New Phytologist

*Methods*

# SapFlower: an automated tool for sap flow data preprocessing, gap-filling, and analysis using deep learning

**Jiaxin Wang and Heidi J. Renninger**

Department of Forestry, Mississippi State University, Mississippi State, MS 39762, USA

Author for correspondence:
*Jiaxin Wang*
*Email: jiaxinwang362@gmail.com; jiaxin.wang@vt.edu*

Received: *28 January 2025*
Accepted: *7 March 2025*

*New Phytologist* (2025) **246:** 2324–2345
**doi**: 10.1111/nph.70107

**Key words:** ecosystem water balance, hardwood, machine learning, plant water use, *Populus*, random forest, recurrent neural networks, thermal dissipation probes.

## Summary

- Sap flow, a critical process in plant water use and ecosystem water cycles, is often measured using thermal dissipation probes (TDP) due to their ease of installation and continuous data collection. However, sap flow data frequently include noise, outliers, and gaps, creating challenges for analysis and requiring substantial manual processing.
- We developed SapFlower, a tool that automates data preprocessing, model training, gap-filling, sapwood area scaling and modeling, and water use analysis. It integrates autocleaning, machine learning and deep learning models (e.g. random forest, Gaussian process regression, long short-term memory (LSTM), bidirectional LSTM (BiLSTM)), and efficient workflows to process sap flow data.
- SapFlower can remove over 90% of noisy data while preserving legitimate variations and achieve high accuracy in gap-filling based on user-determined parameters. Random forest, LSTM, and BiLSTM models reduced root mean square error to 10% or less for long-term gaps. Model training and prediction can be performed efficiently within seconds.
- SapFlower significantly enhances the efficiency and accessibility of TDP data analysis by automating complex tasks, enabling researchers without programming expertise to employ advanced techniques. Future improvements will focus on species-specific corrections for TDP and support for additional measurement methods. SapFlower is openly available on GitHub (https://github.com/JiaxinWang123/SapFlower) and Zenodo (doi: 10.5281/zenodo.13665919).

## Introduction

Carbon and water cycles are fundamental to the Earth's climate system, with plants playing a key role through processes such as transpiration and carbon sequestration. Measuring plant water use is essential for understanding how ecosystems respond to changing environmental conditions, particularly in the context of climate change (Niu *et al.*, 2011). Sap flow, a measure of the movement of water through a plant's vascular system, is a critical indicator of plant transpiration and water use (Meinzer *et al.*, 2004). Understanding sap flow dynamics is essential for studying plant physiology, plant hydraulic functioning, budget of watersheds, ecosystem water cycles, and their responses to environmental stressors such as drought (Wilson *et al.*, 2001; Meinzer *et al.*, 2004; Steppe *et al.*, 2015; Zhu *et al.*, 2017). While recent efforts have advanced the synthesis of global sap flow data, such as SAPFLUXNET data, challenges in raw data cleaning and gap-filling continue to limit its broader accessibility and utility within the global research community (Poyatos *et al.*, 2016; Peters *et al.*, 2018).

The thermal dissipation probe (TDP) method, introduced by Granier (1985), is a widely used, cost-effective technique for measuring sap flow in plants, particularly trees, by quantifying thermal dissipation (TD) caused by xylem sap flow (Granier, 1987; Smith & Allen, 1996). It operates on the principle that sap flow cools a heated probe inserted into the sapwood, with the cooling rate proportional to sap flow velocity. The method employs two radially inserted probes: a heated probe, continuously warmed by an electric current, and a reference probe, which measures ambient wood temperature. The temperature difference ($\Delta T$) between the probes decreases as sap flow increases, with the maximum temperature difference ($\Delta T_M$) occurring when sap flow is absent. Sap flux density ($F$) is commonly expressed as Eqn 1.

$$F = \alpha \times K^{\beta} \quad \text{Eqn 1}$$

where $F$ represents sap flux density, and $\alpha$ and $\beta$ are the empirical coefficients derived from Granier's original formulation. The flow index $K$ is defined as Eqn 2.

Present address: Jiaxin Wang, Department of Forest Resources and Environmental Conservation, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061, USA.

$$K = \frac{\Delta T_M - \Delta T}{\Delta T} \quad \text{Eqn 2}$$

This method is particularly effective for studying plant–water relations and transpiration, although accurate probe placement, calibration, and consideration of wood thermal properties are crucial for reliable measurements (Lu *et al.*, 2004; Masmoudi *et al.*, 2012; Alizadeh *et al.*, 2018). Despite TDP's widespread adoption, sap flow data collected via this method often contain noise, outliers, and gaps due to environmental factors, sensor failures, or data collection interruptions (Oishi *et al.*, 2016). These challenges necessitate efficient data processing, cleaning, and gap-filling methods to ensure robust, accurate, and reliable analysis.

Current tools and methods for sap flow data processing range from simple spreadsheet-based approaches to more sophisticated software/packages designed for specific datasets (Oishi *et al.*, 2016; Peters *et al.*, 2021). However, existing tools require substantial manual intervention, particularly for data cleaning, modeling, and gap-filling. These manual processes often introduce variability in data quality, increasing the risk of generating inconsistent or unstandardized data across different sap flow processing approaches. This, in turn, can lead to inaccurate outcomes. Moreover, the absence of standardized protocols significantly hinders the feasibility of large-scale or long-term studies, in which reliable and automated approaches are essential. Recent studies have shown that state-of-the-art machine learning and deep learning algorithms are superior in gap-filling time-series flux measurements such as eddy covariance and sap flow data, and the most used algorithms are recurrent neural networks (RNNs) and machine learning models, such as support vector regression (SVR) and random forest (RF) (Li *et al.*, 2022; Zhang *et al.*, 2023; Lucarini *et al.*, 2024; Yu *et al.*, 2024). However, most of those algorithms are not available or accessible to those who have no or limited programming skills. The necessity for standardized, automated, and flexible tools for sap flow data processing, gap-filling, and analysis using state-of-the-art techniques has become increasingly apparent. Large ecological datasets, particularly those involving continuous long-term monitoring, are challenging to manage without efficient and reproducible workflows. Automated tools that incorporate advanced machine learning techniques for gap-filling and analysis have the potential to substantially improve the accuracy and efficiency of sap flow data handling, providing new insights into plant water use under varying environmental conditions.

In this study, we present SapFlower, a novel application designed to meet the growing demand for automated sap flow data processing. SapFlower integrates a comprehensive pipeline for data cleaning, preprocessing, model training, gap-filling, sapwood area modeling, and water use analysis. By leveraging advanced machine learning models, including RNNs and RF, SapFlower predicts and fills missing data points using time stamps and user-defined key environmental variables as predictors. By providing a standardized framework, it streamlines sap flow data processing, reducing inconsistencies and minimizing the impact of methodological variations. Beyond automation, SapFlower offers a versatile tool for researchers to analyze sap flow dynamics across different species, ecosystems, and environmental conditions. This study highlights SapFlower's effectiveness in efficiently processing and analyzing sap flow data, supporting its application in long-term plant water use studies and ecosystem monitoring.

## Materials and Methods

### Sap flow data collection

Sap flow data used in this study were collected in 2022 from one of the Advancing *Populus* Pathways in the Southeast (APPS) project sites located in Pontotoc (34°8′N, 88°59′W), Mississippi, USA (Renninger *et al.*, 2024). Eastern cottonwood (*Populus deltoides* W. Bartram ex Marshall) and hybrid poplar (*P. deltoides* × *P. maximowiczii* (A. Henry) (D × M), *P. deltoides* × *P. nigra* (L.) (D × N), *P. deltoides* × *P. trichocarpa* (Torr. & A. Gray ex Hook.) (D × T), a combination of *P. deltoides*, *P. nigra*, and *P. maximowiczii* (D × N × M), *P. trichocarpa* × *P. maximowiczii* (T × M), and *P. trichocarpa* × *P. deltoides* (T × D)) genotypes were planted in 2019 at a spacing of 1.83 × 2.16 m to study their feasibility as crops for bioenergy feedstock in southeastern United States. We began measuring sap flow rates at the start of the second growing season in 2020 using a customized solar panel-powered field sap flow data collection system (Fig. 1). The trees had an average diameter at breast height of 34.9 mm and an average height of 4.78 m. A total of 96 trees were selected for sap flow measurements across two replicate blocks, and the main materials and methods used were described by Renninger *et al.* (2023).

Specifically, the installation of TD sap flow sensors involved inserting two probes (heated and reference, laboratory-made sensors) into the tree trunk, as shown in the side and top views of Fig. 1. Sensors were installed from the north side of the stems to avoid direct heating from the sun. The heated sensor was installed *c.* 10 cm above the reference sensor, with the heated sensor housed in an aluminum tube inserted after drilling precise holes into the trunk. Heat sink paste was applied to the heated sensor to ensure efficient heat dissipation. The sensors were connected to a data acquisition system, specifically an AM16/32B multiplexer (Campbell Scientific Inc., Logan, UT, USA), which interfaced with a CR1000 data logger (Campbell Scientific Inc.). The system was powered by a pair of 12-V batteries charged via solar panels, designed to ensure continuous monitoring in field conditions. An electrical circuit board manages the heating rate and strives to maintain a constant current or wattage to the sensors of 0.2 W for 20-mm sensors. To prevent external heating from sunlight, a piece of reflective insulation covered the sensor pairs. This setup allowed for precise, continuous sap flow measurement by capturing TD as sap moved through the xylem of poplar trees.

### Design of SapFlower

When designing SapFlower, the initial objectives were to standardize and automate the whole process for data previewing and cleaning, state-of-the-art model training, gap-filling, and sap flow

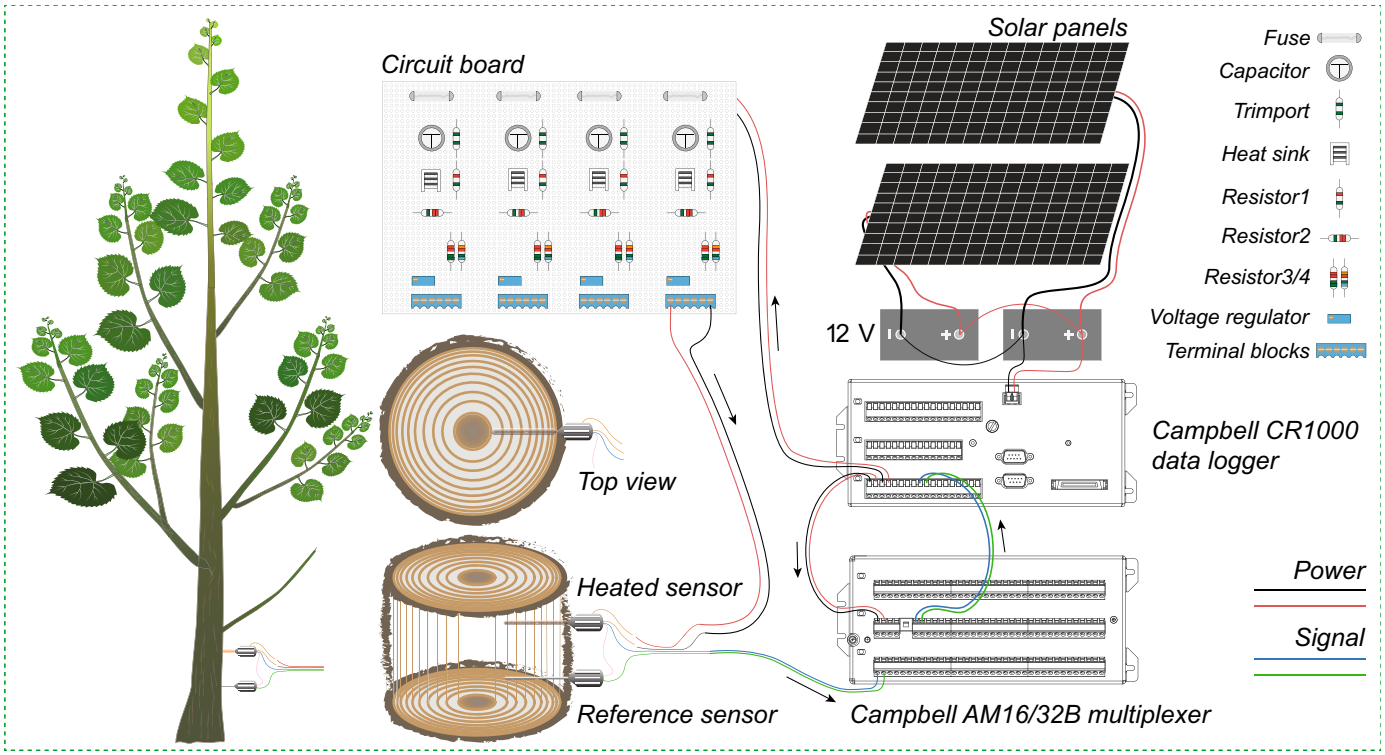



**Fig. 1** Schematic diagram of configuration and installation of thermal dissipation probes. Generally, heated sensors and reference sensors are connected to a circuit board for power control, a multiplexer, and a datalogger for signal collection. Solar panels and deep-cycle 12 -V batteries are used to power the system. Species under investigation belong to the genus *Populus* (poplars).

parameter calculation. Therefore, we proposed the following automated framework, including project configuration, data preprocessing, model development and training, gap-filling, sapwood area modeling, and water use analysis (Fig. 2).

**Project configuration** SapFlower enables users to create, save, and open project files for maintaining consistency, efficiency, and organization in complex tasks, involving linking data files, setting filtering parameters, and saving and reopening edited results. Having project files ensures the project can be easily replicated or modified without reentering details. Saving and reopening project configurations offer consistency, efficiency, and flexibility. Overall, project configurations support streamlined workflows, reproducibility, and organization, making research and project management more efficient and manageable.

**Data preprocessing** Data, including sap flow and the raw temperature differential measurements, preprocessing can be performed either manually or automatically in SapFlower. Users can use their expertise and experience to determine which data are realistic or need to be removed for further gap-filling or analysis based on the environmental data plot (Fig. 3). Alternatively, users can utilize autocleaning by setting the proper parameters for autocleaning before gap-filling. To ensure high-quality data, the raw sap flow measurements underwent several autocleaning methods as follows:

Outliers were detected using the interquartile range (IQR) method. The IQR is defined as Eqn 3.

$$\mathrm{IQR} = Q_3 - Q_1 \quad \text{Eqn 3}$$

where $Q_1$ is the first quartile (25$^{\text{th}}$ percentile) and $Q_3$ is the third quartile (75$^{\text{th}}$ percentile).

Any data points that fell outside of the range in Eqn 4, where $k$ is a threshold multiplier, were flagged as outliers.

$$\left[Q_1 - k \times \mathrm{IQR},\ Q_3 + k \times \mathrm{IQR}\right] \quad \text{Eqn 4}$$

This approach ensured that anomalous data points – either too low or too high – were excluded from further analysis. Typical values for $k$ range between 0 and the maximum threshold of users defined sap flow (raw sap flow measurement, i.e. $\mathrm{d}T/\mathrm{d}V$), depending on the sensitivity of outlier detection.

To address periods of high variability or stagnation in the sap flow data, rolling windows of a specified size were used to calculate the mean and SD within each window. The window size (i.e., 24 or 72 h) can be defined by users based on their specific data and objectives. The rolling mean, $\mu_{\text{window}}$, and rolling SD, $\sigma_{\text{window}}$, were computed using Eqns 5 and 6.

$$\mu_{\text{window}} = \frac{1}{n}\sum_{i=1}^{n} x_i \quad \text{Eqn 5}$$

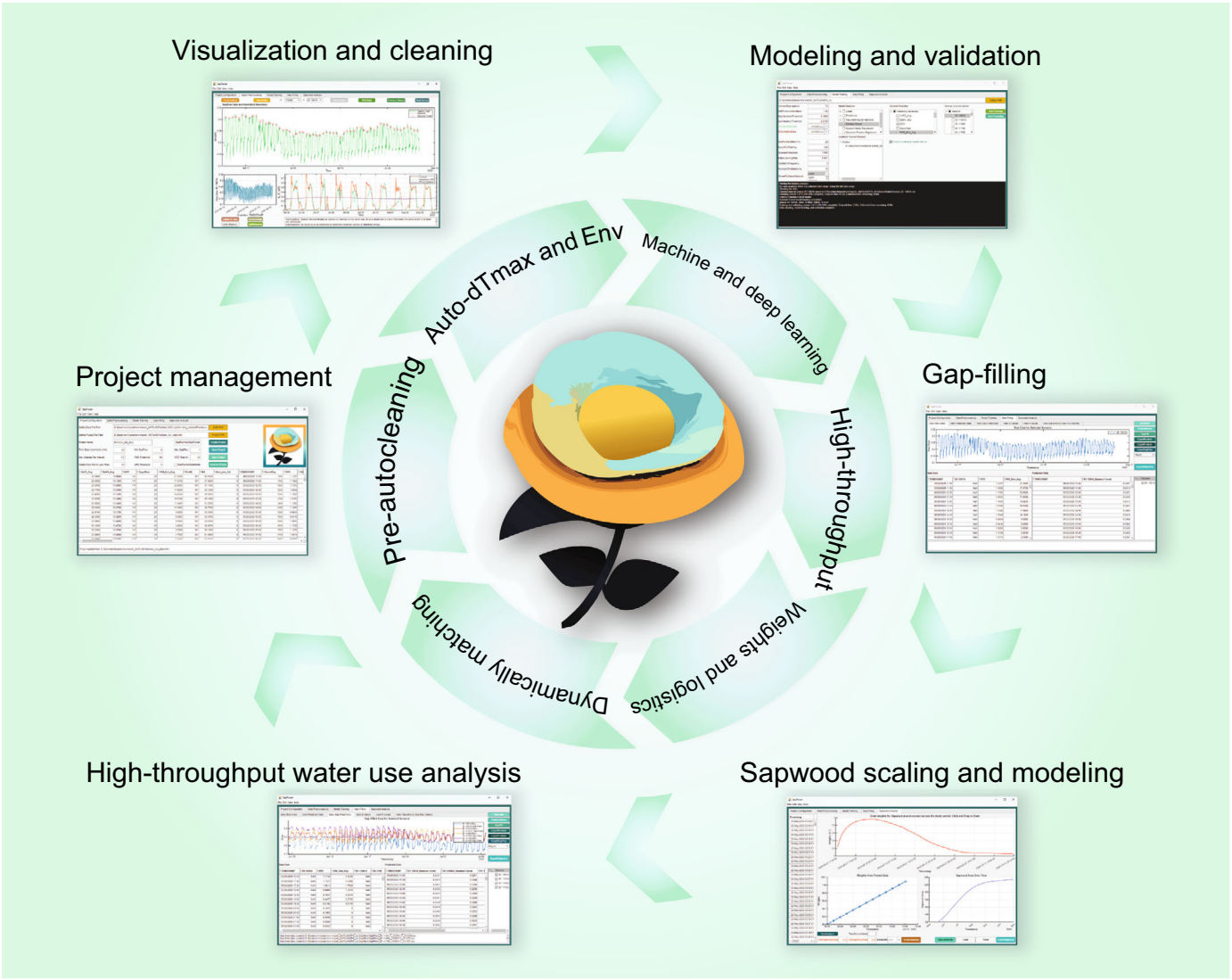



**Fig. 2** Workflow and main functionalities of SapFlower for sap flow raw measurements and SAPFLUXNET data analysis.

$$\sigma_{\text{window}} = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \mu_{\text{window}})^2} \quad \text{Eqn 6}$$

where $n$ is the number of data points in the user-defined window and $x_i$ are the individual sap flow raw measurement values.

Windows where $\sigma_{\text{window}}$ exceeded a predefined threshold for high variation were flagged and removed. Similarly, windows with very low $\sigma_{\text{window}}$ values, indicating sensor stagnation or drift, were also excluded from further analysis. These measures ensured that periods of excessive noise or inactivity were filtered out. Users can set and adjust the thresholds to determine which data should be removed or kept.

Due to low power, datalogger malfunction, or other issues, TDP data sometimes contain inversed measurements (Fig. 4a), such as when $\Delta T_M$ appeared at noon instead of nighttime, and those measurements should be treated carefully. Here, we extracted the nearest neighbors (six points at each side) of sap flow baseline points, identified the maximum of the nearest points, compared the baseline points with three-fourths (empirical range based on our 96 tree measurements) of maximum of the nearest points, and removed the all-day data in which baseline points were identified smaller than maximum of the nearest points. We set three-fourths of the maximum of the nearest points to filter the reverse data because the data occasionally contain spikes that can exceed the TDP measurements at $\Delta T_M$. Detailed formulas are given as Eqns 7–11.

$$\text{LeftId}x_i = \max(1, i - \text{numNeighbors}) \quad \text{Eqn 7}$$

$$\text{RightId}x_i = \min(n, i + \text{numNeighbors}) \quad \text{Eqn 8}$$

$$\text{NearestPoints}_i = \text{sensorData}[\text{LeftId}x_i : \text{RightId}x_i] \quad \text{Eqn 9}$$

$$\text{MaxNearest}_i = \max(\text{NearestPoints}_i, \text{omitnan}) \quad \text{Eqn 10}$$

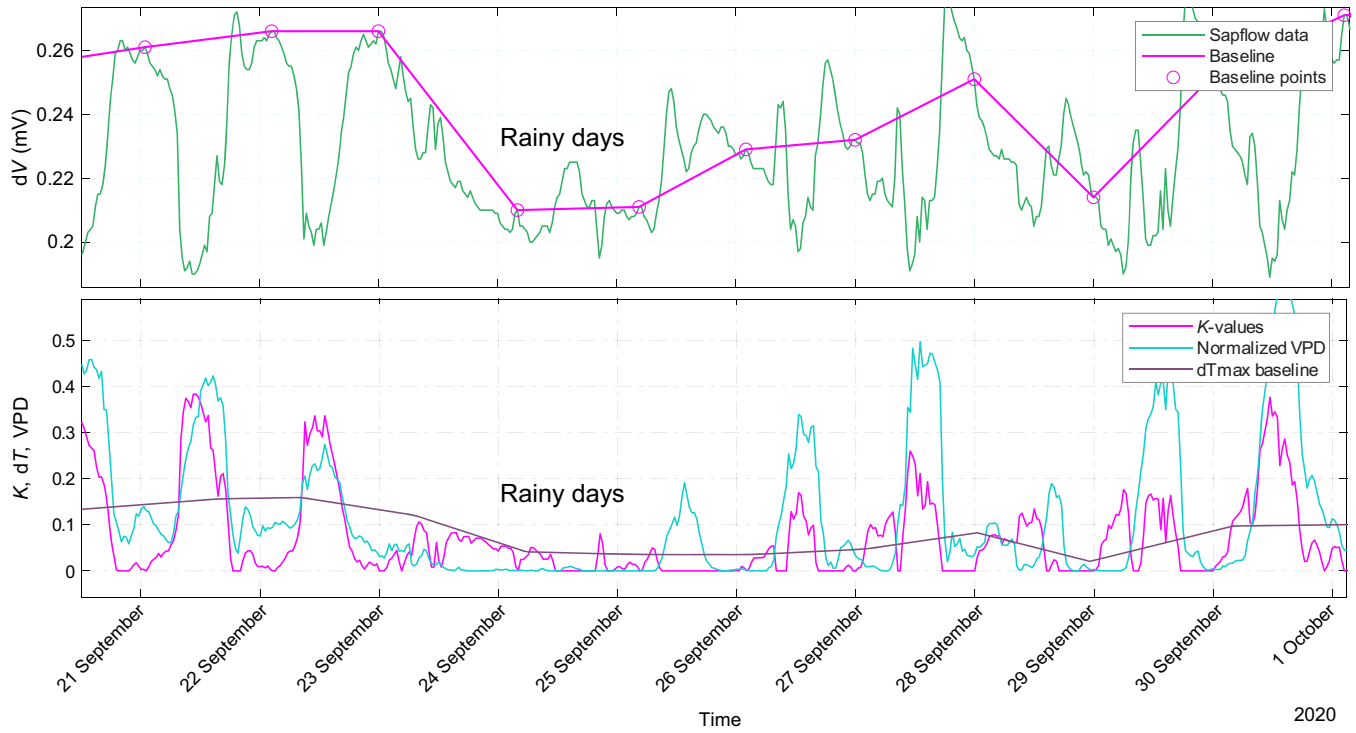



**Fig. 3** Data visualization and cleaning examples of sap flow measurements in SapFlower. The top panel allows users to edit sap flow data, while the bottom panel enables visual inspection to verify whether the data are realistic in relation to environmental conditions. Vapor pressure deficit (VPD) is normalized and plotted with Granier's *K* values to help users' determine whether they should clean sap flow measurements based on environmental conditions. Species under investigation belong to the genus *Populus* (poplars).

$$\text{reversedBaselineValues}_i < \frac{3}{4} \times \text{MaxNearest}_i \quad \text{Eqn 11}$$

## Model development and training

**Gap-filling models** Several models were used to predict and fill gaps in the sap flow data, including both classical statistical models and state-of-the-art machine learning and deep learning approaches. Each model was trained using the cleaned TDP measures and environmental variables, and its performance was evaluated based on prediction accuracy. Summary and comparison of the below-listed models are presented in Table 1. We used the methods described by Oishi *et al.* (2008, 2016) to calculate and identify $\Delta T_M$. Specifically, to calculate $\Delta T_M$, the maximum temperature difference at zero sap flow, we need to identify $\Delta T$ during periods meeting the following conditions:

(**1**) Nighttime hours, defined as periods with near-zero radiation ($R < R_{threshold}$), and $R$ can be set using the photosynthetically active radiation (PAR) threshold in SapFlower.

(**2**) Stable $\Delta T$, where the coefficient of variation (CV) satisfies $CV < CV_{threshold}$.

(**3**) Low vapor pressure deficit ($VPD < VPD_{threshold}$).

Since $K$ in Granier's equation is a dimensionless ratio, using the raw measurement (e.g. millivolt readings) instead of converting to temperature will not affect $K$ itself.

*Simple linear regression.* Simple linear regression was used to model the relationship between sap flow ($y$) and a single predictor variable (e.g. the natural log of VPD) or using one working sensor to gap-fill another sensor; however, since sap flow often saturates at higher VPD, a linear approach may not always be appropriate. The model assumes a linear relationship between the predictor and the response variable, expressed as Eqn 12.

$$y = \beta_0 + \beta_1 x + \epsilon \quad \text{Eqn 12}$$

where $y$ is the predicted TDP measurement, $x$ is the predictor variable (e.g. VPD), $\beta_0$ is the intercept, $\beta_1$ is the slope of the regression line, and $\epsilon$ represents the residual error.

This model was fitted using the least square method, which minimizes the sum of the squared residuals (Eqn 13).

$$\text{RSS} = \sum_{i=1}^{n} \left(y_i - \widehat{y}_i\right)^2 \quad \text{Eqn 13}$$

where $y_i$ is the observed TDP measurement and $\widehat{y}_i$ is the predicted value.

*Multiple linear regression.* Multiple linear regression extends simple linear regression to include multiple predictor variables (e.g. VPD, PAR, soil moisture, and leaf area index (LAI)). The model is expressed as Eqn 14.

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p + \epsilon \quad \text{Eqn 14}$$

where $y$ is the predicted TDP measurement, $x_1$, $x_2$, …, $x_p$ are the predictor variables, $\beta_0$ is the intercept, $\beta_1$, $\beta_2$, …, $\beta_p$ are the

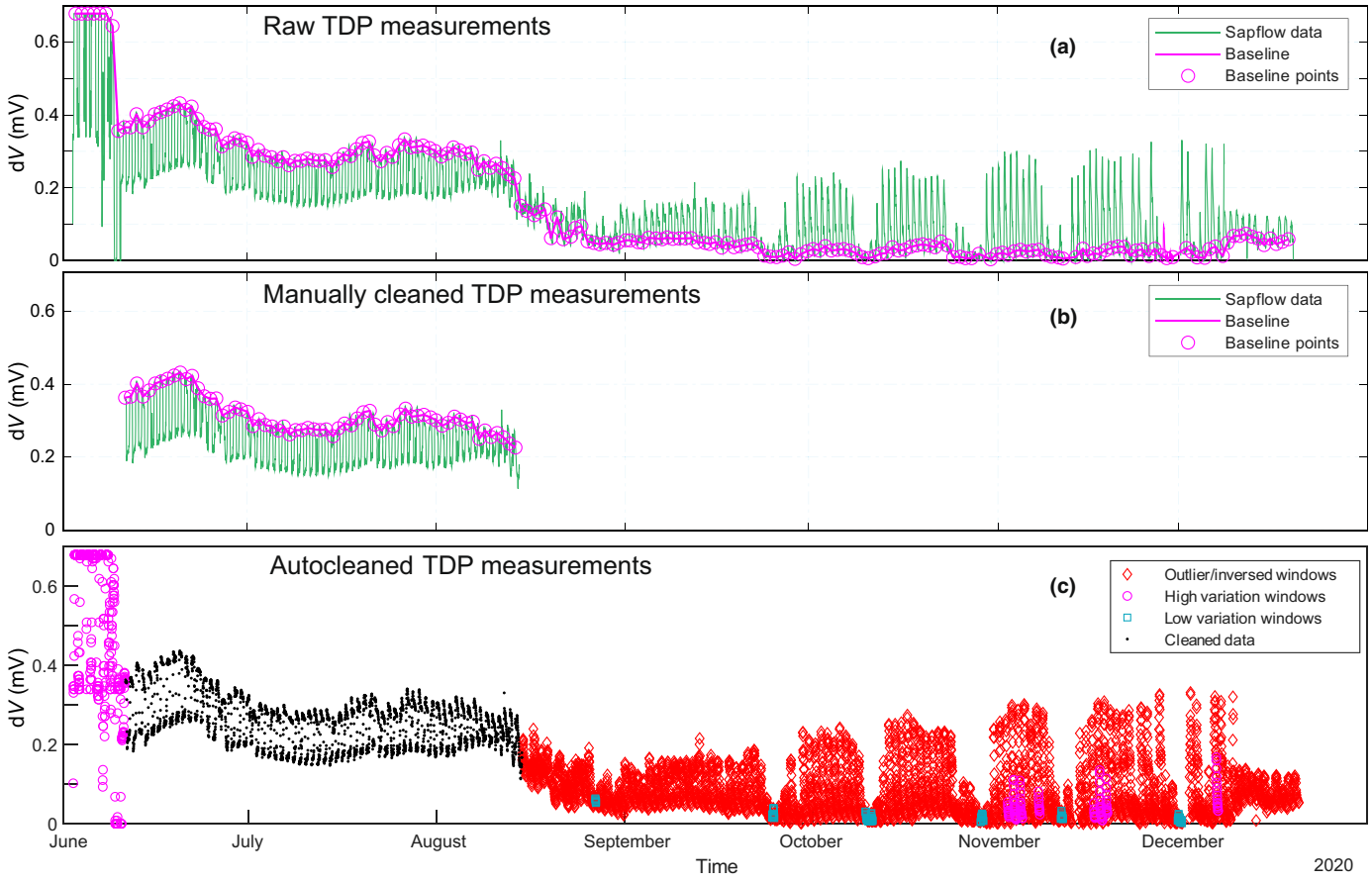



**Fig. 4** Example of raw, manually cleaned, and autocleaned sap flow measurements. It should be noted that the extreme values for raw (a), manually cleaned (b), and autocleaned (c) sap flow measurements have been removed during importing and loading data using users defined thresholds of minimum and maximum sap flow measurement. Inversed data represent data that have flipped day and night measurements due to datalogger malfunction.

coefficients representing the relationship between each predictor and the response, and $\epsilon$ is the residual error.

This model was also fitted using the least square method. Multiple linear regression allowed for the inclusion of various environmental variables to provide more accurate predictions of sap flow.

*Autoregressive with exogenous input model.* The autoregressive with exogenous input (ARX) model is a time-series model that uses past values of the TDP measurement (autoregressive terms) and current values of exogenous variables (environmental data, such as VPD and PAR) to predict future TDP measurements. The ARX model is defined as Eqn 15.

$$y(t) = \sum_{i=1}^{p} a_i y(t-i) + \sum_{j=1}^{q} b_j x(t-j) + \epsilon(t) \quad \text{Eqn 15}$$

where $y(t)$ is the predicted TDP measurement at time $t$, $y(t-i)$ are past TDP values, $x(t-j)$ are the exogenous inputs, $a_i$ and $b_j$ are the model coefficients, $p$ and $q$ are the number of past TDP measurements, and $\epsilon(t)$ is the error term.

The ARX model captures both the time dependence of TDP measurements and the influence of environmental factors, making it suitable for short-term gap-filling in time series with strong temporal correlations.

*Autoregressive moving average with exogenous input model.* The autoregressive moving average with exogenous input (ARMAX) model extends the ARX model by incorporating a moving average component that accounts for the noise into the model. The ARMAX model is defined as Eqn 16.

$$y(t) = \sum_{i=1}^{p} a_i y(t-i) + \sum_{j=1}^{q} b_j x(t-j) + \sum_{k=1}^{r} c_k \epsilon(t-k) + \epsilon(t)$$

Eqn 16

where $c_k$ are the moving average coefficients that model the noise, $r$ represents the number of past TDP measurements, and the other terms are defined similarly to the ARX model.

The ARMAX model is particularly useful for modeling time-series data with both autocorrelation and noise, allowing for more robust predictions in complex TDP data.

*Long short-term memory model.* The long short-term memory (LSTM) model, a specialized form of RNNs, is built to

**Table 1** Summary and comparison of models integrated into SapFlower.

| Model | Description | Advantages | Disadvantages | Best use cases |
|---|---|---|---|---|
| Simple linear regression (SLR) | Models sap flow as a linear function of a single predictor (e.g. VPD) | Simple, interpretable, computationally efficient | Fails when relationships are nonlinear, limited flexibility | Using one sensor/tree to gap-fill another |
| Multiple linear regression (MLR) | Extends SLR to multiple predictors (e.g. VPD, PAR, soil moisture, LAI) | Accounts for multiple factors affecting sap flow | Assumes linearity, sensitive to multicollinearity | Using one sensor/tree to gap-fill another with environmental variables |
| Autoregressive with exogenous inputs (ARX) | Uses past TDP values and environmental variables for short-term prediction | Captures temporal dependencies, suitable for time series | Limited to short-term predictions, requires parameter tuning | Short-term gap-filling with time-series dependence |
| Autoregressive moving average with exogenous inputs (ARMAX) | Enhances ARX by modeling noise through a moving average component | Handles both autoregressive relationships and noise | Computationally intensive, sensitive to noise parameters | Short-term gap-filling with time-series dependence |
| Long short-term memory (LSTM) | Deep learning model capturing long-term dependencies in time series | Handles long-term dependencies, effective for complex time series (day and night dynamics) | Requires large datasets, computationally expensive | Short-/long-term gap-filling for multiple year measurements |
| Bidirectional LSTM (BiLSTM) | Extends LSTM by learning bidirectional dependencies in data | Captures bidirectional dependencies especially effective for complex time series (day and night dynamics) | Computationally intensive due to dual LSTM networks | Short-/long-term gap-filling for multiple year measurements |
| Gated recurrent unit (GRU) | Simplified LSTM with fewer parameters, improving computational efficiency | Computationally efficient alternative to LSTM | Less powerful than LSTM for long-term dependencies | Short-/long-term gap-filling for multiple year measurements |
| Random forest (RF) | Ensemble of decision trees used for robust nonlinear predictions | Handles complex, nonlinear relationships, and reduces overfitting, and effective training and prediction | Prone to overfitting if not properly tuned, sensitive to feature selection, and less effective in capturing day and night dynamics of sap flow measurement | Short-/long-term gap-filling for less varied environmental conditions |
| Support vector regression (SVR) | Machine learning model using a kernel function to map data to higher dimensions | Performs well on high-dimensional data, flexible kernel choices, and effective training and prediction | Requires careful kernel selection and parameter tuning, and less effective in capturing day and night dynamics of sap flow measurement | Using high dimensions of environmental variables for gap-filling |
| Gaussian process regression (GPR) | Probabilistic model providing uncertainty estimates for predictions | Provides confidence intervals for predictions, and effective training and prediction | Computationally expensive, sensitive to kernel choice, and less effective in capturing day and night dynamics of sap flow measurement | Situations requiring uncertainty quantification |
| Kernel regression (KR) | Nonparametric method estimating target values based on kernel functions | Flexible, adapts to nonlinear relationships, and effective training and prediction | Sensitive to bandwidth parameter, performance depends on kernel choice, and less effective in capturing day and night dynamics of sap flow measurement | Short-/long-term gap-filling for less varied environmental conditions |

LAI, leaf area index; PAR, photosynthetically active radiation; TDP, thermal dissipation probe; VPD, vapor pressure deficit.

capture long-term dependencies in time-series data (Hochreiter, 1997; Tai *et al.*, 2015). Unlike conventional RNNs, LSTM networks contain memory cells that store information over extended periods, making them particularly effective for predicting TDP measurement dynamics over time. The model operates through three key gates: the input gate ($i_t$), forget gate ($f_t$), and output gate ($o_t$). The equations are Eqns 17–22 for $i_t$, $f_t$, cell state update ($\tilde{C}_t$), cell state ($C_t$), $o_t$, and $h_t$.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad \text{Eqn 17}$$

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad \text{Eqn 18}$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \quad \text{Eqn 19}$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \quad \text{Eqn 20}$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad \text{Eqn 21}$$

$$h_t = o_t \odot \tanh(C_t) \quad \text{Eqn 22}$$

where $\odot$ is the elementwise multiplication, $W$ and $b$ are the weight matrices and bias terms, and $\sigma$ is the sigmoid activation function.

The LSTM model was trained on the sap flow data using past values of both TDP measurements and environmental variables to predict future values and fill missing data.

*Bidirectional LSTM model.* The bidirectional LSTM (BiLSTM) model contains forward LSTM, backward LSTM, and combined forward and backward states, based on equations given as Eqns 23–33 (Schuster & Paliwal, 1997; Tai *et al.*, 2015). The forward LSTM equations for time step $t$ are given as Eqns 23–27 for $i_t$, $f_t$, $C_t$, $o_t$, and $h_t$.

$$i_t^{\text{fwd}} = \sigma\left(W_i^{\text{fwd}} \cdot \left[h_{t-1}^{\text{fwd}}, x_t\right] + b_i^{\text{fwd}}\right) \quad \text{Eqn 23}$$

$$f_t^{\text{fwd}} = \sigma\left(W_f^{\text{fwd}} \cdot \left[h_{t-1}^{\text{fwd}}, x_t\right] + b_f^{\text{fwd}}\right) \quad \text{Eqn 24}$$

$$C_t^{\text{fwd}} = f_t^{\text{fwd}} \odot C_{t-1}^{\text{fwd}} + i_t^{\text{fwd}} \odot \tanh\left(W_C^{\text{fwd}} \cdot \left[h_{t-1}^{\text{fwd}}, x_t\right] + b_C^{\text{fwd}}\right) \quad \text{Eqn 25}$$

$$o_t^{\text{fwd}} = \sigma\left(W_o^{\text{fwd}} \cdot \left[h_{t-1}^{\text{fwd}}, x_t\right] + b_o^{\text{fwd}}\right) \quad \text{Eqn 26}$$

$$h_t^{\text{fwd}} = o_t^{\text{fwd}} \odot \tanh\left(C_t^{\text{fwd}}\right) \quad \text{Eqn 27}$$

Similarly, the backward LSTM equations are given as Eqns 28–32.

$$i_t^{\text{bwd}} = \sigma\left(W_i^{\text{bwd}} \cdot \left[h_{t+1}^{\text{bwd}}, x_t\right] + b_i^{\text{bwd}}\right) \quad \text{Eqn 28}$$

$$f_t^{\text{bwd}} = \sigma\left(W_f^{\text{bwd}} \cdot \left[h_{t+1}^{\text{bwd}}, x_t\right] + b_f^{\text{bwd}}\right) \quad \text{Eqn 29}$$

$$C_t^{\text{bwd}} = f_t^{\text{bwd}} \odot C_{t+1}^{\text{bwd}} + i_t^{\text{bwd}} \odot \tanh\left(W_C^{\text{bwd}} \cdot \left[h_{t+1}^{\text{bwd}}, x_t\right] + b_C^{\text{bwd}}\right) \quad \text{Eqn 30}$$

$$o_t^{\text{bwd}} = \sigma\left(W_o^{\text{bwd}} \cdot \left[h_{t+1}^{\text{bwd}}, x_t\right] + b_o^{\text{bwd}}\right) \quad \text{Eqn 31}$$

$$h_t^{\text{bwd}} = o_t^{\text{bwd}} \odot \tanh\left(C_t^{\text{bwd}}\right) \quad \text{Eqn 32}$$

The output at time $t$ is obtained by concatenating the hidden states from both the forward and backward LSTM as shown in Eqn 33.

$$h_t^{\text{BiLSTM}} = \left[h_t^{\text{fwd}}, h_t^{\text{bwd}}\right] \quad \text{Eqn 33}$$

where $h_t^{\text{fwd}}$ is the hidden state from the forward LSTM at time $t$, and $h_t^{\text{bwd}}$ is the hidden state from the backward LSTM at time $t$.

*Gated recurrent unit model.* The gated recurrent unit (GRU) is a simplified version of the LSTM model that reduces computational complexity by combining the $f_t$ and $i_t$ into a single update gate (Chung *et al.*, 2014). The GRU model is defined as Eqns 34–37 for update gate ($z_t$), reset gate ($r_t$), candidate hidden state ($\tilde{h}_t$), and $h_t$.

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t]) \quad \text{Eqn 34}$$

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t]) \quad \text{Eqn 35}$$

$$\tilde{h}_t = \tanh(W_h \cdot [r_t \odot h_{t-1}, x_t]) \quad \text{Eqn 36}$$

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \tilde{h}_t \quad \text{Eqn 37}$$

The GRU model, such as LSTM and BiLSTM, was used to model temporal dependencies in TDP measurements, but with fewer parameters, making it computationally faster while maintaining predictive accuracy.

*Random forest.* Random forest learns from multiple decision trees and combines them into an ensemble, outputting either the mode for classification or the mean for regression (Breiman, 2001). The RF regression model is based on Eqn 38.

$$\hat{y} = \frac{1}{T} \sum_{t=1}^{T} f_t(x) \quad \text{Eqn 38}$$

where $\hat{y}$ is the predicted sap flow rate or TDP measurement, $T$ is the number of decision trees, and $f_t(x)$ is the prediction from the $t$-th decision tree for input $x$.

Each decision tree is trained on a bootstrap sample of the training data, and at each split in the tree, a random subset of features is considered to reduce overfitting and increase model robustness.

*Support vector regression.* Support vector regression is a type of machine learning algorithm that finds the best fitting function by mapping data to a higher dimensional space using a kernel function (Smola & Schölkopf, 2004). The goal is to find a function that deviates from the true value by less than a certain threshold while maintaining the margin between support vectors. The SVR model can be described as Eqn 39.

$$y = \sum_{i=1}^{N} \alpha_i K(x_i, x) + b \quad \text{Eqn 39}$$

where $y$ is the predicted sap flow rate or TDP measurement for input $x$, $K(x_i, x)$ is the kernel function that measures the similarity between the data points $x_i$ and $x$, $\alpha_i$ are the Lagrange multipliers, $b$ is the bias term, and $N$ is the number of support vectors.

*Gaussian process regression.* Gaussian process regression (GPR) is a probabilistic model that provides a distribution over possible functions, making it well-suited for tasks involving uncertainty (Quinonero-Candela & Rasmussen, 2005). GPR assumes that the data points are sampled from a multivariate Gaussian distribution, and it estimates the underlying function based on the data. The GPR model is defined as Eqn 40.

$$\mathbf{y} = \mathbf{K}(\mathbf{X}, \mathbf{X}) + \epsilon \quad \text{Eqn 40}$$

where $\mathbf{y}$ is the vector of sap flow rate or TDP measurement values, $\mathbf{K}(\mathbf{X}, \mathbf{X})$ is the covariance matrix computed using the kernel function $k(x, x')$, which defines the relationship between data

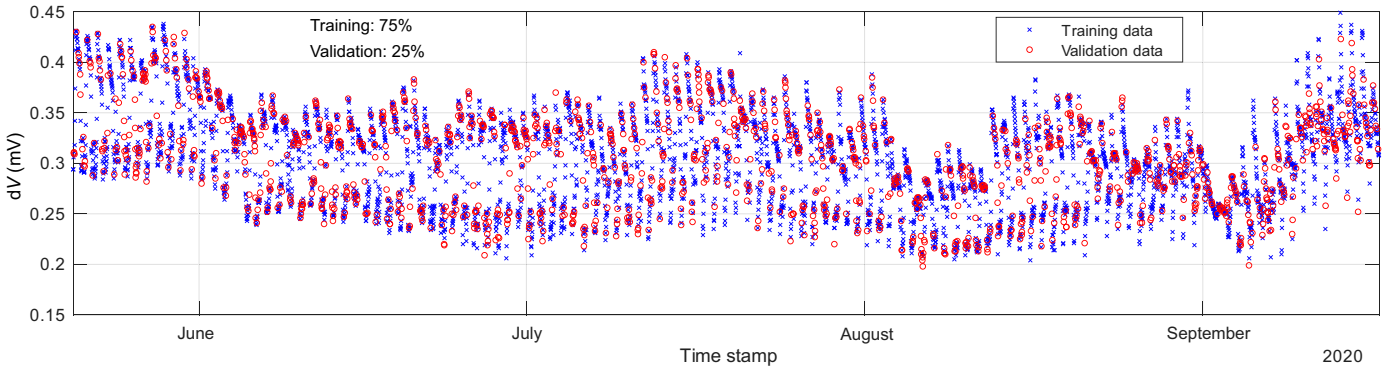



**Fig. 5** Example of data splitting for gap-fill model training and validation. Species under investigation belong to the genus *Populus* (poplars).

points, and $\epsilon$ is the noise term, usually assumed to be Gaussian. For prediction at a new test point $x_*$, the posterior distribution is given by Eqn 41.

$$\mathbf{y}_* \mid \mathbf{X}, \mathbf{y}, x_* \sim \mathcal{N}\left(\mu_*, \sigma_*^2\right) \quad \text{Eqn 41}$$

where $\mu_* = \mathbf{k}(x_*, \mathbf{X})^T \left(\mathbf{K}(\mathbf{X}, \mathbf{X}) + \sigma_n^2 \mathbf{I}\right)^{-1} \mathbf{y}$ is the mean of the posterior distribution, $\sigma_*^2 = k(x_*, x_*) - \mathbf{k}(x_*, \mathbf{X})^T (\mathbf{K}(\mathbf{X}, \mathbf{X}) + \sigma_n^2 \mathbf{I})^{-1} \mathbf{k}(x_*, \mathbf{X})$ is the variance. $k(x_*, X)$ is a vector of covariances between the test point $x_*$ and all training points $X$, computed using the kernel function $k(x, x')$. $K(X, X)$ is the covariance matrix of the training data points $X$, also computed using the kernel function. $\sigma_n^2$ is the noise variance, which accounts for observation noise in the training data. $I$ is the identity matrix, ensuring numerical stability when inverting the matrix. $y$ is the observed values (sap flow rate or TDP measurement values) corresponding to the training data points $X$.

*Kernel regression.* Kernel regression (KR) is a nonparametric method that estimates the conditional expectation of the target variable based on a kernel function (Härdle & Vieu, 1992). KR uses a weighted average of the target values, in which the weights are determined by the similarity between the test point $x$ and each training point $x_i$. The general form of KR is given as Eqn 42.

$$\widehat{y}(x) = \frac{\sum_{i=1}^{n} k(x, x_i) y_i}{\sum_{i=1}^{n} k(x, x_i)} \quad \text{Eqn 42}$$

where $k(x, x_i)$ is also a kernel function that measures the similarity between $x$ and $x_i$ as described previously for GPR, and $y_i$ is the sap flow rate or TDP measurement value corresponding to $x_i$.

**Training procedure** The cleaned dataset was split into training and validation sets to prevent overfitting, and users can determine their splitting ratio based on specific objectives in SAPFLOWER. To obtain better training and validation datasets, the splitting process was designed to ensure that no interpolated or missing data points were included in the training data. During training, the split training and validation datasets will be plotted for users to check the data distribution across the study period if the user has enabled the plotting function (Fig. 5). We used a fixed seed value of 42 when splitting the data for training and validation to ensure that users can reproduce their analyses consistently on the same dataset.

For sap flow modeling and prediction using RNNs, time stamps (including day of the year (DOY), hours, and minutes) are typically decomposed into discrete time components and encoded with one-hot encoding. This process transforms each time component into a binary vector, where '1' represents the specific value and all other positions are marked as '0'. This approach allows the model to capture the temporal patterns in sap flow data without introducing artificial relationships between time points, enabling the RNNs to more effectively recognize periodic trends and cyclical behavior in TDP measurements over time. The manual provides detailed procedures and guidelines for setting and selecting appropriate parameters for deep learning models (Wang, 2024).

To standardize the environmental variables (e.g., VPD, PAR, air temperature, and relative humidity), users can determine whether each variable should be transformed/scaled to have a zero mean and a unit variance by checking feature scaling for model training. The scaling transformation is given in Eqn 43.

$$X_{\text{scaled}} = \frac{X - \mu}{\sigma} \quad \text{Eqn 43}$$

where $X$ represents the original feature value, $\mu$ is the mean of the feature across the training set, and $\sigma$ is the SD. This normalization prevents large-valued variables from dominating the training process and helps the model converge faster.

The adaptive moment estimation (Adam), stochastic gradient descent with momentum, and root mean square propagation can be selected and used to minimize the mean squared error (MSE)

loss function during model (e.g. RNNs) training. The loss function is defined as Eqn 44.

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \widehat{y}_i\right)^2 \quad \text{Eqn 44}$$

where $y_i$ is the observed TDP measurement, $\widehat{y}_i$ is the predicted value, and $n$ is the total number of data points.

The performance of the models was assessed using four key evaluation metrics: mean absolute error (MAE, Eqn 45), root mean square error (RMSE, Eqn 46), Akaike information criterion (AIC, Eqn 47), and Bayesian information criterion (BIC, Eqn 48).

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \widehat{y}_i\right| \quad \text{Eqn 45}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y_i - \widehat{y}_i\right)^2} \quad \text{Eqn 46}$$

$$\mathrm{AIC} = 2k - 2\log_e(L) \quad \text{Eqn 47}$$

$$\mathrm{BIC} = k\log_e(n) - 2\log_e(L) \quad \text{Eqn 48}$$

$$L = \prod_{i=1}^{n}\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{\left(y_i - \widehat{y}_i\right)^2}{2\sigma^2}\right) \quad \text{Eqn 49}$$

The MAE quantifies the average absolute difference between the observed and predicted values. RMSE gives more weight to large errors, making it useful for highlighting cases in which predictions deviate significantly from actual values. AIC evaluates model fit while considering complexity, with lower values indicating better models. BIC introduces a stricter penalty for model complexity by incorporating the sample size $n$, helping to prevent overfitting. In both AIC and BIC, $k$ represents the number of model parameters, and $L$ is the maximum likelihood of the model, which can be calculated as Eqn 49. The $y_i$ are the observed values, and $\widehat{y}_i$ are the predicted values, and $\sigma^2$ is the estimated variance of residuals.

**Gap-filling** Once users have trained the model, they will be able to make predictions and perform gap-filling. Depending on the model that users selected and trained, the prediction and gap-filling will be conducted based on trained model and prediction variables used for model training. To maintain the 'good' raw data intact, gap-filling will only be conducted for missing and filtered (bad data) gaps. SapFlower allows users to train multiple models and make predictions in a high-throughput way, and then select one of the trained and predicted models for gap-filling for each tree or sensor. Besides using RMSE, AIC, and BIC, users can also review the cleaned raw data and gap-filled data through tables and plots to decide whether to train a new model.

**Sap flow calculation** Once users complete gap-filling, they can choose to export the calculated $K$ and/or $F$ values for further sap flow calculations, or they can obtain the $K$ and $F$ values directly in SapFlower. Specifically, based on users' data structure and objectives, they calculate hourly and daily sap flux density using Eqns 50–52.

$$F_{\mathrm{interval}} = F \times T_s \quad \text{Eqn 50}$$

$$F_{\mathrm{hour}} = \sum_{i=1}^{N_{\mathrm{hour}}} F_{\mathrm{interval},i} \quad \text{Eqn 51}$$

$$F_{\mathrm{day}} = \sum_{j=1}^{N_{\mathrm{day}}} F_{\mathrm{interval},j} \quad \text{Eqn 52}$$

where $F$ is sap flux density, $T_s$ is user-defined time step in seconds. The $F_{\mathrm{interval}}$, $F_{\mathrm{hour}}$, and $F_{\mathrm{day}}$ are the sap flux density in a user-defined time step, 1 h, and 1 d, respectively. $N_{\mathrm{hour}}$ and $N_{\mathrm{day}}$ are the number of intervals in 1 h and 1 d, respectively.

The calculated sap flux density $F$ gives the volume of water transported per unit of conducting area per time unit. Users can easily export the $K$ and $F$ values to calculate total sap flow for further water use analysis.

**Sapwood area scaling and modeling** Once users complete gap-filling, they may need sapwood area measurements to integrate with the exported $F$ values for calculating sap flow rate (water use) in SapFlower. Since sapwood area can vary throughout the growing season, users may consider scaling or modeling their sapwood area measurements to obtain daily or finer scale estimates. To support this, SapFlower provides three methods for scaling and modeling sapwood area measurements across the study period. First, we assume that most species exhibit a faster sapwood area growth rate during the early months of the growing season than the later months (Fig. 6). This assumption is based on the factors such as LAI, canopy volume, heat stress, resource availability, and sapwood allocation mechanisms (Schippers *et al.*, 2015; Decuyper *et al.*, 2020; Helluy *et al.*, 2020). Research suggests that the early-growing season provides optimal conditions for sapwood development, and during this time, trees allocate substantial resources to produce new sapwood (or earlywood), which functions as the living xylem responsible for water transport (Utsumi *et al.*, 2003). Second, we introduced a weight curve approach to estimate sapwood area increments throughout the growing season or study period (Fig. 6). Specifically, users can draw a curve to represent the sapwood area increment over time. If initial and final sapwood area measurements are available, these weights can be applied to scale the sapwood area across the study period (Eqn 53).

$$A_i = A_{\mathrm{start}} + \frac{I \times W_i}{T} \quad \text{Eqn 53}$$

where $A_{\mathrm{start}}$ is the starting sapwood area, $I$ is the increment ($A_{\mathrm{end}} - A_{\mathrm{start}}$), $W_i$ is the weight value for the $i$-th timestamp, and $T$ is the total sum of all weights.

For users who lack knowledge of the general growth pattern or do not have detailed sapwood area measurements, SapFlower

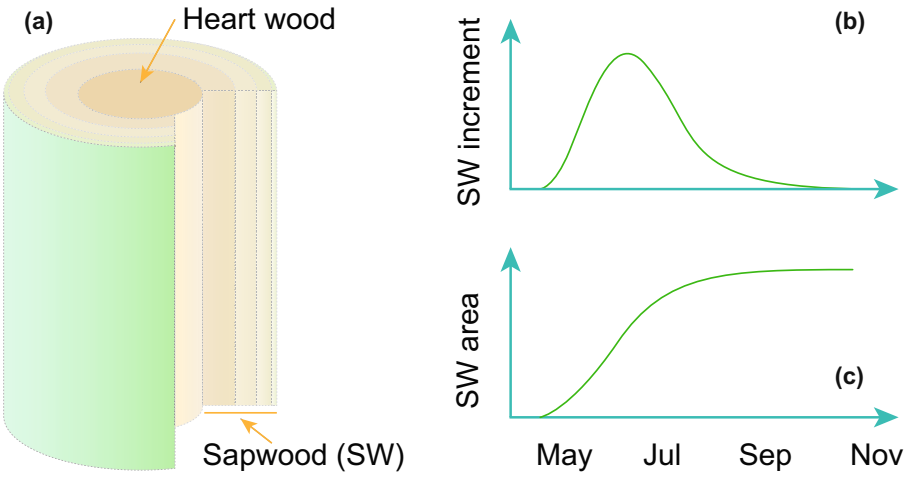



**Fig. 6** Assumptions of sapwood area increment and sapwood area dynamics across the growing season. Specifically, trees produce more sapwood (earlywood) during the early-growing season than the later growing season (a), sapwood area increment follows a sinusoidal pattern (b), and total sapwood area follows a logistic growth model (c).

offers the option to use the generalized logistic model (Eqn 54), provided they have at least six sapwood area measurements during the study period to get reasonable predictions.

$$y(t) = A + \frac{(K-A)}{\left(1 + \nu \cdot e^{-C\cdot(t-B)}\right)^{Q}}^{\frac{1}{\nu}} \quad \text{Eqn 54}$$

where $A$ represents the lower asymptote, which is the baseline value of the quantity when growth is minimal; $K$ is the upper asymptote, representing the maximum value the quantity can approach as growth reaches a saturation point. The parameter $C$ controls the rate at which the growth occurs, and $B$ indicates the time of maximum growth or the inflection point of the curve. The term $v$ introduces a symmetry factor, adjusting the curve's shape and allowing for flexibility in the steepness of the transition from slow to rapid growth. $Q$ controls the growth distribution, affecting how the growth rate is distributed over time. Finally, the parameter $\nu$ further modulates the steepness and sharpness of the curve, making it more adaptable to different growth scenarios.

**Inference and efficiency** Considering that data loading and writing speeds are largely influenced by hardware performance, we focused our testing on SapFlower's data cleaning, model training, and prediction processes. Users can also evaluate SapFlower's efficiency by reviewing the output text area after each action. To assess SapFlower's capability to handle large datasets, we utilized a data frame containing measurements from 96 sensors recorded continuously between May 20, 2020, and October 31, 2020. These data were logged at 30-min intervals, resulting in *c.* 7900 total records. Additionally, to evaluate the impact of model training hyperparameters on both training and inference times, we tested various data splitting ratios, training epochs, learning rates, and the number of hidden layers for RNNs.

### Hardware, software, and statistics

We developed SapFlower using Matlab R2024a on a customized desktop running Windows 11, equipped with an Intel Core i7-12700KF CPU, 128 GB of memory, and a 12 GB NVIDIA RTX 3080 Ti GPU. For macOS compatibility, we compiled SapFlower on a MacBook Pro with macOS 14.5, featuring an M3 PRO Chip and 36 GB of memory. All testing related to data cleaning, model training, and gap-filling was conducted on the Windows desktop. All data calculations and visualization were conducted in SapFlower and Python (v.3.12.4).

## Results

### Data cleaning and preprocessing

SapFlower, integrated manual and automated visual data cleaning and preprocessing functionalities, can efficiently clean and improve the quality of the raw sap flow data. The autocleaning feature, utilizing the IQR method to identify outliers, removed over 90% of the noisy data across all tested 96 sensors, depending on the specific filtering parameters (Fig. 7; Video 1). Manual validation of the results showed that the automatic cleaning retained legitimate variations in sap flow while excluding problematic data points. Fig. 4 presents an example of sap flow raw data, manually cleaned data, and autocleaned data. Adjusting the filtering parameters automatically removed outliers, spikes, and measurements with high and low variation. Rolling window analysis further removed stagnation points in the data, improving the overall data quality. To help users determine whether to delete some bad sap flow data, SapFlower provides a preview of $K$ values, normalized VPD, and $\mathrm{d}T_{max}$ (for $\Delta T_M$) baseline on one plot (Fig. 3; Video 1).

### Model training and gap-filling

We tested SapFlower's model training and gap-filling capabilities using precleaned data and found that machine learning and deep learning models, such as RF and RNNs, are effective for short- and long-term gap-filling (Fig. 8), while linear models are particularly useful for predicting one sensor's data based on another (Fig. S1). Users can train well-fitted models by adjusting hyperparameters, including the validation data split percentage, the number of epochs (iterations), gradient thresholds, learning rate,

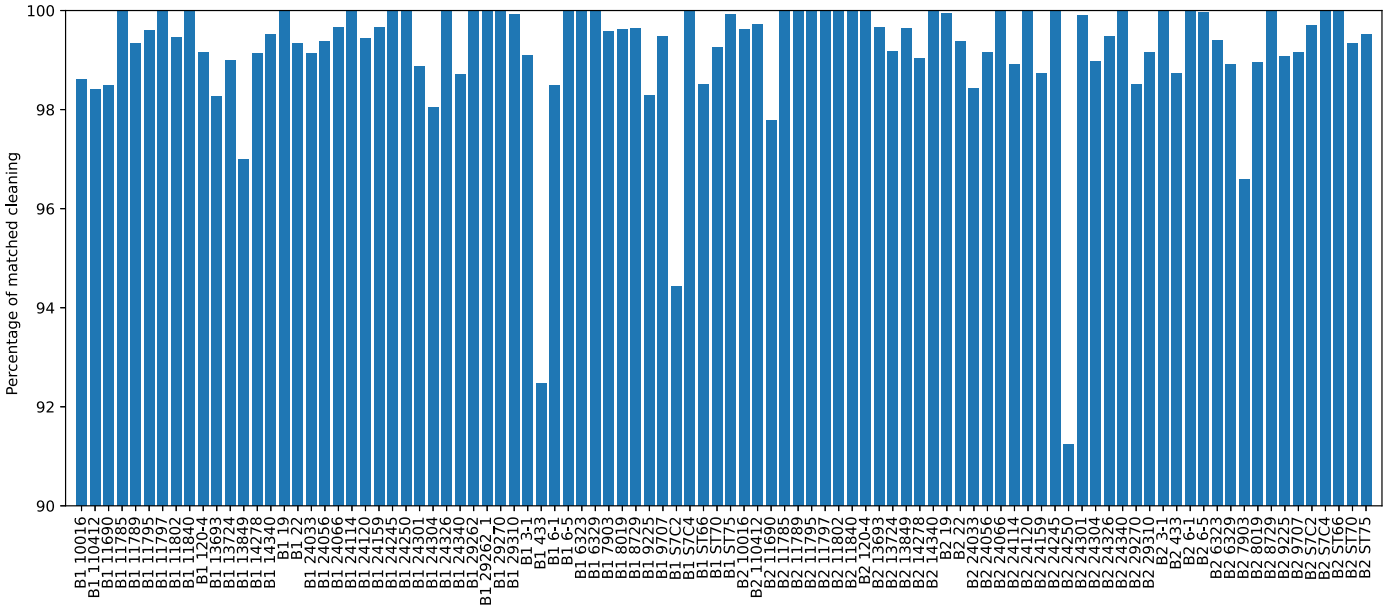
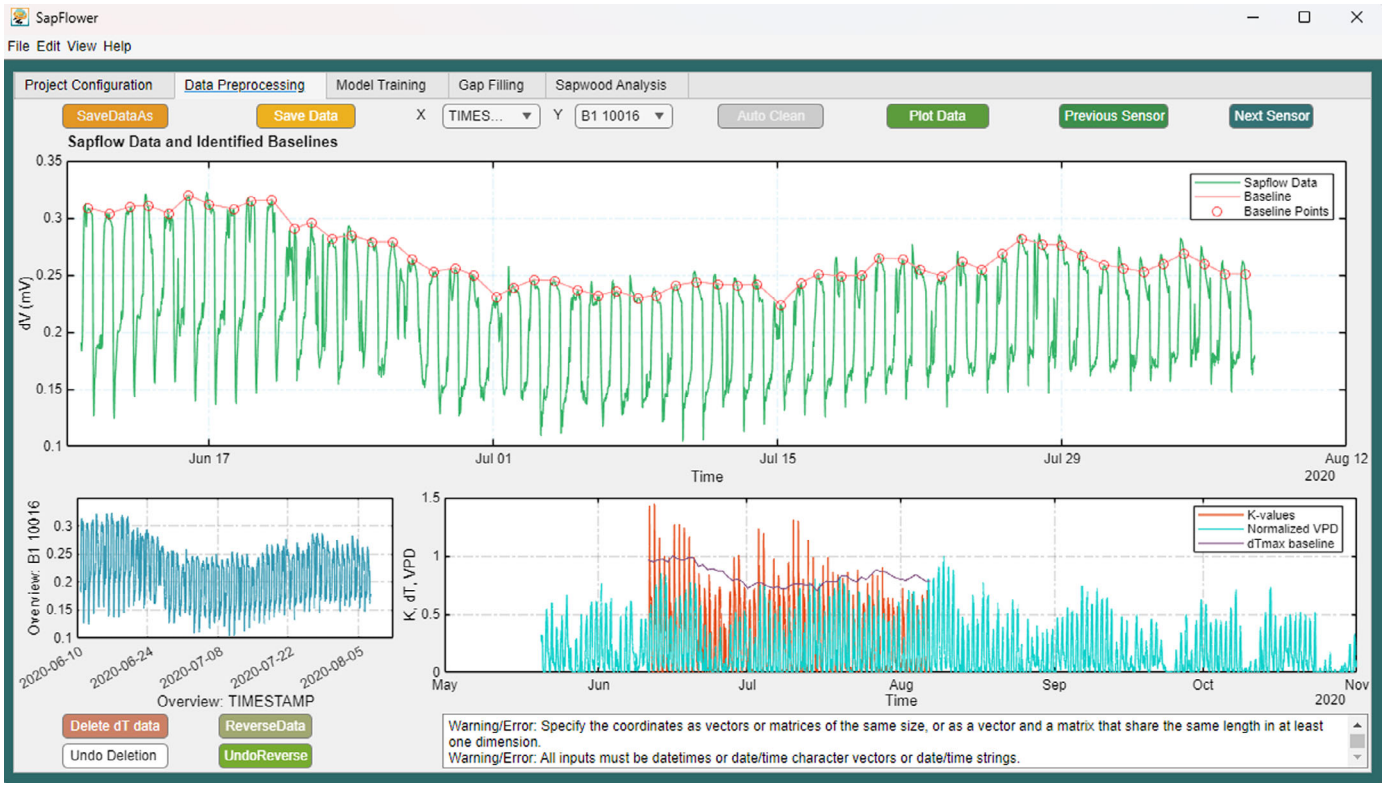



**Fig. 7** Comparison between manually cleaned and autocleaned sap flow raw measurements across 96 sensors. Species under investigation belong to the genus *Populus* (poplars).




**Video 1** Data cleaning in SapFlower.

and the number of hidden units for RNN training (Fig. 9; Video 2). Detailed descriptions on how to set these hyperparameters are given in the manual (Wang, 2024). During model training for gap-filling, we fine-tuned hyperparameters to achieve an RMSE of *c.* 10% or less of the average raw sap flow measurements (e.g. in $\Delta V$, millivolts). Trained models were then used to make predictions based on selected variables, such as VPD, PAR, and time stamp (e.g. DOY and TimeOfDay, to prevent

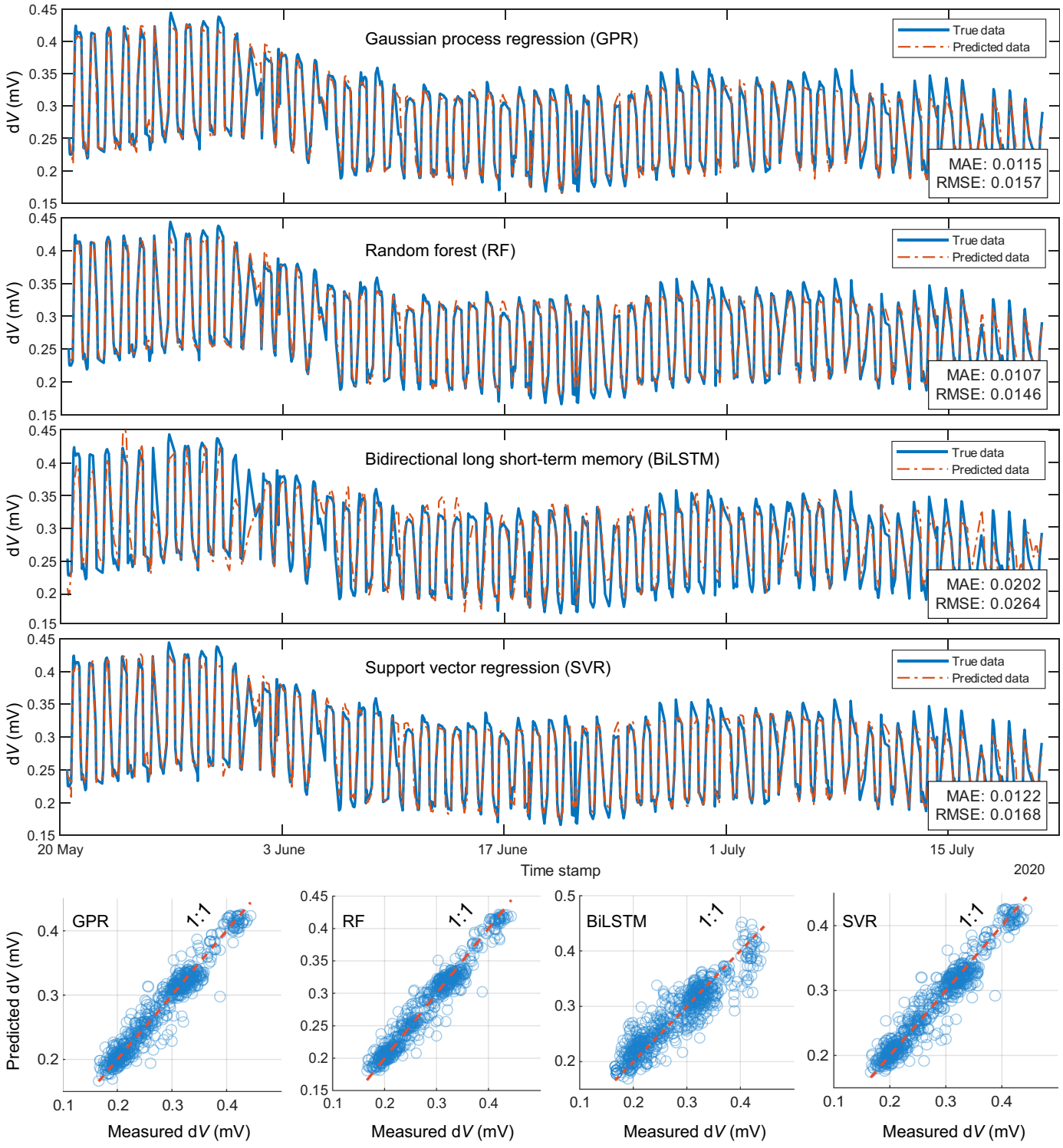



**Fig. 8** Examples of machine- and deep learning model validation results. MAE, mean absolute error; RMSE, root mean square error. Species under investigation belong to the genus *Populus* (poplars).

misalignment of day and night sap flow measurements) for the entire growing season or available environmental data. After predictions were made, cleaned raw sap flow measurements and predicted sap flow data were combined to fill gaps (Fig. 10; Video 2). We also evaluated SapFlower's performance using SAPFLUXNET data and confirmed its effectiveness in high-throughput data management and gap-filling functionalities. For example, users can examine environmental data, train models, and gap-fill sap flow rate data using environmental data provided by SAPFLUXNET. Additionally, they can gap-fill the

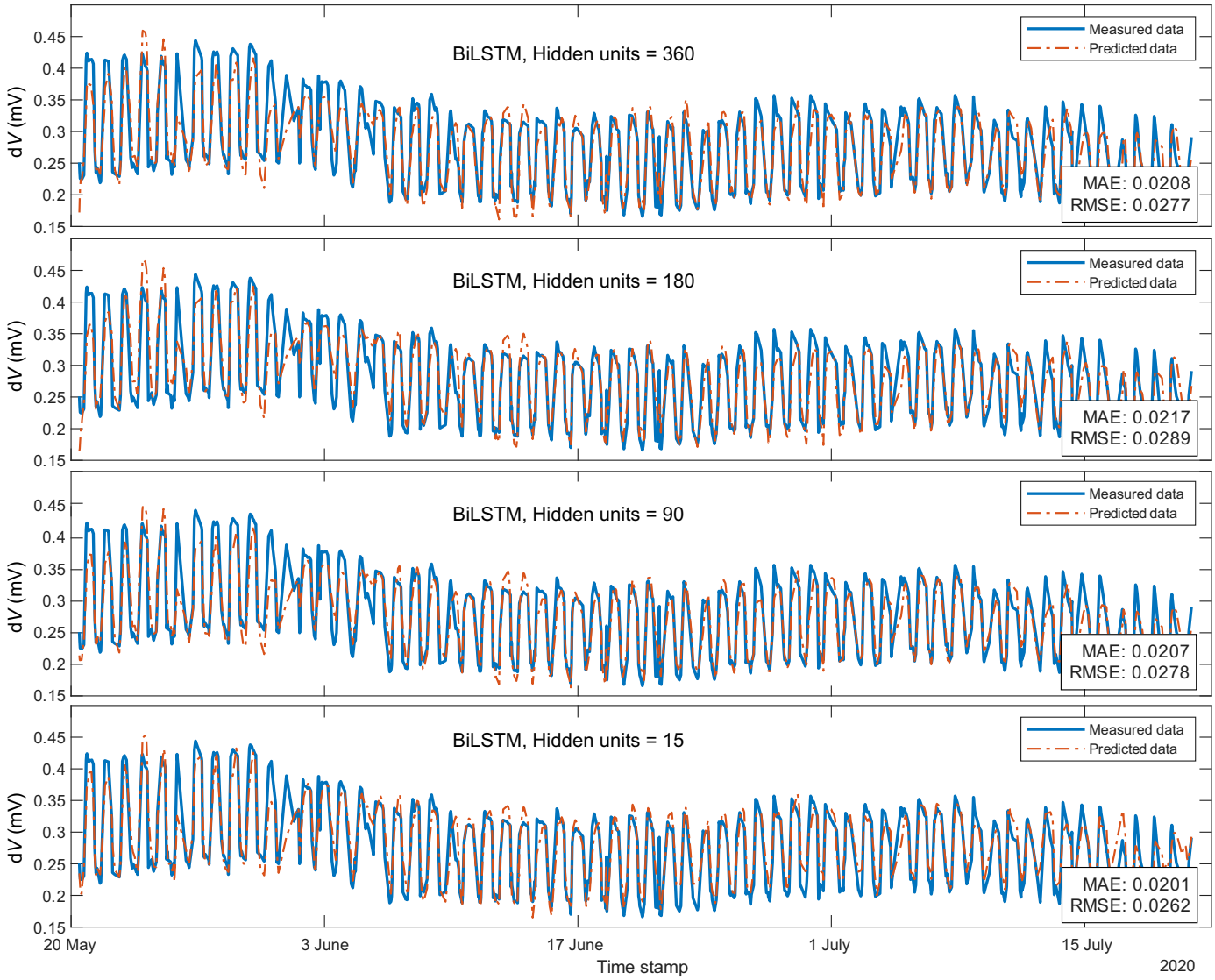



**Fig. 9** Examples of training deep learning models, such as bidirectional long short-term memory (BiLSTM), with various hyperparameters to improve performance. Mean absolute error (MAE) and root mean square error (RMSE) are used as evaluation metrics. Species under investigation belong to the genus *Populus* (poplars).

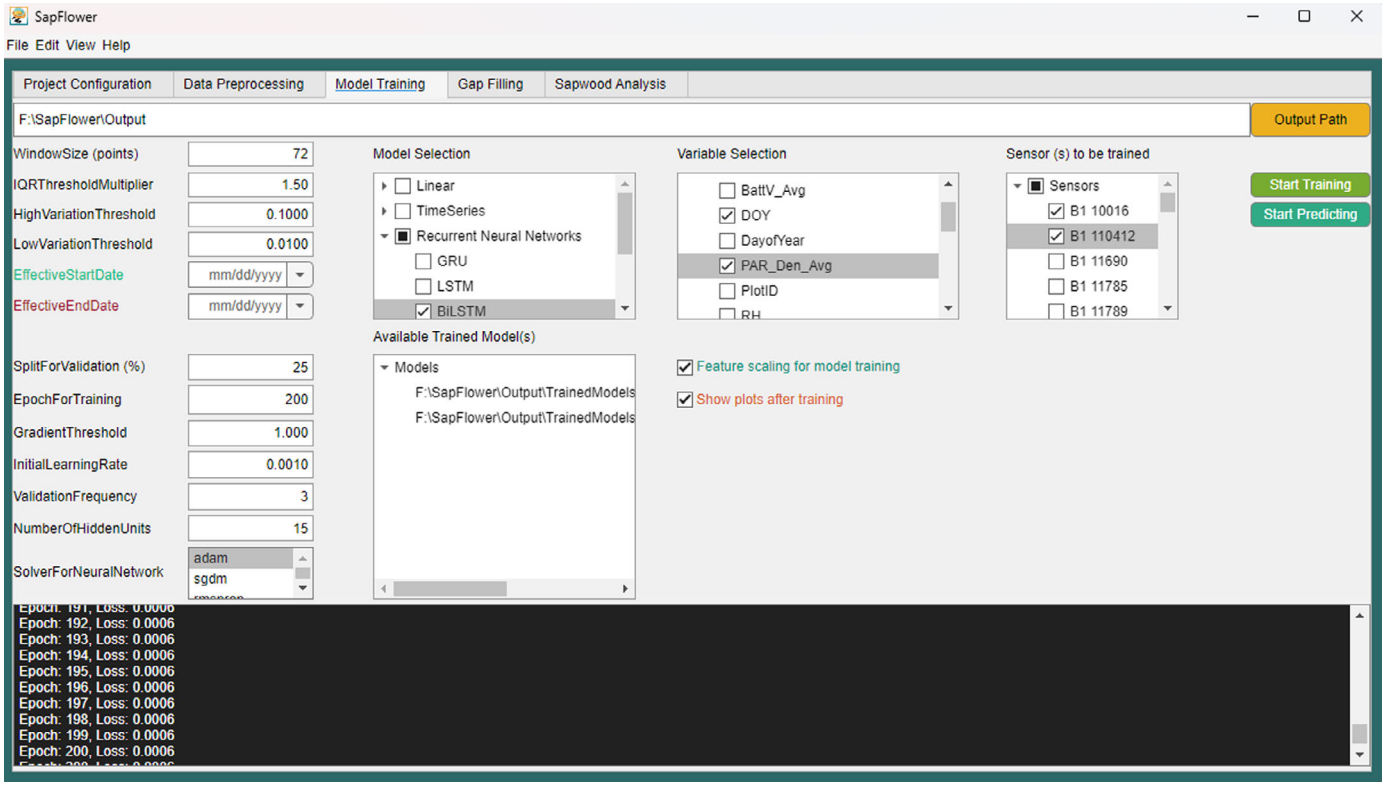
sap flow rate of one sensor or tree using data from another (Figs S1, S2; Video 3).

## SapFlower efficiency

The evaluation of SapFlower's performance demonstrated strong efficiency across data loading, cleaning, model training, and prediction tasks (Table 2). Data loading, filtering, and plotting were completed in 6.38 s, with single sensor data cleaning requiring only 0.1 s, showing rapid preprocessing capabilities. SVR is the fastest algorithm for model training and prediction (followed by RF), with a training and prediction time of $< 0.5$ s. In comparison, GPR and KR require 3–4 s for model training and *c.* 0.2 s for prediction. More complex models, including ARX, ARMAX, GRU, LSTM, and BiLSTM, trained efficiently even with increased model complexity and longer epochs. For instance, GRU, LSTM, and BiLSTM models with 200 epochs and 15 hidden units required 22–31 s, while increasing to 30 hidden units extended the training time to 23–36 s. Despite this, prediction times remained very fast, with all models, including complex ones such as BiLSTM, generating predictions in 0.25–0.36 s. These results indicate that SapFlower effectively manages both simple and advanced models, maintaining high efficiency even when handling large datasets and complex RNN architectures.

## Sapwood area scaling and modeling

Depending on the resolution of the user's sapwood area measurements, SapFlower offers three methods to scale and model sapwood area data for calculating total water use at daily, hourly, or even higher temporal resolutions. We tested these methods and found that all three effectively scale and model




**Video 2** Model training and gap-filling in SapFlower.

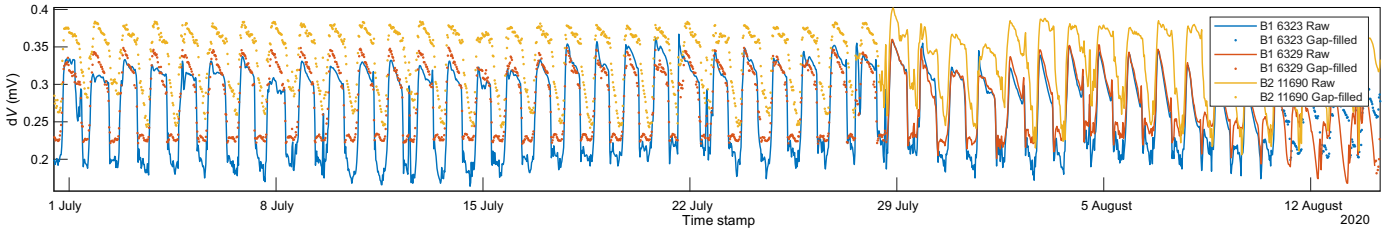



**Fig. 10** Example of gap-filling of three sensors. Species under investigation belong to the genus *Populus* (poplars).

sapwood area across the study period. Among them, weighted curves and logistic modeling proved particularly helpful and valuable when high-resolution sapwood area measurements were challenging to obtain throughout the study period (Fig. 11; Video 4).

### Sap flow/water use calculations

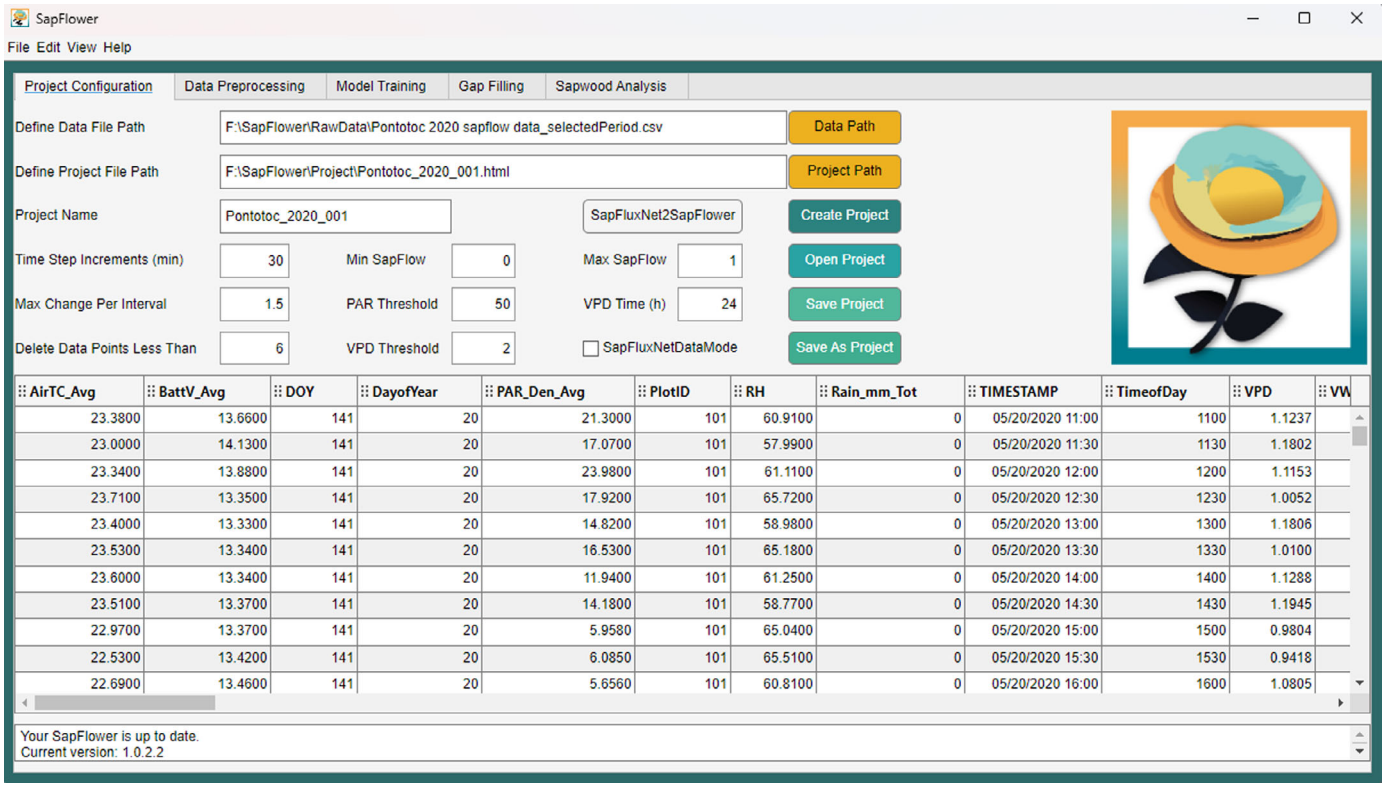
After training and prediction, users can easily calculate and export Granier's $K$. Depending on their objectives and species, they can adjust the coefficients ($\alpha$ and $\beta$) to calculate and export sap flux density data at per-second, hourly, or daily resolution. Specifically, once the model is trained and predictions are made, cleaned raw and predicted data become available. This allows users to load the raw data, apply predictions for gap-filling, calculate $K$ and $F$ for all trees, and export the results in a high-throughput manner.

If users have sapwood area data scaled for the study period, they can calculate per-second, hourly, and daily water use in SapFlower after gap-filling. Once users have exported Granier's $F$ values and clicked the ExportWaterUse button, they will be prompted to provide the path in which the scaled sapwood area data are stored. SapFlower will then match sensor/tree data with the sapwood area data based on time stamps and calculate water use at the selected resolution – per- user defined/selected timestep, such as second, hourly, and/or daily (Fig. 12; Video 5).

## Discussion

### Utility of SapFlower for automated sap flow data processing

SapFlower marks a significant advancement in automating sap flow data analysis, tackling a major challenge in plant physiological research – the labor-intensive and time-consuming processes




**Video 3** SAPFLUXNET data processing in SapFlower.

of data cleaning, gap-filling, and sap flow calculation. By providing a standardized framework, it streamlines data processing, reducing inconsistencies, and mitigating the impact of methodological variations. Integrating these tasks into a single, cohesive platform minimizes the need for manual intervention, allowing researchers to focus more on data interpretation and analysis. The platform's autocleaning feature is particularly valuable, utilizing the IQR and rolling windows methods to identify and remove noise and outliers in sap flow data. This automated cleaning retains the natural variability of sap flow while discarding problematic data, ensuring the quality of data used for subsequent analysis.

Furthermore, SapFlower allows users to customize data cleaning parameters, including setting thresholds for high and low variation and adjusting the window size for filtering, making it adaptable to a wide range of research scenarios. This flexibility is essential in large-scale ecological monitoring programs in which datasets are collected over long periods and across various environmental conditions. Such programs require consistency and automation in data processing to minimize errors and ensure that the data remain comparable over time. By offering automated solutions for these critical tasks, SapFlower ensures that researchers can manage large datasets efficiently while maintaining data integrity.

Additionally, SapFlower's ability to provide visual previews of key metrics – such as Granier's $K$ values, normalized VPD, and $\Delta T_M$ baseline – further enhances its utility (Fig. 3). This feature allows users to quickly assess the quality of the cleaned data and make informed decisions about data retention or deletion. As a result, users can be confident that the sap flow data they are working with are of the highest quality, which is crucial for accurate downstream analyses.

SapFlower also supports SAPFLUXNET data and can be used to clean (if necessary) and gap-fill sap flow rate data. Specifically, SapFlower includes functions to automatically transform SAPFLUXNET format data into a format supported by SapFlower. After compiling the SAPFLUXNET environmental and sap flow data, users can visualize, clean, train gap-filling models, and fill out missing sap flow rate data. It should be noted that when cleaning and visualizing SAPFLUXNET sap flow rate data, users may still see the normalized VPD, $K$, and $dT_{max}$ baselines (although $K$ and $dT_{max}$ baselines should not be presented for sap flow rate data). Additionally, the $K$ values will exhibit a 'reciprocal' pattern with normalized VPD compared with TDP raw measurements (Fig. S2).

## Superiority of machine learning models for gap-filling

The results of our study feature the superior performance of machine learning and deep learning models, particularly RF, SVR, GPR, and BiLSTM networks, in gap-filling sap flow time-series data. These models are specifically designed to capture long-term dependencies and temporal patterns in sequential data, making them particularly effective for datasets such

**Table 2** Efficiency testing results of SapFlower for data loading, model training, and prediction.

| Action | Model | Data split for validation | Epoch/no. of learning cycles | Learning rate | Hidden units | Time (s) |
|---|---|---|---|---|---|---|
| All data preprocessing | | 25% | na | na | na | 6.38 |
| Single sensor data cleaning | | 25% | na | na | na | 0.10 |
| Training and validation | Simple linear | 25% | na | na | na | 0.29 |
| | Multiple linear | 25% | na | na | na | 0.61 |
| | ARX | 25% | na | na | na | 0.36 |
| | ARMAX | 25% | na | na | na | 0.34 |
| | GRU | 25% | 100 | 0.001 | 15 | 11.00 |
| | LSTM | 25% | 100 | 0.001 | 15 | 12.00 |
| | BiLSTM | 25% | 100 | 0.001 | 15 | 17.00 |
| | RF | 25% | 100 | na | na | 0.77 |
| | SVR | 25% | na | na | na | 0.24 |
| | GPR | 25% | na | na | na | 3.40 |
| | KR | 25% | na | na | na | 3.68 |
| | GRU | 25% | 200 | 0.001 | 15 | 22.00 |
| | LSTM | 25% | 200 | 0.001 | 15 | 22.00 |
| | BiLSTM | 25% | 200 | 0.001 | 15 | 31.00 |
| | RF | 25% | 200 | na | na | 1.53 |
| | GRU | 25% | 200 | 0.001 | 30 | 23.00 |
| | LSTM | 25% | 200 | 0.001 | 30 | 23.00 |
| | BiLSTM | 25% | 200 | 0.001 | 30 | 36.00 |
| | RF | 25% | 200 | na | na | 1.53 |
| | GRU | 15% | 200 | 0.001 | 30 | 23.00 |
| | LSTM | 15% | 200 | 0.001 | 30 | 24.00 |
| | BiLSTM | 15% | 200 | 0.001 | 30 | 37.00 |
| | RF | 15% | 200 | na | na | 1.59 |
| | SVR | 15% | na | na | na | 0.43 |
| | GPR | 15% | na | na | na | 3.41 |
| | KR | 15% | na | na | na | 3.43 |
| Prediction | Simple linear | na | na | na | na | 0.03 |
| | Multiple linear | na | na | na | na | 0.02 |
| | ARX | na | na | na | na | 0.04 |
| | ARMAX | na | na | na | na | 0.04 |
| | GRU | na | na | na | 15 | 0.28 |
| | LSTM | na | na | na | 15 | 0.25 |
| | BiLSTM | na | na | na | 15 | 0.36 |
| | GRU | na | na | na | 30 | 0.28 |
| | LSTM | na | na | na | 30 | 0.27 |
| | BiLSTM | na | na | na | 30 | 0.36 |
| | RF | na | na | na | na | 0.17 |
| | SVR | na | na | na | na | 0.05 |
| | GPR | na | na | na | na | 0.22 |
| | KR | na | na | na | na | 0.22 |

The models tested include linear: ARMAX, autoregressive moving average with exogenous inputs; ARX, autoregressive with exogenous inputs; BiLSTM, bidirectional LSTM; GPR, Gaussian process regression; GRU, gated recurrent unit; KR, kernel regression; LSTM, long short-term memory; RF, random forest; SVR, support vector regression. na, not applicable. All tests were conducted with the 'ShowPlotsAfterTraining' option disabled.

as sap flow measurements, in which environmental and physiological conditions vary over time. Traditional statistical models, such as simple and multiple linear regression or ARX models, while useful for short-term gap-filling, were found to be insufficient for capturing the complexity and variability of sap flow data across an entire growing season, albeit linear models are valuable when users try to use one sensor's data to gap-fill another (Fig. S1).

RF, SVR, GRU, and BiLSTM models excel at recognizing these long-term patterns and are particularly adapted at filling out missing data over extended periods, ensuring a more reliable and continuous dataset for further analysis (Amir *et al.*, 2021; Li *et al.*, 2022). This is especially important in ecological studies in which missing data can lead to biased conclusions or gaps in understanding plant responses to environmental stressors. The ability of deep learning models to handle larger volumes of data and capture subtle, long-term variations makes them invaluable for ecological research (Perry *et al.*, 2022).

Another key advantage of using machine- and deep learning models for gap-filling is the reduction in user bias. Manual gap-filling, which has traditionally been used in sap flow studies, is prone to inconsistencies and subjectivity (Poyatos *et al.*, 2016). Researchers may unintentionally introduce errors by filling gaps based on personal assumptions or limited data. By automating

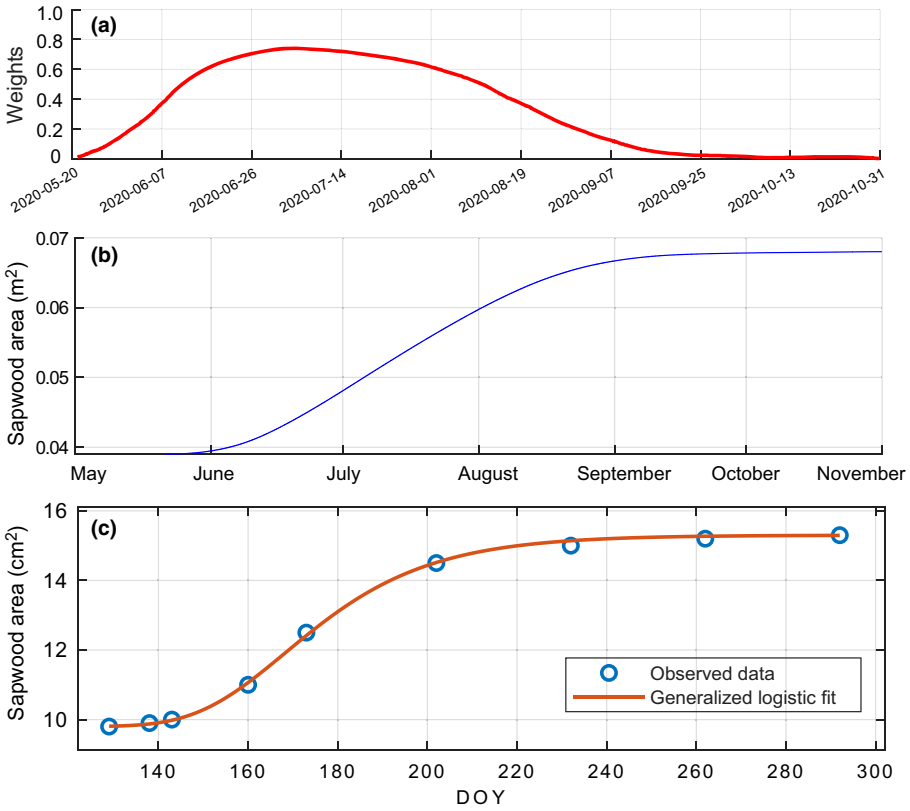
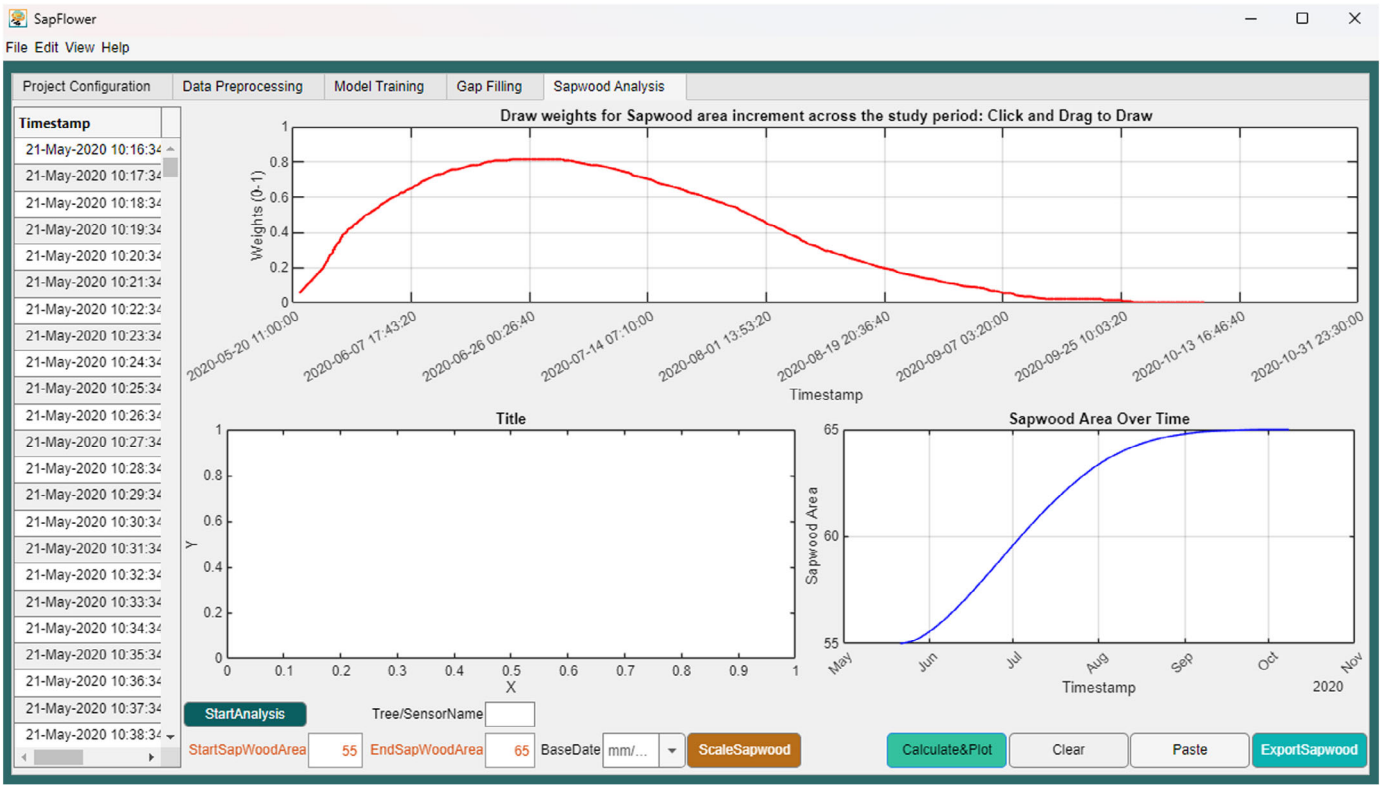



**Fig. 11** Weight curve (a), sapwood area scaling (b), and modeling (c) in SapFlower.




**Video 4** Sapwood area scaling and modeling in SapFlower.

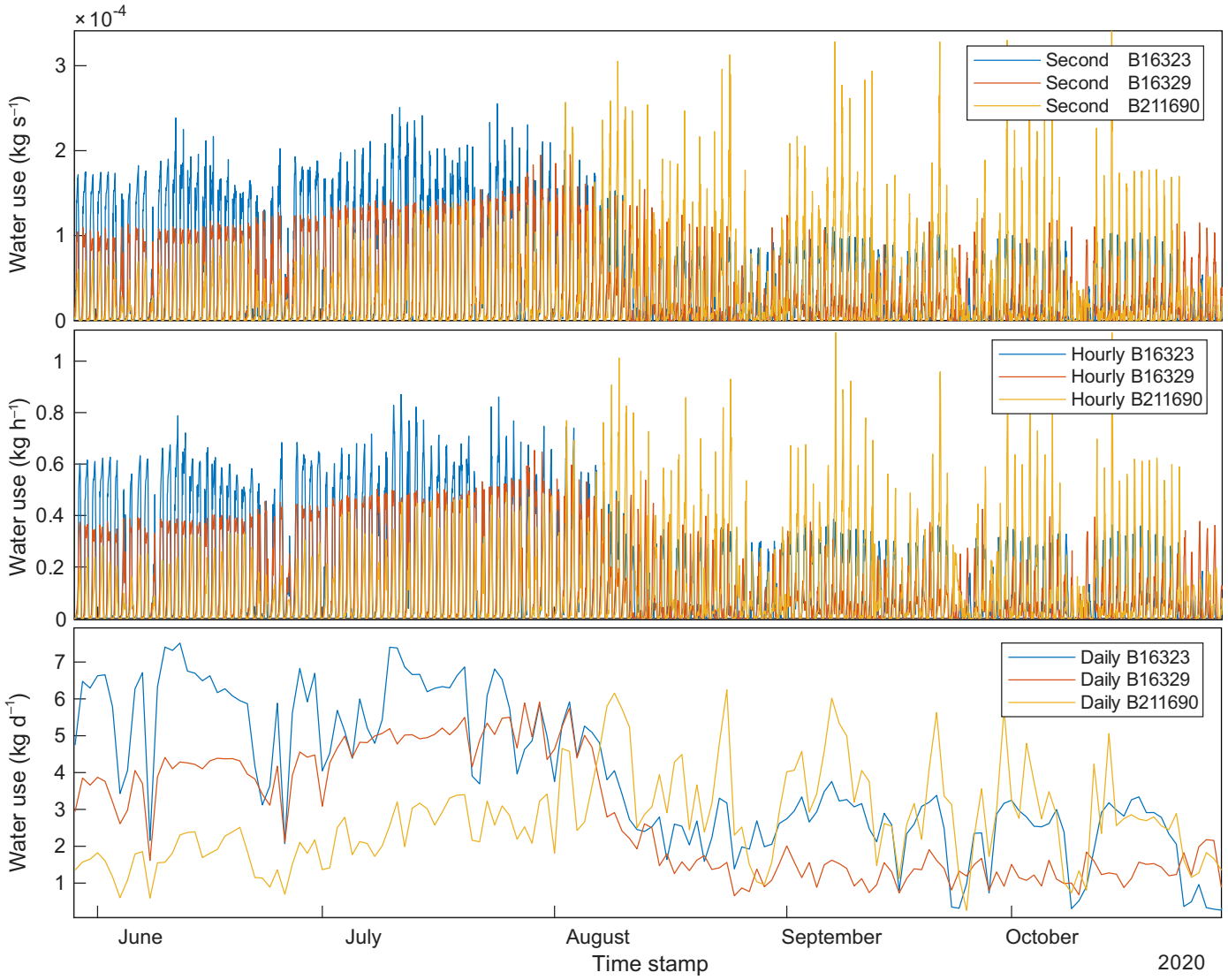



**Fig. 12** Multiscale water use analysis and visualization in SapFlower. Coefficients ($\alpha$ and $\beta$) used are 0.11899 and 1.231. Species under investigation belong to the genus *Populus* (poplars).

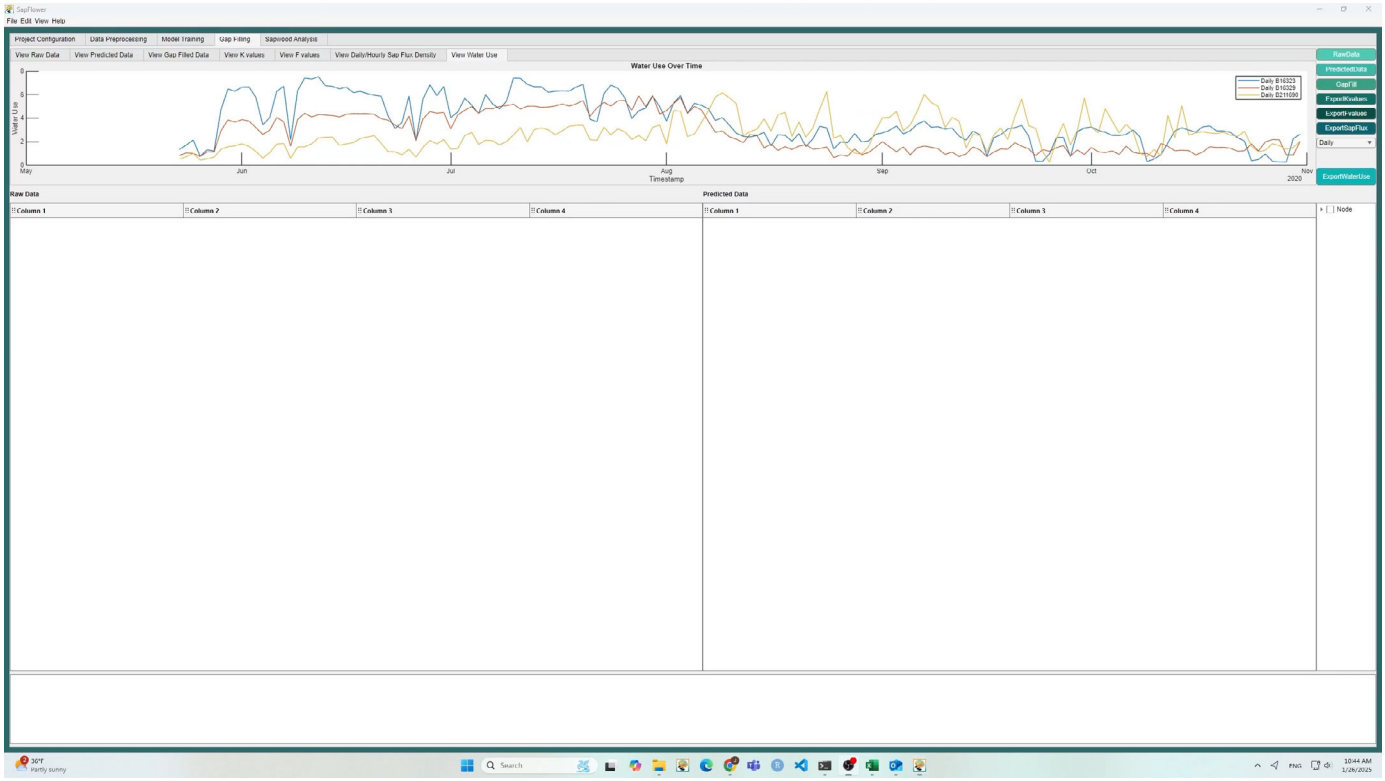
this process through models such as BiLSTM, SapFlower eliminates these inconsistencies and ensures that the gap-filling process is both reproducible and objective. As a result, the quality of the gap-filled data is substantially improved, leading to more reliable interpretations of sap flow dynamics.

## Applicability to various ecological studies

SapFlower's flexible and adaptable framework makes it highly applicable to a wide range of ecological studies. One of its core strengths lies in its ability to handle diverse environmental conditions and species. Users can define their own environmental variables, such as VPD, PAR, and temperature, and incorporate these variables into the model training process. This flexibility allows SapFlower to be used in a variety of experimental setups, from controlled laboratory studies to field-based research in natural ecosystems.

SapFlower's ability to handle data from different tree species and ecosystems – such as eastern cottonwood and *Populus* hybrids in this study – demonstrates its versatility. By incorporating project-specific configuration settings, researchers can save and replicate their workflows, making it easier to conduct long-term monitoring studies. This feature is especially useful in ecological studies that require consistent data processing over several growing seasons, in which replicability and workflow standardization are essential for ensuring data comparability.

Additionally, SapFlower's user-friendly interface democratizes the use of advanced machine learning methods. Although there are some R packages and code available for sap flow gap-filling, the complexity of training and applying models such as convolutional neural networks, LSTM, and RF would have been a barrier for researchers without a strong programming background (Amir *et al.*, 2021; Peters *et al.*, 2021; Li *et al.*, 2022). SapFlower addresses this gap by offering an




**Video 5** Water use analysis in SapFlower.

intuitive interface for model training and data analysis, empowering researchers from diverse disciplines to utilize advanced computational techniques in their studies. Furthermore, SapFlower seamlessly integrates project configuration, data visualization, modeling, gap-filling, sapwood area scaling, and water use analysis, providing a comprehensive solution for the entire sap flow data analysis workflow. This innovation has the potential to transform data analysis into plant physiology and ecology, enabling researchers to derive deeper insights from their data without requiring specialized computational expertise.

## Limitations and future improvements

Despite its advantages, SapFlower has some limitations that present opportunities for future improvement. One area for improvement is the integration of cloud computing capabilities. As ecological datasets grow larger and more complex, the need for high-performance computing becomes increasingly important. While SapFlower is currently optimized for local computation, incorporating cloud-based processing would allow it to handle even larger datasets more efficiently and speed up model training for complex deep learning models. Additionally, expanding SapFlower's support for additional sap flow measurement methods is a key future direction.

Another challenge is handling data anomalies arising from phenological changes, extreme weather events, and sensor signal degradation during multiyear installations. These factors can introduce biases or lead to the misclassification of valid data as outliers or small variations, ultimately affecting the accuracy of sap flow calculations. While SapFlower allows users to adjust parameters and thresholds, it is crucial to carefully assess how measurements respond to environmental conditions to ensure reliable interpretations. Additionally, although SapFlower includes a sapwood area calculation feature, further refinement is needed to account for site-specific and temporal variations in sapwood growth. During periods of rapid tree growth, sensor effectiveness may decline, impacting data accuracy. A critical consideration is also needed for species-, age-, and site-specific calibrations to correct for differences in wood anatomy, growth rates, and thermal properties – factors that significantly influence sap flux calculations.

While SapFlower simplifies data processing, users should remain mindful of the broader uncertainties inherent in the TD method, including those related to probe placement, wood-specific thermal properties, and scaling from point measurements to whole-tree water use. Future versions of SapFlower could integrate species-specific correction factors tailored to gymnosperms, diffuse-porous, and ring-porous species, as well as site- and age-dependent calibration adjustments. These enhancements would improve the robustness and accuracy of sap flow estimates across diverse ecological and environmental contexts while ensuring that automation does not overlook critical methodological considerations.

## Conclusions

SapFlower has proven to be an exceptionally efficient and versatile tool for automating sap flow data analysis, effectively streamlining the processes of data cleaning, gap-filling, sapwood area scaling and modeling, and multiscale water use analysis. By integrating advanced machine learning and deep learning models such as RF, SVR, and BiLSTM, SapFlower delivers highly accurate gap-filling over long-term datasets, enhancing the reliability and quality of sap flow measurements. Its customizable features and user-friendly interface make it accessible to a broad range of researchers, from ecologists monitoring large forests to agronomists studying crop water use. While there are opportunities for future enhancements, such as extending support to other sap flow methods and incorporating cloud-based processing, SapFlower already offers substantial benefits by reducing manual effort and minimizing user bias. By democratizing access to sophisticated analytical techniques, SapFlower empowers researchers to gain deeper insights into plant water use, thereby contributing valuable knowledge to the fields of ecology and agriculture. Overall, SapFlower stands out as a cornerstone tool in sap flow research, addressing critical challenges in ecosystem and crop management amidst the evolving demands of climate change.

## Acknowledgements

This study was funded by the USDA National Institute of Food and Agriculture (USDA-NIFA) through the APPS grant (Advancing *Populus* Pathways in the Southeast; 2018-68005-27636), United States Department of Energy (DOE) through the PoSIES (*Populus* in the Southeast for Integrated Ecosystem Services; DE-EE0009280), and USDA-NIFA McIntire Stennis grant (MISZ-067050). This publication is a contribution of the Forest and Wildlife Research Center, Mississippi State University. We thank the four anonymous reviewers for their thoughtful and constructive feedback. JW thanks Xuening Lu, Adam Coates, and Virginia Polytechnic Institute and State University for supporting, supervising, and funding for open access publication of this work.

## Competing interests

None declared.

## Author contributions

JW was involved in conceptualization, methodology, formal analysis, investigation, data curation, software, validation, visualization, and writing – original draft, review and editing. HJR was involved in project administration, funding acquisition, supervision, and writing – review and editing.

## ORCID

Heidi J. Renninger https://orcid.org/0000-0002-2485-9835
Jiaxin Wang https://orcid.org/0000-0003-4808-5085

## Data availability

The data that support the findings of this study are openly available in Zenodo at doi: 10.5281/zenodo.13665919.

## References

**Alizadeh A, Toudeshki A, Ehsani R, Migliaccio K. 2018.** Potential sources of errors in estimating plant sap flow using commercial thermal dissipation probes. *Applied Engineering in Agriculture* **34**: 899–906.

**Amir A, Butt M, Van Kooten O. 2021.** Using machine learning algorithms to forecast the sap flow of cherry tomatoes in a greenhouse. *IEEE Access* **9**: 154183–154193.

**Breiman L. 2001.** Random forests. *Machine Learning* **45**: 5–32.

**Chung J, Gulcehre C, Cho K, Bengio Y. 2014.** Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv*: 1412.3555.

**Decuyper M, Chávez RO, Čufar K, Estay SA, Clevers JG, Prislan P, Gričar J, Črepinšek Z, Merela M, De Luis M. 2020.** Spatio-temporal assessment of beech growth in relation to climate extremes in Slovenia – an integrated approach using remote sensing and tree-ring data. *Agricultural and Forest Meteorology* **287**: 107925.

**Granier A. 1985.** Une nouvelle méthode pour la mesure du flux de sève brute dans le tronc des arbres. *Annales des Sciences forestières: EDP Sciences* **42**: 193–200.

**Granier A. 1987.** Evaluation of transpiration in a douglas-fir stand by means of sap flow measurements. *Tree Physiology* **3**: 309–320.

**Härdle W, Vieu P. 1992.** Kernel regression smoothing of time series. *Journal of Time Series Analysis* **13**: 209–232.

**Helluy M, Prévosto B, Cailleret M, Fernandez C, Balandier P. 2020.** Competition and water stress indices as predictors of *Pinus halepensis* Mill. radial growth under drought. *Forest Ecology and Management* **460**: 117877.

**Hochreiter S. 1997.** *Long short-term memory*. Cambridge, MA, USA: Neural Computation MIT-Press.

**Li Y, Ye J, Xu D, Zhou G, Feng H. 2022.** Prediction of sap flow with historical environmental factors based on deep learning technology. *Computers and Electronics in Agriculture* **202**: 107400.

**Lu P, Urban L, Zhao P. 2004.** Granier's thermal dissipation probe (TDP) method for measuring sap flow in trees: theory and practice. *Acta Botanica Sinica English Edition* **46**: 631–646.

**Lucarini A, Cascio ML, Marras S, Sirca C, Spano D. 2024.** Artificial intelligence and Eddy covariance: a review. *Science of the Total Environment* **950**: 175406.

**Masmoudi M, Mahjoub I, Lhomme J-P, Ben Mechlia N. 2012.** Sap flow measurement by a single thermal dissipation probe in transient regime: implementation of the method and test under field conditions. *Annals of Forest Science* **69**: 773–781.

**Meinzer FC, James SA, Goldstein G. 2004.** Dynamics of transpiration, sap flow and use of stored water in tropical forest canopy trees. *Tree Physiology* **24**: 901–909.

**Niu S, Xing X, Zhang Z, Xia J, Zhou X, Song B, Li L, Wan S. 2011.** Water-use efficiency in response to climate change: from leaf to ecosystem in a temperate steppe. *Global Change Biology* **17**: 1073–1082.

**Oishi AC, Hawthorne DA, Oren R. 2016.** Baseliner: an open-source, interactive tool for processing sap flux data from thermal dissipation probes. *SoftwareX* **5**: 139–143.

**Oishi AC, Oren R, Stoy PC. 2008.** Estimating components of forest evapotranspiration: a footprint approach for scaling sap flux measurements. *Agricultural and Forest Meteorology* **148**: 1719–1732.

**Perry GL, Seidl R, Bellvé AM, Rammer W. 2022.** An outlook for deep learning in ecosystem science. *Ecosystems* **25**: 1700–1718.

**Peters RL, Fonti P, Frank DC, Poyatos R, Pappas C, Kahmen A, Carraro V, Prendin AL, Schneider L, Baltzer JL. 2018.** Quantification of uncertainties in conifer sap flow measured with the thermal dissipation method. *New Phytologist* **219**: 1283–1299.

**Peters RL, Pappas C, Hurley AG, Poyatos R, Flo V, Zweifel R, Goossens W, Steppe K. 2021.** Assimilate, process and analyse thermal dissipation sap

flow data using the TREX R package. *Methods in Ecology and Evolution* **12**: 342–350.

**Poyatos R, Granda V, Molowny-Horas R, Mencuccini M, Steppe K, Martínez-Vilalta J. 2016.** SAPFLUXNET: towards a global database of sap flow measurements. *Tree Physiology* **36**: 1449–1455.

**Quinonero-Candela J, Rasmussen CE. 2005.** A unifying view of sparse approximate Gaussian process regression. *Journal of Machine Learning Research* **6**: 1939–1959.

**Renninger HJ, Pitts JJ, Rousseau RJ. 2023.** Comparisons of biomass, water use efficiency and water use strategies across five genomic groups of *Populus* and its hybrids. *GCB Bioenergy* **15**: 99–112.

**Renninger HJ, Pitts JJ, Wang J. 2024.** Leaf-level physiological strategies related to productivity and plasticity of *Populus* in the Southeastern United States. *Frontiers in Forests and Global Change* **7**: 1467381.

**Schippers P, Vlam M, Zuidema PA, Sterck F. 2015.** Sapwood allocation in tropical trees: a test of hypotheses. *Functional Plant Biology* **42**: 697–709.

**Schuster M, Paliwal KK. 1997.** Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing* **45**: 2673–2681.

**Smith D, Allen S. 1996.** Measurement of sap flow in plant stems. *Journal of Experimental Botany* **47**: 1833–1844.

**Smola AJ, Schölkopf B. 2004.** A tutorial on support vector regression. *Statistics and Computing* **14**: 199–222.

**Steppe K, Vandegehuchte MW, Tognetti R, Mencuccini M. 2015.** Sap flow as a key trait in the understanding of plant hydraulic functioning. *Tree Physiology* **35**: 341–345.

**Tai KS, Socher R, Manning CD. 2015.** Improved semantic representations from tree-structured long short-term memory networks. *arXiv*: 1503.00075.

**Utsumi Y, Sano Y, Funada R, Ohtani J, Fujikawa S. 2003.** Seasonal and perennial changes in the distribution of water in the sapwood of conifers in a sub-frigid zone. *Plant Physiology* **131**: 1826–1833.

**Wang J. 2024.** SapFlower: an automated tool for sap flow data preprocessing, gap filling, and analysis using deep learning. *Zenodo*. doi: 10.5281/zenodo.13665919.

**Wilson KB, Hanson PJ, Mulholland PJ, Baldocchi DD, Wullschleger SD. 2001.** A comparison of methods for determining forest evapotranspiration and its components: sap-flow, soil water budget, eddy covariance and catchment water balance. *Agricultural and Forest Meteorology* **106**: 153–168.

**Yu C, Yao Y, Yang H, Wang X. 2024.** An improved transformer model for sap flow prediction that efficiently utilizes environmental information. *Agricultural Research* 1–12. doi: 10.1007/s40003-024-00807-6.

**Zhang B, Zhang D, Feng Z, Zhang L, Zhang M, Fu R, Wang Z. 2023.** Assessment of the potential of indirect measurement for sap flow using environmental factors and artificial intelligence approach: a case study of *Magnolia denudata* in Shanghai urban green spaces. *Forests* **14**: 1768.

**Zhu L, Hu Y, Zhao X, Zeng X, Zhao P, Zhang Z, Ju Y. 2017.** The impact of drought on sap flow of cooccurring *Liquidambar formosana* Hance and *Quercus variabilis* Blume in a temperate forest, Central China. *Ecohydrology* **10**: e1828.

## Supporting Information

Additional Supporting Information may be found online in the Supporting Information section at the end of the article.

**Fig. S1** Linear model training and validation to predict one sensor/tree sap flow rate with another sensor/tree using SAPFLUXNET data.

**Fig. S2** Examples of patterns in Granier's *K* values and normalized vapor pressure deficit during the processing of thermal dissipation probe raw measurements and sap flow rate data.

Please note: Wiley is not responsible for the content or functionality of any Supporting Information supplied by the authors. Any queries (other than missing material) should be directed to the *New Phytologist* Central Office.