slide widow with form $\mathbf{x}^*$ over the time series observations, and by applying the SVR prediction to obtain the prediction result. We make the prediction by LS-SVR, **ν**-SVR and **ε**-SVR by only an RBF kernel. Because the RBF kernel function outperforms other kernel functions, in the following experiments we only show the results with an RBF kernel function.

Step 3.2: For each input vector $\mathbf{x}^*$ *calculate and record the predict results*

In each experiment we apply the parallel grid search method for each time lag to find the optimal result. The grid search parameters range is on different (γ ($\gamma = 2\sigma^2$ in equation 7), C):[2^{-4},...2^{5}][2^{0},...,2^{9}] values. We apply all the vectors obtained by each time lag s in set S and calculate the RMSE of testing data, so that the entire evaluating indicators can be saved. The RMSE results of the three SAR-SVR models are shown in Figure 5.

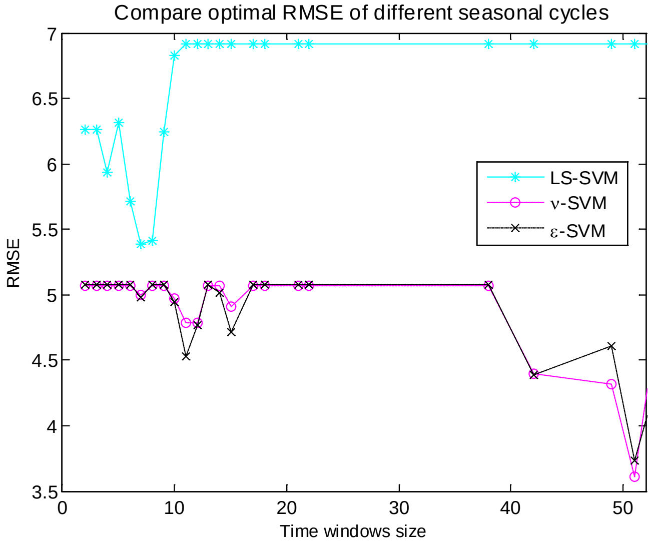


Figure 5 The RMSE results of SAR-SVR

With each SVR model we apply the parallel grid search and apply time lags in S. The minimum RMSE results for different SVR regressions are different and can be observed in Figure 5. For ν-SVR and ε-SVR we use the LIBSVM Matlab package[44], and for the LS-SVR we use our own code. In ε-SVR, the ε is set as 0.01 according to the method in reference[45].

Step 4: Find the optimal lag s and the predicting vector for the final prediction

Step 4.1: Estimate result of the entire predicting vector

We obtain optimal results by both SAR+ν-SVR and SAR+ε-SVR methods and list the top 3 results in Table 1. The three groups of optimal results are sorted according to the RMSE values. The seasonality parameters *s*, maximum time lag, input windows size are parameters associated with input vector and C and γ are parameters associated with the parallel grid search. The training time and predicting time of the models are also listed in the table. The total searching time is the time used when searching the optimal parameters of (C,γ) throughout the time lag *s* in *S*. The input vector can be established from parameters *p, P* and *s*. The vector length values, the window size, show that we don't need a very long input vector to obtain better results. From the results, we discovered that the optimal result is obtained by the SAR(1,2)51+ε-SVR model which will use $x_{t-1}, x_{t-51}, x_{t-52}, x_{t-102}, x_{t-103}$ to predict x_t. It means that the maximum time lag is 103 in this model and input vector size is only 5. We find that both SAR+ε-SVR and SAR+ν-SVR reach their top 3 optimal performances have seasonal time lag *s* as 51 week which is almost one year and is consistent to the time series nature. For SAR+LS-SVR the optimal performance time lags are 7, 6 and 8 weeks and the SAR+LS-SVR result has the larger RMSE values.

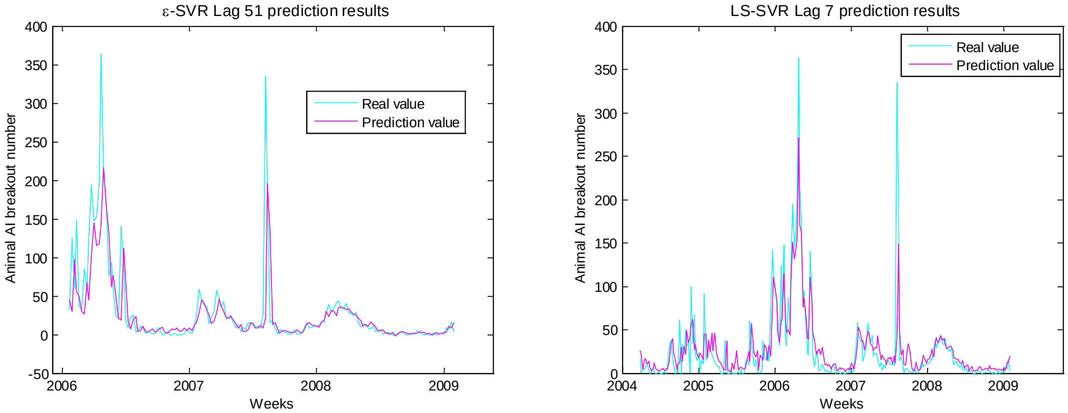
Table 1 Top 3 optimal performance of different SAR(p,P)s+SVR methods

Model descriptions SAR(p,P)s+SVR		top#	RMSE	Seasonals	Max lag	Window size	C	γ	Training time	Predicting time	Total searching time
LS-SVR+	SAR(1,1)7	1	**5.5771**	7	8	3	2	32	0.00502		97.78
	SAR(2,1)6	2	5.9199	6	8	5	1	32	0.00508		
	SAR(1,1)8	3	6.0117	8	9	3	8	16	0.00484		
ε-SVR+ (ε=0.01)	SAR(1,2)51	1	**3.9048**	51	103	5	128	0.0625	0.00943	0.00117	604.86
	SAR(2,2)42	2	4.3666	42	86	8	2	0.0625	0.00305	0.00174	
	SAR(2,2)49	3	4.9028	49	100	8	1	0.0625	0.00213	0.0014	
ν-SVR+ (ν=0.5)	SAR(1,2)51	1	**3.9641**	51	103	5	128	0.0625	0.01704	0.00091	4221.08
	SAR(2,2)42	2	4.6418	42	86	8	16	0.0625	0.01704	0.00091	
	SAR(2,2)49	3	4.6549	49	100	8	4	0.0625	0.0024	0.00091	

Finally, we can select from the above models to predict future event counts. Though SAR+ε-SVR and SAR+ν-SVR with time lag 51 have the almost the same optimal results, but SAR+ε-SVR search faster. SAR+LS-SVR has the better speed but lower accuracy. If there is not sufficient data, SAR(1,2)51+ν-SVR should not be used because it will lose 103 data.

Step 4.2: Apply the optimal vector and parameters to predict

Following this process, we use the selected parameters to make the prediction. SAR(1,1)7+LS-SVR, SAR(1,2)51+ε-SVR and SAR(1,2)51+ν-SVR have the optimal performances. Because the last two models are very similar to each other and we only show the prediction results of SAR(1,2)51+ε-SVR in addition to SAR(1,1)7+LS-SVR in Figure 6.



(a) SAR(1,2)51+ε-SVR prediction result with lag=51, input vector (t-1, t-51, t-52, t-102, t-103)

(b) SAR(1,1)7+LS_SVR prediction result with lag=11, input vector (t-1, t-7, t-8)

Figure 6 SAR-SVR prediction results with different parameters

5. Comparison with other models

We compare the optimal results of the proposed SAR-SVR with the Box-Jenkins model, two layer feed forward NN model (input-hidden-output) and SVR model. All the experiments are conducted on cluster nodes of UTS, which have 64 bits processor of 3.33GHz and 12G memory space with the operation system of Red Hat Enterprise Linux 5 (64bit). The data applied for comparing is the weekly breakout counts time series of avian influenza data. The details of each model are discussed as follows:

(1) Box-Jenkins model. We follow the model estimating processes to estimate the ARMA model parameters. In addition to AR(1), AR(2) and ARMA(1,1), we also apply ARMA(1,2) and ARMA(2,1) model to do the prediction. The RMSE value, the parameters of each model are listed in Table 2. We also identified SARMA models of different parameters with time lag from 2 to 52 and we only list the top 5 optimal results in Table 3.

Table 2 The detailed results of ARMA models

Model	RMSE	A(B)y(t) = ε(t) or A(B)y(t) = C(B)ε(t) *(B is the backshift operator)*	
AR(1)	6.0698	A(B) = 1 - 0.7502 B^-1	
AR(2)	5.7524	A(B) = 1 - 0.65 B^-1 - 0.1336 B^-2	
ARMA(1,1)	**5.5594**	**A(B) = 1 - 0.8635 B^-1**	**C(B) = 1 - 0.277 B^-1**
ARMA(1,2)	5.5742	A(B) = 1 - 0.9333 B^-1	C(B) = 1 - 0.3469 B^-1 - 0.1779 B^-2
ARMA(2,1)	5.7143	A(B) = 1 - 1.401 B^-1 + 0.424 B^-2	C(B) = 1 - 0.8071 B^-1

Table 3 The detailed results of top 5 SARMA models

top#	SARMA(p,q)X(P,Q)S	RMSE	Seasonality	A(B)y(t) = C(B)ε(t)	
1	SARMA(1,0)X(1,1)3	5.6671	3	A(B) = 1 - 0.6212 B^-1 - 0.4281 B^-3 + 0.1291 B^-4	C(B) = 1 - 0.3895 B^-3
2	SARMA(0,1)X(1,1)2	5.6810	2	A(B) = 1 - 0.8666 B^-2	C(B) = 1 + 0.5994 B^-1 - 0.4818 B^-2 - 0.2108 B^-3
3	SARMA(1,0)X(1,1)4	5.8192	4	A(B) = 1 - 0.6334 B^-1 - 0.6537 B^-4 + 0.3466 B^-5	C(B) = 1 - 0.5012 B^-4
4	SARMA(1,0)X(1,1)47	5.8418	47	A(B) = 1 - 0.7578 B^-1 - 0.26 B^-47 + 0.1292 B^-48	C(B) = 1 - 0.2536 B^-47
5	SARMA(1,0)X(1,1)44	5.9337	44	A(B) = 1 - 0.7588 B^-1 - 0.3935 B^-44 + 0.244 B^-45	C(B) = 1 - 0.3725 B^-44

(2) Two layer feed forward NN model. We have tried different input vectors from 1 to 52 and different hidden layer size from 1 to 52 with three different output activation functions, three different hidden layer activation functions, three different training algorithms and three different learning rates. The neural network results are illustrated in Table 4. Each row represents the optimal result of a NN structure which is obtained by searching different input window size from 1 to 52 and different number of hidden neurons from 1 to 52. The total searching time is the time cost of this searching process. The activation function of hidden layer and output layer, the learning rate, input window size, hidden layer size, training and predicting time, and total searching time are listed, the average is 5543.49 seconds. We only listed the optimal NN structure for each combination of hidden layer activation function, output activation function, training method and learning rate. The training epoch is set to 500. The training time and predicting time is time spent only for the listed optimal NN structure. The average RMSE of the optimal results is 6.67 and the minimum values is 5.687 which comes from the five inputs, five hidden layer units and one output NN model which apply sigmoid-linear activation functions. It takes a very long time to finish searching for all the combinations.

Table 4 The detailed optimal results of NN models

Hidden layer activation function	Output activation function	Training method	Learning rate	Time window size	Hidden neurons	RMSE	Training Time	Predicting time	Total searching time
Tan-Sigmoid	Sigmoid	BFGS quasi-Newton	0.1	3	11	7.24	3.631	0.00541	14079.245
Tan-Sigmoid	Sigmoid	BFGS quasi-Newton	0.01	6	5	7.1509	2.1974	0.00543	14061.048
Tan-Sigmoid	Sigmoid	BFGS quasi-Newton	0.001	5	10	7.2303	4.2682	0.00545	14378.538
Tan-Sigmoid	Sigmoid	Resilient backpropagation	0.1	2	18	6.4769	1.4153	0.00537	1245.8677
Tan-	Sigmoid	Resilient	0.01	6	6	6.9445	1.3518	0.00544	1248.5256

Sigmoid		backpropagation							
Tan-Sigmoid	Sigmoid	Resilient backpropagation	0.001	6	15	6.351	1.4112	0.00533	1240.777
Tan-Sigmoid	Sigmoid	Levenberg-Marquardt backpropagation	0.1	2	11	7.1493	2.2678	0.00529	1585.7188
Tan-Sigmoid	Sigmoid	Levenberg-Marquardt backpropagation	0.01	6	3	6.7623	1.9511	0.00512	1669.63
Tan-Sigmoid	Sigmoid	Levenberg-Marquardt backpropagation	0.001	2	18	7.3941	2.5068	0.00544	1567.3736
Sigmoid	Sigmoid	BFGS quasi-Newton	0.1	5	8	6.8759	3.9227	0.00546	13213.198
Sigmoid	Sigmoid	BFGS quasi-Newton	0.01	3	26	7.1054	7.5783	0.00578	13575.352
Sigmoid	Sigmoid	BFGS quasi-Newton	0.001	3	21	6.9909	5.8639	0.00547	13270.377
Sigmoid	Sigmoid	Resilient backpropagation	0.1	2	8	6.7324	1.326	0.00529	1245.9023
Sigmoid	Sigmoid	Resilient backpropagation	0.01	2	14	6.8431	1.3738	0.0053	1245.7101
Sigmoid	Sigmoid	Resilient backpropagation	0.001	3	3	6.4067	1.2631	0.00518	1244.4933
Sigmoid	Sigmoid	Levenberg-Marquardt backpropagation	0.1	2	11	7.108	2.2432	0.00527	1506.5944
Sigmoid	Sigmoid	Levenberg-Marquardt backpropagation	0.01	5	7	6.8171	2.185	0.00547	1542.4265
Sigmoid	Sigmoid	Levenberg-Marquardt backpropagation	0.001	2	12	7.0543	2.2414	0.00541	1503.0771
Tan-Sigmoid	linear	BFGS quasi-Newton	0.1	5	5	6.0294	2.7992	0.00538	13311.874
Tan-Sigmoid	linear	BFGS quasi-Newton	0.01	5	5	6.5736	2.8925	0.00525	13434.59
Tan-Sigmoid	linear	BFGS quasi-Newton	0.001	5	3	6.6768	2.4771	0.00523	13312.791
Tan-Sigmoid	linear	Resilient backpropagation	0.1	5	3	6.036	1.2439	0.00512	1222.5135
Tan-Sigmoid	linear	Resilient backpropagation	0.01	5	24	6.5985	1.4073	0.00522	1220.0749
Tan-Sigmoid	linear	Resilient backpropagation	0.001	3	15	5.9606	1.3763	0.00532	1226.7326
Tan-Sigmoid	linear	Levenberg-Marquardt backpropagation	0.1	2	14	6.3478	2.3747	0.00546	1958.7774
Tan-Sigmoid	linear	Levenberg-Marquardt backpropagation	0.01	2	22	6.1772	2.5817	0.00536	1956.88
Tan-Sigmoid	linear	Levenberg-Marquardt backpropagation	0.001	2	20	6.317	2.6392	0.00541	1905.957
Sigmoid	linear	BFGS quasi-Newton	0.1	5	11	6.3952	4.4533	0.00531	13720.693
Sigmoid	linear	BFGS quasi-Newton	0.01	2	5	6.3248	2.7734	0.00534	13755.963
Sigmoid	linear	BFGS quasi-Newton	0.001	5	8	6.2138	3.5443	0.00528	13848.919
Sigmoid	linear	Resilient backpropagation	0.1	4	22	6.4651	1.3917	0.0052	1227.8344
Sigmoid	linear	Resilient backpropagation	0.01	6	4	6.8795	1.2872	0.00525	1241.6193
Sigmoid	**linear**	**Resilient backpropagation**	**0.001**	**5**	**5**	**5.687**	**1.2697**	**0.00513**	**1231.0638**
Sigmoid	linear	Levenberg-Marquardt	0.1	2	21	7.3152	2.6137	0.00532	1862.0502

		backpropagation							
Sigmoid	linear	Levenberg-Marquardt backpropagation	0.01	2	21	6.532	2.6501	0.00543	1881.2463
Sigmoid	linear	Levenberg-Marquardt backpropagation	0.001	2	9	6.9728	2.1032	0.00525	1822.3828

(3) SVR models. The SVR model we applied are LS-SVR, ε-SVR and ν-SVR. We also traversed the experiments from 1 to 52 time lags for each method and we only list the top 10 results in Tables 5-7. The kernel function is RBF. For each time lag we do the parallel grid search to find the best C and γ parameters. The total searching time from 1 to 52 time lag is listed on the right side of each table, which includes the parallel grid search of C and γ parameters for each time lag. The training time and predicting time is only for the listed optimal model. We observe that the time consuming for SVR is quite less than the 2 layer NN model and that is one reason we rather choose SVR instead of NN model to combine with SAR model. In each table, we only list the top 10 results of each model.

Table 5 The LS_SVR top 10 results

Top 10	Lag s	RMSE	C	γ	Training time	predicting time	
1	14	6.3781	1	8	0.00319	0.00182	Total searching time: 27.18 seconds
2	5	6.4245	32	0.0625	0.00262	0.00127	
3	9	6.48	4	4	0.00315	0.00165	
4	8	6.596	1	8	0.00316	0.00171	
5	11	6.599	1	8	0.00321	0.00179	
6	7	6.6102	1	8	0.0031	0.0017	
7	6	6.6213	16	0.0625	0.00277	0.00116	
8	10	6.6262	4	4	0.00305	0.00165	
9	13	6.6817	1	8	0.00313	0.00175	
10	12	6.8838	2	4	0.00302	0.00167	

Table 6 The ε-SVR (ε=0.01) top 10 results

Top 10	Lag s	RMSE	C	γ	Training time	predicting time	
1	9	5.2174	8	2	0.014	0.00366	Total searching time: 110.43 seconds
2	1	5.2419	16	32	0.011	0.00252	
3	2	5.2536	128	8	0.0688	0.00281	
4	6	5.2657	256	0.0313	0.0204	0.00314	
5	4	5.2716	512	0.0313	0.0249	0.00294	
6	10	5.2815	256	0.0078	0.018	0.00343	
7	7	5.2849	512	0.0078	0.0195	0.00315	
8	3	5.3408	512	0.0078	0.0085	0.00276	
9	8	5.3439	128	0.0313	0.0188	0.0034	
10	5	5.3563	512	0.0313	0.0305	0.00299	

Table 7 The ν _SVR(ν=0.5) top 10 results

Top 10	Lag s	RMSE	C	γ	Training time	predicting time	
1	6	5.2411	512	0.00781	0.0208	0.00212	Total searching time: 555.59 seconds
2	7	5.2416	512	0.00781	0.025	0.00219	
3	2	5.2676	64	8	0.0882	0.00219	
4	9	5.2853	8	2	0.0658	0.00385	
5	3	5.3074	512	0.00781	0.0141	0.00184	
6	12	5.3309	256	0.00781	0.0197	0.00272	

7	5	5.334	512	0.00781	0.0214	0.002	
8	13	5.338	512	0.00781	0.0348	0.00295	
9	8	5.3383	512	0.00781	0.0263	0.00228	
10	4	5.3562	64	0.5	0.0509	0.00217	

(4) SARSVR models. Finally, the top 10 results of SAR-SVR methods have been listed in Tables 8-10. The part of top 3 results of different SAR+SVR methods have been listed in Table 1. We can observe that SAR+ε-SVR and SAR+ν-SVR have the similar better performances than SAR+LS_SVR, but SAR+ν-SVR has very long total searching time of 4221.08. This searching time is long but still less than the average of NN average searching time of 5543.49. If we take 4221.08 second, we can find the best performance model of SAR+ν-SVR but each total searching time of NN is just the time to find only one combination of NN structure among 36 combinations listed in Table 4. Each parameter in Tables 8-10 table has the same meaning as in Table 1.

Table 8 The top 10 results of SAR+LS-SVR

	LS-SVR+ (ν=0.5)	RMSE	Seasonal Lag s	Max lag	Window size	C	γ	Training time	Predicting time	Total searching time
1	SAR(1,1)7	5.5771	7	8	3	2	32	0.0032	0.00178	97.78 seconds
2	SAR(2,1)6	5.9199	6	8	5	1	32	0.00328	0.00182	
3	SAR(1,1)8	6.0117	8	9	3	8	16	0.00317	0.0017	
4	SAR(1,2)4	6.107	4	9	5	4	8	0.00313	0.00163	
5	SAR(1,2)2	6.4245	2	5	5	32	0.0625	0.00263	0.00125	
6	SAR(2,1)3	6.4245	3	5	5	32	0.0625	0.00267	0.00125	
7	SAR(2,1)5	6.4716	5	7	5	1	16	0.00316	0.00174	
8	SAR(2,1)9	6.7476	9	11	5	8	8	0.00307	0.00159	
9	SAR(2,1)10	6.9793	10	12	5	512	0.0625	0.00262	0.0011	
10	SAR(2,1)12	7.0299	12	14	5	512	0.0625	0.00261	0.00109	

Table 9 The top 10 results of SAR+ε-SVR

	ε-SVR+ (ν=0.5)	RMSE	Seasonal Lag s	Max lag	Window size	C	γ	Training time	Predicting time	Total searching time
1	SAR(1,2)51	3.9048	51	103	5	128	0.0625	0.0094	0.001165	604.86 seconds
2	SAR(2,2)42	4.3666	42	86	8	2	0.0625	0.003	0.001741	
3	SAR(2,2)49	4.9028	49	100	8	1	0.0625	0.0021	0.0014	
4	SAR(2,1)11	4.9307	11	13	5	1	8	0.006	0.002911	
5	SAR(2,1)12	4.9763	12	14	5	2	8	0.0068	0.002835	
6	SAR(1,2)15	5.1124	15	31	5	2	16	0.0077	0.002769	
7	SAR(2,1)10	5.1518	10	12	5	4	4	0.009	0.002878	
8	SAR(1,1)3	5.1777	3	4	3	8	4	0.0101	0.002956	
9	SAR(2,1)7	5.1994	7	9	5	4	8	0.0104	0.003143	
10	SAR(1,2)4	5.2351	4	9	5	128	0.125	0.0246	0.002851	

Table 10 The top 10 results of SAR+ν-SVR

	ν-SVR+ (ν=0.5)	RMSE	Seasonal Lag s	Max lag	Window size	C	γ	Training time	Predicting time	Total search

										ing time
1	SAR(1,2)51	3.9641	51	103	5	128	0.0625	0.01704	0.0009	4221.0 8 secon ds
2	SAR(2,2)42	4.6418	42	86	8	16	0.125	0.00803	0.0012	
3	SAR(2,2)49	4.6549	49	100	8	4	0.0625	0.00314	0.001	
4	SAR(2,1)11	4.9923	11	13	5	1	8	0.00872	0.0027	
5	SAR(2,1)12	5.0651	12	14	5	64	2	0.1265	0.0029	
6	SAR(1,1)7	5.1351	7	8	3	32	8	0.06132	0.0025	
7	SAR(2,1)10	5.1901	10	12	5	4	4	0.02274	0.0028	
8	SAR(1,2)15	5.2486	15	31	5	1	16	0.01007	0.0026	
9	SAR(2,0)2	5.2676	2	2	2	64	8	0.08641	0.0021	
10	SAR(2,0)3	5.2676	3	2	2	64	8	0.08616	0.0021	

(5) Model results comparison. It is obvious that the combined method SAR-SVR is superior to the SVR and NN models, and other classic Box-Jenkins models in accuracy. In order to compare with the different results of the prediction models, we decide to select the most similar window size. We select the best performances of 5 inputs, 5 hidden neurons with sigmoid activation functions and one output node with linear function as the comparing model. We also select ν-SVR model with input vector size of 5. There are no time lag as 5 in SAR-SVR top 10 results, so we just apply the time lag 7 as the comparison model, the SAR(1,1)7+ν-SVR(0.5) model which just apply $x_{t-1,}x_{t-7,}x_{t-8}$ as inputs. Because the prediction series have different length, we set the lost data as zeros. The comparison of the most optimal prediction results from each model groups are illustrated in Figure 7. The models are ARMA(1,1) model, NN(5-5sigmoid-1linear), ν-SVR(window size 5) and SAR(1,1)7+ν-SVR .

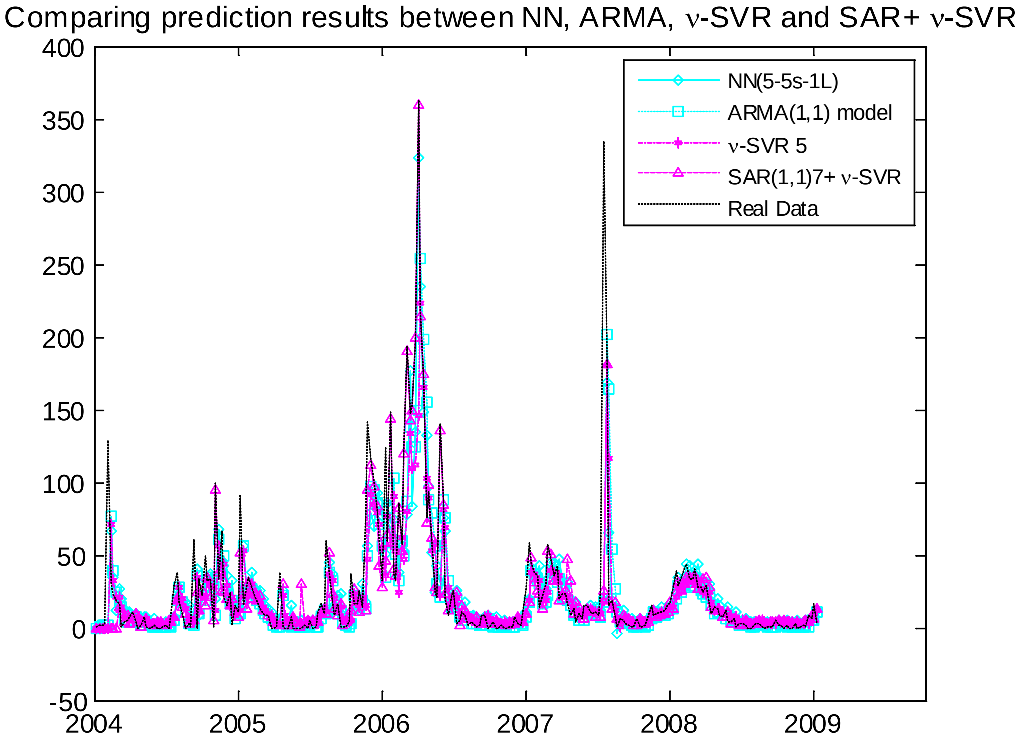
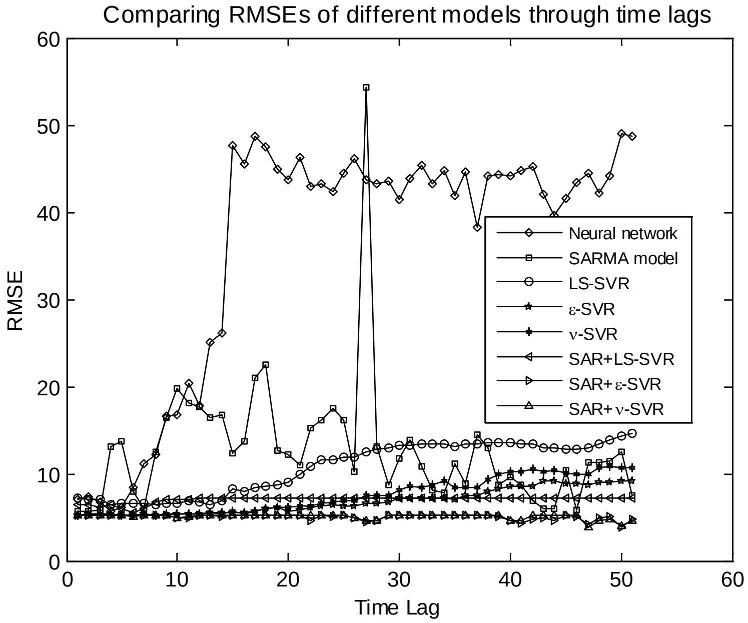
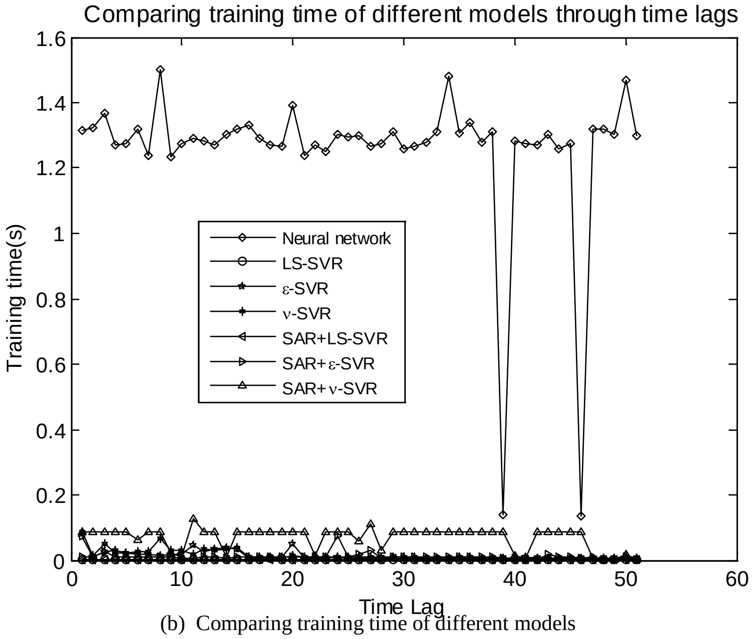


Figure 7 Comparing the prediction results of the optimal models

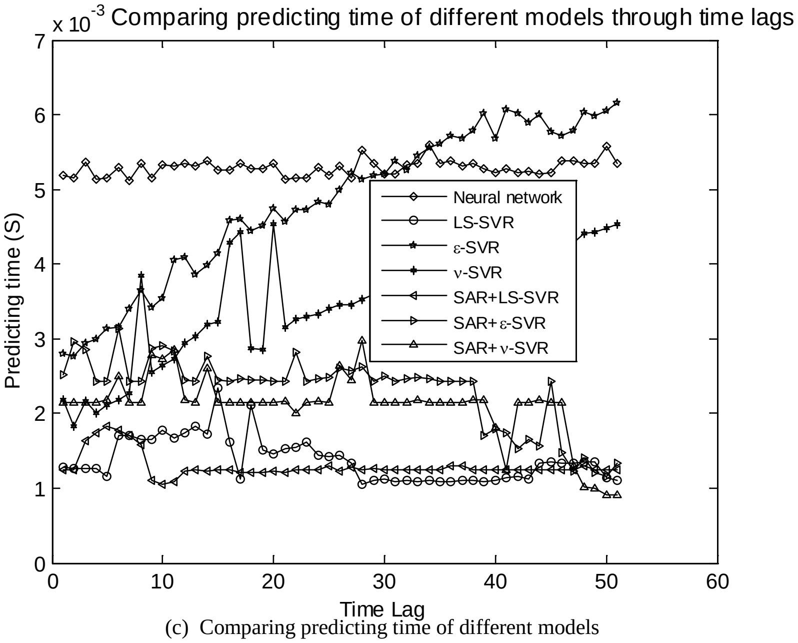
We also compare the RMSEs, training time and predicting time of optimal model results from 1 to 52 time lags in Figure 8. In Figure 8 (a), we listed the RMSE results of 1 to 52 time lags of every model in which the NN model are the model combination comprise the best performance NN with input-sigmoid-linear activation function. We can observe the NN model have small RMSEs with small time lag but it increases sharply when the time lag reach about 14. RMSEs of SVR models also increase with the time lag but slower than NN models. The RMSEs of SARMA models fluctuate with the time lags and reach a sharp peak at about 28, but the RMSE has a decrease trend. It is reasonable that RMSEs of the SAR-SVR will decrease with the time lag and achieve the optimal performances. In Figure 8(b) 8(c), the training time and the predicting time of each model is the training time and predicting time of a specific model without the time of searching the optimal parameters. The training time of NN doesn't include searching from different hidden neurons while the training time of SVR and SARSVR doesn't include the grid parameters searching time. The predicting and training time are just obtained from the optimal model to each time lag. From Figure 7, we can find out that NN model has larger training time and predicting time than SVR and SARSVR models. The training time of SARMA model is the time of estimating the model parameters to each time lag.



(a) Comparing RMSEs of different models



(b) Comparing training time of different models



(c) Comparing predicting time of different models

Figure 8 Comparing the performances different models

The summary parameters of all optimal methods are listed in Table 11, which is picked up from Tables 2-10. The listed results show that SAR+ε-SVR method can better depict the nature of the complex time series of avian influenza outbreak count series with the least RMSE and less time cost. The searching time of SAR(1,2)51+ν-SVR takes more than 4221 seconds, the longest, while the searching of SAR(1,2)51+ε-SVR(ε=0.01) only takes 604.86 second. SAR+ε-SVR is the best choice to perform the prediction tasks, which can have high accuracy and high speed. SVR models have the relative less RMSEs with very high searching speed. The classical ARMA doesn't have very high accuracy, but it only need estimate the parameters and easy to apply. NN model has the lower accuracy and it takes great time and experiences of selecting the optimal network structure and the optimal parameters. Though the optimal NN has the minimum RMSE of 5.687 but the average of NN is 6.67. The optimal NN was obtained by spending much time to exclude the unsuitable NN model structure. It means that the searching time for the optimal NN (5-5sigmoid-1linear) takes 1231.06 seconds provided we know at the beginning to apply sigmoid function in hidden neurons and linear function in output node. Otherwise we must do many other experiments to find out the optimal activation function and training method combinations. Actually, it takes a very long time to exclude other activation functions and training method combinations but still we cannot traverse all NN structures and parameters. LS-SVR has the worst accuracy but the highest speed. Finally we can conclude that SAR+ε-SVR is better choice in avian influenza counts series prediction.

Table 11 The summary of optimal results of different prediction methods

	Model descriptions	Min RMSE	Time lag	window size	Training time	Predicting time	Total searching time
ARMA	ARMA(1,1)	5.5594	n/a	n/a	n/a	n/a	n/a
SARMA	SARMA(1,0)X(1,1)3	5.6671	n/a	n/a	n/a	n/a	382.24
NN model	NN(5-5-1)	5.687	5	5	1.2697	0.00513	1231.06
SVR models	LS-SVR	6.3781	14	14	0.00319	0.00182	27.18
	ε-SVR(ε=0.01)	5.2174	9	9	0.014	0.00366	110.43
	v-SVR(v=0.5)	5.2411	6	6	0.0208	0.00212	555.59
SAR-SVR Method	SAR(1,1)7+LS-SVR	**5.5771**	7	3	0.0032	0.00178	97.78
	SAR(1,2)51+v-SVR(v=0.5)	**3.9641**	51	5	0.01704	0.0009	4221.08
	SAR(1,2)51+ε-SVR(ε=0.01)	**3.9048**	51	5	0.0094	0.001165	604.86

6. Analysis and discussions

We will conduct further analysis on the SAR-SVR methods. Comparing our SAR-SVR with the SVR model, we obtained results with different time lags from 2 to 52 (Figure 9). If we apply the SVR method, we found that RMSE increases with the window size increasing. With the proposed SAR-SVR method, the RMSE values will not always increase, but will be limited to certain values and may even decrease. It is obvious the proposed SAR-SVR is significantly better then the SVR method for the avian influenza data series. However, the question arises as to why our method limits the RMSE. We observed the results of the proposed SAR-SVR method and found that the RMSE level of values have input vectors defined by the model as *SAR(2,0)s*; therefore the time lag *s* is useless and the input vector is only $\{x_{t-1}, x_{t-2}\}$ so the prediction results will always be the same.

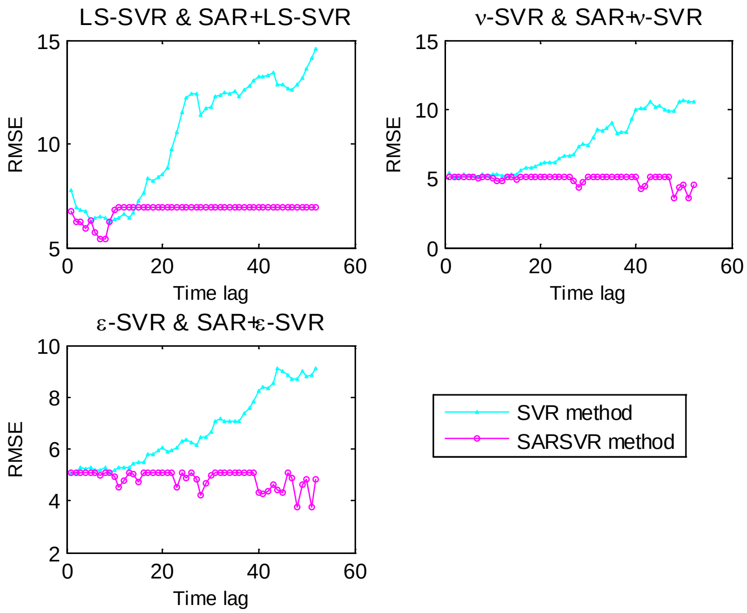


Figure 9. Comparing the proposed SAR-SVR models with SVR models

If we compare the results of SAR-SVR between with time lag in *S* and with time lags of all values from 2 to 52 in Figure 10, the red solid line indicates the results of the proposed SAR-SVR with time lags in S and the green dashed line indicates the results of SVR methods. We found that applying the proposed SAR-SVR almost reaches all the lowest point, which means that the lag set S is good enough for the prediction to reach the optimal results.

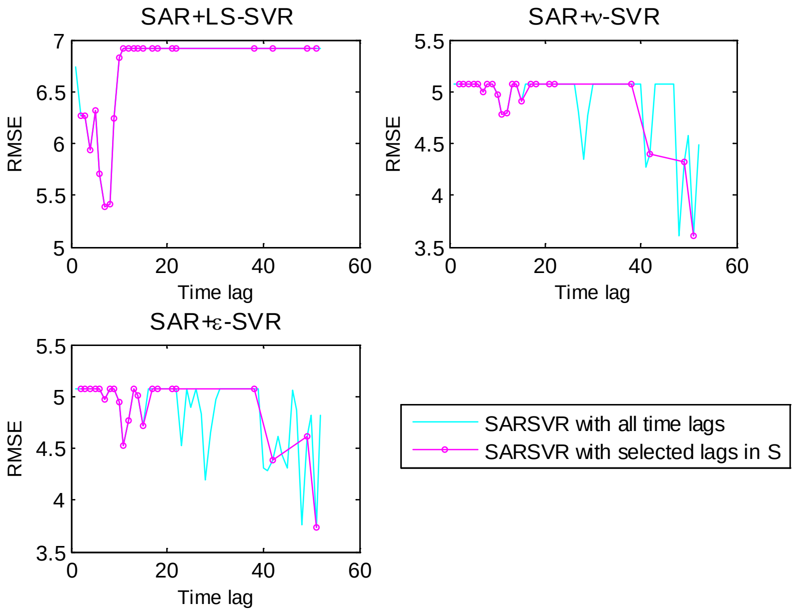


Figure 10 Comparing SAR-SVR with all time lags and SAR-SVR with selected time lags in set S.

7. Conclusions and further study

In this paper, we proposed a new SAR-SVR prediction method by combining the SAR and SVR models on the basis of FFT and ACF analysis. SVR will obtain better time series prediction if there are no seasonal behaviours. Instead of eliminating the seasonal dynamics in the series, we applied FFT and ACF values to identify the seasonality and auto-correlations to create the input vector to the SVR by the SAR model. The experiments show that the proposed SAR-SVR significantly outperforms both two layer feed forward NN and other classic methods.

The proposed SAR-SVR method is more efficient and more effective for the prediction of avian influenza count time series than other methods. Compared to SVR, the SAR-SVR method applies a suitable input vector to reach the better result instead of applying all the previous data series as input. If the time lag *s* is large, the advantage of the proposed SAR-SVR would be more observable by an even shorter input vector. Compared with 2 layer feed forward NN the proposed SAR-SVR has benefits in both accuracy and stability. The accuracy is noticeable, and moreover, the proposed SAR-SVR produces stable results with same parameters, whereas NN parameters always change after each training process.

The SVR-based method has been introduced for the first time into the avian influenza domain. Compared to other SVR models and classic models, **ν-**SVR and ε**-**SVR achieves the optimal performance, while LS-SVR is faster than the other two methods but with lowest accuracy. Most significantly, we applied SVR in the avian influenza time series for the first time and gained better performance than with the other models.

The proposed method facilitates the prediction issue of time series with complex seasonality, which is common in many practical situations. The advantage of the method is that the seasonality, or the auto-correlations inside the time series, can be identified. Above all, the method can adapt to other epidemic time series with little adjustment and it can be adopted in areas other than epidemic diseases time series such as financial time series, stock price time series, and electricity time series.

Future research and study of this method will aim to find an easy way to tune the parameters of the model to find optimal results quickly. This study did not produce optimized methods to tune the SVR prediction model, but it does demonstrate the advantages of the proposed method. For real prediction, we need to discover a quicker method to support the parameters tuning process.

Acknowledgement:

The work presented in this paper was supported by the Australian Research Council (ARC) under Discovery Project DP088739. Special thanks to Dr Declan Butler for providing an important data-set which was used in this study.

References

1. J. Liu, H. Xiao, F. Lei, Q. Zhu, K. Qin, X. W. Zhang, X. L. Zhang, D. Zhao, G. Wang, Y. Feng, J. Ma, W. Liu, J. Wang and G. F. Gao, Highly pathogenic H5N1 influenza virus infection in migratory birds, *Science* **309** (2005) 1206.
2. H. Chen, G. J. D. Smith, S. Y. Zhang, K. Qin, J. Wang, K. S. Li, R. G. Webster, J. S. M. Peiris and Y. Guan, Avian flu H5N1 virus outbreak in migratory waterfowl, *Nature* **436** (2005) 191-192.
3. M. J. Keeling and K. T. D. Eames, Networks and epidemic models, *Journal of The Royal Society Interface* **2** (2005) 295-307.
4. C. P. Jewell, T. Kypraios, R. M. Christley and G. O. Roberts, A novel approach to real-time risk prediction for emerging infectious diseases: A case study in Avian Influenza H5N1, *Preventive Veterinary Medicine* **91** (2009) 19-28.
5. T. Tiensin, M. Nielen, H. Vernooij, T. Songserm, W. Kalpravidh, S. Chotiprasatintara, A. Chaisingh, S. Wongkasemjit, K. Chanachai, W. Thanapongtham, T. Srisuvan and A. Stegeman, Transmission of the Highly Pathogenic Avian Influenza Virus H5N1 within Flocks during the 2004 Epidemic in Thailand, *The Journal of Infectious Diseases* **196** (2007) 1679-1684.
6. M. Ward, D. Maftei, C. Apostu and A. Suru, Estimation of the basic reproductive number (R0) for epidemic, highly pathogenic avian influenza subtype H5N1 spread, *Epidemiology and Infection* **137** (2009) 219.

7. A. Bouma, I. Claassen, K. Natih, D. Klinkenberg, C. A. Donnelly, G. Koch and M. van Boven, Estimation of Transmission Parameters of H5N1 Avian Influenza Virus in Chickens, *PLoS Pathog* **5** (2009).

8. H. Trottier, P. Philippe and R. Roy, Stochastic modeling of empirical time series of childhood infectious diseases data before and after mass vaccination, *Emerg Themes Epidemiol* **3** (2006) 9.

9. P. Zucs, U. Buchholz, W. Haas and H. Uphoff, Influenza associated excess mortality in Germany, 1985 - 2001, *Emerging Themes in Epidemiology* **2** (2005) 6.

10. T. A. Abeku, S. J. de Vlas, G. Borsboom, A. Teklehaimanot, A. Kebede, D. Olana, G. J. van Oortmarssen and J. D. Habbema, Forecasting malaria incidence from historical morbidity patterns in epidemic-prone areas of Ethiopia: a simple seasonal adjustment method performs best, *Trop Med Int Health* **7** (2002) 851-857.

11. D. C. Medina, S. E. Findley, B. Guindo and S. Doumbia, Forecasting Non-Stationary Diarrhea, Acute Respiratory Infection, and Malaria Time-Series in Niono, Mali, *PLoS ONE* **2** (2007) e1181.

12. G. E. P. Box and G. Jenkins, *Time Series Analysis: Forecasting and Control* (Holden-Day, San Francisco, 1970).

13. B. L. Smith, B. M. Williams and R. Keith Oswald, Comparison of parametric and nonparametric models for traffic flow forecasting, *Transportation Research Part C: Emerging Technologies* **10** (2002) 303-321.

14. R. D. Lee and L. R. Carter, Modeling and Forecasting U. S. Mortality, *Journal of the American Statistical Association* **87** (1992) 659-671.

15. A. Harvey, Trends, Cycles and Autoregressions, *The Economic Journal* **107** (1997) 192-201.

16. C. Chatfield and D. L. Prothero, Box-Jenkins Seasonal Forecasting: Problems in a Case-Study, *Journal of the Royal Statistical Society. Series A (General)* **136** (1973) 295-336.

17. S. Raza Abidi and A. Goh, Applying knowledge discovery to predict infectious disease epidemics. In *PRICAI'98: Topics in Artificial Intelligence*, 1998), pp 170-181.

18. L. Xiao-Yi, X. Zhao-Di and L. Zhen-Peng, Wavelet neural network predication for epidemic outbreak. In *Proceedings of the 21st annual international conference on Chinese control and decision conference*, (IEEE Press Guilin, China, 2009).

19. C. Vairappan, S. Gao, Z. Tang and H. Tamura, Annealed Chaotic Learning for Time Series Prediction in Improved Neuro-Fuzzy Network with Feedbacks, *International Journal of Computational Intelligence and Applications* **8** (2009) 429-444.

20. S. H. Ling, F. H. F. Leung, L. K. Wong and H.-K. Lam, Computational Intelligence Techniques for Home Electric Load Forecasting and Balancing, *International Journal of Computational Intelligence and Applications* **5** (2005) 371-392.

21. S. M. F. M. H. M. S. A. R. Moghaddamnia, Comparison of ANFIS, ANN, GARCH and ARIMA Techniques to Exchange Rate Forecasting, *Journal of Applied Sciences* **9** (2009) 10.

22. I. R. Ruiz and H. Pomares, Soft-computing techniques for time series forecasting, in *ESANN'04,* 2004, 2004) pp. 93-102.

23. R. Samsudin, A. Shabri and P. Saad, A comparison of time series forecasting using support vector machine and artificial neural network model, *J. Applied Sci.* **10** (2010) 950-958.

24. A. Hossain, F. Zaman, M. Nasser and M. Islam, Comparison of GARCH, Neural Network and Support Vector Machine in Financial Time Series Prediction. In *Pattern Recognition and Machine Intelligence*, 2009), pp 597-602.

25. M. Bramer, S. Crone, J. Guajardo and R. Weber, A study on the ability of Support Vector Regression and Neural Networks to Forecast Basic Time Series Patterns. In *Artificial Intelligence in Theory and Practice*, (Springer Boston 2006), pp 149-158.

26. W. Chun-Hsin, H. Jan-Ming and D. T. Lee, Travel-time prediction with support vector regression, *Intelligent Transportation Systems, IEEE Transactions on* **5** (2004) 276-281.

27. K.-j. Kim, Financial time series forecasting using support vector machines, *Neurocomputing* **55** (2003) 307-319.

28. F. E. H. Tay and L. Cao, Application of support vector machines in financial time series forecasting, *Omega* **29** (2001) 309-317.

29. W.-C. Hong, Electric load forecasting by support vector model, *Applied Mathematical Modelling* **33** (2009) 2444-2454.

30. Z. Yuan, Better prediction of protein contact number using a support vector regression analysis of amino acid sequence, *BMC Bioinformatics* **6** (2005) 248.

31. J. Wu, M. Liu and L. Jin, A Hybrid Support Vector Regression Approach for Rainfall Forecasting Using Particle Swarm Optimization and Projection Pursuit Technology, *International Journal of Computational Intelligence and Applications* **9** (2010) 87-104.

32. V. Fernandez, Wavelet- and SVM-based forecasts: An analysis of the U.S. metal and materials manufacturing industry, *Resources Policy* **32** 80-89.

33. A. C. Harvey and N. Shephard, Structural time series models. In *Handbook of Statistics*, (Elsevier 1993), pp 261-302.

34. B. Cazelles, M. Chavez, G. C. d. Magny, J.-F. Guegan and S. Hales, Time-dependent spectral analysis of epidemiological time-series with wavelets, *Journal of The Royal Society Interface* **4** (2007) 625-636.

35. G. Constantin de Magny, J.-F. Guegan, M. Petit and B. Cazelles, Regional-scale climate-variability synchrony of cholera epidemics in West Africa, *BMC Infectious Diseases* **7** (2007) 20.

36. J. A. K. Suykens, J. De Brabanter, L. Lukas and J. Vandewalle, Weighted least squares support vector machines: robustness and sparse approximation, *Neurocomputing* **48** (2002) 85-105.

37. V. N. Vapnik, *Statistical Learning Theory* (Wiley, New York, 1998).

38. S. Bernhard, lkopf, J. S. Alex, C. W. Robert and L. B. Peter, New Support Vector Algorithms, *Neural Comput.* **12** (2000) 1207-1245.

39. G. E. P. Box and N. R. Draper, Empirical model-building and response surfaces / George E. P. Box, Norman R. Draper. In (New York Singapore : John Wiley 1987).

40. L. Stone, R. Olinky and A. Huppert, Seasonal dynamics of recurrent epidemics, *Nature* **446** (2007) 533-536.

41. Y. L. Si, P. Debba, A. K. Skidmore, A. G. Toxopeus and L. Li, Spatial and temporal patterns of global H5N1 outbreaks. In (International Society for Photogrammetry and Remote Sensing 2008).

42. D. U. Pfeiffer, P. Q. Minh, V. Martin, M. Epprecht and M. J. Otte, An analysis of the spatial and temporal patterns of highly pathogenic avian influenza occurrence in Vietnam using national surveillance data, *The Veterinary Journal* **174** (2007) 302-309.

43. M. Gauthier-Clerc, C. Lebarbenchon and F. Thomas, Recent expansion of highly pathogenic avian influenza H5N1: a critical review, *Ibis* **149** (2007) 202-214.

44. C.-c. Chang and C.-J. Lin *LIBSVM: a Library for Support Vector Machines*, 2001.

45. V. Cherkassky and Y. Ma, Practical selection of SVM parameters and noise estimation for SVM regression, *Neural Networks* **17** (2004) 113-126.