Multi-step Weekly Average Forecasting of Reservoir Storage Volume Using Deep Learning

Zachary Herbert[1], Zeeshan Asghar[2], Carlos A. Oroza[2]

[1]Central Utah Water Conservancy District
[2]Dept. Civil and Environmental Engineering, University of Utah

Key Points:

- A deep learning algorithm is developed for multi-step reservoir storage volume forecasting from snow water equivalent.
- The algorithm has higher forecast accuracy compared to common statistical methods (SARIMA, VAR, and TBATS).
- The algorithm performed best for years with large runoff; worst for years with small runoff and late-season snow accumulation.

Corresponding author: Zachary Herbert, `zachary@cuwcd.com`

Abstract

Machine-learning algorithms have shown promise for streamflow forecasts, reservoir operations, and scheduling, but have exhibited lower accuracy in predicting extended time horizons of peak storage volume (PSV). Deep learning algorithms exhibited improved inflow forecasting accuracy, but existing research has been mostly limited to real-time operation and short-term planning. We evaluate a new approach based on a hybrid ResCNN-LSTM Encoder-Decoder algorithm, enabling long-term multi-step reservoir forecasts. The proposed approach provides a three-month, weekly averaged prediction of reservoir storage volume (RSV) during the runoff season based on historical snow water equivalent (SWE). The optimal architecture and hyper-parameters for the model are configured through five-fold cross validation resulting in a twelve-layered residual convolutional neural network (ResCNN) as the encoder and a four-layered long short-term memory (LSTM) neural network as the decoder. We evaluate the algorithm using 30 years of RSV and SWE data at the Upper Stillwater Reservoir located in Utah. The most accurate long-term predictions occurred during periods of large runoff (in excess of 28,000 ac-ft). The periods where the model performed the worst were during small runoff and late-season SWE accumulation. We find that the ResCNN-LSTM consistently outperforms three widely used statistical models, with an average PSV absolute percent error of 2.66% for the proposed algorithm compared to SARIMA (14.22%), TBATS (13.82%), and VAR (18.14%).

1 Introduction

Classical statistical models such as Seasonal Auto-regressive Integrated Moving Average (SARIMA) (Papamichail & Georgiou, 2001), Vector Auto-regression (VAR) (Iddrisu et al., 2016) and Trigonometric Seasonal Box-Cox Transformation with ARMA residuals, Trend, and Seasonal Components (TBATS) (Elizaga et al., 2014) have long been employed for reservoir storage and outflow prediction. These models are well-suited to short-term forecasts, but have limited capacity for long-term forecasts due to the convergence of the auto-regressive part of the model to the mean of the time series (Shumway & Stoffer, 2000). Forecast skill is also confounded by hydro-meteorological predictability in snow-dominated catchments (Anghileri et al., 2016). Machine-learning algorithms provide an alternative approach and are increasingly being used in a variety of related hydrologic fields: rainfall-runoff prediction for ungauged basins (Kratzert et al., 2019), hydropower production forecasting (Stokelj et al., 2002), spatial snow water equivalent (SWE) estimation for mountainous areas (Zheng et al., 2018), and quantifying climate and catchment control on hydrological drought (Konapala & Mishra, 2020).

Machine learning has seen broad application to reservoir forecasting in both direct and multi-step scenarios, and generally resulted in more reliable forecasts of inflow extremes. Coulibaly et al. (2000) trained a feed-forward neural network using an early stopping training approach for real-time reservoir inflow forecasting with lead times of one to seven days. An improvement to daily reservoir inflow forecasts was later made using a robust weighted-average ensemble that takes advantage of three different models: nearest neighbors, a conceptual model, and an artificial neural network (Coulibaly et al., 2005). An additive ensemble for monthly reservoir inflow forecasting was developed by Bai et al. (2015), incorporating an auto-regressive model, least squares support vector machine, and adaptive neuro-fuzzy inference system to subforecast trend, period, and stochastic terms. Bourdin et al. (2014), Wang et al. (2012), and Ahmed et al. (2015) all used ensemble learning methods coupled with meteorological predictions to forecast reservoir inflows with respective forecast horizons of three, eight, and fourteen days. Long-range streamflow forecasts, extending twelve months, were assessed by Bennett et al. (2016) in which calibrated climate forecasts are combined with a conceptual runoff model and a three-staged error model to simulate reservoir inflows. Similarly, Y. Liu et al. (2017) developed a long-term streamflow forecasting scheme, extending nine months, utilizing

Random Forest and support vector regression for precipitation post-processing of numerical weather predictions to feed into a hydrological watershed model.

Research with deep learning algorithms applied to reservoir inflow forecasting has found improved forecast accuracy, but has been mostly limited to real-time operation and short-term planning. Deep belief networks for multiscale feature learning (Bai et al., 2016) improved on prior efforts (Bai et al., 2015) in direct-step forecasting using an additive ensemble approach. Budu (2014) and Chiamsathit et al. (2016) both applied multilayered perceptrons for direct-step forecasting scenarios that achieved reasonable accuracy among daily and monthly timesteps, respectively. Time-lagged recurrent neural networks (TLRNs) have been studied (Muluye & Coulibaly, 2007; Kote & Jothiprakash, 2009; Sattari et al., 2012) where a preceding record of reservoir inflow is used to investigate the performance of a back-propagation through time (BPTT) algorithm. Better prediction of inflow into a reservoir using TLRN was achieved by Kote and Jothiprakash (2009) by modifying the artificial neural network to include seasonal (monsoon) effects, accurate mapping of high and low flows was achieved following a monthly time-step.

However, attempts at long-range forecasts with deep learning algorithms have generally exhibited lower accuracy. Similar to Bennett et al. (2016) and Y. Liu et al. (2017), earlier attempts at long-term forecasting (Muluye & Coulibaly, 2007; Kote & Jothiprakash, 2009) failed to accurately predict the peak storage volume (PSV) at extended time horizons. More recently, multi-step flood forecast models (Chang & Tsai, 2016; Zhou et al., 2019; Kao et al., 2020) have been developed for predicting reservoir inflows using adaptive neuro-fuzzy inference systems and long short-term memory (LSTM) based Encoder-Decoder frameworks. However, the timestep for these forecasts are hourly with forecast horizons only extending four, six, and eight hours.

We evaluate a new approach to multi-step weekly average forecasting of reservoir storage volumes (RSV) based on historical regional SWE data. This approach aims to use existing SWE and RSV time series data to train a hybrid, multi-step ResCNN-LSTM (Encoder-Decoder) (LeCun, 1990; Hochreiter & Schmidhuber, 1997) algorithm, to predict the RSV of future timesteps for the proceeding three months. In contrast to RNNs, convolutional neural networks (CNN) operate independently of previous time steps to capture fixed size contexts, allowing for parallel computation within a given sequence. The stacking of convolutional layers allows for precise control of the dependencies to be modeled by effectively increasing the context size (Gehring et al., 2017). Implementing residual connections in CNNs has been shown to improve model performance by increasing the depth of the model architecture (He et al., 2015). Deep learning frameworks utilizing residual connections in CNNs have seen extensive application in other fields (H. Liu & Song, 2018; Ning et al., 2019; Cengil & Cinar, 2018), but have not yet been evaluated in the context of RSV forecasting. Thus, a key distinction between the proposed model in this paper and the others listed above is that a deep residual CNN (ResCNN) is initially used for feature extraction, which may improve long-range forecast accuracy. The specific aims of this study are:

1. Assess the accuracy of the ResCNN-LSTM algorithm in predicting RSV from SWE.
2. Determine characteristics of years in which the algorithm exhibits high vs. low accuracy.
3. Compare the long-term accuracy of the proposed algorithm against three commonly used RSV forecast methods: SARIMA, VAR, and TBATS.

2 Materials and Methods

2.1 Study Region

This study focuses on the Upper Stillwater reservoir located at the top of the Central Utah Water Conservancy District's (CUWCD) collection system in the Uinta Mountains. CUWCD is one of Utah's four large specialty water districts that provides potable and secondary water to various water associations, conservancy districts, irrigation companies, and local residents. The water district spans eight counties with over $3.5 billion in infrastructure. There are currently ten lakes/reservoirs maintained and operated by CUWCD that house non-potable water in excess of 1.6 million ac-ft. The storage levels for these reservoirs act as a barometer for the state's water resources and provide insight for how to appropriately prepare for future water usage. Figure 1 shows Upper Stillwater (located in the middle of the figure) surrounded by a network of snow telemetry monitoring sites.

2.2 Data

The data used for model training is accessed from the National Resource Conservation Service (NRCS) online Application Programming Interface (NRCS, 2020 (accessed

May 3, 2020)). The two data types used for this study are RSV and SWE (the depth of water (in) that would result if the snowpack were melted). Specific to Upper Stillwater Reservoir, the historical daily data span from January 1990 to the present with new values updated daily. Linear time interpolation was used for gap-filling a limited number of missing data points within the time series. From January 1st 1990 to July 14th 2019, a total of 16 missing daily data points, out of 10,787, required interpolation. These points occurred in 1990, during the first year of data collection.

The RSV time series for Upper Stillwater illustrates a seasonal runoff period, fed by snowmelt, that begins in April and ends in July. A governing assumption of this study is that water managers only have until the end of March to make a final decision regarding the level of storage space to leave vacant in the reservoir for the runoff season. Therefore, the critical period to forecast RSV is a 15-week window between the first of April and the beginning of July. Such an extended forecast horizon requires a model capable of learning long-term dependencies.

SWE data were collected from the NRCS monitoring network of snow telemetry sites for the same period as Upper Stillwater's RSV time series. For each of the available monitoring sites, a maximum of three daily data points required interpolation over the entire period. The dependence of RSV (Figure 2) on SWE (Figure 3) is the primary relationship that the model will attempt to learn.

The daily data were then prepared for training the algorithm. The data were first re-sampled into weekly averages and scaled between 0 and 1 based on the chosen activation function for the model (see Section 2.3). A variety of sliding window lengths (15, 20, and 25 weeks) were then used as inputs to predict the next 15 weeks. This range of window sizes is selected based on the time series data for SWE (depending on the precipitation distribution during a given winter season, the process of SWE accumulation ranges between 15 to 25 weeks). The input window length that yields the greatest performance is selected for the final model.

2.3 Deep Learning Model

The objective of the model is to forecast multiple timesteps forward based on multiple inputs from the past. The inputs are the multiple time series of RSV and SWE and the output is a future RSV sequence prediction starting at the final point in the input data. Therefore, a multivariate sequence-to-sequence prediction model is required. This type of model is broken down into two separate models: one for reading the input sequence and encoding it into a fixed-length vector (Encoder), and a second model for decoding the fixed-length vector and outputting the predicted sequence (Decoder) (Sutskever et al., 2014). Following the Decoder, a time-distributed fully connected layer is used as the final component to condense the output and yield a forecasted sequence of values.

Figure 4 illustrates the architecture of this model, which receives a sliding window input of multiple variables and transmits a sliding window output of a single target vector. This architecture has proven to be effective for numerous sequence-to-sequence problems, including multi-step flood forecasting (Kao et al., 2020), network traffic forecasting (Zhang & You, 2020), weather forecasting (Yuan et al., 2019), and predicting solar performance ratio (Yen et al., 2019). The Encoder-Decoder model is written in Python using the Keras deep-learning library (Gulli & Pal, 2017). Another important feature of the proposed model is how each node sequentially transmits information to the next within each layer. This process is governed by the use of a piece-wise linear activation function. A node or unit that implements this activation function is referred to as a rectified linear unit (ReLu) (Agarap, 2018). The benefit of using ReLu is two-fold: it allows for faster training time due to its near-linear properties, while also addressing the vanishing gradient problem during back-propagation of errors. The Adam optimizer (Kingma & Ba, 2014) is used to adaptively optimize the weights within the network using concepts of momentum (Sutskever et al., 2013) and stochastic gradient descent (Robbins & Monro, 1951).

2.3.1 CNN Encoder

A deep ResCNN is used as the encoder in the Encoder-Decoder architecture. A one-dimensional CNN is a model with one or more hidden layers that operate over a 1D sequence (e.g. sentence or time series) through convolutions. In a multi-layer CNN, the stacking of CNN layers creates a hierarchical structure that provides a shorter path to capture long-range dependencies. Therefore, the model creates hierarchical representations over the input sequence allowing nearby input elements to interact at lower layers while distant elements interact at higher layers (Gehring et al., 2017). Causal padding (van den Oord et al., 2016) is used for each CNN layer to ensure the model does not violate the temporal order (i.e. model does not have look-ahead bias). Residual connections are used between stacked CNN layers to boost model performance with increased network depth (He et al., 2015). A residual block is defined with a single skip connection between CNN layers (Figure 5). Within a residual block, the output from the first CNN layer is passed through the activation function (ReLU) and sent in two separate paths: forward to the next CNN layer in sequence, and around to skip the next CNN layer as a residual from the previous. The residual is added to the output of the second CNN layer prior to its respective output being activated through ReLU. Thus, the model is optimized for a residual mapping of feature extraction from the sliding window of time series input.

The encoder is designed to gradually reduce the dimensionality of the input feature matrix while increasing the number of feature abstractions. This is done using filters (see Section 2.3.3) and pooling layers, whose purpose is to condense a CNN layer's output to the most prominent elements. Max pooling is used at the end of each residual section (Figure 5), in which two residual blocks are connected in sequence for a given number of filters and kernel size. Max pooling and flatten are used at the end of the encoder to downsize the extracted features into a fixed length vector proportional to the number of nodes in the decoder. The final output represents the extracted elements as features from the input sequences that will be fed as a flattened sequence for the decoder.

2.3.2 RNN Decoder

The extracted features from the encoder are fed into the decoder to yield a forecasted sequence of values. This is done using an RNN capable of learning long-term dependencies, which is analogous to context for the case of sequence-to-sequence prediction. RNNs are well suited for time series data as they process each timestep sequentially for modelling non-linear relationships between the input and the output. This is achieved by forming recurrent cycles within the nodes/cells of each hidden layer (Kao et al., 2020). In the case of runoff from snowmelt, a typical approach for determining the type of runoff season to expect is to compare the current snowpack with previous years. The process of observing current data and recalling previous significant events and their outcomes is what the RNN attempts to mimic. However, RNNs fail to connect information from the past (input) to the present (output) data when the gap between the two grows too large, which creates the issue of long-term dependencies. This problem is addressed with the use of LSTM networks (Hochreiter & Schmidhuber, 1997), a special kind of RNN. An essential feature of the LSTM cell is its state that runs directly through the network, enabling addition or removal of information from the cell state via regulated gates. These internal gates are weighted functions that govern how the information flows in the cell, mitigating the vanishing gradient problem.

2.3.3 Model training and assessment

The optimal hyper-parameters for the model (i.e. layers, nodes, filters, kernels, input window, epochs, and batch size, see Table 1) are determined through five-fold cross validation. Layers are arrays of nodes that sequentially transmit information from one to the next. Within each layer are nodes connected by multiple weights for a given number of inputs and outputs. A single node receives input data, processes the input as a weighted sum, and then propagates new information to its successor based on a given activation function. Filters and kernels are interrelated hyper-parameters specific to CNNs. A kernel represents a matrix of weights that slide over the input sequence calculating the dot-product between the sequence values and matrix weights. Therefore, the size of the kernel represents the length of the window it spans for deep feature extraction. A complete tour of a kernel over the input sequence represents a filter; thus, kernel's operating over multiple channels of input establish a filter/feature map. The input window represents the multivariate time series of RSV and SWE accumulation during the winter season until the first of April. Epochs represent the number of full passes of the data set that the model uses during training. The batch size is the fraction of data that the model is exposed to during each epoch. An early stopping algorithm is used during training to prevent over-fitting with excessive epochs: a training session will terminate early if there are 10 consecutive epochs with no improvement in minimizing the mean squared error.

For a given input window length, the training data for each test year spans from Jan 1990 to mid October or November of the previous year. For example, the training data set for the 2015 runoff period with a 20-week input window begins January 1990 and ends November 2014. 80% of the data are used for training, while 20% are used for validation. This inner split between training and validation data allows for learning curves to be developed (to evaluate signs of over vs under-fitting). Across all five test years, the model is trained one configuration at a time to develop the regression metrics: mean absolute error (MAE), root mean squared error (RMSE), median absolute error (MedAE), Nash-Sutcliffe model efficiency coefficient (NSE), and explained variance (ExpVar). The regression metrics are calculated from a held-out data set that is not used during model training. The hold-out set spans 30-40 weeks and consists of two consecutive parts: input (15-25 weeks) and output (15 weeks). The input spans from October or November through May; the output from April through June. For example, the hold-out data set used to forecast the 2015 runoff period with a 20-week input window begins in November 2014 and ends at the start of April 2015.

Finally, a confidence interval is calculated from multiple model runs. Due to the stochastic nature of the model, a slightly different forecast will be returned each time the model is trained. Therefore, for a given configuration, the model is trained multiple times to establish a normal distribution of model predictions for each time-step in the forecast. By design, forecast points will be labeled as outliers if they lie beyond the whiskers of their respective boxplot following the Tukey method (Tukey, 1970).

The forecast plots and regression metrics provide valuable insight into the overall accuracy of the ResCNN-LSTM; however, the primary concern is the total expected runoff (TER). This can be defined as the change in RSV from the end of March to the beginning of July. The TER is the amount of water that is expected to fill the reservoir during the critical runoff period. Hence, the statistics of greatest value lie at the crest of each forecast curve (i.e. the PSV).

2.4 Statistical Methods for Comparison

The forecasts are compared against three widely used statistical models: Seasonal Auto-regressive Integrated Moving Average (SARIMA), Vector Auto-regression (VAR), and Trigonometric Seasonal Box-Cox Transformation with ARMA residuals, Trend, and Seasonal Components (TBATS). Each model is trained on monthly averaged data; therefore, their forecasts for PSV are the maximum monthly average during the runoff period. This change in the timestep frequency is due to the limitations of the statistical models to forecast into such extended horizons.

2.4.1 SARIMA Model

A discrete time series $\mathrm{Z_1, Z_2, Z_3, \ldots, Z_{N-1}, Z_N}$ of measurements at equal time intervals is simulated by a stochastic SARIMA model (Box et al., 2015) given by:

$$\varphi(\mathrm{B})\Phi\left(\mathrm{B^S}\right)(1-\mathrm{B})^{\mathrm{d}\left(1-\mathrm{B^S}\right)\mathrm{DZ_t}} = \theta(\mathrm{B})\Theta\left(\mathrm{B^S}\right)\mathrm{e_t} \tag{1}$$

Here, t represents the discrete time and S denotes the length of each season. The B term corresponds to the backward shift operator which is defined by $\mathrm{BZ_t = Z_{t-1}}$ and $\mathrm{B^S Z_t = Z_{t-S}}$. The independently and normally distributed white noise residual is represented by $\mathrm{e_t} = \left[\mathrm{NID}\left(0, \sigma_{\mathrm{e}}^2\right)\right]$ which has a zero mean, and variance defined by σ_{e}^2. From the left hand side of equation 1, the first two terms φ and Φ represent series expansions given by:

$$\varphi(\mathrm{B}) = 1 - \varphi_1\mathrm{B} - \varphi_2\mathrm{B}^2 - \ldots - \varphi_{\mathrm{p}}\mathrm{B^p} \tag{2}$$

$$\Phi(\mathrm{B^S}) = 1 - \Phi_1\mathrm{B^S} - \Phi_2\mathrm{B^{2S}} - \ldots - \Phi_{\mathrm{p}}\mathrm{B^{PS}} \tag{3}$$

Equation 2 represents the nonseasonal auto-regressive operator of order p and $\varphi_i, = 1, 2, \ldots, p$ depicts the nonseasonal auto-regressive parameters. $(1-B)^d$ is the nonseasonal difference operator of order d which produces nonseasonal stationarity of the dth differenced data, usually d = 0, 1, or 2. Whereas, equation 3 depicts the seasonal auto-regressive operator of order P and Φ_i; herein, i= 1,2, ... , P are the seasonal auto-regressive parameters. $\left(1-B^S\right)D$ is the seasonal differencing operator of the order D to produce seasonal stationarity of the Dth differenced data, usually in the order of D = 0, 1, or 2. From the right hand side of equation 1, the first two terms θ and Θ represent series expansions given by:

$$\theta(\mathrm{B}) = 1 - \theta_1\mathrm{B} - \theta_2\mathrm{B}^2 - \ldots - \theta_{\mathrm{q}}\mathrm{B^q} \tag{4}$$

$$\Theta\left(B^S\right) = 1 - \Theta_1 B^S - \Theta_2 B^{2S} - \ldots - \Theta_Q B^{QS} \tag{5}$$

Equation 4 is the nonseasonal moving average operator of the order q; thus, equation 4 and q are the nonseasonal moving average parameters. Equation 5 is the seasonal moving average operator of order Q and $\Theta_i, i = 1, 2, \ldots, Q$ are the seasonal moving average parameters. Lastly, the natural log of the RSV time series is taken to stabilize the variance of the time series and to transform any skew in the distribution into a normal distribution (Papamichail & Georgiou, 2001). Using an annual seasonal term (S) of 12 months, the SARIMA model parameters (p,d,q)(P,D,Q) are configured using the autoarima function provided in the Pmdarima statistical library in Python 3.0.

2.4.2 VAR Model

VAR is another frequently used model for multivariate time series. The basic VAR model of order p, as suggested by Sims (1980) is given by

$$y_t = A_1 y_{t-1} + A_2 y_{t-2} + \cdots + A_p y_{t-p} + CD_t + u_t \tag{6}$$

Where $y_t = (y_{1t}, y_{2t}, \ldots, y_{Kt})'$ represents a vector of K observable endogenous variables and D_t consists of all deterministic variables which carry a constant, a linear trend, seasonal dummy variables and user-specified variables. u_t is a K-dimensional unobservable, zero-mean, white noise process which has a positive definite co-variance matrix $E\left(u_t u_t'\right) = \sum_u \cdot A_i$ and C are parameter matrices of suitable dimension upon which various restrictions can be imposed. For a K-dimensional auto-regression with an effective sample size N, the optimal lag order p is selected that minimizes the Akaike Information Criteria (AIC) given by the following equation:

$$AIC(p) = ln|\bar{\sum}(p)| + \frac{2}{N}(K^2 p) \tag{7}$$

The $\bar{\sum}(p)$ is the quasi-maximum likelihood estimate of the innovation covariance matrix $\sum(p)$ (Ivanov & Kilian, 2005; Sin & White, 1996). The parameters in equation 6 are estimated by the method of generalized least squares. This is done by first estimating the individual equations of the system by ordinary least squares. The residuals can then be utilized to estimate the white noise co-variance matrix $\sum_u$ as $\widehat{\sum_u} = T^{-1} \sum_{t=1}^{T} \widehat{u}_t \widehat{u}_t$ which is used to compute the generalized least square estimator (Iddrisu et al., 2016). The VAR model is developed through the statsmodels Python module and utilizes the same SWE data as the ResCNN-LSTM.

2.4.3 TBATS Model

BATS is a combination of three methodologies; (i) Exponential Smoothing Method, (ii) Box-Cox Transformation and (iii) ARMA model for residuals. The Box-Cox Transformation helps to deal with non-linear data and ARMA model for residuals can de-correlate the time series data. However, the BATS model does not do well when the seasonality is complex with high frequency. Thus, De Livera et al. (2011) proposed a TBATS model, which includes the trigonometric seasonal component. The trigonometric expression of seasonality terms serves to reduce the parameters of model when seasonal frequencies (e.g. annual streamflow or annual peak flow) are high and improves the model flexibility (i.e. lower bias with higher variance), enabling it to handle complex seasonality.

The TBATS model is comprised of the following terms:

$$
\begin{aligned}
&y_t^{(\lambda)} = l_{t-1} + \phi b_{t-1} + \textstyle\sum_{i=1}^{T} s_{t-m_i}^{(i)} + d_t \\
&l_t = l_{t-1} + \phi b_{t-1} + \alpha d_t \\
&b_t = \phi b_{t-1} + \beta d_t \\
&d_t = \textstyle\sum_{i=1}^{p} \varphi_i d_{t-i} + \sum_{i=1}^{q} \theta_i e_{t-i} + e_t
\end{aligned}
$$

Where:

$y_t^{(\lambda)} -$ time series at moment t (Box-Cox transformed)

$s_t^{(i)} - i$ th seasonal component

l_t - local level

b_t - trend with damping

$d_t - ARMA(p, q)$ process for residuals

e_t - Gaussian white noise

Seasonal part:

$$
\begin{aligned}
&s_t^{(i)} = \textstyle\sum_{j=1}^{(k_i)} s_{j,t}^{(i)} \\
&s_{j,t}^{(i)} = s_{j,t-1}^{(i)} \cos(\omega_i) + s_{j,t-1}^{*(i)} \sin(\omega_i) + \gamma_1^{(i)} d_t \\
&s_{j,t}^{*(i)} = -s_{j,t-1}^{(i)} \sin(\omega_i) + s_{j,t-1}^{*(i)} \cos(\omega_i) + \gamma_2^{(i)} d_t \\
&\omega_i = 2\pi j / m_i
\end{aligned}
$$

Model parameters:

T - Amount of seasonalities

m_i - Length of ith seasonal period

k_i - Amount of harmonics for ith seasonal period

λ - Box-Cox transformation

$\alpha, \beta-$ Smoothing

ϕ - Trend damping

$\varphi_i, \theta_i - ARMA(p, q)$ coefficients

$\gamma_1^{(i)}, \gamma_2^{(i)}$ - Seasonal smoothing (two for each period)

Based on a Fourier series, each seasonality is modeled by a trigonometric representation. Advantages of this model framework consist of (i) improved parameter estimation and (ii) an ability to handle complex components and features intrinsic to the time series. As an innovations state-space model, TBATS admits a larger parameter space with the possibility of better forecasts (Hyndman, 2008). Likewise, TBATS involves a much simpler and efficient estimation procedure than other state-space models (e.g. SARIMA and VAR). With the use of trigonometric functions, the model is capable of handling non-integer seasonal frequencies (e.g. 365.25 days in a year), while also accommodating both nested and non-nested seasonal components. Additionally, the model handles nonlinear features typically seen in time series while taking into account any auto-correlation within the residuals (De Livera et al., 2011). The TBATS model is implemented using the tbats package in Python 3.0 and incorporates quarterly, bi-annual, and annual seasonal periods.

3 Results

3.1 Model Architecture

The optimal architecture for the ResCNN-LSTM was found to be a twelve-layered ResCNN as the encoder and a four-layered LSTM as the decoder. Through five-fold cross validation, the optimal number of filters in each ResCNN section was found to be 16 (filter1), 32 (filter2), and 64 (filter3), while the LSTM layers required 64 nodes/cells per layer. The time-distributed fully connected layer is configured with 32 nodes to begin condensing the output from the decoder. Table 1 outlines the optimal configuration and architecture for the model. The maximum number of epochs is set to 100, but the average final converged number ranges between 90 and 100 for a given test year.

3.2 Model Performance

This section is organized by each test year starting from 2015 and ending in 2019. During each test year, the date range for each time series input stems from the second week in November through the end of March. Figure 6 illustrates the forecasted runoff period which is composed with the actual values (solid green), forecasted values (dotted red), reservoir spill limit (dashed black), 95% confidence interval (shaded grey) and boxplots. The box plots and regression metrics are based on a sample population of 30 individual model trainings per test year. Regression metrics for the ResCNN-LSTM forecasts are shown in Table 2 below. Results from the analysis of statistical models are shown in Figure 7. The graphs represent the storage volume forecasts for all three models across the five consecutive test years.

Regardless of the low snowpack, the 2015 forecast proved to be very accurate. In addition to the low errors summarized in Table 2, the 15-week forecast nearly managed to identify the exact week that the reservoir was going to fill. The forecast's accuracy is highest near the end of the period and lowest in the beginning. The 2016 forecast is an improvement over the previous year, with smaller errors for the MAE, RMSE, and MedAE, and improved values for NSE and ExpVar. The 2015 and 2016 forecasts are more skillful in the long-term projections occurring after May rather than earlier in April. Among all of the test years, the forecasts for 2015, 2016 and 2019 were the most accurate with the lowest reported errors. The forecast for 2017 fails to identify when the RSV will first increase, along with when the storage volume will peak. However, the forecast still manages to achieve an accurate prediction for the final storage volume at the end of the period. Similar to the forecast for 2017, the 2018 forecast fails to identify when the RSV will first increase, along with when the storage volume will peak. On the other hand, the forecast maintains an accurate prediction for the final expected volume in the reservoir. The observed runoff volumes for all other test years were in excess of 28,000 ac-ft, whereas 2018 demonstrated a runoff volume on the low end of 18,000 ac-ft.

Results from the other statistical models are shown in Figure 7. The graphs represent the monthly storage volume forecasts for all three models across the five consecutive test years. SARIMA and TBATS exhibited higher accuracy in capturing seasonal fluctuations than VAR, as shown respectively in 2015, 2017, and 2019; and 2016, 2018, and 2019. VAR generally demonstrated the lowest accuracy but occasionally excelled in forecasting extreme inflow events (e.g. in 2017 and 2019). Overall, ResCNN-LSTM outperforms the statistical methods except for the 2018 test year (Figures 6 and 7, and Table 2). The ResCNN-LSTM produces accurate PSV forecasts for each test year, which lie within the distribution bounds of the batch forecasts (Figure 8, comparing the PSV for each test year in comparison to the observed amount). The distributions for the PSV are skewed upwards towards the reservoir spill limit during each test year. Table 3 summarizes the absolute percent errors at the edges and center of the confidence interval for each test year, along with forecasts from the statistically-based models. With the exception of SARIMA in 2016 and VAR in 2019, the ResCNN-LSTM consistently outperforms all three statistical models when using the upper bound of the confidence interval.

4 Discussion

While the ResCNN-LSTM approach exhibited higher accuracy overall, there was significant inter-annual variability, which appears to relate to SWE characteristics. A relatively low snowpack was observed during the 2014-2015 winter and spring seasons (Figure 3). The buildup of SWE begins early in October and peaks in the middle of March. The improved forecast for 2016 may be attributed to a more typical winter experienced from the SWE monitoring sites. The forecast most similar to 2016 is 2019 in which the times series for SWE both steadily climb through the winter into the spring season. The buildup of SWE for 2017 is different from other years: it appears to peak early in March but then continues to linger through April until finally dropping off at the start of May. In comparison to the winters experienced in 2016 and 2019, the buildup of SWE peaks at the start and middle of April, respectively. The most accurate test periods (2015, 2016, 2019) are likely a result of the accumulated SWE tapering off between March and April. These periods occur at either the tail-end or just outside of the input window, making them close enough for the model to effectively map the resulting runoff period assuming that no further SWE accumulation takes place. Thus, the timing of snow ablation dictates model performance as it marks the transition period from SWE accumulation to runoff fed by snow melt. The 2017 test period is a testament to this theory as the accumulation of SWE lingers well past April resulting in very poor regression metrics.

The proposed model's forecasts are also speculated to be influenced by non-linear relationships between RSV (Figure 2) and SWE (Figure 3). Similar to 2015, the 2018 test year experienced a winter with below average SWE accumulation. Likewise, both 2015 and 2018 test years have peak SWE at the middle and end of March, respectively. However, the total observed reservoir inflows are significantly different with 29,287 ac-ft in 2015 and 18,590 ac-ft in 2018. This indicates the model's learned ability to map non-linear relationships among the multivariate dataset which statistical methods fail to capture (e.g. the 2018 VAR forecast, which exhibits low accuracy). Moreover, the model's best long-term forecasts (Figure 6) were during years of large runoff, making it most effective during years when accurate prediction is most important. The model's high predictive performance is likely associated with these events because they historically occur most frequently. Across the entire 30-year dataset for RSV, the most frequently observed reservoir inflow volumes (taken between first of April and beginning of July) were in excess of 28,000 ac-ft. The ResCNN-LSTM is a data-driven model designed to remember essential features over time; therefore, this behavior is entirely expected.

In addition to inter-annual dependencies, forecasts also depend on reservoir operations, hydrological features, and other structural characteristics of the water system (Anghileri et al., 2016). All of these components factor into long-term decision making by water managers who depend on reliable estimates for peak inflows. The proposed ResCNN-LSTM proves to effectively forecast the PSV at Upper Stillwater by (i) taking into account the spill limit as a physical limiter, and (ii) improving accuracy through statistical confidence intervals. The ResCNN-LSTM exhibits a higher bias in forecasting the PSV by consistently producing a realistic value close to but never exceeding what the reservoir can physically store (Figure 6). In contrast to the statistical models (Figure 7), the forecasted PSV is widely over-predicted during some years (e.g. SARIMA: 2015 and 2018; VAR: 2015) and under-predicted during others (e.g. TBATS: 2015 - 2017 and 2019; VAR: 2016 and 2019). This behavior is most likely attributed to the fact that Upper Stillwater has a spill limit to account for storage capacity. Thus, the spill limit acts as an asymptote for the model when producing each forecast, whereas other statistical models fail to recognize it. Likewise, this physical feature would account for the upwards skew in distribution for each test year's PSV forecast (Figure 8). The confidence interval also provides valuable insight for future decision making. The most challenging year for the model was 2018 due to an abnormally low snowpack throughout the winter. With this taken into consideration, the model's forecast for PSV in 2018 still lies within a physically realistic value and statistically confident interval. With an average absolute percent error of 2.66% in the center and 1.82% in the upper bound (Table 3), the confidence interval's upper bound consecutively proves to be the most accurate in predicting the PSV for all five test years. Therefore, water managers would be best served by using the forecasted upper bound for PSV in their forecasts.

Geared towards strengthening water manager abilities to manage and conserve RSV, this research improves on prior multi-step reservoir forecasting efforts by effectively increasing the context size to capture temporal dependencies. Previous studies including direct-step (Coulibaly et al., 2005; Sattari et al., 2012; Bai et al., 2016) and multi-step (Coulibaly et al., 2000; Muluye & Coulibaly, 2007; Kao et al., 2020) deep-learning algorithms have improved forecasting accuracy, but have yet to accurately forecast peak inflows at extended long-term horizons. For a lead time of one week, Coulibaly et al. (2000) forecasted peak flows ranging from an underprediction of 4.3% to an overprediction of 3.8% on average. Similarly, extended four month forecasts by Muluye and Coulibaly (2007) demonstrated reasonable predictions of low and medium reservoir inflows, but then either under or over-predicted the peaks. By contrast, our proposed model consistently manages to confidently predict realistic PSV values at a three-month lead time with an average absolute percent error of 2.66%. Similar to Coulibaly et al. (2000), the long-term forecasts at the end of the forecasted runoff period were more accurate than the initial short-term values. Periods where the proposed model struggled the most are irregular runoff seasons with abnormally dry hydrologic conditions and late-season SWE accumulation. An extended horizon of three months leaves wide potential for changes in hydrologic conditions and can amount to multiple different scenarios for runoff. A topic for future research may involve the inclusion of scenario-based model runs incorporating meteorological predictions from outside entities.

5 Conclusion

Given the comparatively high performance of the proposed algorithm in the study region, the ResCNN-LSTM architecture warrants further study for multi-step RSV forecasting. Considerations for future research include (i) further experiments with model architecture, (ii) investigating additional independent variables, and (iii) modeling additional reservoirs influenced by snowmelt runoff. Potential experiments with model architecture may include implementing batch normalization (Ioffe & Szegedy, 2015) between layers to reduce training time and increase predictive accuracy, and utilizing an attention mechanism (Bahdanau et al., 2014) to observe the intermediate states of the encoder, rather than only the final states. The inclusion of other independent variables, such as atmospheric temperature and solar radiation, may further improve accuracy. Finally, modeling additional reservoirs will provide valuable insight into model transferability.

Acknowledgments

This research was supported by the Central Utah Water Conservancy District and University of Utah. A special thanks to Blake Buehler, Heath Clark, Jared Hansen, and Zac

Brekenridge for their useful comments and discussion. Geographic Information System (GIS) mapping was provided by Lindsy Bentley. All reservoir storage volume (RSV) and snow water equivalent (SWE) data is made publicly available by the National Resource Conservation Service online repository at https://www.nrcs.usda.gov/. The Python code and training data, including the scripts used to compile them, are available through figshare (10.6084/m9.figshare.12756014).

References

Agarap, A. F. (2018). Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*.

Ahmed, S., Coulibaly, P., & Tsanis, I. (2015). Improved spring peak-flow forecasting using ensemble meteorological predictions. *Journal of Hydrologic Engineering*, *20*(2), 04014044.

Anghileri, D., Voisin, N., Castelletti, A., Pianosi, F., Nijssen, B., & Lettenmaier, D. P. (2016). Value of long-term streamflow forecasts to reservoir operations for water supply in snow-dominated river catchments. *Water Resources Research*, *52*(6), 4209–4225.

Bahdanau, D., Cho, K., & Bengio, Y. (2014). *Neural machine translation by jointly learning to align and translate.*

Bai, Y., Chen, Z., Xie, J., & Li, C. (2016). Daily reservoir inflow forecasting using multiscale deep feature learning with hybrid models. *Journal of Hydrology"*, *532"*, 193 - 206.

Bai, Y., Wang, P., Xie, J., Li, J., & Li, C. (2015). Additive model for monthly reservoir inflow forecast. *Journal of Hydrologic Engineering*, *20*(7), 04014079.

Bennett, J. C., Wang, Q. J., Li, M., Robertson, D. E., & Schepen, A. (2016). Reliable long-range ensemble streamflow forecasts: Combining calibrated climate forecasts with a conceptual runoff model and a staged error model. *Water Resources Research*, *52*(10), 8238-8259.

Bourdin, D. R., Nipen, T. N., & Stull, R. B. (2014). Reliable probabilistic forecasts from an ensemble reservoir inflow forecasting system. *Water Resources Research*, *50*(4), 3108-3130.

Box, G. E., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M. (2015). *Time series analysis: forecasting and control.* John Wiley & Sons.

Budu, K. (2014). Comparison of wavelet-based ann and regression models for reservoir inflow forecasting. *Journal of Hydrologic Engineering*, *19*(7), 1385-1400.

Cengil, E., & Cinar, A. (2018). A deep learning based approach to lung cancer identification. , 1–5.

Chang, F.-J., & Tsai, M.-J. (2016). A nonlinear spatio-temporal lumping of radar rainfall for modeling multi-step-ahead inflow forecasts by data-driven techniques. *Journal of Hydrology*, *535*, 256 - 269.

Chiamsathit, C., Adeloye, A. J., & Bankaru-Swamy, S. (2016). Inflow forecasting using artificial neural networks for reservoir operation. *Proceedings of the International Association of Hydrological Sciences*, *373*, 209–214.

Coulibaly, P., Anctil, F., & Bobée, B. (2000). Daily reservoir inflow forecasting using artificial neural networks with stopped training approach. *Journal of Hydrology*, *230*(3), 244 - 257.

Coulibaly, P., Haché, M., Fortin, V., & Bobée, B. (2005). Improving daily reservoir inflow forecasts with model combination. *Journal of Hydrologic Engineering*, *10*(2), 91-99.

De Livera, A. M., Hyndman, R. J., & Snyder, R. D. (2011). Forecasting time series with complex seasonal patterns using exponential smoothing. *Journal of the American statistical association*, *106*(496), 1513–1527.

Elizaga, N. B., Maravillas, E. A., & Gerardo, B. D. (2014). Regression-based inflow forecasting model using exponential smoothing time series and backpropagation methods for angat dam. In *2014 international conference on humanoid, nanotechnology, information technology, communication and control, environment and management (hnicem)* (pp. 1–6).

Gehring, J., Auli, M., Grangier, D., Yarats, D., & Dauphin, Y. N. (2017). Convolutional sequence to sequence learning. In *Proceedings of the 34th international conference on machine learning-volume 70* (pp. 1243–1252).

Gulli, A., & Pal, S. (2017). *Deep learning with keras.* Packt Publishing Ltd.

He, K., Zhang, X., Ren, S., & Sun, J. (2015). *Deep residual learning for image recognition.*

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735–1780.

Hyndman, R. J. (2008). *Forecasting with exponential smoothing : the state space ap-*

proach.

Iddrisu, W., Nokoe, K. S., & Osei-Akoto, I. (2016, 09). Modelling the trend of flows with respect to rainfall variability using vector autoregression. *International Journal of Advanced Research*, *4*, 125-140.

Ioffe, S., & Szegedy, C. (2015). *Batch normalization: Accelerating deep network training by reducing internal covariate shift.*

Ivanov, V., & Kilian, L. (2005). A practitioner's guide to lag order selection for var impulse response analysis. *Studies in Nonlinear Dynamics Econometrics*, *9*(1).

Kao, I.-F., Zhou, Y., Chang, L.-C., & Chang, F.-J. (2020). Exploring a long short-term memory based encoder-decoder framework for multi-step-ahead flood forecasting. *Journal of Hydrology*, 124631.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Konapala, G., & Mishra, A. (2020). Quantifying climate and catchment control on hydrological drought in the continental united states. *Water Resources Research*, *56*(1), e2018WR024620.

Kote, A. S., & Jothiprakash, V. (2009). Monthly reservoir inflow modeling using time lagged recurrent networks. *International Journal of Tomography and Statistics*, *12*(F09), 64–84.

Kratzert, F., Klotz, D., Herrnegger, M., Sampson, A. K., Hochreiter, S., & Nearing, G. S. (2019). Toward improved predictions in ungauged basins: Exploiting the power of machine learning. *Water Resources Research*.

LeCun, Y. e. a. (1990). Handwritten digit recognition with a back-propagation network. *Advances in Neural Information Processing Systems*, 396–404.

Liu, H., & Song, B. (2018). Stock price trend prediction model based on deep residual network and stock price graph. , *02*, 328-331.

Liu, Y., Sang, Y.-F., Li, X., Hu, J., & Liang, K. (2017). Long-term streamflow forecasting based on relevance vector machine model. *Water*, *9*(1).

Muluye, G. Y., & Coulibaly, P. (2007). Seasonal reservoir inflow forecasting with low-frequency climatic indices: a comparison of data-driven methods. *Hydrological sciences journal*, *52*(3), 508–522.

Ning, D., Liu, G., Jiang, R., & Wang, C. (2019). Attention-based multi-scale transfer resnet for skull fracture image classification. , *11198*, 111980D–111980D-5.

NRCS. (2020 (accessed May 3, 2020)). Nrcs online api [Computer software manual]. Retrieved from `https://www.nrcs.usda.gov/"`

Papamichail, D. M., & Georgiou, P. E. (2001). Seasonal arima inflow models for reservoir sizing 1. *JAWRA Journal of the American Water Resources Association*, *37*(4), 877–885.

Robbins, H., & Monro, S. (1951). A stochastic approximation method. *The annals of mathematical statistics*, 400–407.

Sattari, M. T., Yurekli, K., & Pal, M. (2012). Performance evaluation of artificial neural network approaches in forecasting reservoir inflow. *Applied Mathematical Modelling*, *36*(6), 2649 - 2657.

Shumway, R. H., & Stoffer, D. S. (2000). Time series analysis and its applications. *Studies In Informatics And Control*, *9*(4), 375–376.

Sims, C. A. (1980). Macroeconomics and reality. *Econometrica: journal of the Econometric Society*, 1–48.

Sin, C.-Y., & White, H. (1996). Information criteria for selecting possibly misspecified parametric models. *Journal of Econometrics*, *71*(1), 207–225.

Stokelj, T., Paravan, D., & Golob, R. (2002). Enhanced artificial neural network inflow forecasting algorithm for run-of-river hydropower plants. *Journal of Water Resources Planning and Management*, *128*(6), 415–423.

Sutskever, I., Martens, J., Dahl, G., & Hinton, G. (2013). On the importance of initialization and momentum in deep learning. In *International conference on machine learning* (pp. 1139–1147).

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Proceedings of the 27th international conference on neural information processing systems - volume 2* (pp. 3104–3112). Cambridge, MA, USA: MIT Press.

Tukey, J. W. (1970). *Exploratory data analysis.* Addison-Wesley.

van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., ... Kavukcuoglu, K. (2016). *Wavenet: A generative model for raw audio.*

Wang, F., Wang, L., Zhou, H., Saavedra Valeriano, O. C., Koike, T., & Li, W. (2012). Ensemble hydrological prediction-based real-time optimization of a multiobjective reservoir during flood season in a semiarid basin with global numerical weather predictions. *Water Resources Research*, *48*(7).

Yen, C.-F., Hsieh, H.-Y., Su, K.-W., & Leu, J.-S. (2019). Predicting solar performance ratio based on encoder-decoder neural network model. In *2019 11th international congress on ultra modern telecommunications and control systems and workshops (icumt)* (pp. 1–4).

Yuan, M., Ji, X., Lu, T., Chen, P., & Zhang, H. (2019). A novel two-factor attention encoder-decoder network through combining temporal and prior knowledge for weather forecasting. In *2019 international joint conference on neural networks (ijcnn)* (pp. 1–8).

Zhang, X., & You, J. (2020). A gated dilated causal convolution based encoder-decoder for network traffic forecasting. *IEEE Access*, *8*, 6087–6097.

Zheng, Z., Molotch, N. P., Oroza, C. A., Conklin, M. H., & Bales, R. C. (2018). Spatial snow water equivalent estimation for mountainous areas using wireless-sensor networks and remote-sensing products. *Remote sensing of environment*, *215*, 44–56.

Zhou, Y., Guo, S., & Chang, F.-J. (2019). Explore an evolutionary recurrent anfis for modelling multi-step-ahead flood forecasts. *Journal of Hydrology*, *570*, 343 - 355.

6 Tables

Table 1. Optimal architecture and configuration for the model

Parameter	Selected Value	Tested Values
Input Window	20	15,20,25
Epochs	100	50,75,100,150
Batch Size	32	16,32,64
Filter1 (CNN)	16	16,32,64
Filter2 (CNN)	32	16,32,64
Filter3 (CNN)	64	16,32,64
Kernel (CNN)	6	4,5,6,7,8
Cells (LSTM)	64	16,32,64,128
LSTM Layers	4	1,2,3,4,5
CNN Layers	12	3,6,12,18

Table 2. Regression metrics for ResCNN-LSTM hold-out sets

	MAE (%)	RMSE (%)	MedAE (%)	NSE	ExpVar
2015	22.949	28.090	19.161	0.889	0.932
2016	12.295	16.695	8.132	0.968	0.972
2017	32.863	43.216	29.509	0.792	0.836
2018	60.759	70.335	48.331	0.412	0.851
2019	21.515	35.361	7.208	0.812	0.862

Table 3. Forecasted PSV absolute percent error

	Lower Limit	Center	Upper Limit	SARIMA	VAR	TBATS
2015	2.979	2.136	1.294	11.874	14.343	14.144
2016	3.529	2.703	1.878	1.526	28.256	16.396
2017	6.029	4.195	2.362	4.792	3.452	17.607
2018	7.020	2.301	2.419	47.989	43.745	12.602
2019	2.858	1.995	1.133	3.830	0.924	8.356

7 Figures

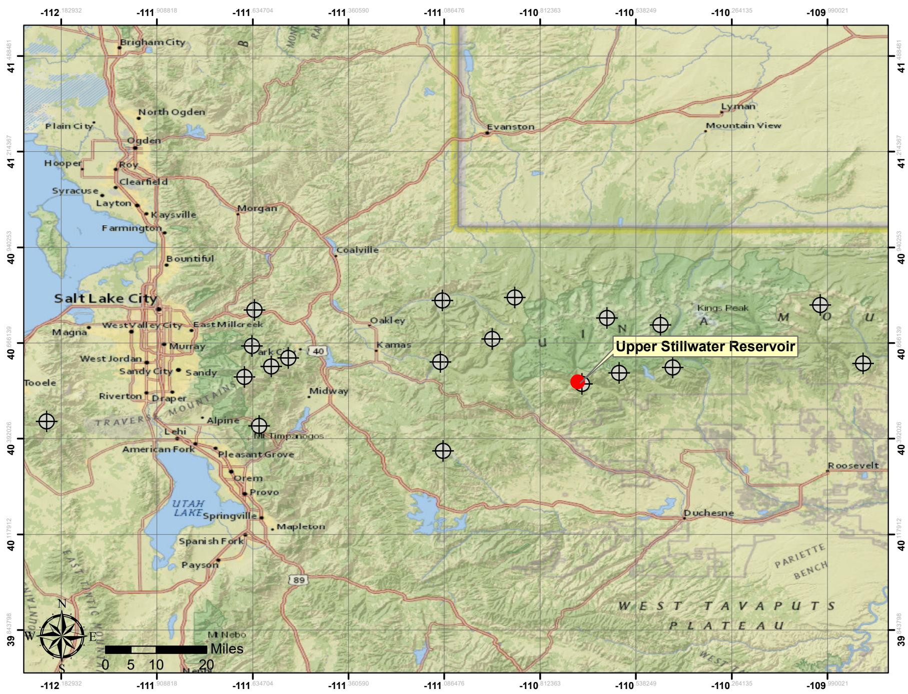

Figure 1. Study site location: Upper Stillwater reservoir, Utah

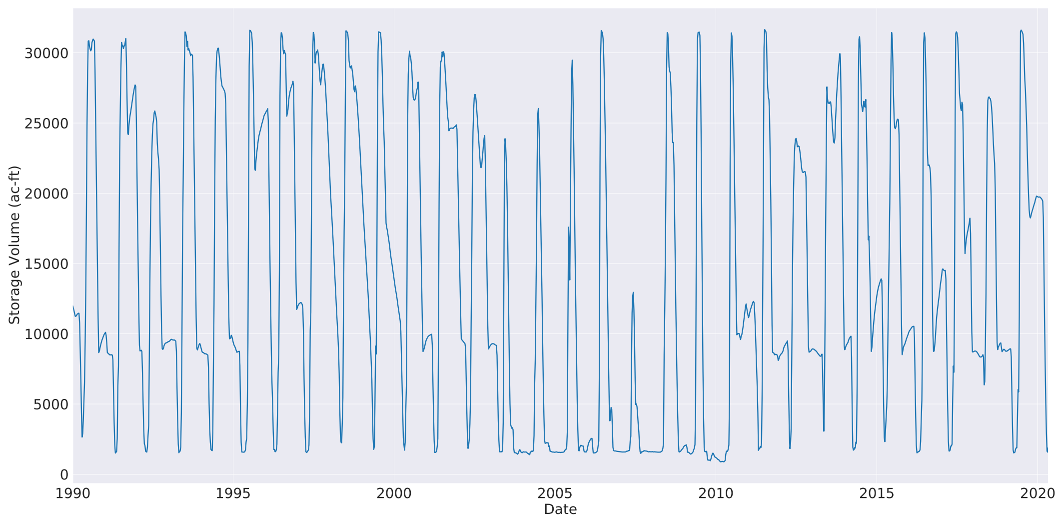

Figure 2. Upper Stillwater historical RSV (ac-ft)

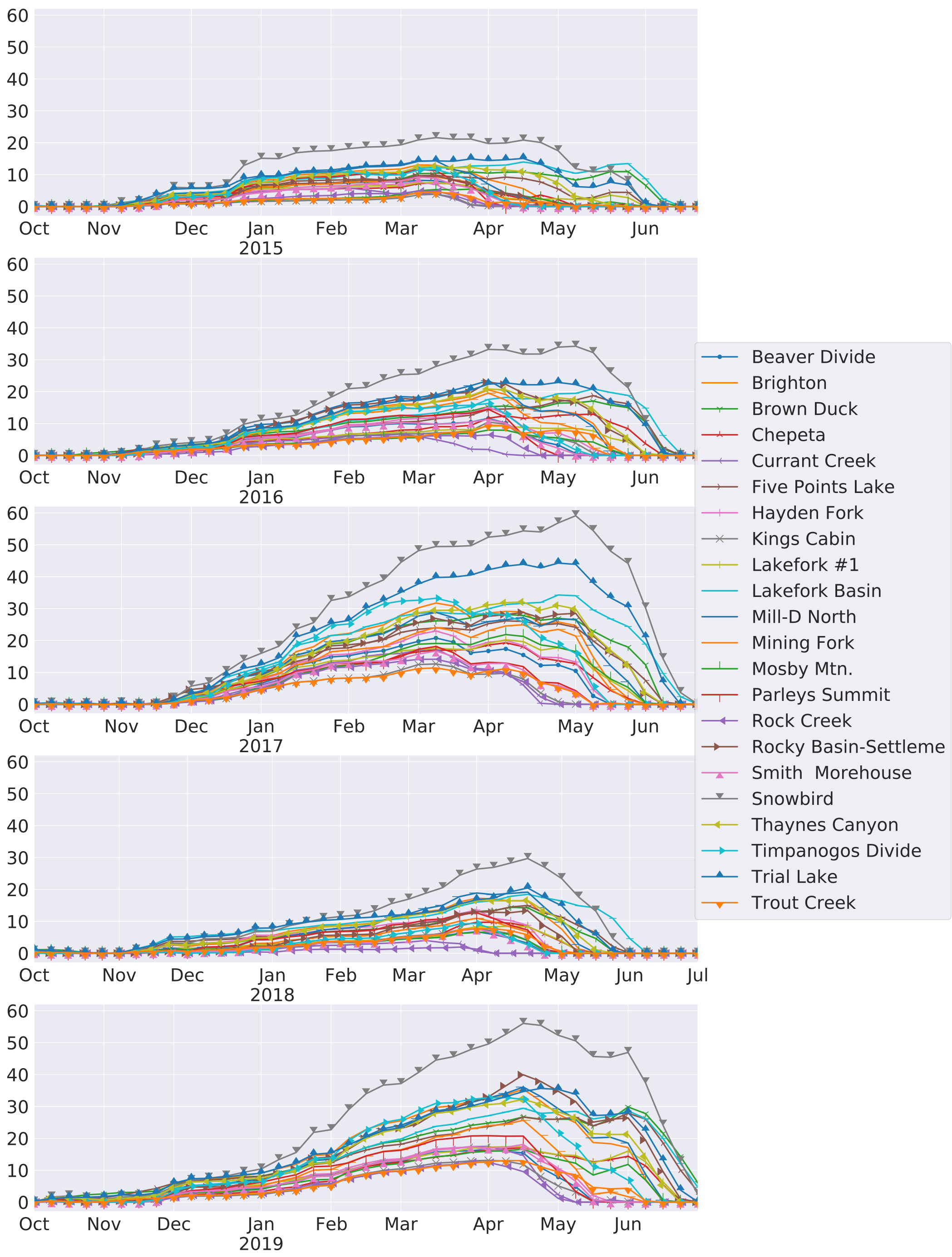

Figure 3. Recorded SWE (in) model inputs for five consecutive test years

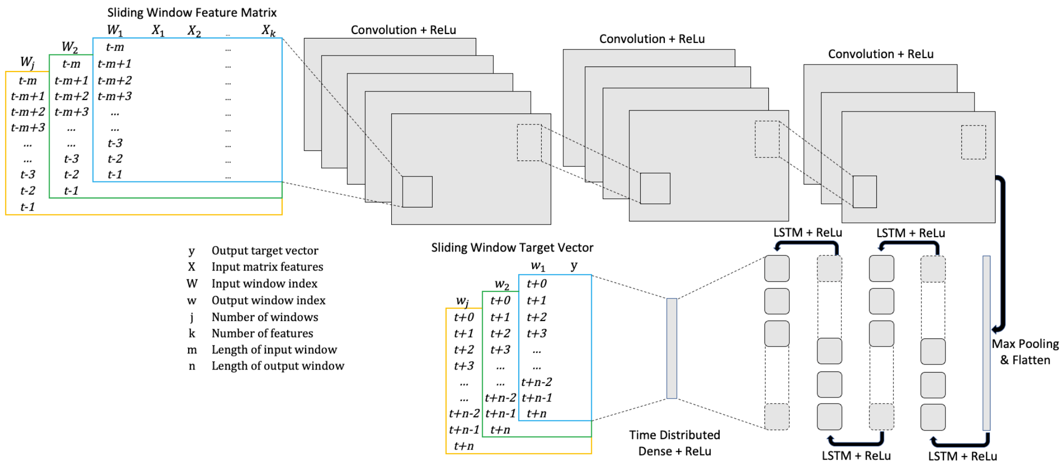

Figure 4. Deep learning model architecture

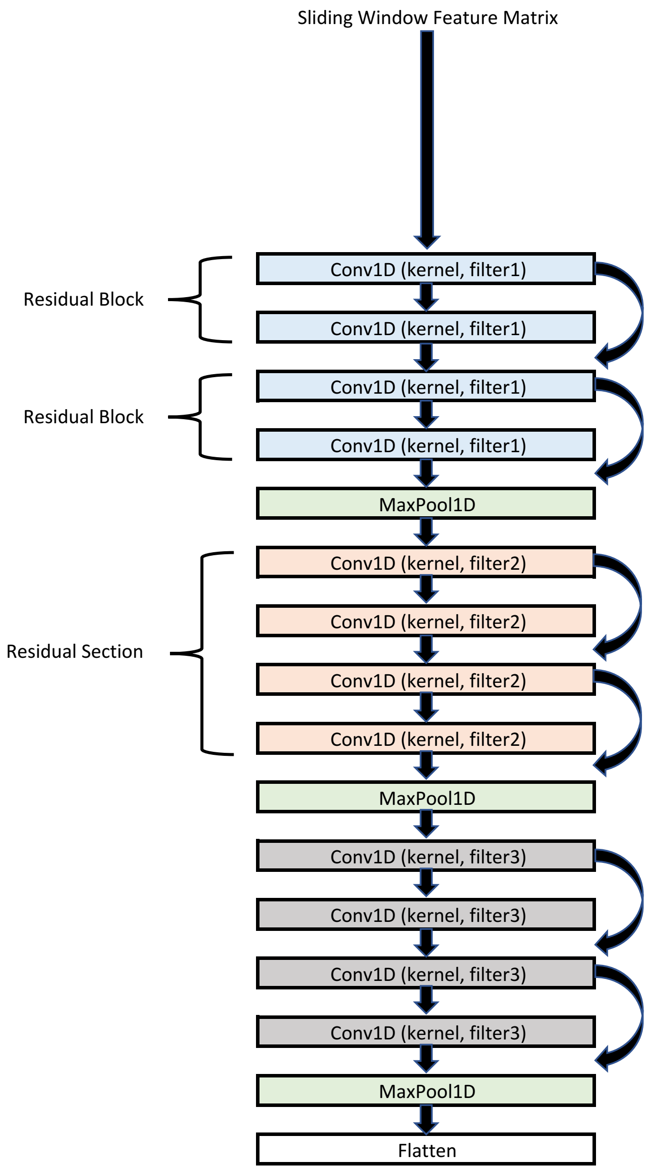

Figure 5. Deep residual framework for CNN encoder

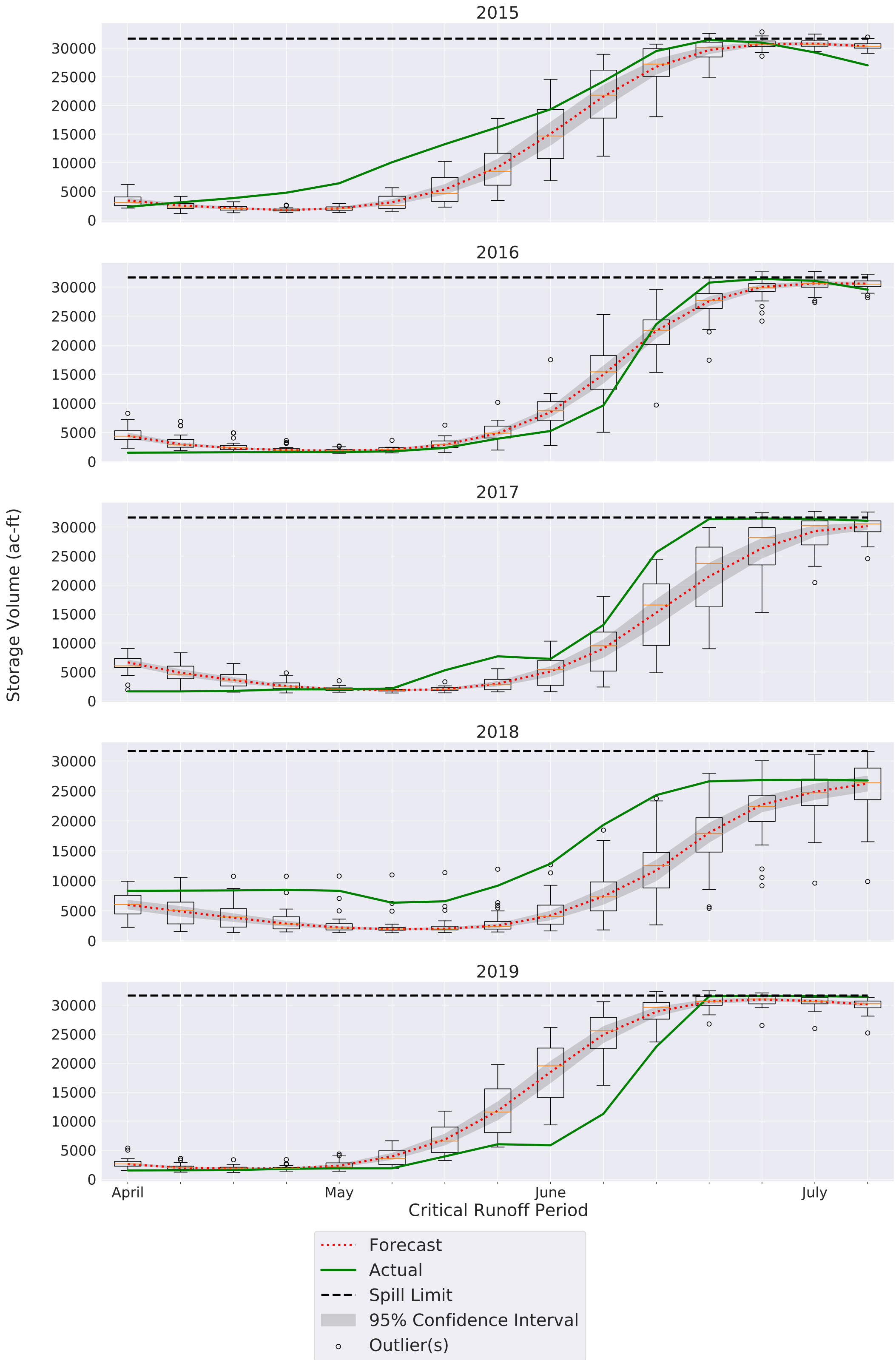

Figure 6. Multi-step storage volume forecasts from 2015 through 2019 using ResCNN-LSTM following a weekly timestep frequency

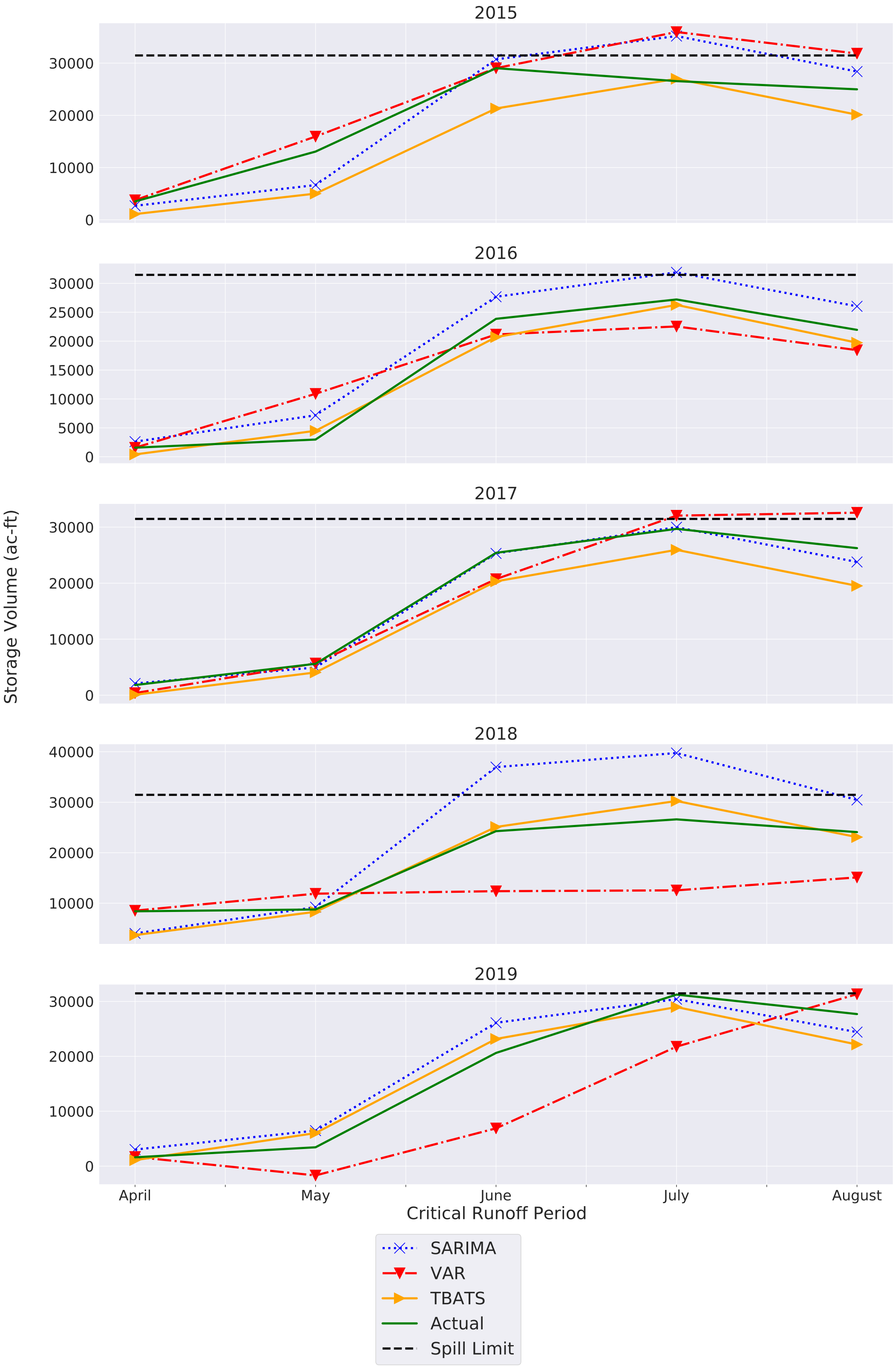

Figure 7. Multi-step storage volume forecasts from 2015 through 2019 using classic statistical methods following a monthly timestep frequency

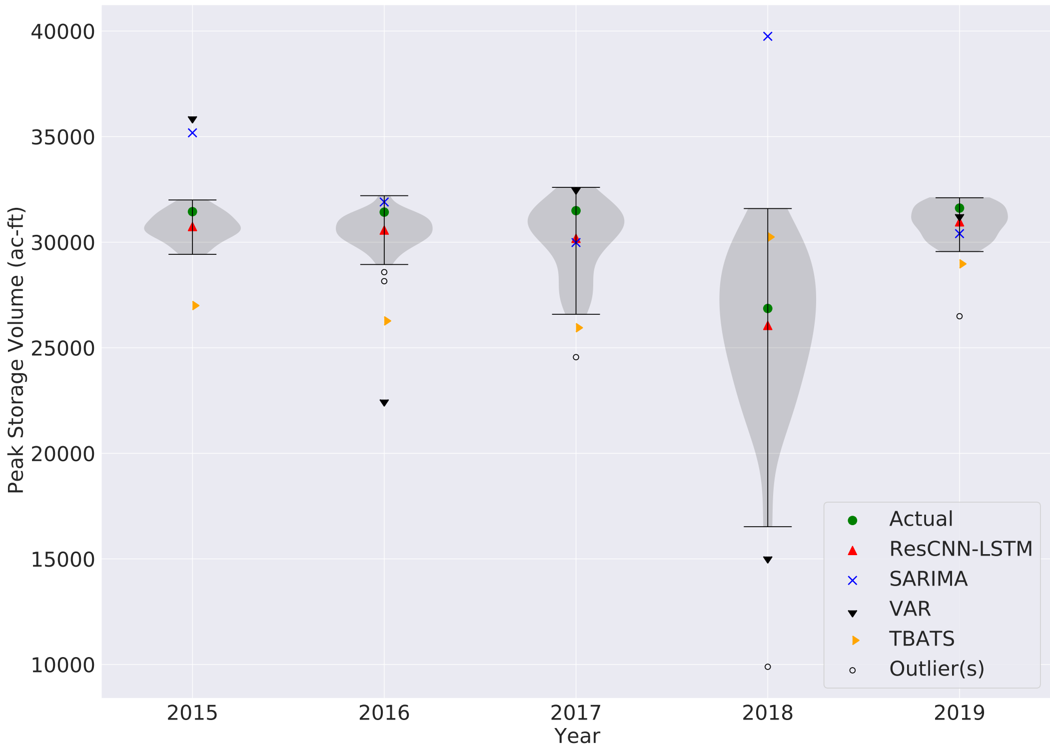

Figure 8. Comparison of various model's forecasted PSV