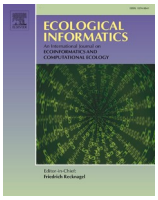
Ecological Informatics 86 (2025) 103014

Contents lists available at ScienceDirect

Ecological Informatics

journal homepage: www.elsevier.com/locate/ecolinf



A comparative study of four deep learning algorithms for predicting tree stem radius measured by dendrometer: A case study

Guilherme Cassales [a,*], Serajis Salekin [b], Nick Lim [a], Dean Meason [b], Albert Bifet [a], Bernhard Pfahringer [a], Eibe Frank [a]

[a] TAIAO, University of Waikato, Hamilton, New Zealand
[b] Scion (New Zealand Forest Research Ltd.), Rotorua 3046, New Zealand

ARTICLE INFO

Keywords:
Point dendrometer
Tree stem radius
Time series forecasting
Deep learning
Artificial intelligence
CNN
LSTM
Transformer
ETSFormer

ABSTRACT

As a dominant terrestrial ecosystem, forests play a pivotal role, which is substantially challenged by climate extremes. At the same time, the practice of patient science to investigate and understand different intricate climate-driven phenomena is no longer an option. On the other hand, recent technological advancements enable scientists to simultaneously collect and analyse a large volume of complex data. High-resolution tree stem radius measurements and predictive simulation through machine learning algorithms offer powerful opportunities for understanding these dynamics. However, when these machine learning methods are applied without careful consideration of data quality, model biases, and other critical factors, their potential is often compromised. In this study, we aimed to evaluate four Deep Learning algorithms (namely CNN, LSTM, Transformer, and ETSFormer), using automatically measured and high temporal resolution tree stem radius data. We explore the complexities of handling voluminous and authentic datasets required by these algorithms. Initial experiments show that it is possible to achieve an MAE as small as 0.0026 mm on the full data. Furthermore, our study delves into the temporal resolution of data, demonstrating the feasibility of using reduced datasets without compromising accuracy levels. Our best result showed that a reduction of 97 % in collection events increases the MAE by only 6 % with the LSTM model, demonstrating that resource use optimisation can be achieved by slightly reducing the temporal resolution of data collection with marginal error increase. This also shows that LSTM can effectively capture longer-term and complex dependencies, which indicates promising future work with additional environmental data.

1. Introduction

Globally, forests are considered the dominant terrestrial ecosystem (Schmitt et al., 2009) and play an important role in regulating water, carbon, energy, and nutrient cycling while providing an important hub for biodiversity conservation (Bar-On et al., 2018; Cazzolla Gatti et al., 2022; Keenan and Williams, 2018). These roles have been substantially challenged and complicated by frequent and intense global changes such as weather extremes, wildfires, and changes in societal demand (Bonan, 2016). Therefore, forests have been receiving increased scientific attention to understand the complex ecological processes and their interactions that drive this system (Bennett et al., 2009). However, achieving such an understanding of a complex system in a precise manner requires both high-quality spatial-temporal data and adequate methods to analyse it and enable forestry stakeholders to respond accordingly to current and predicted scenarios. In particular, a variety of measurements and analyses are needed to enable a data-intensive enquiry into the potential of forests to act as a bio-based solution for global climate change (Seddon, 2022).

Recent rapid technological advancement has greatly improved the ability to capture data with high temporal and spatial resolution. The larger datasets captured with such new technology provide unique opportunities to study complex forestry systems (Baldocchi et al., 2001; Bayne et al., 2017; Hock et al., 2016). For instance, automatically measured Stem Radius (SR) data can be considered real-time big data, and it is well established that it can be turned into environmentally

* Corresponding author at: Te Ipu o Te Mahara AI Institute, Department of Computer Science – Tari Rorohiko, Private Bag 3105, Hamilton 3240, New Zealand.
E-mail addresses: guilherme.cassales@waikato.ac.nz (G. Cassales), serajis.salekin@scionresearch.com (S. Salekin), nick.lim@waikato.ac.nz (N. Lim), Dean.Meason@scionresearch.com (D. Meason), albert.bifet@waikato.ac.nz (A. Bifet), bernhard.pfahringer@waikato.ac.nz (B. Pfahringer), eibe.frank@waikato.ac.nz (E. Frank).

https://doi.org/10.1016/j.ecoinf.2025.103014
Received 5 May 2024; Received in revised form 9 January 2025; Accepted 11 January 2025
Available online 13 January 2025

related biological information (Zweifel et al., 2016, 2020, 2021). Nevertheless, any automatically collected continuous big data comes with major challenges, namely: 1) cleaning the raw datasets to remove missing, corrupt, and ambiguous values; 2) processing and separating the data into meaningful scientific variations (Luković et al., 2022); and 3) analysing it promptly to provide actionable feedback for stakeholders in a timely manner. Moreover, data of this nature demands significant financial and personnel resources for proper upkeep. These challenges intensify and have a greater impact over a longer time. Thus, it is crucial to select and implement a suitable method at the initial phase of experimental execution to ensure the optimal utilisation of such big data.

Along with the availability of big data, advancements in hardware and software technology have sparked a revolution in nearly all scientific fields, with the integration of artificial intelligence (AI) methods into all kinds of tasks (Jordan and Mitchell, 2015). Environmental research specific to forest science is no different and has embraced the latest Machine Learning (ML) and Deep Learning (DL) algorithms. However, despite the rising popularity, the use of ML algorithms in different domains is continuously going through the debate on its applicability toward distinct aspects and still requires the creation of a framework that enables sound evaluation of alternatives according to the problem at hand (Pichler and Hartig, 2023).

Traditional methods for time series (TS) forecasting, such as Autoregressive Integrated Moving Average (ARIMA) (Dimri et al., 2020; Kaur et al., 2023), Seasonal ARIMA (SARIMA) (Dimri et al., 2020), and classical machine learning approaches like Support Vector Machines (SVM) and Random Forests (RF) (Camastra et al., 2022; Cervantes et al., 2020; Jamali, 2019; Salekin et al., 2021), have been widely employed for forecasting environmental data. However, these methods present limitations in dealing with the complex and non-linear patterns (Hyndman and Athanasopoulos, 2021) typical of dendrometer data, which often exhibit seasonal and daily trends, as well as long-term dependencies.

While some of ARIMA's seasonal variants are capable of capturing multiple seasonal patterns with additional support methods, they are time-consuming and may require extensive feature engineering to achieve good results using the available tools (Hyndman and Athanasopoulos, 2021; Zhang, 2003). In addition, these models rely on large amounts of data to correctly extract the seasonal patterns (Shumway et al., 2000), which is often rare in the case of natural resources management and optimisation discipline. Moreover, ARIMA's seasonal variations do not produce reliable outcomes when the data acquisition site differs; therefore, adjusting each location separately is time-consuming. Similarly, applying classical machine learning methods like SVMs and Random Forests to environmental data (Camastra et al., 2022; Cervantes et al., 2020; da Rocha et al., 2024; Jamali, 2019; Jemeļjanova et al., 2024; Lin et al., 2023; Salekin et al., 2021; Tang and Li, 2023) requires substantial feature engineering and human intervention, leading to a large amount of computational time to find a good solution. Yet, these models often fail to capture the temporal dependencies of the TS data effectively.

As an alternative, Convolutional Neural Networks (CNNs) offer a powerful technique to extract higher-level features from TS data through convolutions. Moreover, they are particularly useful in capturing local patterns such as short-term dependencies and abrupt changes (Borovykh et al., 2018). However, convolutions may not be as effective in capturing long-term dependencies compared to more advanced models (Bai et al., 2018).

While CNN has shown good performance in many problems featuring spatial and/or temporal data, more advanced models have been adopted as the solution for such problems. For instance, the Long Short-Term Memory (LSTM) architecture (Hochreiter and Schmidhuber, 1997) can model longer-term dependencies that might be missed by CNNs due to the architecture design to model sequential and temporal relationships. Specifically, LSTMs preserve information over long sequences using the cell state. More recently, the Transformers architecture (Vaswani et al., 2017) has benefitted from an even bigger "memory" in the form of the attention mechanism, which finds relationships based on relevance by computing dependencies between all input elements. This approach avoids the sequential bottleneck of earlier architectures.

Many applications in Natural Language Processing and Computer Vision have benefitted substantially from these techniques. Regarding TS, the Exponential Smoothing Transformers for Time-series Forecasting (ETSFormer) (Woo et al., 2022) have emerged as a noteworthy exemplar of Transformer specialisation by adeptly combining established statistical methodologies with the Transformer architecture. The results presented in the work introducing ETSFormer showcase better forecasting capabilities on diverse domains such as electricity, exchange market, road traffic, weather, and illnesses. All four architectures, CNNs, LSTMs, Transformers, and ETSFormers, are DL approaches that can fill the gaps from the classic auto-regressive and ML models. For instance, once trained, these models may be able to better generalize at different locations, and forecasting could be faster than auto-regressive methods. Moreover, CNNs, LSTMs, and Transformers are widely used and present good performance in other TS forecasting problems (Bai et al., 2018; Borovykh et al., 2018; Che et al., 2018; Gopali et al., 2021; Hewamalage et al., 2021; Jemeļjanova et al., 2024; Khosravi et al., 2025; Pedro et al., 2021; Pölz et al., 2024). At the same time, ETSFormer is a promising alternative since it combines Exponential Smoothing methods with the Transformer architecture, making it particularly suitable to model both seasonal and trend components in TS data (Woo et al., 2022).

The main aim of this study is to investigate the application and performance of the four DL algorithms mentioned above for forecasting SR. To the best of our knowledge, this is the first study to analyse the transferability of DL models for SR forecasting into slightly different locations through an exploratory experiment without having access to topographic and climate data. Moreover, this study pioneers the analysis of the impact of different temporal resolutions on the forecasting performance of SR using these DL models.

Understanding the influence of temporal resolution is crucial for optimising monitoring strategies, as it allows forest managers and researchers to determine the most appropriate data collection intervals that balance accuracy with cost efficiency. By identifying the optimal temporal resolution, current and future infrastructures for monitoring plantation forests can be designed to operate more effectively, reducing unnecessary data collection and associated expenses while maintaining high forecasting precision. Moreover, while capturing the intra-diurnal behaviour of trees provides valuable insights into short-term physiological responses, understanding long-term dependencies is more critical in the context of plantation forestry, where the period from planting to harvesting can extend over several decades.

Our approach allows for a comprehensive analysis of how different DL models handle both short-term variations and long-term trends, shedding light on their respective strengths and limitations. This contributes to a deeper understanding of model performance under varying conditions and offers guidance for future research on model configuration, parameter selection, and the broader implications of these choices for forest management and monitoring practices.

2. Materials and methods

2.1. Experimental site description, establishment and data acquisition

The plantation forest catchment used in this study is shown in Fig. 1, it spans over 38 ha converted from pasture to forest. The catchment is on its second forest rotation of *Pinus radiata* (D. Don), with approximately 88 % of the total planted in 1999 and the remaining 12 % planted in 2006. The area experiences an annual precipitation of 800 mm and an annual mean daily temperature of 12.2 °C. Geologically, the catchment sits on greywacke-cast conglomerate terrain, with elevations ranging

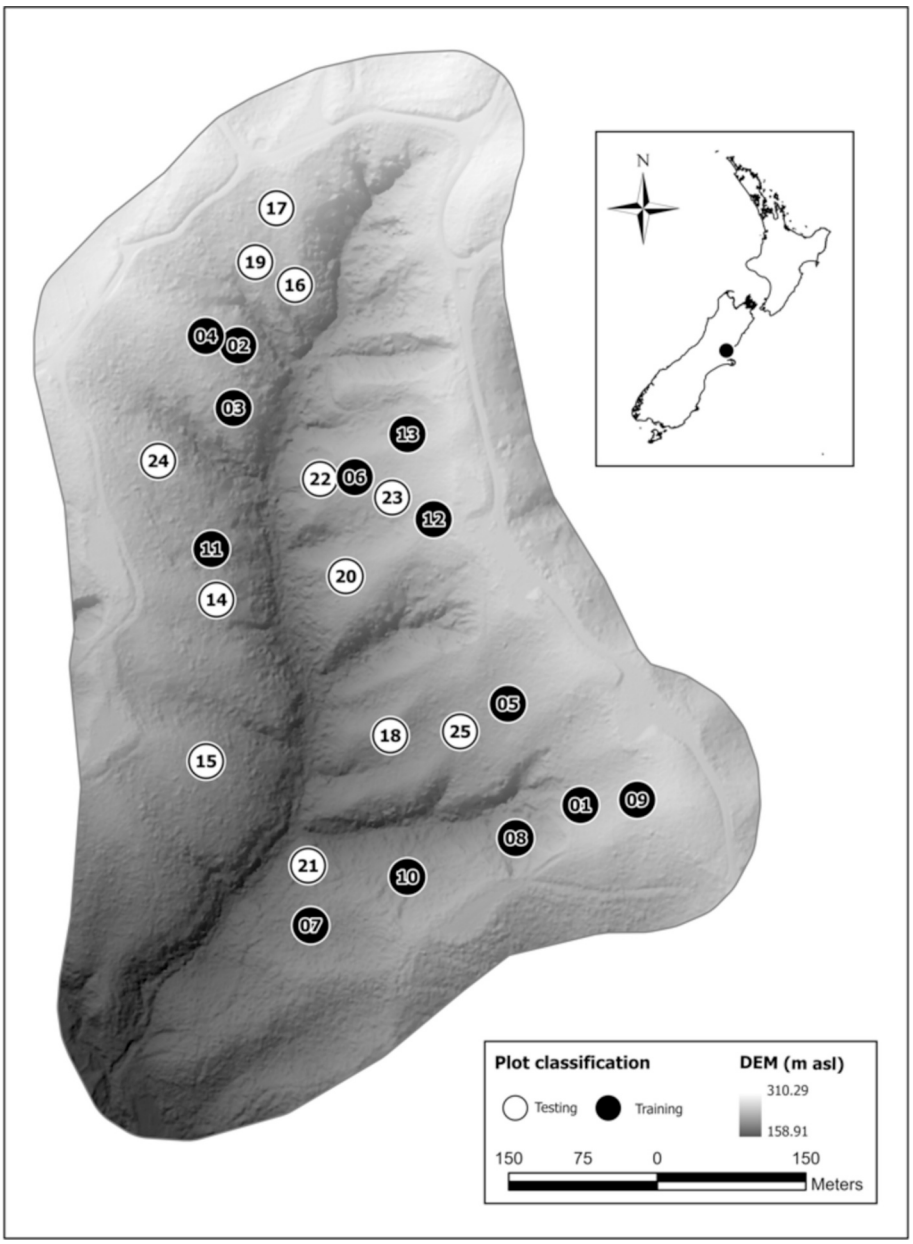


Fig. 1. Study site and plots (train and testing) locations (Salekin et al., 2021).

from 170 to 300 m above sea level.

Site-specific digital elevation model surface of 1 m resolution was sourced from the forest company. A total of 25 plots were established measuring 20 m × 20 m across the catchment by following the methodology presented in (Salekin et al., 2021). In broad terms, the authors created three hundred random points using DEM surface. These points were strategically spaced at least 20 m apart and excluded any point within a 15 m radius of the catchment boundary or anthropogenic structures like forest roads. Topographic metrics are then derived from the DEM at each point, and a cluster analysis is conducted to group the random points. Ultimately, 25 distinct groups of points are identified to determine plot locations, based on the assumption that each catchment inherently comprises 25 characteristic locations.

At each one of the twenty-five experimental plots, four trees with long slender bole and healthy canopies were equipped with point dendrometers, each comprising a potentiometer with a retractable plunger. A button situated at the plunger's end evenly distributes pressure to minimise damage to the tree stem, thus preventing any potential wound response. Potentiometers function by measuring resistance (in microvolts, μV); a fully extended plunger results in maximum resistance, while a fully retracted plunger exhibits no resistance. As the tree stem undergoes growth and contraction, these movements alter the plunger's position on the potentiometer, subsequently modifying the voltage output. These voltage fluctuations are then converted into micrometres (μm) of movement. Moreover, the potentiometer is connected to a chip that stores manually uploaded calibration data necessary for converting μV to μm. Each sensor records readings of stem radius fluctuations (SR) at 5-min intervals, with the readings from the time series used in this work representing the cumulative SR expansion in millimetres (mm) since the installation of the sensor. While the dendrometers' installation data may differ slightly, the earliest data used in this study was collected on the 16th of August 2021, and the latest data used is from the 20th of July 2022.

The dendrometers transmit their data wirelessly to a dedicated processing and storage datalogger node (FlowLab, Infact, Christchurch, New Zealand). Each plot is equipped with a single node, responsible for

relaying data to a gateway node connected to both the cellular network and the internet. At the site, only one gateway node exists, tasked with forwarding data to an Internet of Things (IoT) Platform, which subsequently distributes the data to traditional database systems through Apache Kafka. Data is published into the Kafka topics once daily, with each sensor type (e.g., dendrometer).

2.2. Data mesh and apache Kafka

The Data Mesh presents an innovative approach to organising data across distributed systems, aiming to improve data management by breaking down the data space into domains (Zhamak Dehghani, 2022). Each domain is responsible for managing data related to specific tasks, promoting more agile development through the data-as-a-product paradigm, which encourages the sharing of pre-processed data among domains. The system used in the experiments was built on top of Apache Kafka, a streaming platform widely used in industry and academia. Apache Kafka serves as an open-source distributed event streaming platform designed for efficiently handling data pipelines, streaming analytics, data integration, and essential applications (Apache Kafka, 2023). It operates through multiple servers known as *brokers*, which manage the storage and transmission of data by partitioning it into *topics*. These topics contain messages produced by data sources, which are distributed to subscribed users. This setup allows high-speed and low-delay data processing, making it ideal for real-time applications. Notably, Kafka ensures data integrity by safely storing data across distributed systems, making it a suitable choice for various data science applications.

Fig. 2 depicts the workflow of the Kafka system utilised in the application discussed in this paper. The cylinders indicate topic names, which serve as Kafka abstraction to categorise stored data into distinct "buckets." Ellipses symbolise the numerous sensors deployed in the field to collect data. Round rectangles denote special nodes responsible for transmitting data from the sensors to the backbone. The large rectangle representing the Kafka system outlines the extent of the data it manages. Diamonds represent various processes running within the system, with directed connections indicating the flow of data as input and output for each process.

2.3. Data pre-processing

This study uses the dendrometer readings in the form of a univariate

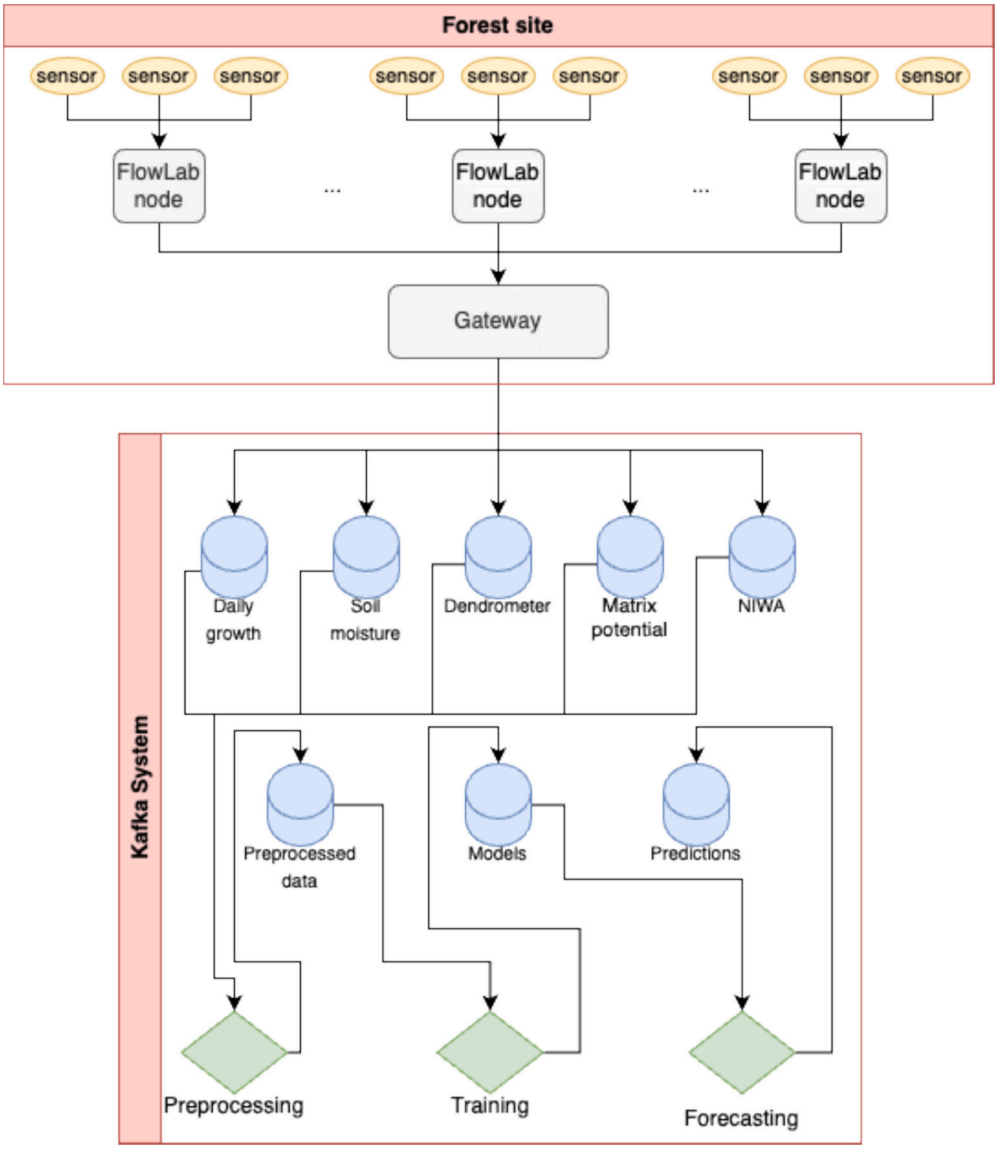


Fig. 2. Data workflow for the system.

time series where each dendrometer corresponds to an independent time series. Formally, a univariate time series consists of a single variable observation at each time (Hyndman and Athanasopoulos, 2021).

The basic pre-processing method for general DL methods uses the sliding window notion, where multiple time steps are grouped to form an input vector and an output is added to this vector. This input vector formed from multiple time steps is also called a lookback (LB) sequence, and this will be the adopted name throughout this manuscript. Although some models can handle missing values and there are methods specifically designed to cope with this challenge, it is recommended to avoid missing values and large gaps when utilising these models(Che et al., 2018; García-Laencina et al., 2010). In this context, ensuring that each sensor is pre-processed in a way to avoid such issues is crucial as they can hinder the model's performance.

In summary, input data consists of a sequence of data points within a set time threshold, while the output data, also called ground truth (gt), consists of the data point 24 h in the future. Next, the steps to reach this data format are described in detail.

2.3.1. Data curation

The first process applied to the data is the dendrometer readings sanitisation, which includes tasks such as removing outliers from the time series and adjusting the values to account for sensor limitations. In essence, the dendrometers' potentiometer has a limit to how much it can contract, and once the limit is reached, the readings will not reflect the real state of the tree. To avoid this, the sensors are reset when close to this limit to allow continuation of the monitoring. However, this process causes the time series to use smaller values, which need to be adjusted by adding the value before the reset to all readings starting from that point to reflect the cumulative SR expansion. This stage is agnostic of the ML algorithm used later to build the model, as its goal is primarily to make the data biologically consistent.

2.3.2. Building time series data

Deploying large monitoring infrastructures in places that are not easily accessible may raise challenges, such as gaps in the data due to sensor and communication failures. To address this, data must go through an interpolation process to fill in any missing values before creating the required input-output pairs. Filling data gaps poses a significant challenge, with various approaches discussed in the literature, as reviewed by Lepot et al. (2017).

While it is possible to address missing data in dendrometers by interpolating the full sensor time series at the 5-min resolution (i.e., a time series with 288 data points per day), it appears more appropriate to interpolate at the level of individual timestamps (i.e., creating 288 daily time series from the same dendrometer at different times of the day). This approach is particularly effective when dealing with dendrometer data due to trees' sinusoidal growth pattern following the day-night cycle. Consequently, each dendrometer time series is divided into 288 distinct time series, each representing readings collected at specific timestamps of the day (e.g., 01:00 pm and 01:05 pm). This approach serves two purposes: (i) it facilitates easier interpolation of a time series by simplifying the overall sinusoidal trend from the dendrometer into a linear trend; and (ii) it minimises data gaps, as a full day of missing data (288 consecutive records) becomes a single missing record in each of the 288 daily time series. After interpolating the gaps, all the 288 time series are merged and ordered by timestamp to form a contiguous sequence for each sensor, as shown in Fig. 3 where the original and the interpolated time series can be seen. The time series pre-processing removes the outliers and interpolates the gaps.

Subsequently, each time series undergoes pre-processing to create the lookback sequences based on the chosen lookback period, which is facilitated by the DL libraries, and the intended output (i.e., data point 24 h in the future) is also added to the vector. The challenge at this stage stems from the experimental design, where data from multiple time series are utilised to build the training dataset. Therefore, it is essential to individually create the lookback sequences for each time series before merging them into one comprehensive training dataset.

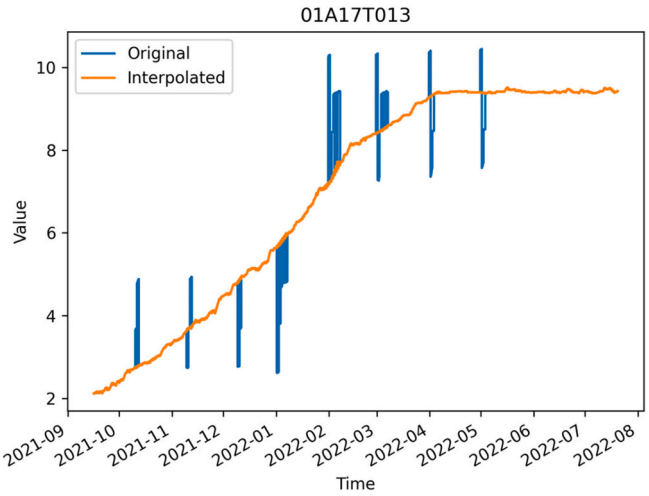


Fig. 3. The original and the interpolated time series for a sensor (01A17T013).

The average length of each one of the 100 interpolated time series is 78,860.39, with a standard deviation of 12,933.94. Also, there are 7.8 M data points across all the time series.

2.4. Algorithms

Neural networks (NN), the most prominent class of models for DL, have been widely applied in for modelling and predicting complex system dynamics, pattern recognition, insight extraction, and making precise predictions from large datasets (Lindemann et al., 2021).

While NNs are known for their high computational demands, requiring significant resources and lengthy training processes, recent advancements in hardware, especially the development of high-performance Graphical Processing Units (GPUs), have considerably reduced processing times. This development, along with the availability of large datasets, has profoundly impacted the fields of AI and ML. A large number of different NN architectures can be found in the literature, each employing unique techniques to capture specific behaviours. In this paper, we focus on three such architectures, which will be described subsequently.

2.4.1. Convolutional neural network

CNNs are widely employed in time series forecasting problems (Bai et al., 2018; Borovykh et al., 2018; Hyndman and Athanasopoulos, 2021) due to their relatively low computational demands and good performance. Fig. 4 presents a sample diagram of a CNN with a single Convolutional layer, a MaxPooling layer (in red), a Flatten layer followed by two Dense layers to form the single point output. The idea in this architecture is that the Convolutional layers will extract features.

2.4.2. Long short-term memory

The Long Short-Term Memory (LSTM) network, a variant of Recurrent Neural Networks (RNNs), was developed to address the limitations of traditional RNNs, particularly in handling long-term dependencies. Initially proposed by Hochreiter and Schmidhuber (1997), this architecture incorporates a forget gate mechanism to discard unnecessary information, aiding in the neural network's convergence and learning process. LSTMs have risen to prominence because of their impressive performance when capturing and analysing temporal dependencies in sequential tasks, including time series forecasting and language modelling. This is primarily due to their design being focused on enabling them to deal with long-range dependencies and preserving information over lengthy intervals. This makes them particularly

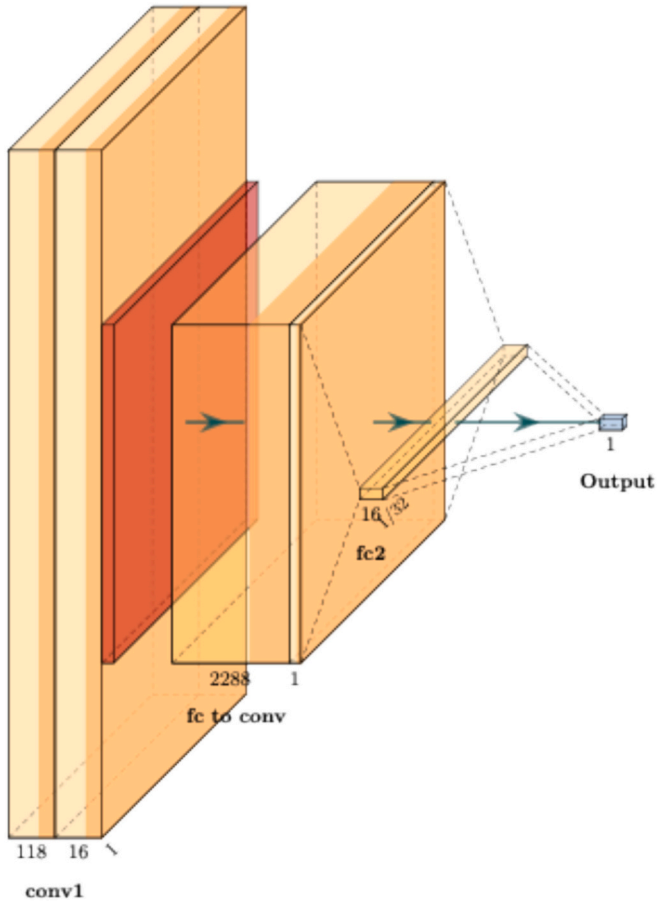


Fig. 4. A diagram representing a CNN architecture drawn using PlotNeuralNet[a] tool.
[a]https://github.com/HarisIqbal88/PlotNeuralNet.

promising candidates for modelling the intricate and nonlinear temporal patterns found in forestry data, such as tree growth patterns and forest ecosystem dynamics.

Fig. 5 shows the LSTM cell architecture. In addition, LSTM's mathematical formulations are presented in Eqs. (1) to (5) using an arbitrary activation function denoted by σ. The subscripts represented by t, $f, i, o,$ *and* c represent the timestamp, forget gate, input gate, output gate, and cell state, respectively. The symbols W, U *and* b represent weight vectors.

$$f_t = \sigma_g(W_f x_t + U_f h_{t-1} + b_f) \tag{1}$$

$$i_t = \sigma_g(W_i x_t + U_i h_{t-1} + b_i) \tag{2}$$

$$o_t = \sigma_g(W_o x_t + U_o h_{t-1} + b_o) \tag{3}$$

$$\tilde{c_t} = \sigma_c(W_c x_t + U_c h_{t-1} + b_c) \tag{4}$$

$$h_t = o_t \odot \sigma_h(c_t) \tag{5}$$

The Forget gate employs Eq. (1) to decide what information needs to be removed from the cell state using the previous timestamp hidden state (h_{t-1}, represented by block A on the left) and the current input through the σ activation function. The Input gate employs Eq. (2) and uses current input and the previous timestamp hidden state (h_{t-1}) to update the cell state. Eq. (3) represents the output gate, used to provide the hidden state for the next timestamp (h_{t+1}, represented by block A on the right). The Candidate Cell State Eq. (4) calculates a candidate value for the Cell State at timestamp t. Lastly, Eq. (5) shows the final output formula (h_t), used to calculate the output of the current unit that should be used as both the prediction and the input of the hidden state at the next timestamp (h_{t+1}).

2.4.3. *Transformer*

The Transformer is an NN architecture that has revolutionised natural language processing and other sequence-to-sequence tasks with its attention mechanism. By encoding temporal information, the Transformer's attention network enables it to selectively focus on relationships within the data, effectively capturing long-range dependencies. Introduced by (Vaswani et al., 2017), the Transformer architecture employs a self-attention mechanism that can simultaneously attend to different positions in the input sequence, thereby effectively weighing the importance of different elements for predictions. Due to their complex structure, Transformers are typically trained using learning rate schedulers with a warmup phase, where the learning rate increases each epoch, followed by a decay phase, where the learning rate starts to decay after reaching a peak. This approach accelerates the initial training

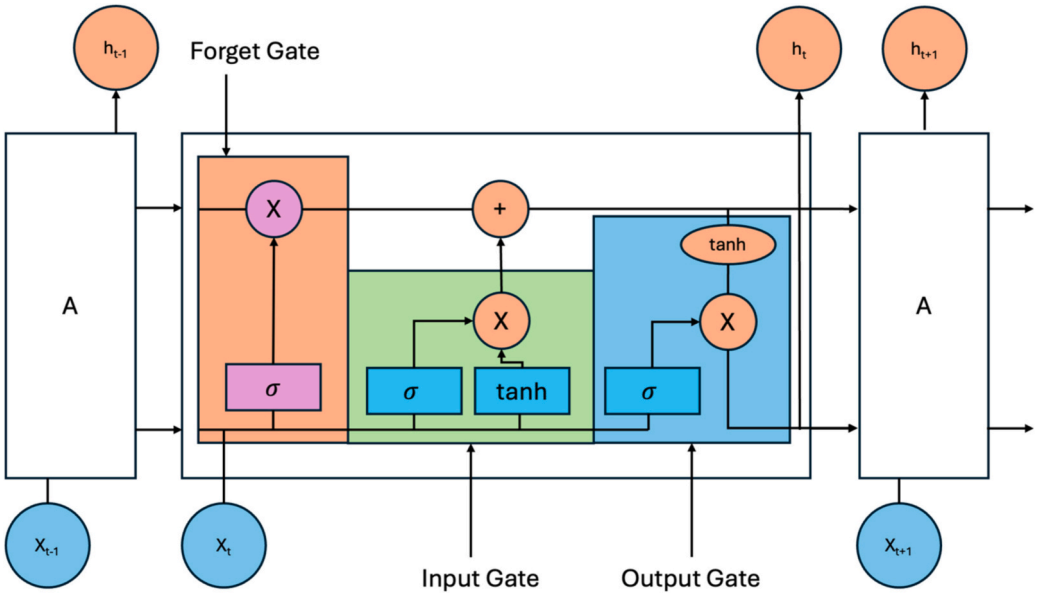


Fig. 5. A diagram of the LSTM cell architecture adapted from (Steve Jerome Lawrence, 2023).

phase to quickly find a promising region, then slows down to fine-tune the solution within that region. Fig. 6 shows the Transformer's general architecture.

2.4.4. ETSformer

ETSformer: Exponential Smoothing Transformers for Time Series Forecasting (ETSformer) is a specialised adaptation of the transformer architecture tailored for time series forecasting. The unique feature of ETSformer lies in its utilisation of the exponential smoothing principle. By incorporating novel exponential smoothing attention (ESA) and frequency attention (FA) mechanisms, ETSformer replaces the traditional self-attention mechanism in standard Transformers, resulting in improved accuracy and efficiency in time series forecasting by furnishing them with an enhanced capability to leverage the periodic and seasonal patterns often found in time series data (Woo et al., 2022).

3. Experimental set up

All experiments were performed using an NVIDIA GeForce GTX 1080Ti with 11GB, allocated to a node with an i7-6700K CPU at 4.00GHz and 62GB RAM. The results present the Mean Average Error (MAE) and the Mean Squared Error (MSE), along with statistical measures of the results aggregation. The MAE measure indicates the average prediction error in absolute terms (real value), while MSE represents the average squared error (Hodson, 2022).

3.1. Experiment 1 - evaluation of the four methods with 5-min resolution

In the first experiment, three DL methods were evaluated in the task of forecasting dendrometer-measured SRs for individual sensors 24 h ahead under different parameter configurations. The 25 plots were split into two groups. Group 1 comprised sensors from plots 1 through 13 and was used for training the models, while Group 2 comprised sensors from plots 14 through 25 and was used for testing. This arguably small training split was chosen for a few reasons, such as to assess the generalisation capacity of the trained models, the risk of overfitting, and data availability. Generalisation capability is important because one of the aims of this study is to understand how well a model can forecast data from plots with different topographic characteristics while not having access to such information. The preliminary results using the "simple" CNN baseline indicate the risk of overfitting because the model achieves a stable loss in a few epochs and does not improve after this as seen in Fig. 7. Moreover, this behaviour happens in most methods, indicating the risk of undertraining is small. Finally, even with only 52 sensors, the data has 3.4 M samples available with a 5-min resolution. The utilised models are described below.

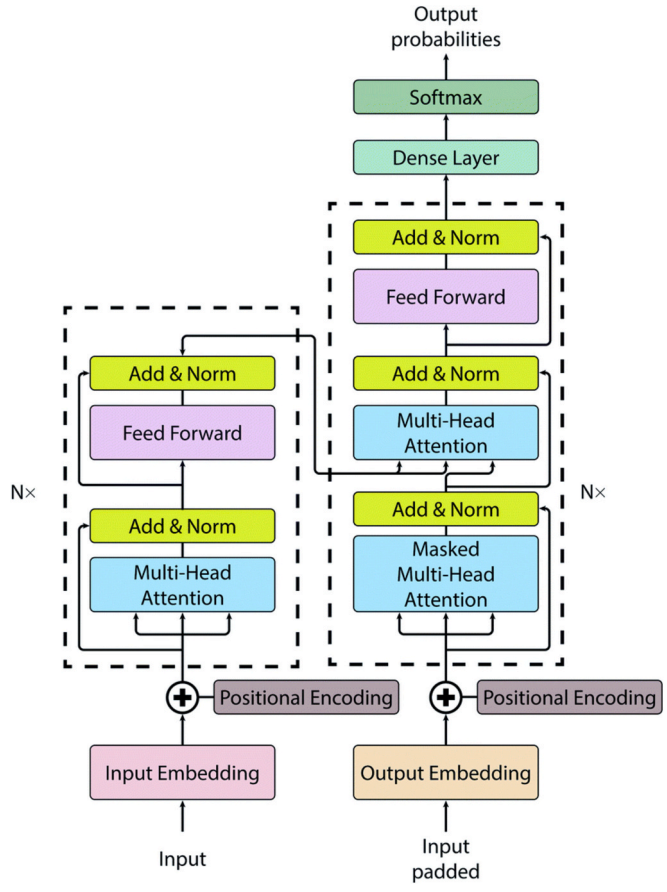


Fig. 6. A diagram of the vanilla Transformer architecture (Zhumagambetov et al., 2021).

Two CNN architectures were included in this study, their layer composition is described in Tables 1 and 2. Architecture SingleConvL (a single pair of convolutional 1D and MaxPooling 1D) has 15,201 total parameters, while Architecture DoubleConvL (two pairs of convolutional 1D and MaxPooling 1D) has 8049 total parameters. Both architectures were trained with a combination of the parameters: a) Lookback hours – 10, 24, and 48 h; b) Learning Rate – 0.001 and 0.0001; and c) Configuration – SingleConvL and DoubleConvL (output shapes differ based on lookback).

In the LSTM models, we varied the parameters units, lookback hours (LB), and learning rate (LR) according to Table 3. Every LSTM model consists of a single layer. The units parameter specifies how many LSTM cells the model has in parallel, determining the complexity of relationships that can be learned. The LB parameter indicates how much recent history information the model is using to output a prediction in the future. Longer lookback sequences provide more information to the model in the form of an increased receptive field. The receptive field refers to the specific region of the input data that a particular neuron in the network can "see" or respond to. Finally, the learning rate determines the step size during the optimisation process, with each step aimed at bringing the model closer to a local minimum in parameter space.

In the Transformer models, we varied the parameters feedforward dimensions (*ff_dim*), LB, and LR according to Table 4. The *ff_dim* parameter serves the role of enhancing the model's capacity to capture complex patterns and dependencies in the data; however, increasing it too much may reduce the generalisation ability of the model and increase computational resource usage (i.e., memory and time) as it will have more filters to compute and keep in memory. As stated in Subsection Transformers, LR schedulers are commonly applied during training. We evaluate two schedules, which we call 'Small' and 'Large,' respectively. These schedulers have the warm-up phase with a linear increase from the initial LR to the maximum LR, followed by a cosine decay from the maximum LR to the final LR. The Large scheduler configuration corresponds to an initial LR of 10–6, a maximum LR of 10–4, and a final LR of 10–5, while the Small configuration corresponds to an initial LR of 10^{-7}, a maximum LR of 10^{-5}, and a final LR of 10^{-6}.

Since ETSFormer builds on the standard Transformer, the results obtained using the above configurations for the Transformer were utilised to reduce the number of configurations tested for ETSFormer. We copy the best-performing Transformer configuration and vary only the LR, as shown in Table 5.

3.2. Experiment 2 - temporal resolution analysis

In the second experiment, the configurations with the most promising results from Experiment 1 were used to assess the impact of reducing the temporal resolution (i.e., increasing the collection interval) of the sensors. The dendrometers in the infrastructure collect and transmit readings to a station in 5-min intervals, which can raise several

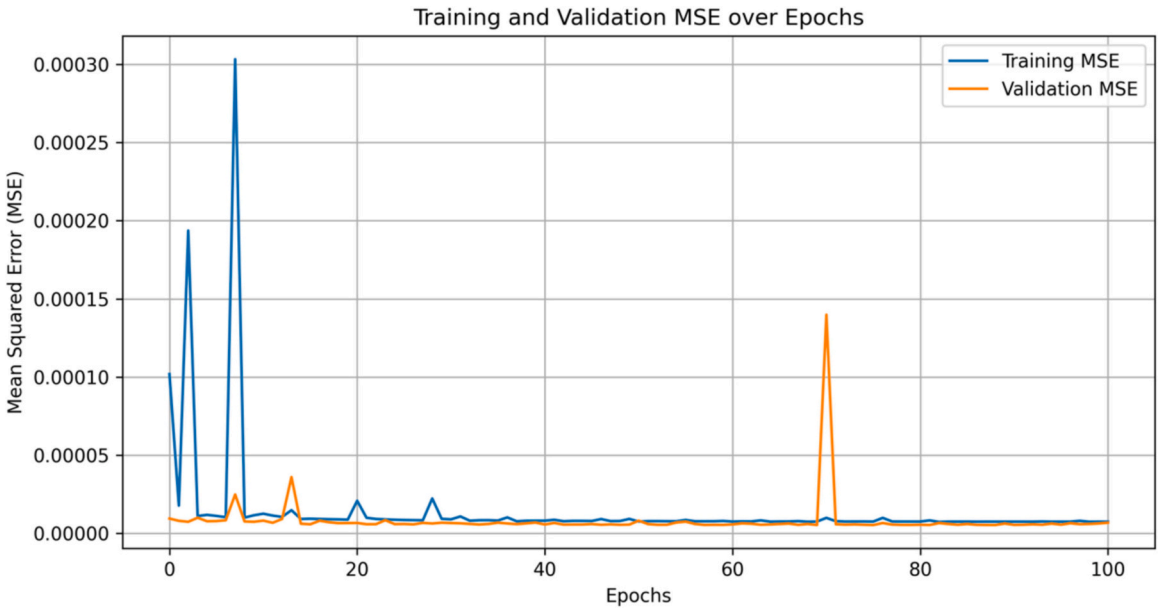


Fig. 7. Training and Validation MSE over epochs for the SingleConvL CNN model with a 10-h lookback.

Table 1
Sample CNN architecture SingleConvL for 10-h lookback.

Layer (type)	Output Shape	Param #
Convolutional 1D	(None, 118, 16)	64
Max Pooling 1D	(None, 59, 16)	0
Flatten	(None, 944)	0
Dense	(None, 16)	15,120
Dense	(None, 1)	17

Table 2
Sample CNN architecture DoubleConvL for 10-h lookback.

Layer (type)	Output Shape	Param #
Convolutional 1D	(None, 118, 16)	64
Max Pooling 1D	(None, 59, 16)	0
Convolutional 1D	(None, 57, 16)	784
Max Pooling 1D	(None, 28, 16)	0
Flatten	(None, 448)	0
Dense	(None, 16)	7,184
Dense	(None, 1)	17

Table 3
LSTM configurations.

Configuration	Units	LB	LR
LSTM 1	25	10	0.01
LSTM 2	25	10	0.001
LSTM 3	25	24	0.01
LSTM 4	25	24	0.001
LSTM 5	50	10	0.01
LSTM 6	50	10	0.001
LSTM 7	50	24	0.01
LSTM 8	50	24	0.001
LSTM 9	100	10	0.01
LSTM 10	100	10	0.001
LSTM 11	100	24	0.01
LSTM 12	100	24	0.001

issues such as faster battery drainage, replicated data, and a lack of failure tolerance in case of communication failures. If the same forecast quality could be obtained using smaller temporal resolution, dendrometer batteries would last longer, manual intervention frequency would be reduced, there would be less replicated data, and data would be less likely to be discarded in the event of communication failures. In summary, costs would be reduced. Thus, data was processed to represent intervals of 30 min, 60 min, 3 h, 6 h, and 12 h. Training and testing were performed in the same conditions as experiment 1.

Table 4
Transformer configurations.

Configuration	ff_dim	Lookback hours	LR scheduler
Transformer 1	8	10	Large
Transformer 2	8	10	Small
Transformer 3	8	24	Large
Transformer 4	8	24	Small
Transformer 5	16	10	Large
Transformer 6	16	10	Small
Transformer 7	16	24	Large
Transformer 8	16	24	Small
Transformer 9	32	10	Large
Transformer 10	32	10	Small
Transformer 11	32	24	Large
Transformer 12	32	24	Small

Table 5
ETSFormer configurations.

Configuration	Ff_dim	Lookback hours	LR scheduler
ETSFormer 1	8	10	Large
ETSFormer 2	8	10	Small

4. Results

4.1. Experiment 1 - comparison of three methods

Each model configuration is evaluated by forecasting 48 individual time series. Results for CNN, LSTM, Transformers, and ETSFormer are presented in Tables 6, 7, 8, and 9, respectively. The tables display the mean and standard deviation for the MAE and MSE metrics extracted from the 48 test time series. Additionally, we show the top 5 results and the worst performing configuration for CNN, LSTM and Transformer, while both results for ETSFormer.

In addition, scatter plots for the best version of the models can be

Table 6
The top 5 and the worst CNNs according to MAE for a 5-min temporal resolution.

Architecture	LR	LB	MAEmean	MAEstd	MSEmean	MSEstd
DoubleConvL	0.001	10	0.027743	0.006653	0.000089	0.000043
DoubleConvL	0.0001	24	0.029133	0.006024	0.000096	0.000042
SingleConvL	0.001	24	0.031778	0.004375	0.000107	0.000037
SingleConvL	0.001	10	0.034277	0.005224	0.000123	0.000043
SingleConvL	0.0001	48	0.034471	0.014244	0.000139	0.000109
SingleConvL	0.0001	24	0.036289	0.013808	0.000150	0.000106

Table 7
The top 5 and the worst LSTMs according to MAE for a 5-min temporal resolution.

Units	LR	LB	MAEmean	MAEstd	MSEmean	MSEstd
100	0.001	10	0.02663	0.00555	0.000081	0.000038
25	0.001	24	0.02666	0.00620	0.000083	0.000042
25	0.01	10	0.02678	0.00525	0.000081	0.000037
25	0.001	10	0.02700	0.00545	0.000083	0.000038
100	0.001	24	0.02709	0.00588	0.000084	0.000040
100	0.01	24	0.02925	0.00694	0.000095	0.000044

Table 8
The top 5 and the worst Transformers according to MAE for a 5-min temporal resolution.

FF_Dim	LR	LB	MAEmean	MAEstd	MSEmean	MSEstd
8	Large	10	0.033187	0.021621	0.000159	0.000321
32	Large	10	0.034859	0.023479	0.000176	0.000344
16	Small	24	0.046663	0.035619	0.000327	0.000689
8	Small	24	0.046789	0.034887	0.000319	0.000638
16	Large	10	0.047282	0.033305	0.000305	0.000565
8	Large	24	0.065495	0.049804	0.000612	0.001295

Table 9
The results for ETSFormers according to MAE for a 5-min temporal resolution.

LR	LB	MAEmean	MAEstd	MSEmean	MSEstd
Large	10	0.02606	0.005627	0.001262	0.000404
Small	10	0.06224	0.016772	0.004965	0.002069

seen in Fig. 8, where the points are represented in blue, with a red line fitting all validation set points.

A general trend is observed in the four methods, indicating that a lookback sequence of 10 h is sufficient when using the 5-min temporal resolution. This behaviour correlates with the sinusoidal nature of dendrometer SR. With a 10-h lookback, the model has enough data to estimate the trend in the sequence and accurately forecast the target value. Conversely, using a 24-h lookback adds more input information (i.e., parameters) to the model in the form of a longer sequence. Based on the results, a longer LB requires a smaller LR to successfully model the behaviour, which leads to a slower conversion (requiring more epochs). Additionally, all methods with variation in the LB present the best result with a 10-h LB.

In CNNs, it is possible to see a sharp drop in performance from the best to the fourth-best model, in which the difference is only the architecture. This indicates that the Double Convolution layer architecture can better model the time series with higher-level features.

In addition, while the worst-performing model is not the one with 48-h LB, the best-performing 48-h LB model presents a big increase in error compared to the best-performing models for 10 and 24 LB. The bad overall performance of 48-h LB models coupled with the larger number of parameters (in CNNs the number of parameters varies according to the LB sequence), indicate that the increased LB may be causing the model to overfit training data.

For LSTM models, the differences across all the models are much smaller, suggesting that the architecture is successful in modelling both the daily trend and the seasonal trend successfully independently of the parameter configuration. There is no clear pattern except that when using more units, a smaller LR is required to avoid convergence problems due to the increased number of parameters.

The presented Transformer results suggest either a small or a large ff_dim parameter, instead of the middle ground 16, yields better forecasts. Specifically, the Transformer model with 10^{-6} initial LR, a 10-h LB, and a *ff_dim* of 8 exhibits the best overall performance, followed closely by the model with Large LR, 10-h LB and 32 ff_dim. After these two models, there is a sharp drop in performance. We hypothesise that reducing the number of intermediate parameters from 16 to 8 diminishes the risk of overfitting spurious relationships in the training data, and increasing it to 32 may create enough depth in the model to be able to capture general patterns instead of overfitting to the training data. Additionally, the smaller LR seems to work better with the longer LB, confirming the previous model's results.

In the case of ETSFormer, the configuration with the Large LR

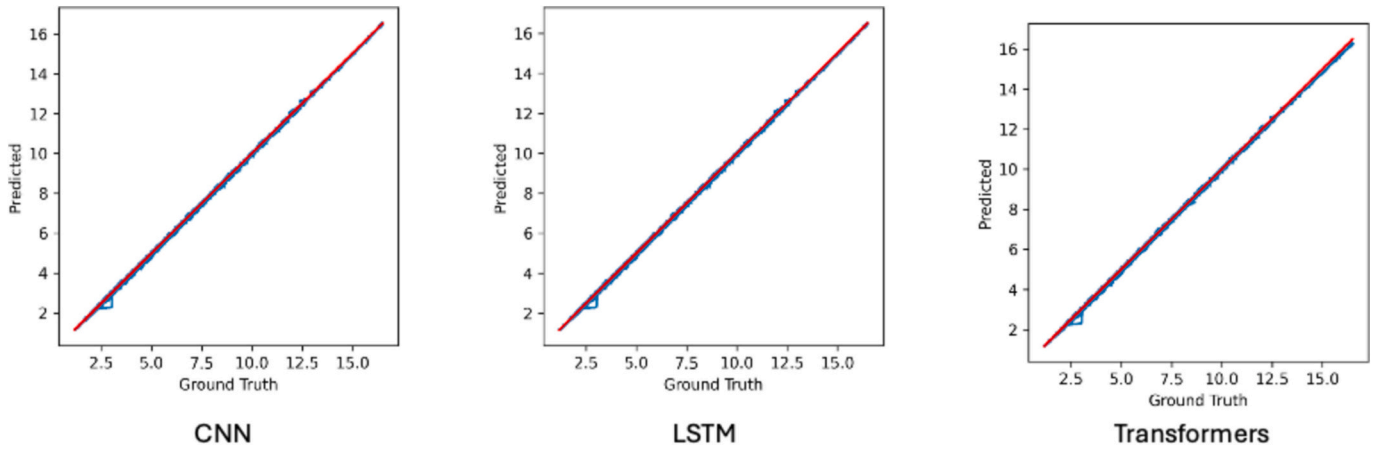


Fig. 8. Scatter plot and linear fit for the best version of each algorithm using all validation points.

yielded the best results. The optimal ETSFormer model follows the same trend showed with Transformers where the Large LR schedule presents superior MAE performance with a 10-h LB. The performance achieved by such a model improves the Transformer performance by incorporating the classical statistical methods, which facilitate time series decomposition and enhance the model.

In general, ETSFormer achieved the best MAE results (0.02606), followed by LSTM (0.02663), CNNs (0.02774), and Transformers (0.03318) with a 5-min temporal resolution. These results suggest that the Transformer model struggles to capture the temporal relationship using only dendrometer readings without time encodings, which is the extraction of each component (e.g., month, day, and hour) from the timestamp string. Moreover, the CNNs present a remarkably good performance, almost on par with LSTM and ETSFormer, indicating the convolutional layers can successfully extract features that are descriptive from the time series. Additionally, despite being the most accurate model, ETSFormer was not the most consistent, as it exhibited a higher standard deviation than LSTM. Lastly, Fig. 9 shows a sample dendrometer time series with the forecasted time series of the best configuration of each model.

4.2. Experiment 2 - temporal resolution analysis

In this experiment, the results from all test time series (48) were grouped according to the temporal resolution and model. Only the models with the best MAE in the previous experiment (5-min resolution) are presented. The aggregated MAE for each temporal resolution is presented in Table 10 where all models are shown together for easier comparison. Additionally, we include the violin plot showing the error dispersion for the best version of all four models at the temporal resolutions of 5, 30, and 60 min in Fig. 10. Violin plots use the Absolute error of each prediction to calculate a few statistics, the red rectangle indicates the quartiles, the black line indicates the mean, the long blue lines indicate the minimum and maximum values, and the shade indicates the frequency that a prediction had that error. The larger the shade the more common that value was among all the predictions. The X-axis ticks are named after the combination of the model and the time resolution. Moreover, the plots use a symlog where the information up to 0.05 uses a linear scale and above that value it uses a log scale.

Due to the experimental design using time-defined lookback sequences (10 h and 24 h), we adjust the input sequence of the coarse resolutions according to time instead of using the same number of time steps for all temporal resolutions. Precisely, a 10-h lookback has 120 data points using a 5-min resolution, while the same 10-h lookback has 10 data points using a 60-min resolution. As a result, the input sequence of the coarser resolutions (i.e., 360 min and 720 min) contains fewer than 5 items. Unfortunately, both CNN and ETSFormer architectures require more than five data points to work due to their convolution and pooling layers, which causes execution errors with coarse resolutions.

While the 5-min resolution experiments show that CNNs can deliver a similar performance to the other models, the reduced temporal resolution experiments show the shortcomings of the CNN models when dealing with scarce data. While the more complex models suffer a decrease in performance, the speed of deterioration is much faster for CNNs. This could be attributed to how CNNs handle temporal patterns compared to the more robust LSTMs.

CNNs depend on local patterns and are primarily designed to capture local dependencies within a small window of data using convolutions (Borovykh et al., 2018). Reducing the data frequency effectively reduces the receptive field of CNNs, which makes CNNs struggle because they no longer have enough local patterns to extract meaningful features (Bai et al., 2018; van den Oord et al., 2013). On the other hand, LSTMs use their internal memory state to capture long-term dependencies, making them more robust to reductions in temporal resolution (Hochreiter and Schmidhuber, 1997).

Considering the increase in error from the 5-min resolution, the only viable resolution alternatives for LSTMs are 30-min and 180-min intervals. The LSTM model shows practically no deterioration in forecast quality when comparing the 5-min and 30-min resolutions and exhibits only a slight increase in error at the 180-min resolution.

For the Transformer model, only the 360-min resolution yields an MAE (0.0366) close to that of the 5-min resolution (0.0332). Having a good MAE with a coarser resolution suggests that the model is overfitting when using the 5-min resolution, which could also explain the large standard deviation (0.0216) by having large errors in samples where the time series behaviour deviates from the ones seen in the

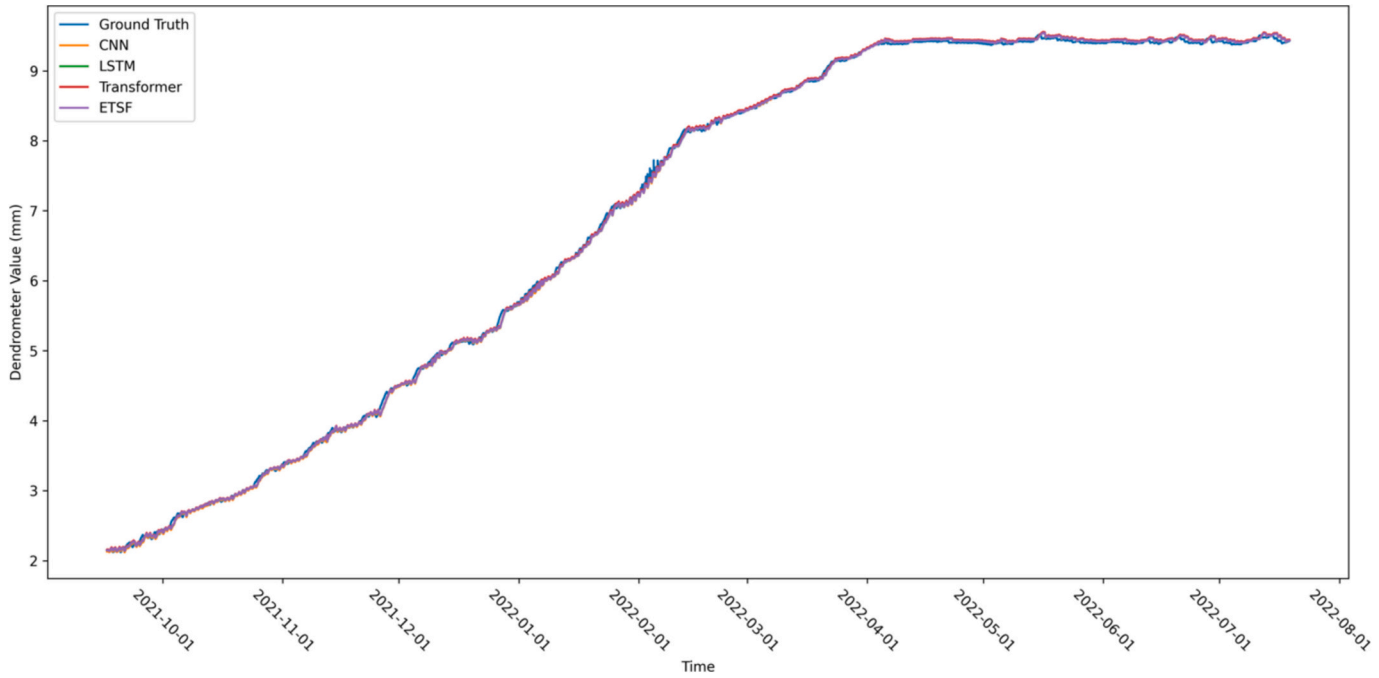


Fig. 9. A dendrometer time series and the forecasted time series from each model.

Table 10
Temporal resolution (Tres) results for the best models from Experiment 1. The best MAE results per temporal resolution are highlighted in bold. Averages are calculated from 48 samples (test sensors).

	CNN		LSTM		Transformer		ETSFormer	
Tres	MAEmean	MAEstd	MAEmean	MAEstd	MAEmean	MAEstd	MAEmean	MAEstd
5	0.0277	0.0066	0.0266	0.00555	0.0332	0.0216	**0.0261**	0.0056
30	0.1207	0.0233	**0.0294**	0.0051	0.0404	0.0105	0.0640	0.0180
60	0.2866	0.0514	0.0432	0.0032	0.0393	0.0047	**0.0266**	0.0198
180	–	–	**0.0313**	0.0066	0.0389	0.0077	0.0757	0.0356
360	–	–	**0.0359**	0.0096	0.0366	0.0082	–	–
720	–	–	**0.0428**	0.0059	0.0489	0.0078	–	–

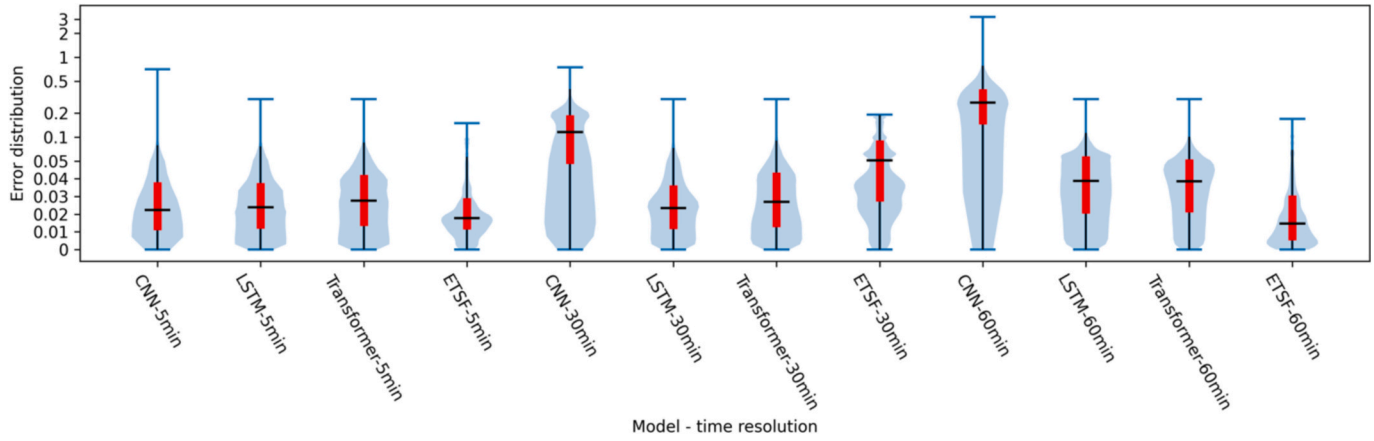


Fig. 10. Dispersion of the Absolute Errors from LSTM models prediction.

training set.

The ETSFormer model achieves excellent performance at the 60-min resolution (0.0266), exhibiting an MAE on par with the 5-min resolution LSTM and better than the 5-min resolution Transformer (0.0332) and CNN (0.0277). This performance advantage arises from employing classic statistical methods to enhance the NN's forecasting ability, enabling it to accurately model the dendrometer behaviour. It is noteworthy that the performance significantly deteriorates at 30-min and 3-h resolutions. One hypothesis for this performance discrepancy is related to the sinusoidal behaviour of trees and their seasonal variation, which may require different lookback sequences to accurately model the data under various temporal resolutions. This optimisation is not in the scope of this work. Another reason could be the reduction in the ratio of interpolated data compared to the total data available, which, despite our best efforts, could still be the cause for outliers and misrepresented behaviours.

5. Discussion

In general, the results of this study confirm that more complex models (i.e., models with more parameters) can capture intricate and long-term relationships better. To the best of our knowledge, our study one of the first to use the Transformer and ETSFormer models to forecast environmental data. Indeed, similar studies conducted in this field use classic models such as SVM and Random Forests, with a few of them touching simple Multilayer Perceptron (MLP) and LSTMs (Camastra et al., 2022; Cervantes et al., 2020; da Rocha et al., 2024; Jamali, 2019; Jemeļjanova et al., 2024; Khosravi et al., 2025; Tang and Li, 2023). In this sense, besides pioneering the use of more advanced models, we also pioneered the study of temporal resolution in the problem of forecasting complex environmental data.

A key result from our experiments shows that while the more complex models can capture the intricate relationships, they require a smaller LR. The smaller LR is required because the model has more parameters to optimise. This phenomenon is well-documented, as highlighted by Bengio (2012), who noted that excessively small LRs lead to slow convergence, while overly large ones can cause divergence. Slow convergence comes from the fact that the steps toward the minimum error are very small and take many epochs to reach convergence. The divergence would be the opposite, with a very large step that overshoots the minimum error, and thus, the model fails to find the best configuration.

Tuning the LR remains an open research topic. Recent works, such as Cyclic Learning Rates (CLR) (Smith, 2017), super-convergence (Smith and Topin, 2019), and non-monotonic LR schedulers (Seong et al., 2018), address the challenges of monotonic learning rates and propose techniques for faster convergence. However, these approaches still rely on hyperparameter tuning, given the hyperparameters associated with the scheduling process (e.g., initial LR, maximum LR, minimum LR, and warm-up epochs). Yedida et al. (2019) proposed another technique aimed at automatically computing an adaptive LR, backed by theoretical foundations, which could offer a promising avenue for enhancing results in future work without solely relying on hyperparameter tuning.

Despite the reports of forecasting accuracy and computational efficiency of Transformers being comparable to LSTMs and CNNs (Gopali et al., 2021; Pedro et al., 2021; Pölz et al., 2024), our evaluation indicates that using the basic architecture of Transformers for forecasting a univariate time series 24 h ahead was not the most efficient use of the attention mechanism. Pedro et al. (2021) noted that while Transformers excelled in capturing longer-term dependencies, they are generally more challenging to parametrise and exhibit higher result variability. These model characteristics, combined with the data characteristics of exhibiting daily and seasonal trends, could explain why the Transformer models in this study exhibited the worst performance among the four tested models. Analysing a single variable without proper context to infer if the tree is in an active or dormant period becomes very challenging.

Our CCN results confirm the general assumptions of the architecture.

The CNNs have presented a good performance at the 5-min resolution. However, their performance deteriorates quickly as the temporal resolution diminishes. As stated before, CNNs excel at capturing short-term dependencies. When data is more sparse, the convolutions struggle to extract good information. This is aligned with biophysical systems, specifically trees, where the daily sinusoidal growth cycle (short-term) appeared to be consistent (Allen et al., 2019; Enright, 1970; Olsen and Degn, 1985). On the other hand, as the update frequency becomes smaller, the seasonality of data ceases to be daily and starts becoming annual, representing the active and dormant states.

Based on our findings, it becomes evident that capturing long-term dependencies does not guarantee good performance in this problem. In addition to capturing such dependencies, the model has to cope with the daily seasonality and the seasons variance with the active and dormant growth states of the trees. For such reasons, specialised transformer architectures designed with unique methodologies to enhance the capture of complex temporal dependencies within data sequences may be an avenue to improve performance with this type of data. For instance, the W-transformer, introduced by Sasal et al. (2022), utilises a maximal overlap discrete wavelet transformation, facilitating the extraction of intricate nonlinear patterns across extended spans of the input sequence. Similarly, the Informer model, as proposed by Zhou et al. (2021) introduces a novel ProbSparse self-attention mechanism coupled with a distilling process and a generative-style decoder, enabling the prediction of longer time series in a single forward operation, thus eliminating the need for step-by-step inference.

Regarding ETSFormer, the decomposition of the time series to aid in the forecast appears to enhance forecasting accuracy, as evidenced by promising results with 5- and 60-min intervals. However, the results for 30- and 180-min intervals significantly deviate from this trend, which could be attributed to a lack of data, both in the time span of the dataset and the size of the lookback sequence, to perform proper smoothing and TS decomposition techniques. It is known that such techniques require data volume to provide robust and reliable results, and while this study uses large amounts of data, it does not span several years as usually found in TS studies.

Overall, our findings indicate that it is possible to reduce the temporal resolution of dendrometer SR with minimal performance loss. These results have several practical implications for terrestrial sensor networks in remote locations. First, reducing the temporal resolution leads to a cost reduction in storage space and processing time, as a high frequency of collection in many locations simultaneously can quickly accumulate several TBs of data. It also leads to the conservation of IoT resources, including battery life (by triggering less frequent readings) and data transmission over wireless communication. Additionally, terrestrial sensor networks in forests are typically in remote locations. Thus, they are usually hard to access for maintenance, which generates even more costs to manage a high throughput infrastructure.

Despite the promising findings, this research presents limitations that should be addressed in future work. These include the 24-h forecast horizon, the 10-h lookback sequence, and the lack of lookback sequence tuning for the reduced temporal resolution models. While the 24-h forecast horizon of this research is a very good demonstration of forecasting with complex environmental data, it can only provide for decision makers short-term monitoring and early warning of imminent problems with the environmental (e.g. floods). To provide adequate time for planning and response of decision makers and end users, forecasting needs to be weeks or months ahead. Nonetheless, our results show that DL is a viable alternative that could improve decision-making. Although mathematical models are accurate, they may be slow to produce forecasts (Hewamalage et al., 2021) as they usually need to compute interactions between several complex systems In addition, they require specific data and are fine-tuned for each location and species being studied (Hyndman and Athanasopoulos, 2021; Pankratz, 2012). In contrast, the DL models employed in this study effectively forecasted the radial growth of pine trees without the need for location-specific and auxiliary variables as in Baldocchi et al. (2001).

The lookback sequence has an interesting relationship with the forecast horizon. Generally, a longer forecast horizon usually requires a larger lookback sequence to accurately model recent dependencies and project them into the future. Although this study used data collected from 100 trees, the dataset spans less than a year, limiting the use of larger lookback sequences with coarser temporal resolutions and longer forecast horizons. Despite this, the DL methods were able to accurately model the time series without auxiliary environmental variables.

A key insight from this study is the evaluation of how different temporal resolutions affect forecasting quality. Our findings demonstrate that reducing temporal resolution is feasible with minimal performance loss. This approach could be validated by other researchers who have access to long-term datasets to assess whether larger datasets can further mitigate performance loss when using coarser temporal resolutions.

Building on these findings, future studies could incorporate environmental variables that influence plant and tree growth, such as weather conditions, soil properties, and topography. The significance of our results lies in establishing the limits of temporal resolution for dendrometer data, particularly in contexts where multiple variables are monitored at differing temporal frequencies. Understanding these limitations can guide the design of forestry monitoring systems, ensuring proper alignment and integration of variables collected at different resolutions. Ultimately, this knowledge could enhance the predictive accuracy of forecasting models and deepen our understanding of the mechanisms driving forest growth. Moreover, incorporating these insights could improve the models' ability to generalize across forests with varying environmental conditions, which will be a focus of future research.

6. Conclusion

In this paper, we have presented an evaluation of four DL algorithms for forecasting dendrometer time series, a discussion on the impact of hyperparameters concerning the data being modelled, and, to the best of our knowledge, the first study that: (i) uses Transformer and ETSFormer architecture to forecast complex environmental data; (ii) assesses the impact of reducing the temporal resolution of data collection as a means to decrease operational costs in remote sensing. Our results show the feasibility of reducing data collection intervals from 5 min to 180 min while maintaining similar error levels. We also explored the trade-offs associated with temporal resolution and their implications for the tested models. Finally, we suggest future research directions that could further contribute to this field. Our results show that, while CNNs can work very well for forecasting environmental data for higher temporal resolutions (i.e. 5 min resolution), the complex interdependency of daily and seasonal trends in dendrometer data makes them unsuitable for reduced temporal resolutions. Under reduced temporal resolution, our results show that LSTMs overperform the competitors at maintaining forecast quality. In addition, LSTM is a promising architecture for forecasting complex environmental data, as it was able to capture longer-term complex dependencies under reduced resolution.

Funding

This research is supported by the TAIAO project CONT-64517-SSIFDS-UOW (Time-Evolving Data Science / Artificial Intelligence for Advanced Open Environmental Science) and the Forest Flows Endeavour Programme (C04X1905). Both are funded by the New Zealand Ministry of Business, Innovation, and Employment (MBIE). URLs: https://taiao.ai/ https://forestflows.nz/

CRediT authorship contribution statement

Guilherme Cassales: Writing – review & editing, Writing – original

draft, Visualization, Software, Investigation, Data curation, Conceptualization. **Serajis Salekin:** Writing – review & editing, Writing – original draft, Data curation, Conceptualization. **Nick Lim:** Writing – review & editing, Software, Investigation. **Dean Meason:** Writing – review & editing, Funding acquisition. **Albert Bifet:** Writing – review & editing, Funding acquisition. **Bernhard Pfahringer:** Writing – review & editing. **Eibe Frank:** Writing – review & editing.

Acknowledgment

The authors would like to thank Scion's Forest Biometrics team for assisting with the experimental setup and Russell McKinley for manual data annotations and checking.

Data availability

The forest data used in this article is commercially owned by forest owners and companies and, therefore cannot be shared publicly. The codes developed in this study are available at the repository https://github.com/cassales/Ecological_Informatics_Stem_Radius_Forecasting.

References

Allen, S.T., Kirchner, J.W., Braun, S., Siegwolf, R.T.W., Goldsmith, G.R., 2019. Seasonal origins of soil water used by trees. Hydrol. Earth Syst. Sci. 23 (2), 1199–1210. https://doi.org/10.5194/hess-23-1199-2019.

Apache Kafka, 2023, May 1. Apache Kafka. https://kafka.apache.org/.

Bai, S., Kolter, J.Z., Koltun, V., 2018. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. arXiv 1–14. https://arxiv.org/abs/1803.01271.

Baldocchi, D., Falge, E., Gu, L., Olson, R., Hollinger, D., Running, S., Anthoni, P., Bernhofer, C., Davis, K., Evans, R., Fuentes, J., Goldstein, A., Katul, G., Law, B., Lee, X., Malhi, Y., Meyers, T., Munger, W., Oechel, W., Wofsy, S., 2001. FLUXNET: a new tool to study the temporal and spatial variability of ecosystem–scale carbon dioxide, water vapor, and energy flux densities. Bull. Am. Meteorol. Soc. 82 (11), 2415–2434. https://doi.org/10.1175/1520-0477(2001)082<2415:FANTTS>2.3.CO;2.

Bar-On, Y.M., Phillips, R., Milo, R., 2018. The biomass distribution on earth. Proc. Natl. Acad. Sci. 115 (25), 6506–6511. https://doi.org/10.1073/pnas.1711842115.

Bayne, K., Damesin, S., Evans, M., 2017. The internet of things - wireless sensor networks and their application to forestry. N. Z. J. For. 61, 37–41.

Bengio, Y., 2012. Practical recommendations for gradient-based training of deep architectures. In: Neural Networks: Tricks of the Trade, Second, pp. 437–478. https://doi.org/10.1007/978-3-642-35289-8_26.

Bennett, E.M., Peterson, G.D., Gordon, L.J., 2009. Understanding relationships among multiple ecosystem services. Ecol. Lett. 12 (12), 1394–1404. https://doi.org/10.1111/j.1461-0248.2009.01387.x.

Bonan, G.B., 2016. Forests, climate, and public policy: A 500-year interdisciplinary odyssey. Annu. Rev. Ecol. Evol. Syst. 47 (1), 97–121. https://doi.org/10.1146/annurev-ecolsys-121415-032359.

Borovykh, A., Bohte, S., Oosterlee, C.W., 2018. Conditional time series forecasting with convolutional neural networks. In: Proceedings of the International Conference on Artificial Neural Networks, ICANN 2017. https://arxiv.org/abs/1703.04691.

Camastra, F., Capone, V., Ciaramella, A., Riccio, A., Staiano, A., 2022. Prediction of environmental missing data time series by support vector machine regression and correlation dimension estimation. Environ. Model Softw. 150, 105343. https://doi.org/10.1016/j.envsoft.2022.105343.

Cazzolla Gatti, R., Reich, P.B., Gamarra, J.G.P., Crowther, T., Hui, C., Morera, A., Bastin, J.-F., de Miguel, S., Nabuurs, G.-J., Svenning, J.-C., Serra-Diaz, J.M., Merow, C., Enquist, B., Kamenetsky, M., Lee, J., Zhu, J., Fang, J., Jacobs, D.F., Pijanowski, B., Liang, J., 2022. The number of tree species on Earth. Proc. Natl. Acad. Sci. 119 (6). https://doi.org/10.1073/pnas.2115329119.

Cervantes, J., Garcia-Lamont, F., Rodríguez-Mazahua, L., Lopez, A., 2020. A comprehensive survey on support vector machine classification: applications, challenges and trends. Neurocomputing 408, 189–215. https://doi.org/10.1016/j.neucom.2019.10.118.

Che, Z., Purushotham, S., Cho, K., Sontag, D., Liu, Y., 2018. Recurrent neural networks for multivariate time series with missing values. Sci. Rep. 8 (1), 6085. https://doi.org/10.1038/s41598-018-24271-9.

da Rocha, S.J.S., Torres, C.M.M.E., Villanova, P.H., Rufino, M.P.M.X., Romero, F.M.B., Jacovine, L.A.G., de Morais Junior, V.T.M., de França, L.C.J., Schettini, B.L.S., Reis, L.P., Viana, Á.B.T., Albuquerque, T.P., Verly, O.M., Soares, C.P.B., Leite, H.G., 2024. Machine learning methods: modeling net growth in the Atlantic Forest of Brazil. Eco. Inform. 81, 102564. https://doi.org/10.1016/j.ecoinf.2024.102564.

Dehghani, Zhamak, 2022. Data Mesh, vol. 9781492092391. O'Reilly Media, Inc.

Dimri, T., Ahmad, S., Sharif, M., 2020. Time series analysis of climate variables using seasonal ARIMA approach. J. Earth Syst. Sci. 129 (1), 149. https://doi.org/10.1007/s12040-020-01408-x.

Enright, J.T., 1970. Ecological aspects of endogenous rhythmicity. Annu. Rev. Ecol. Syst. 1 (1), 221–238. https://doi.org/10.1146/annurev.es.01.110170.001253.

García-Laencina, P.J., Sancho-Gómez, J.-L., Figueiras-Vidal, A.R., 2010. Pattern classification with missing data: a review. Neural Comput. & Applic. 19 (2), 263–282. https://doi.org/10.1007/s00521-009-0295-6.

Gopali, S., Abri, F., Siami-Namini, S., Namin, A.S., 2021. A Comparative Study of Detecting Anomalies in Time Series Data Using LSTM and TCN Models.

Hewamalage, H., Bergmeir, C., Bandara, K., 2021. Recurrent neural networks for time series forecasting: current status and future directions. Int. J. Forecast. 37 (1), 388–427. https://doi.org/10.1016/j.ijforecast.2020.06.008.

Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural Comput. 9 (8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735.

Hock, B., Heaphy, M., Evans, M., Dunningham, A., Graham, B., 2016. The internet of things for forestry: new concepts, new opportunities A view of the future. NZ J. For. 60 (4).

Hodson, T.O., 2022. Root-mean-square error (RMSE) or mean absolute error (MAE): When to use them or not. Geosci. Model Dev. 15 (14), 5481–5487. https://doi.org/10.5194/gmd-15-5481-2022.

Hyndman, R.J., Athanasopoulos, G., 2021. Forecasting: Principles and Practice, 3rd ed. OTexts OTexts.com/fpp3.

Jamali, A., 2019. A fit-for-purpose algorithm for environmental monitoring based on maximum likelihood, support vector machine and random forest. Int. Arch. Photogramm. Remote. Sens. Spat. Inf. Sci. XLII-3/W7, 25–32. https://doi.org/10.5194/isprs-archives-XLII-3-W7-25-2019.

Jemeljanova, M., Kmoch, A., Uuemaa, E., 2024. Adapting machine learning for environmental spatial data - a review. Eco. Inform. 81, 102634. https://doi.org/10.1016/j.ecoinf.2024.102634.

Jordan, M.I., Mitchell, T.M., 2015. Machine learning: trends, perspectives, and prospects. Science 349 (6245), 255–260. https://doi.org/10.1126/science.aaa8415.

Kaur, J., Parmar, K.S., Singh, S., 2023. Autoregressive models in environmental forecasting time series: a theoretical and application review. Environ. Sci. Pollut. Res. 30 (8), 19617–19641. https://doi.org/10.1007/s11356-023-25148-9.

Keenan, T.F., Williams, C.A., 2018. The terrestrial carbon sink. Annu. Rev. Environ. Resour. 43 (1), 219–243. https://doi.org/10.1146/annurev-environ-102017-030204.

Khosravi, K., Farooque, A.A., Naghibi, A., Heddam, S., Sharafati, A., Hatamiafkoueieh, J., Abolfathi, S., 2025. Enhancing Pan evaporation predictions: accuracy and uncertainty in hybrid machine learning models. Eco. Inform. 85, 102933. https://doi.org/10.1016/j.ecoinf.2024.102933.

Lawrence, Steve Jerome, 2023, December 20. What Is LSTM? - Introduction to Long Short-Term Memory. https://www.scaler.com/topics/deep-learning/lstm/.

Lepot, M., Aubin, J.-B., Clemens, F., 2017. Interpolation in time series: an introductive overview of existing methods, their performance criteria and uncertainty assessment. Water 9 (10), 796. https://doi.org/10.3390/w9100796.

Lin, Y., Salekin, S., Meason, D.F., 2023. Modelling tree diameter of less commonly planted tree species in New Zealand using a machine learning approach. For. Intern. J. Forest Res. 96 (1), 87–103. https://doi.org/10.1093/forestry/cpac037.

Lindemann, B., Müller, T., Vietz, H., Jazdi, N., Weyrich, M., 2021. A survey on long short-term memory networks for time series prediction. Procedia CIRP 99, 650–655. https://doi.org/10.1016/j.procir.2021.03.088.

Luković, M., Zweifel, R., Thiry, G., Zhang, C., Schubert, M., 2022. Reconstructing radial stem size changes of trees with machine learning. J. R. Soc. Interface 19 (194). https://doi.org/10.1098/rsif.2022.0349.

Olsen, L.F., Degn, H., 1985. Chaos in biological systems. Q. Rev. Biophys. 18 (2), 165–225. https://doi.org/10.1017/S0033583500005175.

Pankratz, A., 2012. Forecasting with Dynamic Regression Models. John Wiley & Sons.

Pedro, Lara-Benítez, Gallego-Ledesma, L., C.-G. M, L.-R. J. M, 2021. Evaluation of the transformer architecture for univariate time series forecasting. In: G. and C. F. and C. C. and C. D. and O.-A. M. and M. S. and T. A. and R. J. and G.-M. R, Enrique, Alba, Luque (Eds.), Advances in Artificial Intelligence. Springer International Publishing, pp. 106–115.

Pichler, M., Hartig, F., 2023. Machine learning and deep learning—A review for ecologists. Methods Ecol. Evol. 14 (4), 994–1016. https://doi.org/10.1111/2041-210X.14061.

Pölz, A., Blaschke, A.P., Komma, J., Farnleitner, A.H., Derx, J., 2024. Transformer versus LSTM: A comparison of deep learning models for karst spring discharge forecasting. Water Resour. Res. 60 (4). https://doi.org/10.1029/2022WR032602.

Salekin, S., Bloomberg, M., Morgenroth, J., Meason, D.F., Mason, E.G., 2021. Within-site drivers for soil nutrient variability in plantation forests: A case study from dry sub-humid New Zealand. CATENA 200, 105149. https://doi.org/10.1016/j.catena.2021.105149.

Sasal, L., Chakraborty, T., Hadid, A., 2022. W-transformers: A wavelet-based transformer framework for univariate time series forecasting. In: 2022 21st IEEE International Conference on Machine Learning and Applications (ICMLA), pp. 671–676. https://doi.org/10.1109/ICMLA55696.2022.00111.

Schmitt, C.B., Burgess, N.D., Coad, L., Belokurov, A., Besançon, C., Boisrobert, L., Campbell, A., Fish, L., Gliddon, D., Humphries, K., Kapos, V., Loucks, C., Lysenko, I., Miles, L., Mills, C., Minnemeyer, S., Pistorius, T., Ravilious, C., Steininger, M., Winkel, G., 2009. Global analysis of the protection status of the world's forests. Biol. Conserv. 142 (10), 2122–2130. https://doi.org/10.1016/j.biocon.2009.04.012.

Seddon, N., 2022. Harnessing the potential of nature-based solutions for mitigating and adapting to climate change. Science 376 (6600), 1410–1416. https://doi.org/10.1126/science.abn9668.

Seong, S., Lee, Y., Kee, Y., Han, D., Kim, J., 2018. Towards flatter loss surface via nonmonotonic learning rate scheduling. In: Conference on Uncertainty in Artificial Intelligence, pp. 1020–1030.

Shumway, R.H., Stoffer, D.S., Stoffer, D.S., 2000. Time Series Analysis and its Applications, vol. 3. Springer.

Smith, L.N., 2017. Cyclical learning rates for training neural networks. In: 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 464–472. https://doi.org/10.1109/WACV.2017.58.

Smith, L.N., Topin, N., 2019. Super-convergence: Very fast training of neural networks using large learning rates. In: Pham, T. (Ed.), Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications. SPIE, p. 36. https://doi.org/10.1117/12.2520589.

Tang, Y., Li, H., 2023. Comparing the performance of machine learning methods in predicting soil seed bank persistence. Eco. Inform. 77, 102188. https://doi.org/10.1016/j.ecoinf.2023.102188.

van den Oord, A., Dieleman, S., Schrauwen, B., 2013. Deep content-based music recommendation. In: Burges, C.J., Bottou, L., Welling, M., Ghahramani, Z., Weinberger, K.Q. (Eds.), Advances in Neural Information Processing Systems, 26. Curran Associates, Inc.. In: https://proceedings.neurips.cc/paper_files/paper/2013/file/b3ba8f1bee1238a2f37603d90b58898d-Paper.pdf

Vaswani, A., Brain, G., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I., 2017. Attention is all you need. Adv. Neural Inf. Proces. Syst. 5998–6008. https://papers.nips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Woo, G., Liu, C., Sahoo, D., Kumar, A., Hoi, S., 2022. ETSformer: Exponential Smoothing Transformers for Time-series Forecasting. http://arxiv.org/abs/2202.01381.

Yedida, R., Saha, S., Pilani, B., Goa, B., Prashanth, T., 2019. A novel adaptive learning rate scheduler for deep neural networks LipschitzLR: Using theoretically computed adaptive learning rates for fast convergence. https://doi.org/10.13140/RG.2.2.28333.95201.

Zhang, G.P., 2003. Time series forecasting using a hybrid ARIMA and neural network model. Neurocomputing 50, 159–175. https://doi.org/10.1016/S0925-2312(01)00702-0.

Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., Zhang, W., 2021. Informer: beyond efficient transformer for long sequence time-series forecasting. Proc. AAAI Conf. Artific. Intellig. 35 (12), 11106–11115. https://doi.org/10.1609/aaai.v35i12.17325.

Zhumagambetov, R., Molnár, F., Peshkov, V.A., Fazli, S., 2021. Transmol: repurposing a language model for molecular generation. RSC Adv. 11 (42), 25921–25932. https://doi.org/10.1039/D1RA03086H.

Zweifel, R., Haeni, M., Buchmann, N., Eugster, W., 2016. Are trees able to grow in periods of stem shrinkage? New Phytol. 211 (3), 839–849. https://doi.org/10.1111/nph.13995.

Zweifel, R., Etzold, S., Sterck, F., Gessler, A., Anfodillo, T., Mencuccini, M., von Arx, G., Lazzarin, M., Haeni, M., Feichtinger, L., Meusburger, K., Knuesel, S., Walthert, L., Salmon, Y., Bose, A.K., Schoenbeck, L., Hug, C., De Girardi, N., Giuggiola, A., Rigling, A., 2020. Determinants of legacy effects in pine trees – implications from an irrigation-stop experiment. New Phytol. 227 (4), 1081–1096. https://doi.org/10.1111/nph.16582.

Zweifel, R., Sterck, F., Braun, S., Buchmann, N., Eugster, W., Gessler, A., Häni, M., Peters, R.L., Walthert, L., Wilhelm, M., Ziemińska, K., Etzold, S., 2021. Why trees grow at night. New Phytol. 231 (6), 2174–2185. https://doi.org/10.1111/nph.17552.