An autoencoder wavelet based deep neural network with attention mechanism for multi-step prediction of plant growth

Bashar Alhnaity[a,*], Stefanos Kollias[a], Georgios Leontidis[a], Shouyong Jiang[a], Bert Schamp[b], Simon Pearson[c]

[a] *School of Computer Science, University of Lincoln, Brayford Pool, Lincoln, UK*
[b] *PCS Ornamental Plant Research, Schaessest raat 18, Dest elbergen, Belgium*
[c] *Lincoln Institute for Agri-Food Technology, University of Lincoln, Riseholme, Lincoln, UK*

Abstract

Multi-step-ahead prediction is considered of major significance for time series analysis in many real life problems. Existing methods mainly focus on one-step-ahead forecasting, since multiple step forecasting generally fails due to accumulation of prediction errors. This paper presents a novel approach for predicting plant growth in agriculture, focusing on prediction of plant Stem Diameter Variations (SDV). The proposed approach consists of three main steps. At first, wavelet decomposition is applied to the original data, so as to facilitate model fitting and reduce noise. Then an encoder-decoder framework is developed using Long Short Term Memory (LSTM) and used for appropriate feature extraction from the data. Finally, a recurrent neural network including LSTM and an attention mechanism is proposed for modelling long-term dependencies in the time series data. Experimental results are presented which illustrate the good performance of the proposed approach and that it significantly outperforms the existing models, in terms of error criteria such as RMSE, MAE and MAPE.

Keywords: multi-step prediction, wavelet analysis, LSTM, deep neural networks, attention mechanism, time series analysis, plant growth prediction

*Corresponding author
Email address: alhnaity.bashar@gmail.com (Bashar Alhnaity)

1. Introduction

Time-series analysis and prediction has been a research topic of significance in various fields and real-life applications, including smart agriculture and prediction of plant growth, forecasting of financial stocks, anomaly, or intrusion, detection, medical imaging and air pollution prediction [15, 2, 1]. Time series data are generally produced as series of observations aggregated in chronological order. Their complexity is generally quite high, which makes their analysis a very challenging task [18]. Due to this nature, using shallow machine learning and neural network models to analyze the data has produced many bottlenecks. As a consequence, the development and use of more complex models, which can automatically extract and learn deep representations from time-series, or image data, has been a topic of major recent work [37, 28, 4, 6, 5, 3].

Recently, Deep Learning (DL) models have produced great progress in agricultural tasks, such as crop management and plant growth analysis. Plants, like other bio-systems, are highly complex and dynamic systems. Modelling plant growth dynamics is a unique challenge, due to large data variations, e.g., related to scale of interest, level of description, or integration of environmental parameters [2].

Multi-step-ahead time series prediction refers to prediction of the time series in multiple time steps ahead into the future. In comparison with one step ahead prediction, multi-step-ahead prediction can provide additional benefits to growers; it is, however, more challenging task as it has to address various additional complications [42, 45]. In the literature, there are three primary strategies for managing multi-step-ahead prediction tasks: the recursive strategy, the direct strategy and the multiple output prediction strategy. The recursive strategy is based on consecutive one-step-ahead forecasts; each step ahead prediction uses previously forecasted values as inputs. Recursive strategy methods have few drawbacks, such as error accumulation. The direct strategy predicts separate models for each forecast. Other techniques on hybrid direct-recursive strategies. Moreover, there is the multi-output model strategy, which is designed to fore-

cast the entire time series ahead, in one shot. All strategies include challenges that need to be tackled [44].

This paper proposes a novel deep learning direct strategy approach for effective prediction of plant growth. It consists of three components: Wavelet Transformation (WT), encoding-decoding based on the LSTM model, and prediction using LSTM with an attention mechanism. WT can assist in smoothing the noise effect existing in time series data. The encoder-decoder (ED) part can extract appropriate features from the reconstructed smoothed signal; these features form a compact representation, exploited in the final prediction step. A model composed of LSTM units is blended with Attention Mechanism (AM), in order to implement the final prediction of plant growth. The resulting approach is named WT-ED-LSTM-AM hereafter. The effectiveness of the WT-ED-LSTM-AM model is validated using real datasets provided by European greenhouses. Moreover, the obtained results are compared with those achieved when using Support Vector Regression (SVR), Random Forest Regression (RFR), standard Long-Short Term Memory (LSTM) networks, multi-layer perceptrons (MLP), and networks with gated recurrent units (GRU). An ablation study has also been implemented, by removing either the wavelet transform part (ED-LSTM-AM method), or the attention mechanism part (WT-ED-LSTM method).

In summary, the main contributions of this paper are the following:

- A novel architecture for multi-step prediction of plant growth and stem diameter variations, including wavelet transformation, data encoding-decoding and an LSTM with attention components.
- Improved performance in multi-step prediction on real life data sets, when compared with baseline models and state-of-the-art methods.

The remainder of this paper is organised as follows: Section 2 presents related work. Section 3 describes the proposed pipeline and the utilized models and components. Section 4 provides a detailed presentation of the proposed WT-ED-LSTM-AM approach. Section 5 presents the developed experimental study.

Finally, Section 6 provides conclusions and suggestions for future work.

2. Related Work

This section provides a short description of existing machine learning prediction models applied to horticulture, and in particular, to plant growth analysis, which is crucial for smart farming [47].

Data-driven models (DDM) that are used in signal processing include Machine Learning (ML) models, such as Generalized Linear Models, Artificial Neural Networks [14] and Support Vector Machines [34]. Those methods have many desirable characteristics, such as: imposing few restrictions and assumptions; ability to approximate nonlinear functions; strong predictive capabilities; flexibility to adapt to multivariate system inputs [9]. According to [39] machine learning, linear polarisation, wavelet-based filtering, vegetation indices and regression analysis are the most popular techniques used for analyzing agricultural data. Deep Learning (DL) has obtained great popularity in the last few years [22]. DL involves Deep Neural Networks (DNNs), which can extract hierarchical feature structures and create rich representations of the data. A strong advantage of DL is feature learning, i.e., automatic feature extraction from raw data, with features from higher levels of the hierarchy being formed through composition of lower-level features [22]. Consequently, DL can solve complex real life problems with high accuracy [32], provided there is availability of adequately large data-sets describing the problem. Gonzalez-Sanchez et al. [21] presented a comparative study of ANN, SVR, M5-prime regression, K-nearest neighbor classifiers and Multiple Linear Regression for crop yield prediction in ten crop datasets. In their study, Root Mean Square Error (RMSE), Root Relative Square Error (RRSE), Normalized Mean Absolute Error (MAE) and Correlation Factor (R) were used as accuracy metrics to validate the models. Results showed that M5-Prime regression achieved the lowest errors across the produced crop yield models. The results of that study ranked the techniques from best to worst, as follows: M5-Prime, kNN, SVR, ANN, MLR. Another

study by [10] applied four ML techniques, SVM, Random Forest (RF), Extremely Randomised Trees (ERT) and Deep Learning (DL) to estimate corn yield in Iowa State. Comparison of the validation statistics showed that DL provided the more stable results, overcoming the over-fitting problem. In the current paper (and in [2]) we develop a novel deep learning architecture for prediction of plant growth, using stem diameter variation as a growth indicator.

Stem diameter is considered a parameter of major importance that describes the growth of plants during vegetative growth stage. The variation of stem diameter has been widely used to derive proxies for plant water status and, is therefore used in optimisation strategies for plant-based irrigation scheduling in a wide range of species. Plant stem diameter variation (SDV) refers to plant stem periodic shrinkage and recovery movement during day and night. This periodic variation is related to plant water content and can be used as an indicator of the plant water content changes. During active vegetative growth and development, crop plants rely on the carbohydrate gained from photosynthesis and the translocation of photo-assimilates from the site of synthesis to sink organs [48]. The fundamentals of stem diameter variations have been well documented in the literature [46]. It has been documented that SDV is sensitive to water and nutrient conditions and is closely related to the response of crop plants to changes of environmental conditions [27]. Moreover, stem diameter is a parameter that describes the growth of crop plants under abiotic stress during vegetative growth stage. Therefore, it is important to generate stem diameter growth models able to predict the response of SDV to environmental changes and plant growth under different conditions. Many studies emphasize the need to critically review and improve SDV models for assessment of environmental impact on crop growth [25]. SDV daily models have been developed to accurately predict inter-annual variation in annual growth in balsam fir (Abies balsamea L). Inclusion of daily data in growth-climate models can improve prediction of the potential growth response to climate by identifying particular climatic events that escape to a classical dendroclimatic approach [17]. However, development of models that are capable of predicting SDV and plant growth taking

into consideration environmental variables has so far remained limited.

Since horticulture management decisions become data-driven, DL is continuously gaining popularity as one of the most successful techniques to model obtained data. In this paper, we propose a DL model and a new approach for multi-step prediction of plant growth using wavelet transformation (WT), encoder-decoder based on LSTM and RNN-LSTM prediction with an attention mechanism and we evaluate its performance on real plant growth data.

3. Problem Definition and Components

3.1. Problem definition

The aim of a model for single step time series prediction, T, is to implement a mapping from a sequence of input data, $(\mathbf{x_0}, \mathbf{x_1}, ..., \mathbf{x_t})$, to a single output target value, y_{t+1}:

$$\hat{y_{t+1}} = T(\mathbf{x_0}, ..., \mathbf{x_t}) \tag{1}$$

where $\mathbf{x}_t$ is generally an M-dimensional vector, with elements $(x_t(0), ..., x_t(M-1))$ and $x_t(i), y_t \in \mathbb{R}$; $\hat{y}_t$ is the estimate of the target value y_t. Eq. 2 shows a multi(k)-step prediction:

$$(\hat{y}_{t+1}, ...\hat{y}_{t+k}) = T(\mathbf{x_0}, ..., \mathbf{x_t}) \tag{2}$$

Model T is usually estimated through supervised learning with direct strategy for multi-step prediction, using a collection of training data and respective labels.

3.2. Wavelet transform

The Wavelet transform can be used for data denoising, while handling the non-stationary nature of the collected time series data. In the following we use the wavelet transform for representing, decomposing and reconstructing the original data. Wavelet analysis was firstly introduced by Mallat [31] and since then has been used in various domains for signal processing [33], image recognition [23], remote sensing data decomposition [35], time series decomposition

[16], medical image analysis and medical diagnosis [40]. The Discrete Wavelet Transform decomposes signals into a low frequency approximation set and several high frequency detailed sets. Thus the original time series, represented as $X = [\mathbf{x_0}, ..., \mathbf{x_{N-1}}]$ with $N = 2^J$, is transformed as shown in Eqs. 3 and 4 :

$$W_\varphi(j,n) = \frac{1}{\sqrt{N}} \sum_k \mathbf{x}_k \varphi_{j,n}(k) \tag{3}$$

$$W_\psi(j,n) = \frac{1}{\sqrt{N}} \sum_k \mathbf{x}_k \psi_{j,n}(k) \tag{4}$$

where $j = 0, 1, ..., J-1,\ k = 0, ..., N-1$ and $n = 0, 1, ..., 2^j - 1$; φ and ψ represent the wavelet function and the scaling function, respectively. Mallat proposed filtering the time series using a pair of high-pass and low-pass filters as an implementation of discrete wavelet transform. There are two types of wavelets, father wavelets $\varphi(t)$ and mother wavelets $\psi(t)$, in the Mallat algorithm. Father wavelets $\varphi(t)$ and mother wavelets $\psi(t)$ integrate to 1 and 0, respectively, which can be formulated as:

$$\int \varphi(t)dt = 1, \int \psi(t)dt = 0 \tag{5}$$

The mother wavelets describe high-frequency parts, while the father wavelets describe low-frequency components of a time series. The mother wavelets and father wavelets in the j level can be formulated as:

$$\varphi_{j,k}(t) = 2^{-\frac{j}{2}} \varphi(2^{-j} - k) \tag{6}$$

$$\psi_{j,k}(t) = 2^{-\frac{j}{2}} \psi(2^{-j} - k) \tag{7}$$

Time series data can be reconstructed by a series of projections on the mother and father wavelets with multilevel analysis indexed by $k\epsilon\{0, 1, 2, ...\}$ and by $j\epsilon\{0, 1, 2, ..., J\}$, where J denotes the number of multi-resolution scales. The orthogonal wavelet series approximation to a time series $x(t)$ is formulated by:

$$x(t) = \sum_k s_{J,k}\varphi_{J,k}(t) + \sum_k f_{J,k}\psi_{J,k}(t) + \sum_k d_{J-1,k}(t) + ... + \sum_k d_{1,k}\psi_{1,k}(t) \tag{8}$$

where the expansion coefficients $s_{J,k}$ and $d_{J,k}$ are given by the projections

$$s_{J,k} = \int \varphi_{J,k} x(t) dt \tag{9}$$

$$d_{j,k} = \int \psi_{j,k} x(t) dt \tag{10}$$

The multi-scale approximation of time series $x(t)$ is given as:

$$S_J(t) = \Sigma_k s_{J,k} \varphi_{J,k}(t) \tag{11}$$

$$D_j(t) = \Sigma_k d_{j,k} \psi_{j,k}(t) \tag{12}$$

Then, the brief form of orthogonal wavelet series approximation can be denoted by:

$$x(t) = S_J(t) + D_J(t) + D_{J-1}(t) + . + D_1(t) \tag{13}$$

where $S_J(t)$ is the coarsest approximation of the input time series $x(t)$. The multi-resolution decomposition of $x(t)$ is the sequence of $S_J(t), D_J(t), D_{J-1}(t), ...D_1(t)$. To reconstruct the original series, Eq. 14 is used:

$$\mathbf{x}_k = \frac{1}{\sqrt{N}} \sum_n W_\varphi(j,n) \varphi_{j,n}(k) + \sum_{j=j}^{\infty} \sum_n W_\psi(j,n) \psi_{j,n}(k) \tag{14}$$

There are several wavelet families, such as Daubechies (dbN), Coiflets (CoifN) and Symlets (symN). In this paper we use db2 to decompose the original series into one approximation and two detail sets.

3.3. Support vector regression

Support vector regression (SVR) technique is a prediction method that arises from a nonlinear generalization of the Generalized Portrait algorithm developed by Vapnik [13]. The goal of SVR is to obtain a linear function $f(x) =< w, x > +b$ with $w \epsilon R^N$ and $b \epsilon R$ for a given training set $\{(x_1, y_1), ..., (x_m, y_m)\}$ as follows:

$$\text{minimize } \frac{1}{2}\|w\|^2 + c\sum_{i=1}^{m}(\xi_i + \xi_1^*)$$
$$\text{subject to } \begin{cases} y_i - \langle w, x_i\rangle - b \leq \varepsilon + \xi_i \\ \langle w, x_i\rangle + b - y_i \leq \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0 \end{cases} \tag{15}$$

where ξ_i and ξ_i^* are slack variables introduced to deal with infeasible constraints and C is called the *regularization* parameter. In most cases, the problem can be solved in its dual formulation:

$$\begin{aligned} max_{a,a^*} -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}(a_i - a_i^*)(a_j - a_j^*)K(x_i - x_j) \\ -\varepsilon\sum_{i=1}^{N}(a_i - a_i^*) + \sum_{i=1}^{N} y_i(a_i - a_i^*) \\ \text{subject to:} \\ \sum_{i=1}^{N}(a_i - a_i^*) = 0, a_i, a_i^* \varepsilon\, [0, C] \end{aligned} \tag{16}$$

where $K(x_i, x_j)$ is known as the kernel function, which allows the projection of the original data to a higher dimensional space, so as to become linearly separable. Common kernel functions include the linear, radial basis and polynomial ones. among these, Radial Basis Function (RBF) provides dimensionality reduction, restricting the computational load during training and providing improved generalization capabilities. For these reasons, RBF kernel has been adopted, defined as follows:

$$K(x, x_i) = exp\left(-\frac{1}{\sigma^2}\|x - x_i\|^2\right) \tag{17}$$

where x and x_i are vectors in the input space, i.e., vectors of features from the training or test datasets.

3.4. Random forest regression

Random forest regression (RFR) belongs to the category of ensemble learning algorithms. As a base learner of the ensemble, RFR uses decision trees. The

idea of ensemble learning is that multiple predictors can be more effective in making predictions over the test data, distinguishing noise from patterns. RFR constructs independent regression trees, with a bootstrap sample of the training data being chosen at each regression tree. As a consequence, the regression tree continuously grows until it reaches the largest possible size. Final prediction is a weighted average of all regression trees predictions [8].

3.5. Multilayer perceptrons

Multilayer Perceptrons (MLP) [24] have been the main architecture used for supervised learning and classification tasks in the past; they consist of multiple fully connected layers of neurons, with feedforward spread of information. Their training is performed with the backpropagation algorithm.

3.6. Long-short term memory

Long short-term memory (LSTM) is a variation of the recurrent neural network (RNN) architecture [26]. They have been able to solve the gradient vanishing problem in long-term time series analysis. The LSTM structure contains three modules: the forget gate, the input gate and the output gate. The forget and input gates control which part of the information should be removed/reserved to the network; the output gate uses the processed information to generate the provided output. LSTMs also include a Cell State, which allows the information to be saved for a long time. In the following, we illustrate the operation of LSTM units.

Let i_t and $\tilde{c}_t$ be the values of the input gate and the candidate state of the memory cell at time t, respectively. These are computed as follows:

$$i_t = \sigma(W_i \mathbf{x_t} + U_i h_{t-1} + b_i) \tag{18}$$

$$\tilde{c}_t = \tanh(W_c \mathbf{x_t} + U_c h_{t-1} + b_c) \tag{19}$$

Let us denote by f_t and c_t the value of the forget gate and the state of memory cell at time t. These are, respectively, calculated by:

$$f_t = \sigma(W_f \mathbf{x_t} + U_f h_{t-1} + b_f) \tag{20}$$

$$c_t = i_t * \tilde{c}_t + f_t * c_{t-1} \tag{21}$$

Let, also, o_t and h_t denote the values of the output gate and memory cell at time t, respectively. These are computed as follows:

$$o_t = \sigma(W_o \mathbf{x_t} + U_o h_{t-1} + V_o c_t + b_o) \tag{22}$$

$$h_t = o_t * \tanh(c_t) \tag{23}$$

where $\mathbf{x_t}$ is the input vector to the memory cell at time t; W_i, W_f, W_c, W_o, U_i, U_f, U_c, U_o and V_o are weight matrices; b_i, b_f, b_c and b_o are bias vectors;

3.7. Gated recurrent units

Gated recurrent units (GRU) are simplified LSTMs. They do not include output gates, thus there is no control over the memory content. They can be used instead of LSTMs. Further information can be found in [12].

3.8. LSTM encoder-decoder models

In LSTM Encoder-Decoder models, the encoder part compresses the information from the entire input sequence into a vector composed of the sequence of the LSTM hidden states. Consequently, the encoder summarizes the whole input sequence into the final cell state vector and passes it to the decoder [30, 43]. The latter uses this representation as initial state to reconstruct the time series, denoted as s_t in Fig. 1.

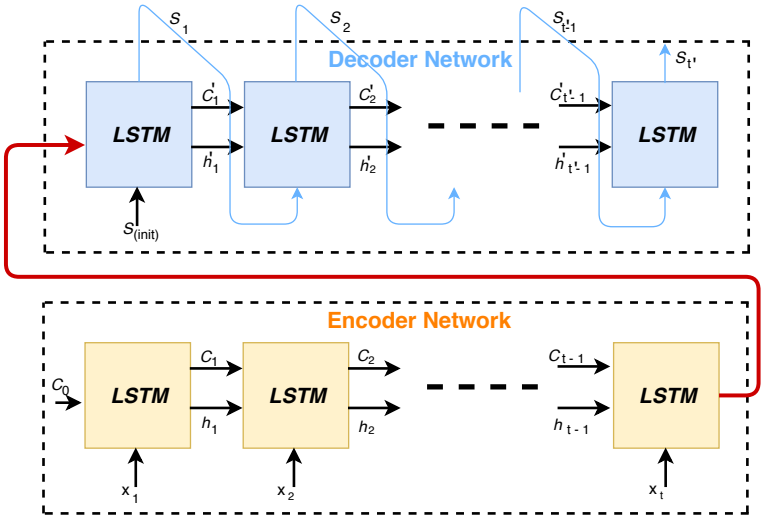

Figure 1: LSTM encoder decoder architecture.

3.9. Attention mechanisms

Attention mechanisms have been used as a means to improve performance in vision and signal processing tasks, by focusing on feature segments of high significance [36]. They are currently implemented through attentive neural network models [20, 49, 19, 38, 11, 41]. Bahdanau et al [7] introduced an attention mechanism to model a long-term dependence, by generating a context vector as a weighted sum of all provided information. In this paper, the attention mechanism is used both across the different internal LSTM layers, as well as over the LSTM output layers. Prediction of the output signal can be derived using the conditional probability distribution of the input signal and of the previous sample of the output signal, i.e.,

$$p(y_i \mid \mathbf{x_1}, ..., \mathbf{x_{i-1}}, y_{i-1}) \tag{24}$$

This distribution is, however, impossible to compute in most real life cases. For this reason, Eq. 24 is approximated by the non-linear function:

$$g(y_i, h_i, C_i) \tag{25}$$

where g is the LSTM function, h_i is the internal state of the LSTM and C_i is the current context, i.e., a vector holding information of which inputs are important at the current step. The context is derived from both the current state, h_i, and the input sequence $\mathbf{x}$. After the LSTM has stepped through the whole input sequence, the attention mechanism of the network decides the attention that should be put on the annotation provided at each step. The transition functions of the attentive neural network are described by Eqs. 26-28. The attention mechanism begins by computing e_t:

$$e_t = v^T . \tanh\left(W_e . h_t + U_e . d_{t-1} + b\right) \tag{26}$$

where $v, b, h_t, d_{t-1} \in \mathbb{R}^n$ and $W_e, U_e \in \mathbb{R}^{n*n}$ and d stands for the input sequence x. The attention score, $a_{t,t'}$, for each t', is computed by the softmax function, as follows:

$$a^{t,t'} = \frac{\exp(e_t)}{\sum_{j=1}^{T} \exp(e_j)} \tag{27}$$

The context vector, C_t, is computed as the weighted sum of all internal LSTM states:

$$C_t = \sum_{t'=1}^{T} a^{t,t'} . h_{t'} \tag{28}$$

4. The Proposed Method

4.1. Setting up the prediction framework

In the following, we use the models and methods, described in the previous Section, in a novel deep prediction framework. The proposed architecture (WT-ED-LSTM-AM) includes wavelet-based transformation of the collected signals, followed by an encoding-decoding step, using LSTM and attention models for final prediction.

Figure 2 shows the proposed approach for SDV multi-step prediction. The target is to predict SDV in multiple hourly steps ahead, based on current information and history sensory signal data.

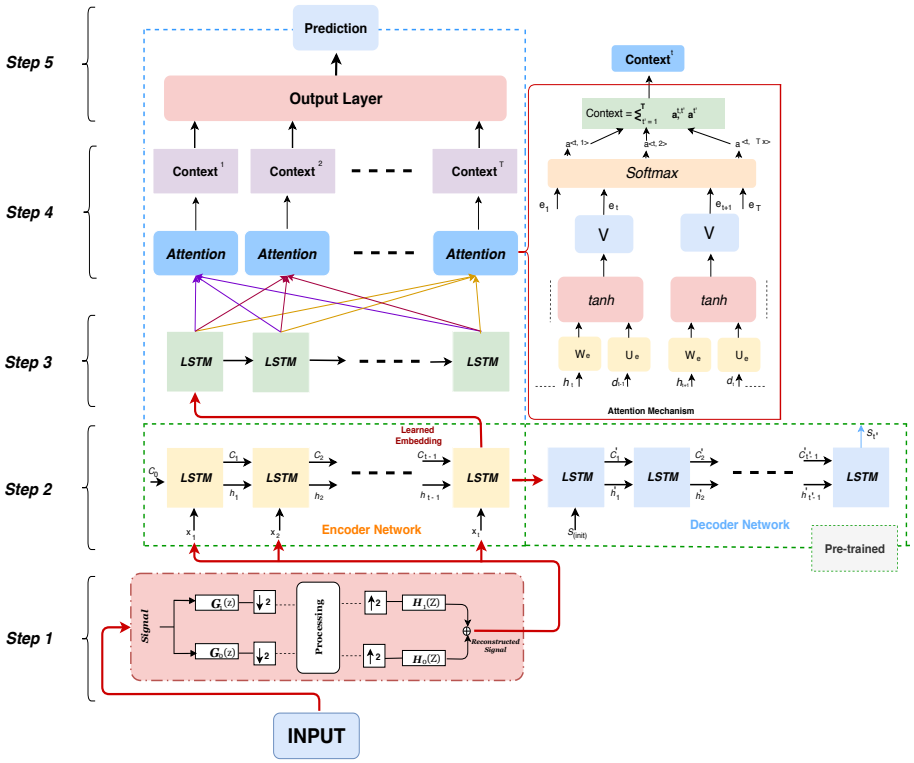

Figure 2: Deep model architecture (WT-ED-LSTM-AM) for SDV prediction.

4.2. The WT-ED-LSTM-AM architecture

The proposed WT-ED-LSTM-AM architecture for plant growth predictionis shown in Figure 2, being composed of five steps:

- Step 1: Data denoising is performed first, using the wavelet transform (WT). In particular, we decompose each input signal in two components, generating a subsampled (by 2) time series approximation and eliminating noise that is present in the high frequency component. By upsampling (by 2) and filtering, a reconstructed signal is obtained, which is provided to the next step of our approach.

- Step 2: The encoder-decoder stage is then implemented. The encoder is pre-trained to extract useful and representative embeddings from the reconstructed time series data (x_t); these embeddings can be used next

for prediction purposes. Two-layer LSTM cells are used in the encoder implementation. Based on the learned embedding states, the decoder learns to generate the (reconstructed) input signal. We designed this step, inspired by the success of video representation learning, where a similar architecture was introduced [43].

- Step 3: The encoder-decoder step constitutes the feature extraction component of the proposed approach. Then, an LSTM network, with attention mechanism, is trained to make single, or multi-step prediction, using the learned embedding as input features. LSTMs use the transition functions $\{h_1, h_2, ..., h_n\}$ of the embedding states learned in Step 3.

- Step 4: As shown in Figure 2, the attention mechanism is applied to the outputs of each LSTM unit to model a respective long-term dependence. The learned embedding states, the attention weights corresponding to these states, and the respective context, are computed as described in the previous Section, are used for implementing the attention mechanism.

- Step 5: A single layer neural network is responsible for the final prediction of the SDV value, as described in Eq. 30.

$$h_s = tanh(W_p C + W_x h_n) \tag{29}$$

$$\hat{y} = W_s h_s + b_s \tag{30}$$

5. The Experimental Study

An extensive experimental study has been carried out to evaluate the performance of the proposed approach, targeting supervised multi-step prediction of SDV in real-world data sets. The obtained results illustrate the effectiveness and efficiency of the proposed approach in predicting the SDV.

5.1. Experimental set up

The proposed architecture was used to predict growth of Ficus plants (Ficus benjamina), based on data collected from four cultivation tables in a 90 m^2 greenhouse compartment of the Ornamental Plant Research Centre (PCS) in Destelbergen, Belgium. Plant density was approximately 15 pots per m^2, where every pot contained cuttings.

The experiment started on 23 March 2016. Greenhouse microclimate was set by controlling the window openings, a thermal screen, an air heating system, assimilation light and a CO2 adding system. Plants were irrigated with an automatic flood irrigation system, controlled by time and radiation sum. Setpoints for microclimate and irrigation control were similar to the ones used in commercial greenhouses. The microclimate of the greenhouse was continuously monitored. Photosynthetic active radiation (PAR) and CO2 concentration were measured with an LI-190 Quantum Sensor (LI-COR, Lincoln, Nebraska, USA) and a carbon dioxide probe (Vaisala CARBOCAP GMP343, Vantaa, Finland), respectively. Temperature and relative humidity were measured with a temperature and relative humidity probe (Campbell Scientific CS215, Logan, UT, USA), which was installed in a ventilated radiation shield.

Stem diameter was continuously monitored on three plants with a linear variable displacement transducer (LVDT, Solartron, Bognor Regis, UK) sensor. The hourly variation rate of stem diameter ($mm\ d^{-1}$) was calculated as the difference between the current stem diameter and the stem diameter recorded on one hour earlier, at a given time point. Thus, the frequency of collected data has been at one hour basis.

We performed experiments on one-step, two-step and three-step forecasting.

In one-step-ahead forecasting, we used input data collected in previous 15 hours, to predict the SDV value in the current hour.

In two-step-ahead, i.e., 6 hours forecasting, we used input data collected in the previous 6 hours, with a 6-hour stride.

In three-step-ahead, i.e., 12 hours forecasting, we used the previous 12 hours, with a 12-hour stride.

In all experiments, we used the first 70% of data samples as training set, the next 10% of data samples as validation set and the rest 20% of data samples as test set.

5.2. Performance evaluation

The Mean Absolute Error (MAE), the Root Mean Squared Error (RMSE) and the Mean Absolute Percentage Error (MAPE) were used to evaluate the performance of prediction models. Formulas of these measures are shown below:

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}\left(\frac{A_t - F_t}{A_t}\right)^2} \tag{31}$$

$$MAE = \frac{1}{n}\sum_{t=1}^{n}|A_t - F_t| \tag{32}$$

$$MAPE = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{A_t - F_t}{A_t}\right| \tag{33}$$

where A_t denotes the actual values and F_t the predicted values.

5.3. Feature normalization

In all experiments we used min-max normalization (min-max scaling) on the extracted features, re-scaling their values in the range [0, 1] through the following formula:

$$g_i = \frac{f_i - min(f)}{max(f) - min(f)} \tag{34}$$

where g_i and f_i are the values of the normalised and original $i-th$ feature and $min(f)$ and $max(f)$ are the minimum and maximum values of the original features.

5.4. Experimental results

The experimental results illustrate the very good performance of the proposed methodology, which outperforms all considered baseline methods. For comparison purposes, we used the same hyper-parameters in the proposed approach and in three baseline models: a two-layer stacked GRU, a LSTM and

a MLP with Stochastic Gradient Descent (SGD); learning rate $ls = 0.001$ and batch size = 32 were adopted. All models were trained for 100 epochs, using the same training, as well as validation and test data sets. In the proposed method, we used a two layer LSTM encoder-decoder structure, with 128 and 32 neurons respectively. In the prediction model, we used a single layer LSTM with 128 neurons. Figure 3 illustrates minimization of the MSE per epoch during the training phase of all models, in all three prediction tasks.

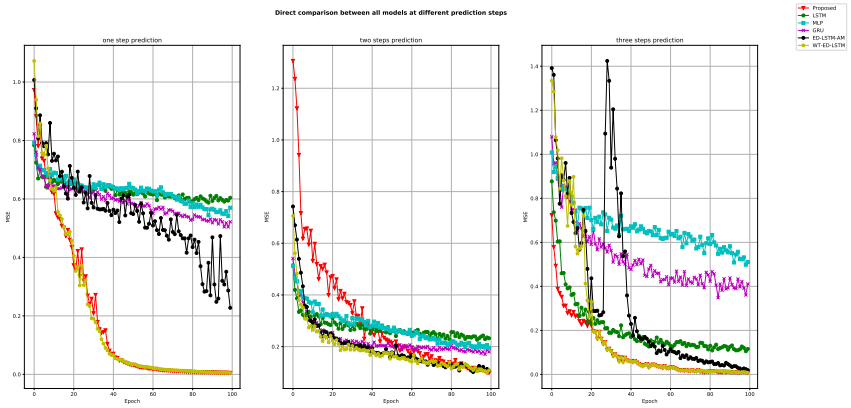

Figure 3: Training comparison of the different models at each epoch.

The obtained accuracy in terms of the three error criteria for the multi-step prediction tasks is shown in Table 1.

Steps	**One Step (1hr)**			**Two steps (6hr)**			**Three steps (12hr)**		
Error / **Models**	**RMSE**	**MAE**	**MAPE**	**RMSE**	**MAE**	**MAPE**	**RMSE**	**MAE**	**MAPE**
SVR	0.65	0.46	4.47	0.70	0.55	2.40	0.82	0.67	1.50
RFR	0.74	0.52	8.31	0.66	0.51	3.27	0.72	0.57	1.80
MLP	0.0034	0.0023	2.72	0.0045	0.0029	2.20	0.0048	0.0027	1.63
GRU	0.0031	0.0022	3.43	0.0039	0.0026	2.74	0.0080	0.0040	1.93
LSTM	0.0031	0.0022	3.27	0.0033	0.0024	2.46	0.0054	0.0031	1.60
WT-ED-LSTM	0.0028	0.0020	2.60	0.0033	0.0026	2.61	0.0042	0.0034	1.46
ED-LSTM-AM	0.0074	0.0034	3.397	0.0030	0.0031	2.45	0.0046	0.0032	1.40
Proposed	**0.0026**	**0.0017**	**2.14**	**0.0028**	**0.0021**	**2.03**	**0.0029**	**0.0023**	**1.35**

Table 1: Performance of the proposed and baseline models for multi-step prediction

Let us first dsicuss the one step prediction results.

The performance of the LSTM and GRU models for one-step-ahead prediction were very similar, with the LSTM model showing an (edge) improvement over GRU one, as far the MAPE criterion was concerned. The MLP model performance was lower than LSTM and GRU when considering RMSE and MAE criteria; it scored better than the LSTM and GRU when MAPE criterion was considered.

The proposed approach (WT-ED-LSTM-AM) outperformed all baseline models on all multi-step prediction tasks. Figure 4 illustrates this achievement, over prediction steps ranging from 1 to 12.

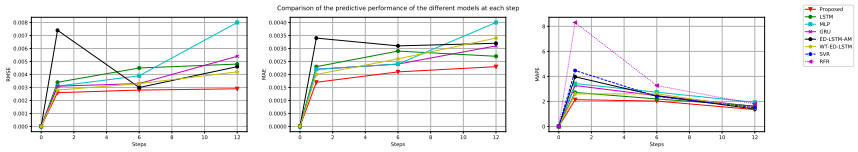

Figure 4: Comparison of the predictive performance of the different models at each step.

Figure 5 shows the accuracy of Ficus growth one-step prediction by all methods for about 600 data samples. It can be seen that the proposed model successfully performs one-step ahead prediction, outperforming the other methods and providing accurate estimates of almost all peak values in the original data.

In addition, to compare the distribution of the prediction errors provided by the baseline models with that of the proposed approach, we performed a statistical analysis of them. The histograms of the produced one-step-prediction errors are shown in Figure 6. It can be seen that in the proposed approach, close to 57% of predictions resulted in prediction errors around 0.00 and the remaining 43% prediction errors ranging between -0.010 and 0.015.

Let us now discuss the results obtained in two-step prediction.

These results are shown in Table 1. The proposed approach outperformed all baseline models, providing lower RMSE, MAE and MAPE values in this case as well. Fig. 7 shows that almost all peak original values are precisely predicted

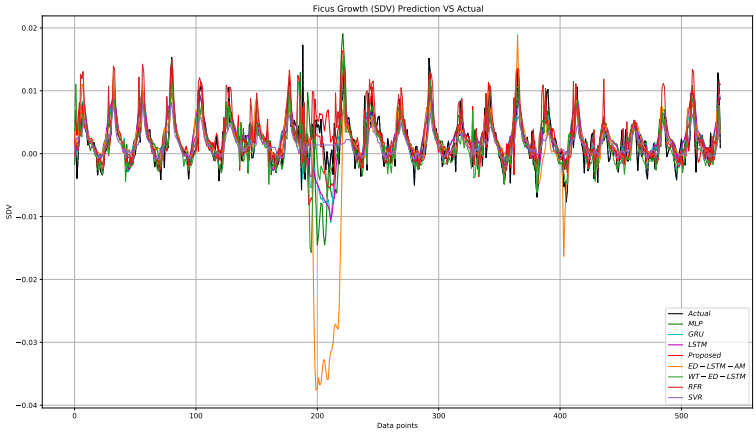

Figure 5: Obtained accuracy in one-step prediction of Ficus growth (SVD) by the proposed and baseline methods

by the proposed approach.

The resulting histograms for two-step-prediction errors are shown in Fig. 8. In the proposed approach, close to 77% of predictions resulted in prediction errors between -0.004 and 0.002; the remaining 23% of prediction errors ranged between -0.008 and 0.008. This greatly outperformed the other baseline models.

The proposed approach also provided a superior three-step-ahead prediction.

In Fig. 9, it can be seen that all baseline models failed to capture the peak at data sample 14, with the proposed approach providing much better estimates of the targeted values than the baseline models. Fig. 10 shows the prediction error distributions for all baseline models, as well for the proposed approach. It shows that close to 56% of predictions provided by the proposed approach, produced prediction errors between -0.002 and 0.002; the remaining 44% of prediction errors ranged between -0.006 and 0.006. In this case, as well, the proposed method outperformed all other baseline models.

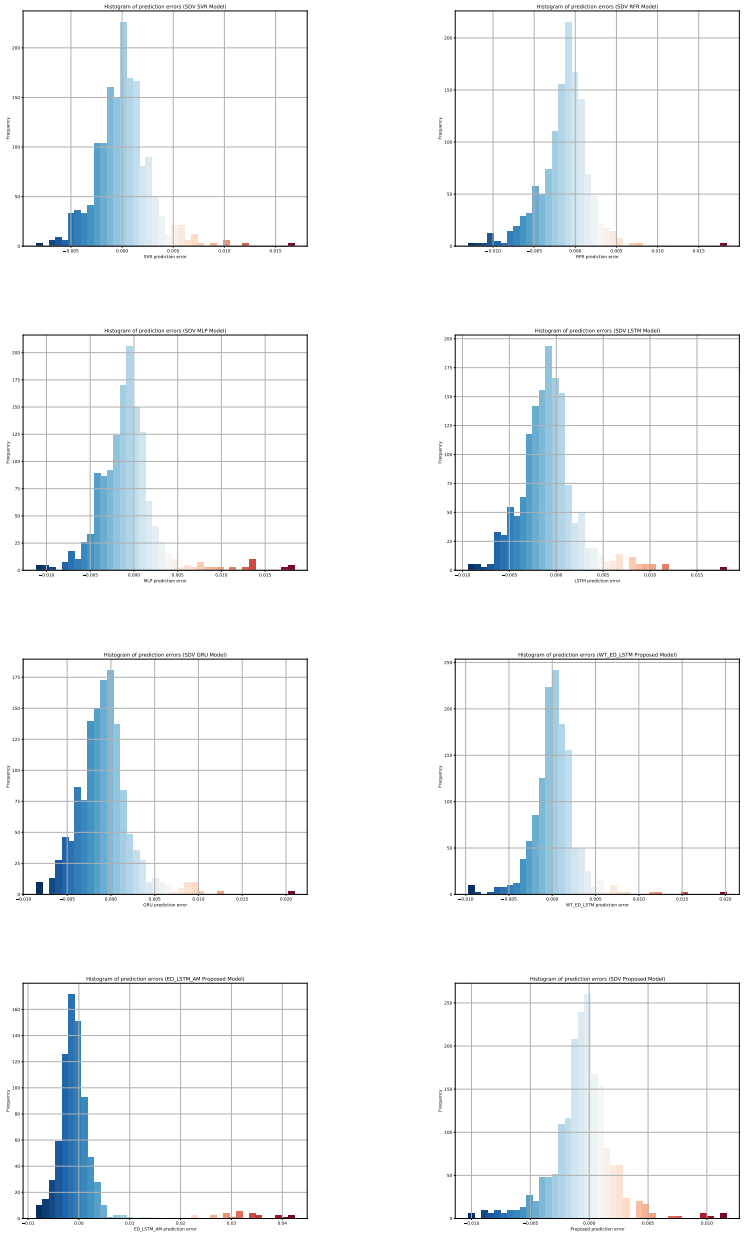

Figure 6: Error distribution for one step prediction (1 hour ahead)

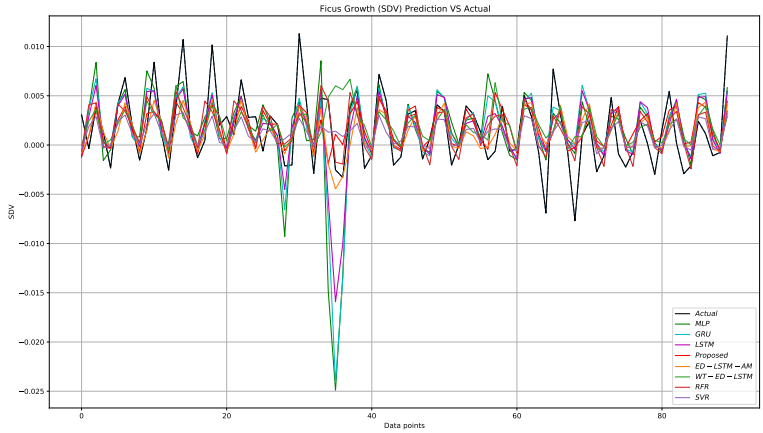

Figure 7: Obtained accuracy in two-step prediction of Ficus growth (SVD) by the proposed and baseline methods

6. Conclusions and Future Work

This paper proposed a novel multi-step-ahead time series prediction approach. The first step of the proposed method has been to use a wavelet transform to decompose and smooth the original data. As a consequence, a better model fitting could be achieved on the reconstructed signals. The second step introduced an encoder-decoder framework based on LSTMs, which managed to effectively produce appropriate features for multi-step prediction. The third step which used LSTMs coupled with an attention mechanism was able to successfully implement the prediction tasks.

The proposed approach was used for multi-step-ahead prediction of Ficus Benjamina stem diameter variations, providing high prediction accuracy.

Real-world data have been collected and formed datasets that were used to evaluate the proposed methodology. Hourly time intervals were used in the input data, as well in our multi-step-ahead predictions. Comparisons were carried out, over these real-world data, with state-of-the-art baseline models, showing that the developed approach provides much better prediction results.

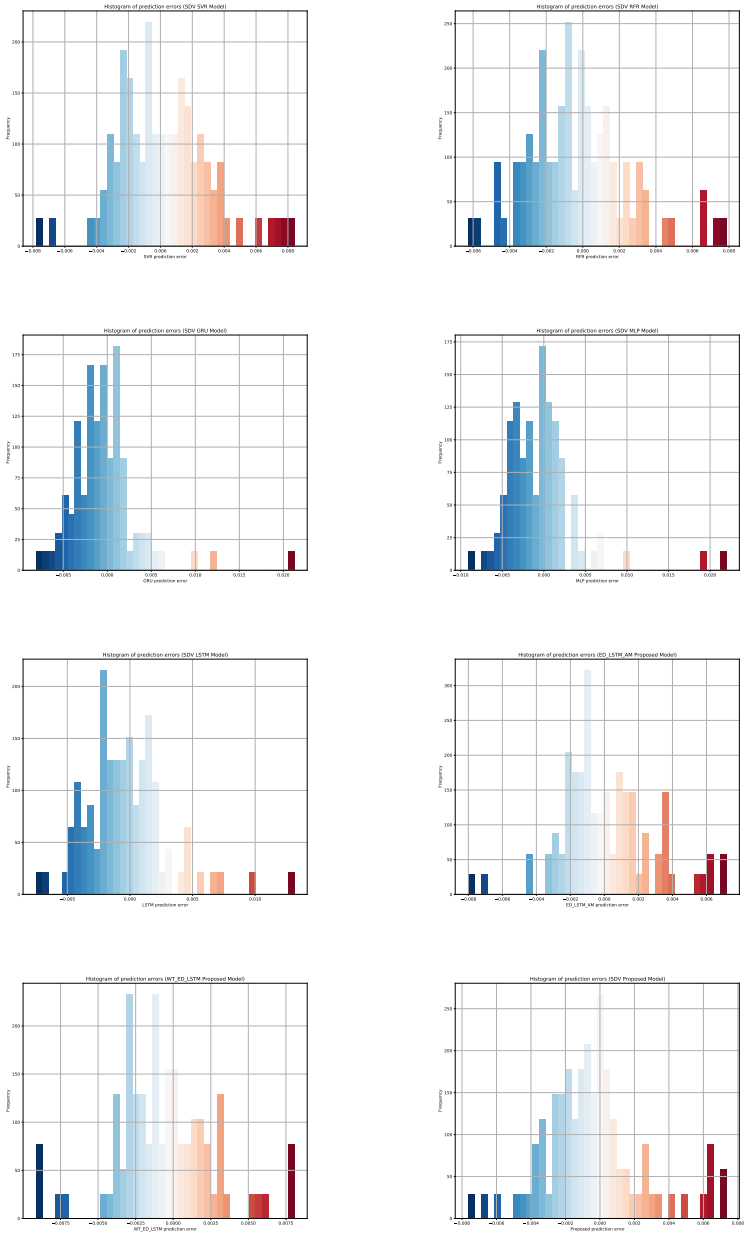

Figure 8: Error distribution for two step prediction (6 hours ahead)

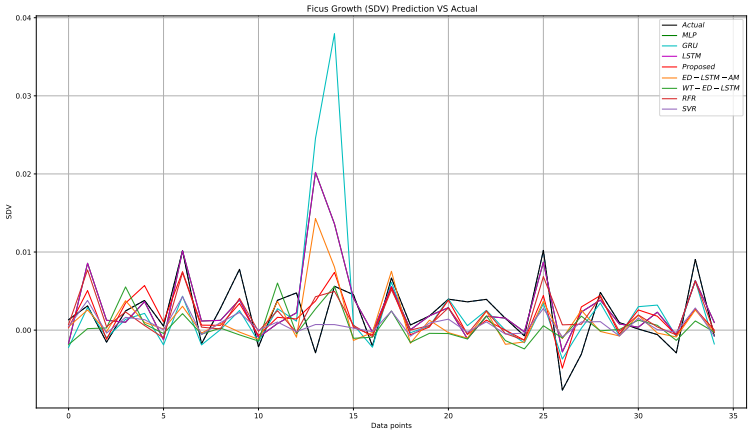

Figure 9: Obtained accuracy in three-step prediction of Ficus growth (SVD) by the proposed and baseline methods

A topic of future research is to merge the data driven approach presented in this paper with knowledge-based ones, especially for modelling the context, i.e., the relations among the considered variables; we will be adapting former research of ours in merging symbolic and subsymbolic approaches [29].

Acknowledgements

This work was done in the framework and with the support of the EU Interreg SMARTGREEN project (2017-2021). We would like to thank all growers (UK and EU), for providing us with the presented data sets. We also wish to thank the reviewers of the paper. Their valuable feedback, suggestions and comments helped us to improve the quality of this work.

References

[1] Alhnaity, B., & Abbod, M. (2020). A new hybrid financial time series prediction model. *Engineering Applications of Artificial Intelligence*, *95*, 103873.

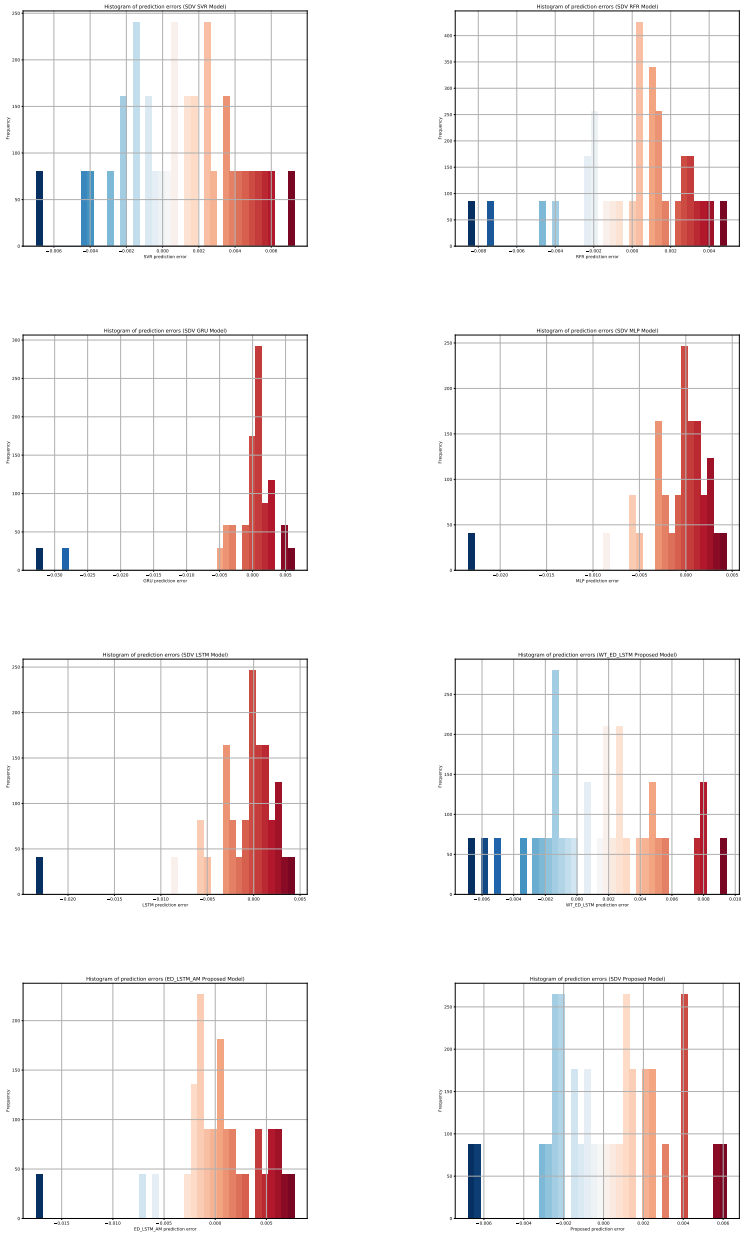

Figure 10: Error distribution for three step prediction (12 hour ahead)

[2] Alhnaity, B., Pearson, S., Leontidis, G., & Kollias, S. (2019). Using deep learning to predict plant growth and yield in greenhouse environments. *arXiv preprint arXiv:1907.00624*, .

[3] Arqub, O. A. (2017). Adaptation of reproducing kernel algorithm for solving fuzzy fredholm–volterra integrodifferential equations. *Neural Computing and Applications*, *28*, 1591–1610.

[4] Arqub, O. A., & Abo-Hammour, Z. (2014). Numerical solution of systems of second-order boundary value problems using continuous genetic algorithm. *Information sciences*, *279*, 396–415.

[5] Arqub, O. A., Al-Smadi, M., Momani, S., & Hayat, T. (2017). Application of reproducing kernel algorithm for solving second-order, two-point fuzzy boundary value problems. *Soft Computing*, *21*, 7191–7206.

[6] Arqub, O. A., Mohammed, A.-S., Momani, S., & Hayat, T. (2016). Numerical solutions of fuzzy differential equations using reproducing kernel hilbert space method. *Soft Computing*, *20*, 3283–3302.

[7] Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, .

[8] Breiman, L. (2001). Random forests machine learning, vol. 45, .

[9] Buhmann, M. D. (2003). *Radial basis functions: theory and implementations* volume 12. Cambridge university press.

[10] Chlingaryan, A., Sukkarieh, S., & Whelan, B. (2018). Machine learning approaches for crop yield prediction and nitrogen status estimation in precision agriculture: A review. *Computers and Electronics in Agriculture*, *151*, 61–69.

[11] Choi, E., Bahadori, M. T., Sun, J., Kulas, J., Schuetz, A., & Stewart, W. (2016). Retain: An interpretable predictive model for healthcare us-

ing reverse time attention mechanism. In *Advances in Neural Information Processing Systems* (pp. 3504–3512).

[12] Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, .

[13] Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine learning*, *20*, 273–297.

[14] Daniel, J., Andrés, P.-U., Héctor, S., Miguel, B., Marco, T. et al. (2008). A survey of artificial neural network-based modeling in agroecology. In *Soft Computing applications in industry* (pp. 247–269). Springer.

[15] De Gooijer, J. G., & Hyndman, R. J. (2006). 25 years of time series forecasting. *International journal of forecasting*, *22*, 443–473.

[16] Du, Z., Qin, M., Zhang, F., & Liu, R. (2018). Multistep-ahead forecasting of chlorophyll a using a wavelet nonlinear autoregressive network. *Knowledge-Based Systems*, *160*, 61–70.

[17] Duchesne, L., & Houle, D. (2011). Modelling day-to-day stem diameter variation and annual growth of balsam fir (abies balsamea (l.) mill.) from daily climate. *Forest Ecology and Management*, *262*, 863–872.

[18] Fu, T.-c. (2011). A review on time series data mining. *Engineering Applications of Artificial Intelligence*, *24*, 164–181.

[19] Gangopadhyay, T., Tan, S. Y., Huang, G., & Sarkar, S. (2018). Temporal attention and stacked lstms for multivariate time series prediction, .

[20] Geng, Z., Chen, G., Han, Y., Lu, G., & Li, F. (2020). Semantic relation extraction using sequential and tree-structured lstm with attention. *Information Sciences*, *509*, 183–192.

[21] González-Chávez, M. d. C. A., Carrillo-González, R., Cuellar-Sánchez, A., Delgado-Alvarado, A., Suárez-Espinosa, J., Ríos-Leal, E., Solís-Domínguez,

F. A., & Maldonado-Mendoza, I. E. (2019). Phytoremediation assisted by mycorrhizal fungi of a mexican defunct lead-acid battery recycling site. *Science of the Total Environment*, *650*, 3134–3144.

[22] Goodfellow, I. (2016). Nips 2016 tutorial: Generative adversarial networks. *arXiv preprint arXiv:1701.00160*, .

[23] Gupta, D., & Choubey, S. (2015). Discrete wavelet transform for image processing. *International Journal of Emerging Technology and Advanced Engineering*, *4*, 598–602.

[24] Haykin, S. (2007). *Neural networks: a comprehensive foundation*. Prentice-Hall, Inc.

[25] Hinckley, T. M., & Bruckerhoff, D. N. (1975). The effects of drought on water relations and stem shrinkage of quercus alba. *Canadian Journal of Botany*, *53*, 62–72.

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

[27] Kanai, S., Adu-Gymfi, J., Lei, K., Ito, J., Ohkura, K., Moghaieb, R. E., El-Shemy, H., Mohapatra, R., Mohapatra, P. K., Saneoka, H. et al. (2008). N-deficiency damps out circadian rhythmic changes of stem diameter dynamics in tomato plant. *Plant science*, *174*, 183–191.

[28] Karpathy, A., & Fei-Fei, L. (2015). Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3128–3137).

[29] Kollia, I., Simou, N., Stafylopatis, A., & Kollias, S. (2010). Semantic image analysis using a symbolic neural architecture. *Image Analysis & Stereology*, *29*, 159–172.

[30] Malhotra, P., Ramakrishnan, A., Anand, G., Vig, L., Agarwal, P., & Shroff, G. (2016). Lstm-based encoder-decoder for multi-sensor anomaly detection. *arXiv preprint arXiv:1607.00148*, .

[31] Mallat, S. G. (1989). A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (pp. 674–693).

[32] Pan, W. (2016). A survey of transfer learning for collaborative recommendation with auxiliary data. *Neurocomputing*, *177*, 447–453.

[33] Pasti, L., Walczak, B., Massart, D., & Reschiglian, P. (1999). Optimization of signal denoising in discrete wavelet transform. *Chemometrics and intelligent laboratory systems*, *48*, 21–34.

[34] Pouteau, R., Meyer, J.-Y., Taputuarai, R., & Stoll, B. (2012). Support vector machines to map rare and endangered native plants in pacific islands forests. *Ecological Informatics*, *9*, 37–46.

[35] Quiroz, R., Yarlequé, C., Posadas, A., Mares, V., & Immerzeel, W. W. (2011). Improving daily rainfall estimation from ndvi using a wavelet transform. *Environmental Modelling & Software*, *26*, 201–209.

[36] Rapantzikos, K., Tsapatsoulis, N., Avrithis, Y., & Kollias, S. (2007). Bottom-up spatiotemporal visual attention model for video analysis. *IET Image Processing*, *1*, 237–248.

[37] Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural networks*, *61*, 85–117.

[38] Shen, T., Zhou, T., Long, G., Jiang, J., & Zhang, C. (2018). Bi-directional block self-attention for fast and memory-efficient sequence modeling. *arXiv preprint arXiv:1804.00857*, .

[39] Singh, A., Ganapathysubramanian, B., Singh, A. K., & Sarkar, S. (2016). Machine learning for high-throughput stress phenotyping in plants. *Trends in plant science*, *21*, 110–124.

[40] Singh, R., & Khare, A. (2014). Fusion of multimodal medical images using daubechies complex wavelet transform–a multiresolution approach. *Information Fusion*, *19*, 49–60.

[41] Song, H., Rajan, D., Thiagarajan, J. J., & Spanias, A. (2018). Attend and diagnose: Clinical time series analysis using attention models. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

[42] Sorjamaa, A., Hao, J., Reyhani, N., Ji, Y., & Lendasse, A. (2007). Methodology for long-term prediction of time series. *Neurocomputing*, *70*, 2861–2869.

[43] Srivastava, N., Mansimov, E., & Salakhudinov, R. (2015). Unsupervised learning of video representations using lstms. In *International conference on machine learning* (pp. 843–852).

[44] Taieb, S. B., & Atiya, A. F. (2015). A bias and variance analysis for multistep-ahead time series forecasting. *IEEE transactions on neural networks and learning systems*, *27*, 62–76.

[45] Taieb, S. B., Bontempi, G., Atiya, A. F., & Sorjamaa, A. (2012). A review and comparison of strategies for multi-step ahead time series forecasting based on the nn5 forecasting competition. *Expert systems with applications*, *39*, 7067–7083.

[46] Vandegehuchte, M. W., Guyot, A., Hubau, M., De Groote, S. R., De Baerdemaeker, N. J., Hayes, M., Welti, N., Lovelock, C. E., Lockington, D. A., & Steppe, K. (2014). Long-term versus daily stem diameter variation in co-occurring mangrove species: Environmental versus ecophysiological drivers. *Agricultural and Forest Meteorology*, *192*, 51–58.

[47] Wolfert, S., Ge, L., Verdouw, C., & Bogaardt, M.-J. (2017). Big data in smart farming–a review. *Agricultural Systems*, *153*, 69–80.

[48] Yu, S.-M., Lo, S.-F., & Ho, T.-H. D. (2015). Source–sink communication: regulated by hormone, nutrient, and stress cross-signaling. *Trends in plant science*, *20*, 844–857.

[49] Yuan, W., Wang, H., Yu, X., Liu, N., & Li, Z. (2020). Attention-based context-aware sequential recommendation model. *Information Sciences*, *510*, 122–134.